

Universidad de Valladolid

E. T. S. DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

**Herramienta Web configurable para el apoyo
a la toma de decisión usando GRASS y JSP**

Alumna: Esperanza González de la Mora

Tutores: José Belarmino Pulido Junquera

Carlos J. Alonso González

Estoy profundamente agradecida con mis tutores José Belarmino Pulido Junquera y Carlos J. Alonso González en primer lugar por brindarme la oportunidad de convertirme en una profesional informática, pero también por transmitirme su metodología, aptitudes y conocimientos con un gran valor añadido como es la cercanía y el buen hacer de las cosas. Sin vuestra constancia y capacidad posiblemente este proyecto no hubiera tenido lugar.

Quiero agradecer a las empresas e instituciones que hicieron posible la realización del trabajo como son Iberdrola y Proes, y a GMV por permitirme la realización de prácticas en empresa.

También a mi familia y amigos por su apoyo incondicional y ánimo en los días más grises. A mis compañeros y amigos del GSI y GMV que me han acompañado cada día y hemos vivido grandes hazañas.

Además quiero agradecer a los posibles lectores que hayan despertado el interés por este trabajo.

Y por supuesto, volveré a repetir: a mi perro Tul, por su paciencia en los días de trabajo, y su alegría en los ratos de descanso.

*A todos vosotros: **Muchas gracias.***

```
If(me_habéis_apoyado){  
    Muchas gracias;  
    }else{  
    También ☺;  
    }
```


ÍNDICE

ÍNDICE	5
ÍNDICE DE FIGURAS.....	15
CAPÍTULO 1: INTRODUCCIÓN	21
1.1 INTRODUCCIÓN	21
1.2 OBJETIVOS.....	22
1.3 ESTRUCTURA DE LA MEMORIA	23
CAPÍTULO 2: AQUITECTURA DEL DSS	25
2.1. INTRODUCCIÓN	25
2.2. PROCESO.....	25
2.3. DATOS DE ENTRADA: LA BASE DE DATOS GEOGRÁFICA Y BASE DE DATOS DE LA APLICACIÓN	29
2.4. TIPOS DE REGLAS DE DECISIÓN PARA REALIZAR LA TOMA DE DECISIÓN	29
2.5. DATOS Y CRITERIOS FIJADOS EN LAS REUNIONES OCEANLIDER	30
2.5.1. <i>Criterios físicos</i>	30
2.5.2. <i>Criterios operacionales</i>	32
2.5.3. <i>Criterios de recurso energético</i>	39
2.5.4. <i>Criterios medioambientales</i>	39
2.6. EL PROTOCOLO DE LA APLICACIÓN DE LOS CRITERIOS	56
CAPÍTULO 3: GENERACIÓN AUTOMÁTICA DE SCRIPTS	61
3.1. CONTEXTO Y OBJETIVOS	61
3.2. ANÁLISIS DEL MODELO DE DATOS	62
3.2.1. <i>Requisitos</i>	62
3.2.2. <i>Modelo entidad-relación</i>	62
3.3. MODELO DE DATOS	67
3.3.1. <i>Modelo relacional</i>	67
3.3.2. <i>Descripción de las entidades y relaciones</i>	74
3.3.2.1 Entidades.....	74
3.3.2.2 Relaciones	77
3.3.3. <i>Modelo estático</i>	77
3.3.4. <i>Modelo dinámico</i>	80
3.4. SISTEMA GESTOR DE BASE DE DATOS.....	83
3.4.1. <i>Herencia en PostgreSQL</i>	84
3.5. OPERACIONES DEFINIDAS EN EL SISTEMA Y PROGRAMACIÓN DE CRITERIOS	86
3.5.1. <i>Operaciones del sistema</i>	86
3.5.2. <i>Estudio de las entradas de las operaciones</i>	95
3.5.3. <i>Tabla Operaciones en la base de datos</i>	100
3.5.4. <i>Programación de criterios</i>	100
3.5.4.1. Programación de los criterios definidos en OceanLider	101

3.5.4.1.1. Criterios físicos	101
4.5.4.1.2. Criterios operacionales	103
4.5.4.1.3. Criterios de recurso energético	110
4.5.4.1.4. Criterios medio ambientales	112
3.6. MÉTODOS DE RAZONAMIENTO.....	132
CAPÍTULO 4: DESARROLLO DE LA HERRAMIENTA	133
4.1. PLAN DE DESARROLLO Y SEGUIMIENTO.....	133
4.1.1. <i>Introducción</i>	133
4.1.1.1. Propósito.....	133
4.1.1.2. Alcance.....	133
4.1.1.3. Definiciones, Acrónimos y Abreviaturas	134
4.1.1.4. Perspectiva general.....	134
4.1.2. <i>Perspectiva General del Proyecto</i>	134
4.1.2.1. Propósito, Alcance y Objetivos del Proyecto	134
4.1.2.2. Suposiciones y Restricciones.....	134
4.1.2.3. Entregables del Proyecto	136
4.1.2.4. Evolución del Plan de Desarrollo de Software	136
4.1.3. <i>Organización del proyecto</i>	137
4.1.3.1. Estructura Organizativa.....	137
4.1.3.2. Interfaces Externas	137
4.1.3.3. Roles y Responsabilidades del Equipo de Desarrollo	137
4.1.4. <i>Gestión de Proceso</i>	138
4.1.4.1. Estimaciones del Proyecto	138
4.1.4.2. Plan de Proyecto	138
4.1.4.2.1. Plan de Fases	138
4.1.4.2.2. Objetivos de las Iteraciones.....	139
4.1.4.2.3. Versiones	140
4.1.4.2.4. Planificación Temporal del Proyecto	141
4.1.4.2.5. Recursos del Proyecto	141
4.1.4.2.5.1. Recursos humanos	141
4.1.4.2.5.2. Recursos software.....	141
4.1.4.2.5.3. Recursos hardware.....	142
4.1.4.3. Seguimiento y Control del Proyecto	142
4.1.4.4. Gestión de Requisitos	142
4.1.4.5. Control de Calidad	142
4.1.4.6. Informes y Medidas	142
4.1.4.7. Gestión de Riesgos.....	142
4.1.4.8. Gestión de configuraciones	142
4.1.5. <i>Plan de Iteraciones</i>	143
4.1.5.1. Plan de fase de Inicio	143
4.1.5.1.1. <i>Introducción</i>	143
4.1.5.1.1.1. Objetivos	143
4.1.5.1.1.2. Alcance.....	143
4.1.5.1.2. Plan de la Iteración	144
4.1.5.1.2.1. Descripción de tareas.....	144
4.1.5.1.2.2. Calendarización de tareas	145
4.1.5.1.3. Criterios de evaluación	146
4.1.5.2. Plan de fase de Elaboración	147
4.1.5.2.1. <i>Introducción</i>	147

4.1.5.2.1.1. Objetivos	147
4.1.5.2.1.2. Alcance	147
4.1.5.2.2. Plan de la Iteración	148
4.1.5.2.2.1. Descripción de tareas	148
4.1.5.2.2.2. Calendarización de tareas	149
4.1.5.2.3. Criterios de evaluación	151
4.1.5.3. Plan de fase de Construcción	151
4.1.5.3.1. Iteración 1	151
4.1.5.3.1.1. Introducción	151
4.1.5.3.1.1.1. Objetivos	151
4.1.5.3.1.1.2. Alcance	152
4.1.5.3.1.2. Plan de la Iteración	152
4.1.5.3.1.2.1. Descripción de tareas	152
4.1.5.3.1.2.2. Calendarización de tareas	153
4.1.5.3.1.3. Criterios de evaluación	154
4.1.5.3.2. Iteración 2	154
4.1.5.3.2.1. Introducción	154
4.1.5.3.2.1.1. Objetivos	154
4.1.5.3.2.1.2. Alcance	155
4.1.5.3.2.2. Plan de la Iteración	155
4.1.5.3.2.2.1. Descripción de tareas	155
4.1.5.3.2.2.2. Calendarización de tareas	156
4.1.5.3.2.3. Criterios de evaluación	157
4.1.5.3.3. Iteración 3	157
4.1.5.3.3.1. Introducción	157
4.1.5.3.3.1.1. Objetivos	157
4.1.5.3.3.1.2. Alcance	158
4.1.5.3.3.2. Plan de la Iteración	158
4.1.5.3.3.2.1. Descripción de tareas	158
4.1.5.3.3.2.2. Calendarización de tareas	159
4.1.5.3.3.3. Criterios de evaluación	160
4.1.5.4. Plan de fase de Transición	160
4.1.5.4.1. Introducción	160
4.1.5.4.1.1. Objetivos	160
4.1.5.4.1.2. Alcance	161
4.1.5.4.2. Plan de la Iteración	161
4.1.5.4.2.1. Descripción de tareas	161
4.1.5.4.2.2. Calendarización de tareas	162
4.1.5.4.3. Criterios de evaluación	162
4.1.6. Seguimiento	163
4.1.6.1. Introducción	163
4.1.6.1.1. Objetivo	163
4.1.6.2. Fase de Inicio	163
4.1.6.2.1. Seguimiento 1ª iteración	163
4.1.6.3. Fase de Elaboración	164
4.1.6.3.1. Seguimiento 1ª iteración	164
4.1.6.4. Fase de Construcción	165
4.1.6.4.1. Seguimiento 1ª iteración	165
4.1.6.4.2. Seguimiento 2ª iteración	165
4.1.6.4.3. Seguimiento 3ª iteración	166

4.1.6.5. Fase de Transición.....	167
4.1.6.5.1. Seguimiento 1ª iteración	167
4.2. INGENIERÍA DEL SOFTWARE	168
4.2.1. Análisis	168
4.2.1.1. Requisitos.....	168
4.2.1.1.1. Requisitos generales.....	168
4.2.1.1.2. Requisitos de base de datos	168
4.2.1.1.3. Requisitos de apariencia.....	169
4.2.1.1.4. Requisitos de autenticación en la aplicación.....	169
4.2.1.1.5. Requisitos de presentación de objetivos	170
4.2.1.1.6. Requisitos de creación de escenarios.....	170
4.2.1.1.7. Requisitos de consulta y ejecución de esenarios.....	170
4.2.1.1.8. Requisitos de presentación de resultados.....	171
4.2.1.2. Casos de Uso	171
4.2.1.2.1. Descripción de casos de uso	171
4.2.1.3. Diagrama de clases	172
4.2.1.3.1. Descripción de las clases	174
4.2.2. Diseño	174
4.2.2.1. Casos de Uso	174
4.2.2.1.1. General	174
4.2.2.1.2. Autenticación usuario.....	175
4.2.2.1.3. Crear escenario.....	175
4.2.2.1.4. Cargar escenario.....	176
4.2.2.1.5. Realizar cálculos	177
4.2.2.1.6. Visualizar resultados.....	177
4.2.2.2. Descripción de casos de uso	178
4.2.2.2.1. General	178
4.2.2.2.2. Autenticación de usuario.....	178
4.2.2.2.3. Crear escenario.....	179
4.2.2.2.4. Cargar escenario.....	179
4.2.2.2.5. Realizar cálculos	180
4.2.2.2.6. Visualizar resultados.....	180
4.2.2.3. Diagrama de clases	180
4.2.2.3.1. Componente datos	181
4.2.2.3.2. Componente entities.....	182
4.2.2.3.3. Componente motor.....	193
4.2.2.3.4. Componente interfaz.....	198
4.2.2.4. Diagramas de secuencia.....	200
4.2.2.4.1. CU-01: Autenticación usuario	200
4.2.2.4.2. CU-02: Crear escenario	201
4.2.2.4.3. CU-03: Cargar escenario	201
4.2.2.4.4. CU-04: Realizar cálculos.....	202
4.2.2.4.5. CU-05: Visualizar resultados.....	202
4.2.2.4.6. CU-6: Modificar y/o seleccionar criterios escenario.....	203
4.2.2.4.7. CU-7: Hacer Loguin	203
4.2.2.4.8. CU-8: Registrarse	204
4.2.2.4.9. CU-9: Salir de sesión	204
4.2.2.4.10. CU-10: Seleccionar Dispositivo	205
4.2.2.4.11. CU-11: Seleccionar zona geográfica	205
4.2.2.4.12. CU-12: Introducir parámetros escenario	206

4.2.2.4.13. CU-13: Modificar y/o seleccionar criterios escenario	206
4.2.2.4.14. CU-14: Guardar escenario	207
4.2.2.4.15. CU-15: Selección escenario	207
4.2.2.4.16. CU-16: Modificar dispositivo	208
4.2.2.4.17. CU-17: Modificar zona geográfica	208
4.2.2.4.18. CU-18: Modificar parámetros escenario	209
4.2.2.4.19. CU-19: Modificar y/o seleccionar criterios escenario	209
4.2.2.4.20. CU-20: Extraer información escenario	210
4.2.2.4.21. CU-21: Traducir/Implementar criterios a operaciones	211
4.2.2.4.22. CU-22: Realización script ejecutable	211
4.2.2.4.23. CU-23: Ejecución de script	212
4.2.2.4.24. CU-24: Modificación estado escenario	212
4.2.2.4.25. CU-25: Mostrar información resultados intermedios y finales	213
4.2.2.4.26. CU-26: Visualizar resultados	213
4.3. TECNOLOGÍAS MÁS RELEVANTES	213
4.3.1. <i>Visión global y tecnología</i>	213
4.3.1.1. Partes del sistema	215
4.3.1.2. Aplicación MVC	216
4.4. INTERFAZ DEL SISTEMA	219
4.4.1. <i>Objetivos generales del Diseño</i>	219
4.4.2. <i>Diseño de la Interfaz de Usuario</i>	220
4.4.2.1. Bocetos	220
a. Página principal	221
b. Página de presentación de Objetivos	222
c. Página de selección del escenario	223
d. Página de configuración de criterios	224
e. Página de visualización de resultados	225
4.5. FUNCIONALIDADES RELEVANTES	227
4.5.1. <i>Generación de scripts operacionales</i>	227
4.5.1.1. Mecanismos de depuración en la ejecución de operaciones SIG	227
4.5.1.2. Ejemplo de script operacional	228
4.5.2. <i>Visualización de mapas en la web</i>	230
4.5.2.1. Introducción	230
4.5.2.2. Generación .MAP para mapas raster y vectoriales	233
4.5.2.2.1. Creación .MAP para mapas vectoriales	234
4.5.2.2.2. Creación .MAP para mapas raster	235
4.5.2.2.3. Mecanismos de debug en MapServer	239
4.5.2.2.4. Generación automática de .MAP en la herramienta DSS	240
4.5.2.2.5. Problemática de visualización de mapas con MapServer y PostGIS 2.0	240
4.5.2.3. Visualización utilizando la biblioteca OpenLayers	241
4.6. OPERACIONES INTERMEDIAS	243
4.6.1. <i>Generación del mapa profundidad como MDE</i>	243
4.6.1.1. Cálculo de un MDE en GvSIG	244
4.6.1.1.1. Rellenar celdas sin datos	245
4.6.1.1.2. Rellenar celdas sin datos (por vecindad)	246
4.6.1.2. Cálculo de un MDE en GRASS	247
4.6.1.2.1. Módulo r.surf.idw	248
4.6.1.2.2. Módulo v.surf.bspline	249
4.6.1.2.3. Método v.surf.rst	250

4.6.1.2.4. Módulo r.surf.contour	251
4.6.1.3. Conclusión.....	252
4.6.2. Estudio de la EMC (Evaluación Multicriterio) en un caso práctico	252
4.6.2.1. Lenguaje de las siglas y paleta de colores de los resultados.....	252
4.6.2.2. Listado de criterios y operaciones que intervienen	253
4.6.2.2.1. Reglas físicas.....	253
4.6.2.2.2. Reglas Operacionales.....	254
4.6.2.2.3. Restricciones de impacto medio ambiental	255
4.6.2.2.4. Regla de Decisión Final (OWA- Media Ponderada Ordenada).....	256
4.6.2.3. Esquema del caso práctico.....	257
4.6.2.4. Consideraciones previas	258
4.6.2.5. Estudio con valores nulos	259
4.6.2.5.1. Aplicación de criterios físicos.....	259
4.6.2.5.2. Aplicación de criterios operacionales de exclusión total.....	260
4.6.2.5.3. Aplicación de criterios operacionales de exclusión parcial.....	261
4.6.2.5.4. Aplicación de criterios medioambientales de exclusión parcial	263
4.6.2.5.5. Toma de decisión final.....	264
4.6.2.5.5.1. Riesgo máximo	264
4.6.2.5.5.2. Riesgo mínimo.....	265
4.6.2.5.5.3. Riesgo moderado mayor	265
4.6.2.5.5.4. Riesgo moderado menor.....	266
4.6.2.5.5.5. Sin riesgo	266
4.6.2.6. Estudio sin valores nulos.....	267
4.6.2.6.1. Aplicación de criterios físicos.....	267
4.6.2.6.2. Aplicación de criterios operacionales de exclusión total.....	268
4.6.2.6.3. Aplicación de criterios operacionales de exclusión parcial.....	269
4.6.2.6.4. Aplicación de criterios medioambientales de exclusión parcial	271
4.6.2.6.5. Toma de decisión final.....	272
4.6.2.6.5.1. Riesgo máximo	272
4.6.2.6.5.2. Riesgo mínimo.....	273
4.6.2.6.5.3. Riesgo moderado mayor	273
4.6.2.6.5.4. Riesgo moderado menor.....	274
4.6.2.6.5.5. Sin riesgo	274
4.6.2.7. Conclusiones	275
4.6.3. Importación de las capas informativas de la BBDD OceanLider a PostgreSQL 9.1 + PostGIS 2.0	275
4.6.3.1. Creación de la BBDD OceanLider en PostgreSQL	275
4.6.3.1.1. Creación de la base de datos utilizando la interfaz gráfica de pgAdminIII	275
4.6.3.1.2. Creación de la base de datos utilizando la consola de ejecución de sentencias SQL de pgAdminIII	278
4.6.3.2. Creación de la estructura de la BBDD de OceanLider con las capas informativas dentro de PostgreSQL.....	279
4.6.3.3. Poblar la BBDD de OceanLider en PostgreSQL.....	281
4.6.3.3.1. Importación de las capas vectoriales de una entidad.....	282
4.6.3.3.1.1. Problemática y resolución en la importación de Shapesfiles con acentos y caracteres especiales en el nombre	286
4.6.3.3.2. Importación de las capas raster de una entidad.....	290
4.6.3.4. Comprobación de resultados.....	291
4.6.3.4.1. Resumen de las capas vectoriales y raster de la BBDD OceanLider proporcionada por PROES	291

4.6.3.4.1.1. Península	291
4.6.3.4.1.2. Baleares	295
4.6.3.4.1.3. Canarias	296
4.6.3.4.2. Capas vectoriales y raster en la BBDDOceanLider ubicadas en PostgreSQL tras la importación	299
4.6.3.4.2.1. Comprobación de Península	299
4.6.3.4.2.2. Comprobación de Baleares	309
4.6.3.4.2.3. Comprobación de Canarias	311
4.6.3.4.3. Importación de las capas fallidas	318
4.6.3.5. Conclusión	319
4.6.4. Unión de capas raster y vectoriales de la BBDDOceanLider en la proyección WGS84	319
4.6.4.1. Contexto del contenido de la base de datos geográfica	319
4.6.4.2. Unión de capas raster y vectoriales de Península, Baleares y Canarias	320
4.6.4.2.1. Unión de capas vectoriales	321
4.6.4.2.2. Unión de capas raster	323
CAPÍTULO 5: VALIDACIÓN Y PRUEBAS.....	329
5.1. RESULTADO DE LAS OPERACIONES DE LOS CRITERIOS Y SUBCRITERIOS PROPUESTOS PARA LA HERRAMIENTA DSS	329
5.1.1.1. Criterios físicos	330
5.1.1.2. Criterios operacionales	337
5.1.1.3. Criterios recurso energético	344
5.1.1.4. Criterios medioambientales	347
5.2. RESULTADOS DE LA COMBINACIÓN DE CRITERIOS	353
5.2.1. Combinación de criterios físicos	353
5.2.1.1. Combinación 4	353
5.2.2. Combinación de criterios operacionales	356
5.2.2.1. Combinación 8	356
5.2.3. Combinación de criterios medioambientales	361
5.2.3.1. Combinación 10	361
5.2.4. Combinación de criterios físicos y criterios operacionales	364
5.2.4.1. Combinación 17	364
5.2.5. Combinación de criterios físicos y criterios de recurso energético	369
5.2.5.1. Combinación 18	369
5.2.6. Combinación de criterios físicos y criterios medioambientales	371
5.2.6.1. Combinación 20	371
5.2.7. Combinación de criterios físicos, criterios operacionales y criterios medioambientales	375
5.2.7.1. Combinación 21	375
5.2.8. Combinación de criterios físicos, criterios operacionales y criterios de recurso energético	383
5.2.8.1. Combinación 22	383
5.2.9. Combinación de criterios físicos, criterios de recurso energético y criterios medioambientales	387
5.2.9.1. Combinación 23	387
5.2.10. Combinación de criterios físicos, criterios operacionales, criterios medioambientales y criterios de recurso energético	393

5.2.10.1. Combinación 24	393
CAPÍTULO 6: MANUALES	401
6.1. MANUAL DE INSTALACIÓN.....	401
6.1.1. <i>Instalación de NetBeans (Glassfish)</i>	401
6.1.1.1. Importación del código fuente y configuración de parámetros en código fuente.....	402
6.1.2. <i>Instalación de GRASS</i>	402
6.1.2.1. Creación de MAPSET y LOCATION	402
6.1.2.2. Configuración del archivo .grassrc6	405
6.1.2.3. Configuración de variables de entorno de Windows	406
6.1.2.4. Posibles conflictos en el lanzamiento de operaciones fuera del entorno de GRASS	407
6.1.3. <i>Instalar y repoblar BBDD sobre PostgreSQL</i>	407
6.1.3.1. Instalar PostgreSQL 9.1 + PostGIS 2.0	407
6.1.3.2. Crear y repoblar la base de datos DSS_nueva.....	408
6.1.3.3. Crear y repoblar la base de datos BBDOceanLider	409
6.1.4. <i>Instalar directorios de la herramienta OceanLider</i>	410
6.2. MANUAL DE USUARIO	410
6.2.1. <i>Registrarse</i>	411
6.2.2. <i>Entrar en el sistema</i>	412
6.2.3. <i>Crear un nuevo escenario</i>	413
6.2.4. <i>Crear un nuevo dispositivo</i>	417
6.2.5. <i>Visualizar características dispositivo</i>	421
6.2.6. <i>Seleccionar área de estudio de un escenario</i>	423
6.2.7. <i>Cargar un escenario creado</i>	423
6.2.8. <i>Cambiar características de un escenario</i>	426
6.2.9. <i>Configuración de criterios</i>	428
6.2.10. <i>Configuración de la toma de decisión final</i>	431
6.2.11. <i>Ejecutar escenario</i>	434
6.2.12. <i>Visualización de resultados</i>	435
6.2.13. <i>Salir del sistema</i>	439
6.3. HERRAMIENTAS PARA EL DESARROLLADOR	439
6.3.1. <i>Configuración para la ejecución de comandos y scripts de GRASS en CMD de Windows</i>	439
6.3.2. <i>Importación de una tabla Excel en PostgreSQL</i>	443
6.3.2.1. Importación de una tabla Excel campo a campo	443
6.3.2.2. Alternativas teniendo en cuenta la variación en el tiempo de las tablas Excel.....	447
6.3.2.2.1. Modelo EAV	447
6.3.2.2.2. XML Y Xpath	448
6.3.2.3. Conclusiones	451
6.3.3. <i>Importar capas vectoriales y raster en PostgreSQL 9.1 + PostGIS 2.0</i>	452
6.3.3.1. Importar capas raster	452
6.3.3.2. Importar capas vectoriales.....	452
CAPÍTULO 7: CONCLUSIONES	457
BIBLIOGRAFÍA	459
APÉNDICE A: CONTENIDO DEL CD-ROM.....	463

APÉNDICE B: DICCIONARIO	465
APÉNDICE C: TOMA DE DECISIÓN	471
1. TIPOS DE REGLAS DE DECISIÓN	471
1.1. <i>Toma de decisión mediante procedimiento de Evaluación MultiCriterio.</i>	471
1.1.1. Combinación lineal ponderada, WLC	474
1.1.2. Media ponderada ordenada (OWA: Ordered Weighted Average)	474
1.1.3. Otros métodos EMC	475
1.2. <i>Toma de Decisión mediante Evaluación MultiCriterio y su combinación con Sistemas de Información Geográfica.</i>	476
APÉNDICE D: SCRIPTS OPERACIONALES DE LOS CASOS PRÁCTICOS	481
1. SCRIPT OPERACIONAL DE CASO PRÁCTICO DE ESTUDIO CON VALORES NULOS	481
2. SCRIPT OPERACIONAL DE CASO PRÁCTICO DE ESTUDIO SIN VALORES NULOS	487
APÉNDICE E: FICHERO SQL DE LA UNIÓN DE DIFERENTES CAPAS VECTORIALES INFORMATIVAS	495
APÉNDICE F: FICHERO SQL DE LA UNIÓN DE DIFERENTES CAPAS VECTORIALES DE RESTRICCIÓN	511
APÉNDICE G: FICHERO .BAT DE LA UNIÓN DE CAPAS RASTER INFORMATIVAS	541
APÉNDICE H: INSTALACIÓN DEL COMPONENTE XMTTOOLS DE EXCEL Y PASOS PARA LA TRANSFORMACIÓN DE UNA TABLA EXCEL A XML	547
APÉNDICE I: VALIDACIÓN Y PRUEBAS COMPLETA	553
1. RESULTADO DE LAS OPERACIONES DE LOS CRITERIOS Y SUBCRITERIOS PROPUESTOS PARA LA HERRAMIENTA DSS	553
1.1. <i>Criterios físicos</i>	554
1.2. <i>Criterios operacionales</i>	568
1.3. <i>Criterios recurso energético</i>	602
1.4. <i>Criterios medioambientales</i>	610
2. RESULTADOS DE LA COMBINACIÓN DE CRITERIOS	677
2.1. <i>Combinación de criterios físicos</i>	678
2.1.1. Combinación 1	678
2.1.2. Combinación 2	686
2.1.3. Combinación 3	692
2.1.4. Combinación 4	702
2.2. <i>Combinación de criterios operacionales</i>	710
2.2.1. Combinación 5	710
2.2.2. Combinación 6	713
2.2.3. Combinación 7	718
2.2.4. Combinación 8	722
2.3. <i>Combinación de criterios medioambientales</i>	737
2.3.1. Combinación 9	737
2.3.2. Combinación 10	760
2.3.3. Combinación 11	763
2.3.4. Combinación 12	775

2.3.5. Combinación 13	778
2.3.6. Combinación 14	793
2.3.7. Combinación 15	796
2.4. <i>Combinación de criterios físicos y criterios operacionales</i>	807
2.4.1. Combinación 16	807
2.4.2. Combinación 17	810
2.5. <i>Combinación de criterios físicos y criterios de recurso energético</i>	815
2.5.1. Combinación 18	815
2.5.2. Combinación 19	817
2.6. <i>Combinación de criterios físicos y criterios medioambientales</i>	820
2.6.1. Combinación 20	820
2.7. <i>Combinación de criterios físicos, criterios operacionales y criterios medioambientales</i>	824
2.7.1. Combinación 21	824
2.8. <i>Combinación de criterios físicos, criterios operacionales y criterios de recurso energético</i>	833
2.8.1. Combinación 22	833
2.9. <i>Combinación de criterios físicos, criterios de recurso energético y criterios medioambientales</i>	838
2.9.1. Combinación 23	838
2.10. <i>Combinación de criterios físicos, criterios operacionales, criterios medioambientales y criterios de recurso energético</i>	845
2.10.1. Combinación 24	845

ÍNDICE DE FIGURAS

Figura 1: Esquema del proceso.....	29
Figura 2: Función de operación difusa lineal.....	31
Figura 3: Valores de idoneidad configurados para cada agrupación por defecto para cada tipo de dispositivo.	33
Figura 4: : Valores de idoneidad por defecto definida para cada agrupación y tipo de dispositivo	33
Figura 5: Función de operación difusa lineal normal, de subida y bajada.....	34
Figura 6: Secuencia modular del protocolo de aplicación de criterios	58
Figura 7: Empleo de las operaciones de los módulos del protocolo de aplicación de los criterios	59
Figura 8: Diagrama Entidad-Relación de la base de datos de la herramienta DSS	65
Figura 9: Modelo estático de la base de datos de la herramienta DSS	78
Figura 10: Diagrama E-R con el modelo dinámico resaltado	81
Figura 11: Aplicación difusa (fuzzy) con una función lineal.....	87
Figura 12: Aplicación difusa (fuzzy) con una función lineal con rampa de subida y de bajada... ..	87
Figura 13: Fórmula operación OWA.....	88
Figura 14: Tabla de operaciones configuradas en el sistema.....	100
Figura 15: Tabla de programación del criterio distancia a la costa.....	101
Figura 16: Tabla de programación del criterio profundidad.....	102
Figura 17: Tabla de programación del criterio Altura de ola significativa	102
Figura 18: Tabla de programación del criterio Velocidad de corriente	103
Figura 19: Tabla de programación del criterio Facilidad de anclaje al fondo (composición del sustrato marino).....	103
Figura 20: Tabla de agrupaciones de la capa vectorial de accidentes geográficos (Formas).....	104
Figura 21: Tabla de programación del criterio Facilidad de anclaje al fondo (morfología del fondo marino).....	104
Figura 22: Tabla de programación del criterio Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español	105
Figura 23: Tabla de programación del criterio Viabilidad de instalación en zonas designadas como Dominio Público Portuario	106
Figura 24: Tabla de programación del criterio Viabilidad de instalación en zonas preseleccionadas para la extracción de áridos	106
Figura 25: Tabla de programación del criterio Viabilidad de instalación en zonas ocupadas por conducciones submarinas	107
Figura 26: Tabla de programación del criterio Viabilidad de instalación en zonas preseleccionadas para el vertido de material de dragado.....	107
Figura 27: Tabla de programación del criterio Viabilidad de instalación en zonas ocupadas por sensores instrumentales de medición de parámetros oceanográficos	108
Figura 28: Tabla de programación del criterio Viabilidad de instalación en zonas ocupadas por dispositivos de separación del tráfico marítimo.....	108
Figura 29: Tabla de programación del criterio Viabilidad de instalación en zonas restringidas para uso militar.....	109

Figura 30: Tabla de programación del criterio Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías	109
Figura 31: Tabla de programación del criterio Viabilidad de instalación en zonas restringidas por la presencia de vertidos.....	110
Figura 32: Tabla de programación del criterio Viabilidad de instalación en zonas restringidas por la existencia de vertidos de material militar.....	110
Figura 33: Tabla de programación del criterio mareas vivas y muertas en condiciones de marea media	111
Figura 34: Tabla de programación del criterio densidad.....	112
Figura 35: Tabla de programación del criterio Geoparques.....	113
Figura 36: Tabla de programación del criterio Reservas de la biosfera.....	113
Figura 37: Tabla de programación del criterio Patrimonio de la humanidad.....	114
Figura 38: Tabla de programación del criterio Red de Áreas Marinas Protegidas del Convenio OSPAR	114
Figura 39: Tabla de programación del criterio Zonas Especialmente Protegidas de Importancia para el Mediterráneo.....	115
Figura 40: Tabla de programación del criterio Sitios reconocidos con el Diploma Europeo de Calidad de Espacios Protegidos.....	116
Figura 41: Tabla de programación del criterio Lugares de Importancia Comunitaria	116
Figura 42: Tabla de programación del criterio Zonas de Especial Protección para las Aves	117
Figura 43: Tabla de programación del criterio Parques Nacionales	117
Figura 44: Tabla de programación del criterio Biotopos Protegidos	118
Figura 45: Tabla de programación del criterio Microrreservas.....	118
Figura 46: Tabla de programación para el criterio Monumentos Naturales.....	119
Figura 47: Tabla de programación para el criterio Parajes Naturales	120
Figura 48: Tabla de programación para el criterio Parques Naturales	120
Figura 49: Tabla de programación para el criterio Reservas con ley de protección propia	121
Figura 50: Tabla de programación para el criterio Reservas Naturales	121
Figura 51: Tabla de programación para el criterio Zonas de Especial Protección de los Valores Naturales	122
Figura 52: Tabla de programación para el criterio Áreas de interés para la conservación de los cetáceos.....	122
Figura 53: Tabla de programación para el criterio Áreas compatibles con la figura de Parque Nacional.....	123
Figura 54: Tabla de programación para el criterio Áreas importantes para las aves.....	124
Figura 55: Tabla de programación para el criterio Zonas propuestas para realizar el Inventario de la biodiversidad marina.....	124
Figura 56: Tabla de programación para el criterio Áreas importantes para las aves marinas	125
Figura 57: Tabla de programación para el criterio Zona marina especialmente sensible	125
Figura 58: Tabla de programación para el criterio Rutas migratorias de cetáceos.....	126
Figura 59: Tabla de programación para el criterio de Instalaciones de acuicultura	127
Figura 60: Tabla de programación para el criterio de Almadrabas	127
Figura 61: Tabla de programación para el criterio de Arrecifes	128
Figura 62: Tabla de programación para el criterio Caladeros.....	128
Figura 63: Tabla de programación para el criterio Viveros	129
Figura 64: Tabla de programación para el criterio Zonas protegidas de interés pesquero	130
Figura 65: Tabla de programación para el criterio Piscifactorías	130
Figura 66: Tabla de programación para el criterio Reservas marinas	131

Figura 67: Tabla de programación para el criterio Zonas de producción de moluscos y de otros invertebrados marinos.....	131
Figura 68: Tabla de roles y responsabilidades del equipo de desarrollo.....	138
Figura 69: Tabla del plan de fases	139
Figura 70: Diagrama de Gantt del plan de fases	139
Figura 71: Tabla de versiones.....	139
Figura 72: Plan calerandizado de la primera iteración.....	146
Figura 73: Diagrama de Gantt de la fase de inicio.....	146
Figura 74: Plan calerandizado de la primera iteración de la fase de elaboración	150
Figura 75: Diagrama de Gantt de la primera iteración de la fase de elaboración	151
Figura 76: Plan calerandizado de la primera iteración de la fase de construcción.....	153
Figura 77: Diagrama de Gantt de la primera iteración de la fase de construcción	154
Figura 78: Plan calerandizado de la segunda iteración de la fase de construcción	156
Figura 79: Diagrama de Gantt de la segunda iteración de la fase de construcción.....	157
Figura 80: Plan calerandizado de la tercera iteración de la fase de construcción	159
Figura 81: Diagrama de Gantt de la tercera iteración de la fase de construcción	160
Figura 82: Plan calerandizado de la primera iteración de la fase de transición	162
Figura 83: Diagrama de Gantt de la primera iteración de la fase de transición	162
Figura 84: Diagrama casos de uso de análisis	171
Figura 85: Diagrama de clases de análisis de la herramienta DSS	173
Figura 86: Casos de uso generales de diseño.....	175
Figura 87: Caso de uso de diseño - Autenticación usuario-.....	175
Figura 88: Caso de uso de diseño - Crear escenario-.....	176
Figura 89: Caso de uso de diseño- Cargar escenario	176
Figura 90: Caso de uso de diseño- Realizar Cálculos-	177
Figura 91: Caso de uso de diseño- Visualizar resultados-	177
Figura 92: Componentes del sistema web y el sistema operacional	181
Figura 93: Componente DAOpattern del diagrama de clases de diseño.....	182
Figura 94: Componente Entities del diagrama de clases de diseño	183
Figura 95: Componente Motor del diagrama de clases de diseño	193
Figura 96: Componente Motor del diagrama de clases de diseño antes de la corrección	197
Figura 97: Componente interfaz del diagrama de clases de diseño	198
Figura 98: Diagrama de secuencia de diseño del CU Autenticación usuario	200
Figura 99: Diagrama de secuencia de diseño del CU Crear escenario	201
Figura 100: Diagrama de secuencia de diseño del CU Cargar escenario.....	201
Figura 101: Diagrama de secuencia de diseño del CU Realizar cálculos	202
Figura 102: Diagrama de secuencia de diseño del CU Visualizar resultados	202
Figura 103: Diagrama de secuencia de diseño del CU Modificar y/o seleccionar criterios escenario	203
Figura 104: Diagrama de secuencia de diseño del CU Hacer login.....	203
Figura 105: Diagrama de secuencia de diseño del CU Registrarse	204
Figura 106: Diagrama de secuencia de diseño del CU Salir de sesión	204
Figura 107: Diagrama de secuencia de diseño del CU Seleccionar dispositivo	205
Figura 108: Diagrama de secuencia de diseño del CU Seleccionar zona geográfica.....	205
Figura 109: Diagrama de secuencia de diseño del CU Introducir parámetros escenario.....	206
Figura 110: Diagrama de secuencia de diseño del CU Modificar y/o seleccionar criterios escenario	206
Figura 111: Diagrama de secuencia de diseño Guardar escenario.....	207

Figura 112: Diagrama de secuencia de diseño del CU Selección escenario	207
Figura 113: Diagrama de secuencia de diseño del CU Modificar dispositivo	208
Figura 114: Diagrama de secuencia de diseño del CU Modificar zona geográfica	208
Figura 115: Diagrama de secuencia de diseño del CU Modificar parámetros escenario	209
Figura 116: Diagrama de secuencia de diseño del CU Modificar y/o seleccionar criterios escenario	209
Figura 117: Diagrama de secuencia de diseño del CU Extraer información escenario.....	210
Figura 118: Diagrama de secuencia de diseño del CU Traducir/Implementar criterios a operaciones	211
Figura 119: Diagrama de secuencia de diseño del CU Realización script ejecutable.....	211
Figura 120: Diagrama de secuencia de diseño del CU Ejecución de script	212
Figura 121: Diagrama de secuencia de diseño del CU Modificación estado escenario	212
Figura 122: Diagrama de secuencia de diseño del CU Mostrar información resultados intermedios y finales.....	213
Figura 123: Diagrama de secuencia de diseño del CU Visualizar resultados	213
Figura 124: Esquema de la arquitectura del DSS.....	215
Figura 125: Esquema de aplicación del patrón Modelo Vista Controlador	217
Figura 126: Esquema de aplicación MVC al DSS para la configuración de escenario.....	218
Figura 127: Esquema de aplicación MVC al DSS para la visualización de resultados.....	219
Figura 128: Vista de la interfaz DSS dividida en contenedores.....	221
Figura 129: Diseño de la página de entrada de la interfaz DSS	222
Figura 130: Diseño de la página de bienvenida de la interfaz DSS	223
Figura 131: Diseño de la página para la selección del escenario de la interfaz DSS	224
Figura 132: Diseño de la página para la configuración de criterios del escenario de la interfaz DSS.....	225
Figura 133: Diseño de la página para la consulta de información de los resultados de un escenario de la interfaz DSS	226
Figura 134: Diseño de la página para la visualización del resultado de un escenario de la interfaz DSS	226
Figura 135: Diagrama de la arquitectura de MapServer [Hexacta, 2013].....	232
Figura 136: Ejemplo de la estructura de un MapFile (.MAP) [MapServer d, 2013].....	234
Figura 137: Parte Layer de un MapFile (.MAP) para definir una capa vectorial ubicada en el sistema de ficheros [MapServer d, 2013].....	235
Figura 138: Parte Layer de un MapFile (.MAP) para definir una capa vectorial ubicada en PostGIS.....	235
Figura 139: Parte Layer de un MapFile (.MAP) para definir una capa raster ubicada en el sistema de ficheros.....	236
Figura 140: Parte Layer de un MapFile (.MAP) para definir una capa raster con reclasificación de valores.....	237
Figura 141: Imagen de la capa vectorial de Batimetría perteneciente a la Península	243
Figura 142: Ventana de dialogo de Sextante para la operación de rasterización	244
Figura 143: Resultado de la rasterización de la operación de rasterización de Sextante en GvSIG	245
Figura 144: Resultado de la operación de rellenar celdas sin datos de Sextante a partir de Batimetría con tensión 0.1	246
Figura 145: Resultado de la operación rellenar celas sin datos por vecindad de Sextante a partir de la Batimetría.....	247
Figura 146: Resultado de Prof_idw en escala de grises con Línea de Costa de referencia.....	248

Figura 147: Resultado de Prof_idw.250 en escala de grises con Línea de Costa de referencia.. 249

Figura 148: Resultado de Prof_rst en escala de grises con Línea de Costa de referencia..... 251

Figura 149: Paleta de colores para el estudio de la EMC en un caso práctico..... 253

Figura 150: Esquema de operación del caso práctico 257

Figura 151: Capa raster de entrada Altura media de ola significativa 260

Figura 152: Capa raster de entrada de potencia del viento 261

Figura 153: Creación de una nueva base de datos utilizando la interfaz gráfica de pgAdminIII 276

Figura 154: Pestaña de Propiedades en el cuadro de dialogo para la creación de una nueva base de datos utilizando la interfaz gráfica de pgAdminIII 277

Figura 155: Pestaña de definición en el cuadro de dialogo para la creación de una nueva base de datos utilizando la interfaz gráfica de pgAdminIII..... 278

Figura 156: Icono de la consola de ejecución de sentencias SQL de pgAdminIII..... 278

Figura 157: Creación de un nuevo esquema de la base de datos a través de la interfaz de pgAdminIII 280

Figura 158: Cuadro de dialogo en la creación de un nuevo esquema de la base de datos utilizando la interfaz gráfica de pgAdminIII 281

Figura 159: Cuadro de dialogo de QuantumGIS para la importación de Shapes a PostgreSQL 287

Figura 160: Error de QGIS en la importación de shape con objetos espaciales tipo MULTILINESTRING 287

Figura 161: Paleta de colores 1..... 329

Figura 162: Modificación del nombre del directorio DATABASE de GRASS 403

Figura 163: Ventana principal de GRASS TclTk GUI..... 403

Figura 164: Ventana de creación de una localización de GRASS con código de proyección EPSG 404

Figura 165: Ventana de GRASS TclTk GUI para la creación de un MAPSET..... 405

Figura 166: Ventana de configuración de variables de entorno de Windows..... 406

Figura 167: Creación de una nueva base de datos utilizando la interfaz gráfica de pgAdminIII 408

Figura 168: Ventana de variables de entorno de Windows 7 440

Figura 169: Ejemplo de script operacional con comandos de GRASS en formato .BAT para Windows..... 442

Figura 170: Cuadro de dialogo de ‘Guardar como’ de Microsoft Excel para seleccionar tipo de fichero..... 444

Figura 171: Ejemplo de una tabla Excel..... 445

Figura 172: Resultado de la importación de una tabla Excel a PostgreSQL..... 445

Figura 173: Resultado al poblar una tabla con un campo numérico 446

Figura 174: Aplicación del modelo EAV sobre el modelo relacional de BBDD [Balteus b, 2012] 448

Figura 175: Importación de capas vectoriales tipo Shape a PostgreSQL desde QuantumGIS ... 453

Figura 176: Cuadro de dialogo para la conexión a PostgreSQL..... 454

Figura 177: Cuadro de dialogo principal para la importación de shapes a PostreSQL en QGIS 455

Figura 178: Cuadro de dialogo de ogr2gui para la importación de una capa vectorial tipo Shape a PostgreSQL..... 456

Figura 179: OWA: Espacio de estrategias de decisión [Jiang e Eastman 2000]..... 475

Figura 180: Sistema de integración entre SIG y EMC para un único objetivo [Gómez y Barredo, 2005]..... 478

Figura 181: Esquema general del proceso de integración de las diversas fases que componen el procedimiento multicriterio/multiobjetivo en un SIG ráster 479

Figura 182: Cuadro de dialogo de Opciones de Microsoft Excel para la instalación de complementos 548

Figura 183: Cuadro de dialogo de Complementos Excel disponibles..... 549

Figura 184: Menú de comandos de Microsoft Excel 549

Figura 185: Error de compilación del complemento XmlTools 550

Figura 186: Cuadro de dialogo de Opciones de Microsoft Excel para activar la pestaña 'Programador' 551

Figura 187: Cuadro de dialogo para almacenar una tabla Excel como archivo XML en Microsoft Excel 552

Figura 188: Paleta de colores 2..... 554

Capítulo

1

INTRODUCCIÓN

1.1 Introducción

El presente trabajo está incluido en el marco del proyecto *CENIT OceanLider* (líderes en energías renovables oceánicas) y es continuación del proyecto fin de carrera de la Ingeniería Técnica en Informática de Gestión con título “*Estudio de herramientas SIG de libre distribución para la selección de emplazamientos de unidades de generación de energía oceánica*” que fue elaborado por la autora y los tutores José Belarmino Pulido Junquera y Aníbal Bregón Bregón. Ha sido realizado como cometido de contrato de beca, contrato profesional y contrato de prácticas en empresa en el Grupo de Sistemas Inteligentes (GSI) del Departamento de Informática de la Universidad de Valladolid y la empresa GMV Sistemas de Valladolid. Por tanto esta memoria es el resumen del trabajo realizado durante año y medio.

En el pasado proyecto se estudiaron diferentes Sistemas de Información Geográficos (SIG) para gestionar, analizar y operar información espacial con el objetivo de crear un **Sistema Inteligente para el Apoyo a la Toma de Decisión (DSS**, por sus siglas en inglés *Decision Support System*) y se construyó un prototipo o prueba de concepto. En el actual trabajo se ha realiza el análisis, diseño, implementación y puesta en marcha del DSS.

El objetivo de la aplicación es proporcionar al usuario los mejores emplazamientos para la ubicación de distintos tipos de Unidades Integradas de Generación de Energía eléctrica en medio Oceánico (UNIGEOs) teniendo en cuenta las restricciones estructurales, geográficas, físicas, legales, sociales y económicas que se disponga en el momento de realizar la toma de decisión.

Mediante la aplicación de diferentes criterios y subcriterios sobre el escenario configurado por el usuario, el sistema experto determinará el conjunto de zonas óptimas y condicionantes para la instalación de los parques de generación de energía en medio oceánico. Para tal objetivo se usa un modelo basado en **técnicas de análisis multicriterio** dentro del entorno de los SIG y del tratamiento digital de imágenes.

Se aplica el análisis multicriterio que es, posiblemente, la técnica de toma de decisiones más utilizada en los SIG en la que permite poder combinar múltiples criterios y objetivos contradictorios.

Para su desarrollo y uso se ha determinado emplear diferentes operaciones procedentes de **GRASS** que es un SIG de libre distribución. Para el tratamiento de criterios que lleven asociado una lógica booleana se emplean operaciones de multiplicación y superposición (entre otras) como una operación estándar en cualquier sistema SIG. Para el tratamiento y combinación de criterios que lleven asociado una lógica borrosa se implementa y aplica operaciones fuzzy no tan comunes en los SIG, y la operación Media Lineal Ponderada para la combinación de varios. La **Media Lineal Ponderada** multiplica cada factor por su peso y suma posteriormente los resultados, los cuales son posteriormente multiplicados por las áreas de exclusión, imágenes booleanas.

Para la toma de decisión final, es necesario aplicar otra regla de decisión adicional utilizando la operación **Media Ponderada Ordenada**, pues permite seleccionar el grado de compromiso y de riesgo mediante la selección de los pesos de orden.

También ha sido necesario el estudio de diferentes sistemas gestores de información que dieran soporte geoespacial. El sistema gestor de información espacial de libre distribución que se ha utilizado para almacenar las capas vectoriales disponibles ha sido **PostGIS**.

De esta manera, se ha conseguido una herramienta capaz de evaluar y determinar las zonas óptimas y condicionantes de los emplazamientos según los criterios y factores configurados por el usuario, siendo capaz de generar y procesar de forma automática operaciones SIG obteniendo los resultados procedentes de una evaluación multicriterio.

1.2 Objetivos

El objetivo principal de este proyecto será el desarrollo y puesta en marcha de un Sistema Inteligente para el Apoyo a la Toma de Decisión. Una vez definidas las técnicas y herramientas que intervengan en su realización, se procederá a analizar, diseñar e implementar la arquitectura del DSS. Más concretamente, en este Trabajo Fin de Grado se pretenden alcanzar los siguientes objetivos:

- Redefinir la arquitectura del prototipo DSS conforme a los requisitos de los clientes.
- Ampliación de las funcionalidades.
- Generación automática de scripts operacionales.
- Implementación basada en web (nuevas tecnologías).

Para la realización de estos objetivos se plantean las siguientes tareas:

- Se hará la instalación e integración de los entornos necesarios para desarrollar este proyecto, como son: GRASS, PostgreSQL y PostGIS.
- Estudio de la base de datos proporcionada en el proyecto *OceanLider* y su almacenaje en una base de datos espacial.
- Se generarán los mapas necesarios que no estén contemplados en la base de datos.
- Fijación de los criterios físicos, operacionales, medioambientales y de recurso energético con las entidades vinculadas al proyecto.

- Se flexibilizará el conjunto de criterios disponibles para el usuario como las reglas de decisión.
- Programación de los criterios con operaciones con el SIG elegido (GRASS).
- Implementación de operaciones SIG para dar cabida a técnicas de evaluación multicriterio.
- Mejorar la interfaz tanto a la hora de introducir datos como de visualizar resultados, adaptándola a la tecnología web.
- Se definirá, analizará, diseñará, implementará, pondrá en marcha y probará la herramienta DSS.

1.3 Estructura de la memoria

La memoria está estructurada en 7 capítulos, junto los apéndices del final de la memoria y la bibliografía. Los capítulos están distribuidos de la siguiente manera:

- **Introducción:** se presentan los objetivos perseguidos con la realización de este proyecto Fin de Grado, así como una descripción de la estructura de esta memoria.
- **Arquitectura DSS:** contiene el estudio sobre la arquitectura que seguirá el Sistema Inteligente para el Apoyo a la Toma de Decisión y se definirán los aspectos esenciales de la herramienta.
- **Generación automática de scripts:** se describe la base donde se encuentra la implementación de los criterios para la programación automática de scripts.
- **Desarrollo de la herramienta:** en este capítulo se describe el plan de desarrollo y seguimiento, el proceso de análisis y diseño del proyecto según la metodología UPEDU¹, las funcionalidades más relevantes de la herramienta DSS y una descripción de las operaciones intermedias que han sido necesarias para su construcción.
- **Validación y pruebas:** se describe el conjunto de pruebas realizado para la verificación del correcto funcionamiento de la herramienta DSS. Se muestran los resultados de las pruebas realizadas para cada criterio y subcriterio fijado por los expertos, y la combinación de los mismos.
- **Manuales:** este capítulo contiene diferentes manuales destinados a los usuarios y desarrolladores: manual de instalación, manual de usuario y diferentes manuales para los desarrolladores.
- **Conclusiones:** contiene las conclusiones finales de este proyecto.
- **Bibliografía:** contiene las referencias bibliográficas empleadas en este proyecto.
- **Apéndices:** se incluye un diccionario de todos los acrónimos que han aparecido en la memoria de este Proyecto Fin de Grado, el contenido del CDROM, diferentes resultados de los scripts de unión de capas raster y vectoriales, y un manual de la instalación del componente XmlTools.

¹UPEDU o UnifiedProcessforEducation: Es un proceso de desarrollo software especializado para la educación.

AQUITECTURA DEL DSS

2.1. Introducción

En este apartado se describe el funcionamiento del DSS, los datos de entrada, los tipos de reglas de decisión para realizar la toma de decisión, los criterios fijados para la herramienta y el protocolo de aplicación de los criterios.

2.2. Proceso

Se definen los pasos a seguir en el funcionamiento del DSS. Posteriormente se ilustra de forma gráfica en la *Figura 1: Esquema del proceso*.

Paso 1: Cuadro de diálogo para definir el escenario.

En primer lugar se proporcionará los datos precisos para crear un escenario. Se elige un dispositivo de los que se han creado y que está ligado a un tipo de dispositivo (eólico, mareomotriz o undimotriz) y varios parámetros, o en su defecto, se crea uno nuevo introduciendo los datos de las características que le atañen. Dependiendo de la elección, se tendrá distintas restricciones estructurales y relaciones con otros usos del mar o actividades.

Una vez seleccionado el dispositivo, de forma visual se elegirá la zona geográfica seleccionando una región del mapa de la costa y/o mar española. Después, se le irán presentando al usuario diferentes apartados (descritos en los siguientes pasos) donde debe elegir los criterios y valores concretos que se usarán en los cálculos, y que pueden depender del tipo de dispositivo elegido (en cualquier caso, se le ofrecerá un valor por defecto que podrá mantener).

De esta manera quedará definido el escenario y se podrá proceder a la consulta. Es en este instante donde el usuario dejará de interactuar con la interfaz de la aplicación. Una vez que el sistema tenga los resultados de la consulta, estarán los resultados a disposición para proceder a su visualización.

Paso 2: Criterios físicos

Son los que delimitan las zonas de interés atendiendo principalmente a los límites operativos de los dispositivos. En este módulo se restringe aquellas regiones en las que los UNIGEOs no pueden trabajar, debido a los requisitos de funcionamiento de los mismos.

Las restricciones físicas se expresan mediante un conjunto de criterios de exclusión total y parcial que dependen del tipo de dispositivo elegido. La interfaz muestra al usuario los criterios en uso. Los criterios totalmente excluyentes se definen mediante reglas booleanas, o reglas hard, que identifican zonas aptas y no aptas. Las zonas no aptas se eliminan del escenario.

Los criterios parcialmente excluyentes se definen mediante reglas fuzzy que identifican zonas aptas, no-aptas y negociables. Cada criterio asigna un valor entre 0 y 1 a las zonas negociables, 0 indicando zona no apta y 1 zona apta.

Se le da la libertad al usuario de escoger el método de razonamiento para la combinación de los criterios parciales, y también, de la combinación de subcriterios dentro de un mismo criterio. Los métodos de razonamiento disponibles son: método **más que pesimista** (multiplicación de los valores), **pesimista** (mínimo de los valores), **optimista** (máximo de los valores) y **neutro** (MLP).

Si los expertos determinan un método de razonamiento concreto para la combinación de criterios parciales o para la combinación de subcriterios dentro de un criterio, se podrá fijar siendo transparente para el usuario.

Además, la herramienta proporcionará al usuario información adicional sobre los criterios que aparecen y podrá parametrizarlos en caso que se requiera. Si elige el método de razonamiento MLP para la combinación de criterios parciales o de los subcriterios dentro de un mismo criterio, el usuario puede definir la importancia o peso que se quiere dar a cada uno asignándole un valor entre 0 y 1. Un valor 0 indica una importancia mínima, mientras que un valor 1 indica una importancia máxima. Se fuerza que la suma de los pesos sea 1.

Paso 3: Criterios operacionales

Se incluyen en este módulo aquellos criterios que no impiden el funcionamiento de los dispositivos, pero que restringen la instalación de los mismos. No se consideran operacionales los criterios medioambientales ni los criterios de recurso energético.

Los criterios operacionales pueden ser total o parcialmente excluyentes y dependen del tipo de dispositivo elegido. La interfaz muestra al usuario los criterios en uso.

Los criterios totalmente excluyentes se definen mediante reglas booleanas, o reglas hard, que identifican zonas aptas y no aptas. Las zonas no aptas se eliminan del escenario.

Los criterios parcialmente excluyentes se definen mediante reglas fuzzy que identifican zonas aptas, no-aptas y negociables. Cada criterio asigna un valor entre 0 y 1 a las zonas negociables, 0 indicando zona no apta y 1 zona apta.

Se le da la libertad al usuario de escoger el método de razonamiento para la combinación de los criterios parciales, y también, de la combinación de subcriterios dentro de un mismo criterio. Los

métodos de razonamiento disponibles son: método **más que pesimista** (multiplicación de los valores), **pesimista** (mínimo de los valores), **optimista** (máximo de los valores) y **neutro** (MLP).

Si los expertos determinan un método de razonamiento concreto para la combinación de criterios parciales o para la combinación de subcriterios dentro de un criterio, se podrá fijar siendo transparente para el usuario.

Además, la herramienta proporcionará al usuario información adicional sobre los criterios que aparecen y podrá parametrizarlos en caso que se requiera. Si elige el método de razonamiento MLP para la combinación de criterios parciales o de los subcriterios dentro de un mismo criterio, el usuario puede definir la importancia o peso que se quiere dar a cada uno asignándole un valor entre 0 y 1. Un valor 0 indica una importancia mínima, mientras que un valor 1 indica una importancia máxima. Se fuerza que la suma de los pesos sea 1.

Paso 4: Criterios medioambientales

Se incluyen en este módulo los criterios relacionados con la evaluación ambiental.

Los criterios medioambientales pueden ser de exclusión total o de exclusión parcial, y dependen del tipo de dispositivo elegido. La interfaz muestra al usuario los criterios en uso.

Los criterios de exclusión total identifican regiones no aptas que son eliminadas del escenario. Los criterios parcialmente excluyentes identifican zonas aptas, no aptas y negociables. Cada criterio asigna un valor entre 0 y 1 a las zonas negociables, 0 indicando zona no apta y 1 zona apta.

Se le da la libertad al usuario de escoger el método de razonamiento para la combinación de los criterios parciales, y también, de la combinación de subcriterios dentro de un mismo criterio. Los métodos de razonamiento disponibles son: método **más que pesimista** (multiplicación de los valores), **pesimista** (mínimo de los valores), **optimista** (máximo de los valores) y **neutro** (MLP).

Si los expertos determinan un método de razonamiento concreto para la combinación de criterios parciales o para la combinación de subcriterios dentro de un criterio, se podrá fijar siendo transparente para el usuario.

Además, la herramienta proporcionará al usuario información adicional sobre los criterios que aparecen y podrá parametrizarlos en caso que se requiera. Si elige el método de razonamiento MLP para la combinación de criterios parciales o de los subcriterios dentro de un mismo criterio, el usuario puede definir la importancia o peso que se quiere dar a cada uno asignándole un valor entre 0 y 1. Un valor 0 indica una importancia mínima, mientras que un valor 1 indica una importancia máxima. Se fuerza que la suma de los pesos sea 1.

Paso 5: Criterios de recurso energético

Este módulo elimina del escenario aquellas regiones en las que el dispositivo no puede trabajar cuando el recurso energético de la zona de estudio no es favorable.

Los criterios de recurso energético pueden ser de exclusión total o parcial, y dependen del tipo de dispositivo elegido. La interfaz muestra al usuario los criterios en uso.

Los criterios totalmente excluyentes se definen mediante reglas booleanas, o reglas hard, que identifican zonas aptas y no aptas. Las zonas no aptas se eliminan del escenario.

Los criterios parcialmente excluyentes se definen mediante reglas fuzzy o fórmulas consensuadas por las entidades vinculadas al proyecto que identifican zonas aptas, no-aptas y negociables. Cada criterio asigna un valor entre 0 y 1 a las zonas negociables, 0 indicando zona no apta y 1 zona apta.

Se le da la libertad al usuario de escoger el método de razonamiento para la combinación de los criterios parciales, y también, de la combinación de subcriterios dentro de un mismo criterio. Los métodos de razonamiento disponibles son: método **más que pesimista** (multiplicación de los valores), **pesimista** (mínimo de los valores), **optimista** (máximo de los valores) y **neutro** (Media Lineal Ponderada).

Si los expertos determinan un método de razonamiento concreto para la combinación de criterios parciales o para la combinación de subcriterios dentro de un criterio, se podrá fijar siendo transparente para el usuario.

Además, la herramienta proporcionará al usuario información adicional sobre los criterios que aparecen y podrá parametrizarlos en caso que se requiera. Si elige el método de razonamiento MLP para la combinación de criterios parciales o de los subcriterios dentro de un mismo criterio, el usuario puede definir la importancia o peso que se quiere dar a cada uno asignándole un valor entre 0 y 1. Un valor 0 indica una importancia mínima, mientras que un valor 1 indica una importancia máxima. Se fuerza que la suma de los pesos sea 1.

Paso 6: Toma de decisión

Este módulo es el responsable de la aplicación de todos los criterios definidos en los módulos anteriores. Se hará la combinación de la salida procedente de la rama de criterios operacionales, la rama de criterios medioambientales y la rama de criterios de recurso energético. Para hacer la combinación de estas tres salidas se aplica la **Media Ponderada Ordenada (MPO)**, que permite establecer los grados de compromiso y riesgo en la toma de decisión final.

La aplicación de la regla de decisión final (en este caso MPO), genera una capa raster de resultado. Este mapa indica si las zonas son aptas, no aptas o negociables, considerando todos los criterios configurados y ponderando los criterios operacionales, medioambientales y de recurso energético. Cada rejilla de la capa tendrá asociado un número real en el intervalo [0,1], significando 0 zona no apta, 1 zona apta y valores intermedios para zonas negociables.

Paso 7: Resultados

Este módulo de resultados permite mostrar todas las capas generadas en los diferentes módulos, permitiendo que el usuario pueda observar el efecto de la aplicación de cada criterio o subcriterio como resultado intermedio o la capa resultado de la toma de decisión como resultado final.

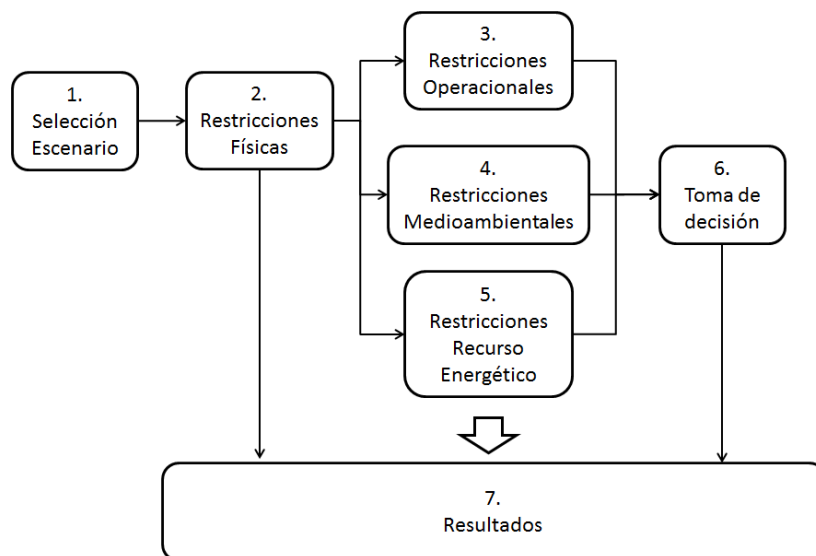


Figura 1: Esquema del proceso

2.3. Datos de entrada: La base de datos geográfica y base de datos de la aplicación

Existen dos procedencias de datos de entrada: según los datos que alberga la propia aplicación, y los procedentes a los datos geográficos.

- **Datos DSS:** es la base de datos donde se almacena los datos vinculantes a la aplicación.
- **Datos Geográficos:** es la base de datos donde se almacenan las capas vectoriales pertenecientes a la base de datos OceanLider.

2.4. Tipos de reglas de decisión para realizar la toma de decisión

En el proceso de decisión, cuando existen diferentes criterios, se pueden distinguir dos tipos de reglas [Hansen, 2005]: las restricciones y los factores.

- **Las restricciones**, también llamadas criterios duros, hard, crisp o booleanos, se corresponden con operaciones que devuelven cierto/falso. Este tipo de reglas imponen criterios de exclusión (se cumple/no se cumple) y generalmente limitarán los análisis (excluyendo zonas de un mapa, delimitando regiones, etc.).
- **Los factores**, llamados criterios o restricciones blandas o soft, están asociadas a algún tipo de incertidumbre en el criterio de decisión. Su utilización implicará usar técnicas como lógica fuzzy o borrosa [Zadeh, 1965], redes de creencias, redes bayesianas o algún otro modelo probabilístico. Indicarán mediante sus funciones de pertenencia un posible

grado de cumplimiento de un criterio. Para poder combinar estos factores será necesario realizar un proceso de normalización de esos grados de incertidumbre (hasta qué punto un criterio se cumple, estando en un intervalo $[0,1]$, por poner un ejemplo).

Al haber intereses contrapuestos, será imposible satisfacer todas las preferencias de distintos actores. Para ello, será necesario aplicar técnicas **de Evaluación Multi-Criterio (MCE)** que asigne pesos de preferencias a dichos criterios y objetivos contradictorios en el proceso de toma de decisión y seguir un criterio de normalización para poder hacer comparaciones. Existen diferentes métodos de MCE que podrán ser consultados en el *Anexo C*.

En lo referente al DSS, la aplicación de criterios booleanos estará ligada a operaciones sobre mapas que pueden entenderse como operaciones sobre teoría de conjuntos: intersección de áreas, unión, diferencia, etc.

En cuanto a factores o criterios soft se necesitarán operaciones de lógica difusa, donde se ha sido necesario la implementación con operaciones de álgebra de mapas raster en GRASS. Según nuestro conocimiento [Ascough y cols. 2002][Malczewski, 2006], pocos productos SIG tienen componentes que realicen MCE:

- IDRISI lleva en sus primeras versiones una técnica denominada Proceso de Análisis Jerárquico ideado por Saaty [Saaty, 1980; Eastman y cols. 1993; Ramírez, 2007].
- Otros SIG tienen rutinas MCE mediante rutinas “plug in”, como en el caso de ARC-INFO [Carver, 1991], que permiten la construcción de algún tipo de MCE.

2.5. Datos y criterios fijados en las reuniones OceanLider

Se han consolidado los criterios pertenecientes al Sistema de Apoyo a la toma de decisión del proyecto OceanLider clasificados en criterios físicos, criterios operacionales, criterios medioambientales y criterios de recurso energético. Cada criterio puede contener uno o varios subcriterios y parámetros asociados que se describen a continuación.

2.5.1. Criterios físicos

- **Distancia a la costa.** De exclusión parcial. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa (valor A, valor B, valor C y valor D)

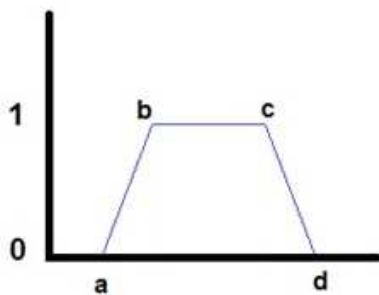


Figura 2: Función de operación difusa lineal

Capa de entrada: Línea de costa (*lin_costa*), es una capa vectorial de tipo lineal perteneciente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor A: 8 Km
 - Valor B: 9 Km
 - Valor C: 20 Km
 - Valor D: 25 Km
- **Profundidad.** De exclusión parcial. Se aplica una operación difusa sobre la profundidad extrayendo los parámetros del dispositivo (parámetros: Profundidad mínima, ProfMin, y profundidad máxima, ProfMax). Las medidas se aplican en metros. Se consideran los siguientes valores que procesa internamente:

valor A:= ProfMin-5m, valor B:= ProfMin, valor C:= ProfMax, valor D:= ProfMax+5m

Nota: Se puede proponer otra constante distinta a ± 5 metros para calcular los valores A y D.

Capa de entrada: Profundidad (*profundidad*), es una capa raster pre-calculada a partir de la capa de batimetría creando un Modelo Digital de Elevaciones (MDE). Se ubica en el sistema de ficheros del servidor en la carpeta Union (ver apartado 4.6.1. *Generación del mapa profundidad como MDE*).

- **Supervivencia del dispositivo.** De exclusión total. El usuario puede seleccionar varios subcriterios. En caso que se seleccione más de un subcriterio se le da la opción al usuario de escoger el método de combinación de los subcriterios: método de razonamiento más que pesimista (multiplicación), pesimista (mínimo), optimista (máximo) y neutro (MLP).

- ✓ **Altura de ola significativa.** Se extrae el valor mayor de ola significativa admitido configurado en el dispositivo, HsMax. Se hace una reclasificación sobre el raster de altura de ola media.

Si $Altura_ola_media(Hs) \geq 0$ & $Hs \leq HsMax$ entonces
 Altura_ola_significante := 1
 Sino
 Altura_ola_significante := 0

Capa de entrada: Hs media (*unionhsmedia*), es una capa raster resultante de la unión de la colección de rasters de altura de ola significativa media en la Península, Baleares y Canarias. Se ubica en el sistema de ficheros del servidor en la carpeta Union.

- ✓ **Velocidad de corriente.** Se extrae el valor máximo de corriente admitido en la configuración del dispositivo, CoMax. Se hace una reclasificación sobre el raster de la velocidad de mareas vivas.

Si $corriente_marea_viva \geq 0$ & $corriente_marea_viva \leq CoMax$ entonces
 Velocidad_corriente := 1
 Sino
 Velocidad_corriente := 0

Capa de entrada: Velocidad media de corriente en condiciones de marea viva (*medviv*), es una capa raster perteneciente a las capas informativas. Se ubica en el sistema de ficheros del servidor en la carpeta Union.

2.5.2. Criterios operacionales

- **Mantenimiento de la posición.** De exclusión parcial. El usuario puede seleccionar varios subcriterios. En caso que se seleccione más de un subcriterio, el método de razonamiento para la combinación de los subcriterios es el más que pesimista (multiplicación).
- ✓ **Facilidad de anclaje al fondo (composición del sustrato marino).** Se extraen los valores de idoneidad de las zonas configuradas para la capa Geofis dependiendo del tipo de dispositivo elegido y se le deja la libertad al usuario para poder cambiar estos valores a través de la interfaz.
 Los valores de idoneidad configurados para cada agrupación por defecto y para cada tipo de dispositivo se reflejan en la siguiente tabla.

NOMBRE	NOMBRE CAPA	METADATOS	Tipo de Capa	CORRIENTES		OLEAJE		
				FUERA DE COSTA (Apoyados sobre base de gravedad o pilotados)	FUERA DE COSTA (Fondeados)	FUERA DE COSTA (Apoyados sobre base de gravedad o pilotados)	FUERA DE COSTA (Flotantes Fondeados)	
MORFOLOGÍA FONDOS MARINOS								
Sustrato de los fondos	Geofis_P.shp	Cartografía vectorial poligonal de los tipos de sustrato de los fondos marinos del litoral de la Península Ibérica, Ceuta y Melilla. Se encuentra en ETRS89 huso 30.	Roca	Excluyente parcial	0,5	0,75	0,5	0,75
			Vegetado		1	1	1	1
			Blando Cohesivo		0	0,5	0	0,5
			Blando no Cohesivo		1	1	1	1

Figura 3: Valores de idoneidad configurados para cada agrupación por defecto para cada tipo de dispositivo.

Capa de entrada: Sustrato de los fondos (*geofis*), es una capa vectorial de tipo polígono perteneciente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

- ✓ **Facilidad de anclaje al fondo (morfología del fondo marino).** Se extraen los valores de idoneidad de las zonas configuradas en la capa Formas dependiendo del tipo de dispositivo elegido y se le deja la libertad al usuario de cambiar los valores según la agrupación para los diferentes tipos que se muestran en la siguiente tabla.

NOMBRE	NOMBRE CAPA	METADATOS	NOMBRE DE GRUPO	TIPO DE ACCIDENTE GEOGRÁFICO	CORRIENTES		OLEAJE	
					FUERA DE COSTA (Apoyados sobre base de gravedad o pilotados)	FUERA DE COSTA (Fondeados)	FUERA DE COSTA (Apoyados sobre base de gravedad o pilotados)	FUERA DE COSTA (Flotantes Fondeados)
Accidentes geográficos submarinos	Formas_P.shp	Cartografía vectorial lineal que se corresponde con los accidentes geográficos submarinos más reseñables de la península Ibérica, Ceuta y Melilla. Se encuentra en ETRS89 huso 30.	Grupo1	Afloramiento rocoso	1	1	1	1
			Grupo1	Marcas de arrastre	1	1	1	1
			Grupo1	Plataforma continental	1	1	1	1
			Grupo1	Playas sumergidas	1	1	1	1
			Grupo1	Ripples	1	1	1	1
			Grupo1	Talud continental	1	1	1	1
			Grupo0	Abanico deltaico	0	0	0	0
			Grupo0	Abanico deltaico profundo	0	0	0	0
			Grupo0	Cañon submarino	0	0	0	0
			Grupo0	Cañon submarino	0	0	0	0
			Grupo0	Llanuras abisales	0	0	0	0
			Grupo0	Megaripples	0	0	0	0
			Grupo0	Megaripples	0	0	0	0
			Grupo0	Morfología de crestas y valles	0	0	0	0
			Grupo0	Ondas de arena	0	0	0	0
			Grupo0	Relieve volcanico	0	0	0	0

Figura 4: : Valores de idoneidad por defecto definida para cada agrupación y tipo de dispositivo

Capa de entrada: Accidentes geográficos submarinos (*formas*), es una capa vectorial de tipo polígono perteneciente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

- **Viabilidad de instalación en zonas ocupadas.** De exclusión parcial. El usuario puede seleccionar varios subcriterios. Si se selecciona más de un subcriterio el método de razonamiento para la combinación de subcriterios es el pesimista (mínimo de los valores).
- ✓ **Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español.** Se aplica una operación difusa sobre la distancia a las zonas ocupadas por las ayudas a la navegación del sistema portuario español. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

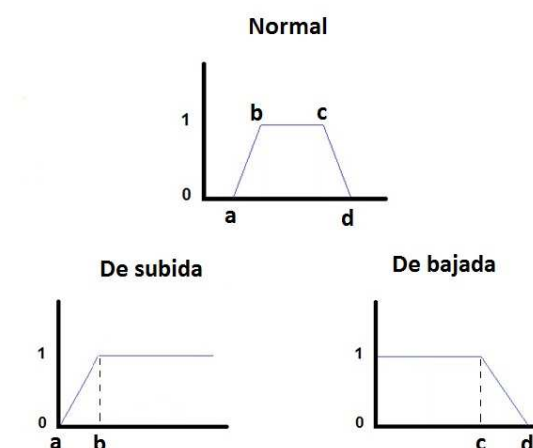


Figura 5: Función de operación difusa lineal normal, de subida y bajada

Capa de entrada: Ayudas a la navegación del sistema portuario español (*aton*), es una capa vectorial de tipo punto perteneciente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor A: 0.5 Km
- Valor B: 1 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Viabilidad de instalación en zonas designadas como Dominio Público Portuario.** Se aplica una operación difusa sobre la distancia a las zonas designadas como Dominio

Público Portuario. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Delimitación del Dominio Público Portuario de los puertos existentes en la Península, Baleares y Canarias (*dpp*), es una capa vectorial de tipo polígono perteneciente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor A: 0 Km
- Valor B: 1 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Viabilidad de instalación en zonas preseleccionadas para la extracción de áridos.** Se aplica una operación difusa sobre la distancia a las zonas preseleccionadas para la extracción de áridos. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Zonas recomendadas para la extracción de áridos (*ext_aridos*), es una capa vectorial de tipo polígono perteneciente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor A: 1 Km
- Valor B: 2 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Viabilidad de instalación en zonas ocupadas por conducciones submarinas.** Se aplica una operación difusa sobre la distancia a las zonas ocupadas por conducciones submarinas. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Cables eléctricos submarinos, emisarios o tuberías (*conducciones*), es una capa vectorial de tipo lineal perteneciente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor A: 1.5 Km

- Valor B: 3 Km
- Valor C: 6 Km
- Valor D: 15 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo normal.

- ✓ **Viabilidad de instalación en zonas preseleccionadas para el vertido de material de dragado.** Se aplica una operación difusa sobre la distancia a las zonas preseleccionadas para el vertido de material de dragado. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Localización de zonas de vertido de material de dragado (*vertido_dragado*), es una capa vectorial de tipo punto perteneciente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor A: 1 Km
- Valor B: 2 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Viabilidad de instalación en zonas ocupadas por sensores instrumentales de medición de parámetros oceanográficos.** Se aplica una operación difusa sobre la distancia a las zonas ocupadas por sensores instrumentales de medición de parámetros oceanográficos. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Localización de sensores instrumentales de medición de parámetros oceanográficos (*rmo*), es una capa vectorial de tipo punto perteneciente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor A: 0.5 Km
- Valor B: 1 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Viabilidad de instalación en zonas ocupadas por dispositivos de separación del tráfico marítimo.** Se aplica una operación difusa sobre la distancia a las zonas ocupadas por dispositivos de separación del tráfico marítimo. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Dispositivos de separación del tráfico marítimo (*dst*), es una capa vectorial de tipo polígono perteneciente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor A: 1.5 Km
- Valor B: 3 Km
- Valor C: 6 Km
- Valor D: 15 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo normal.

- **Viabilidad de instalación en zonas restringidas.** De exclusión parcial. El usuario puede escoger varios subcriterios. Si se selecciona más de un subcriterio el método de razonamiento para la combinación de subcriterios es el pesimista (mínimo de los valores).

- ✓ **Viabilidad de instalación en zonas restringidas para uso militar.** Se aplica una operación difusa sobre la distancia a las zonas restringidas para uso militar. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Zonas de uso militar (*militar*), es una capa vectorial de tipo polígono perteneciente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor A: 1 Km
- Valor B: 2 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías.** Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Zona de cables submarinos (*zona_cables*), es una capa vectorial de tipo polígono perteneciente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor A: 1.5 Km
- Valor B: 3 Km
- Valor C: 6 Km
- Valor D: 15 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo normal.

- ✓ **Viabilidad de instalación en zonas restringidas por la presencia de vertidos.** Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de vertidos. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Vertidos existentes en el litoral (*vertidos*), es una capa vectorial de tipo punto perteneciente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor A: 1 Km
- Valor B: 2 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Viabilidad de instalación en zonas restringidas por la existencia de vertidos de material militar.** Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de vertidos de material militar. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Vertidos de material militar (*vertido_material_militar*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor A: 1 Km
- Valor B: 2 Km
- Valor C: 99999 Km

· Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

2.5.3. Criterios de recurso energético

La influencia del recurso energético en la toma de decisión se calcula mediante un único criterio. Es de exclusión parcial y dependiente del tipo de dispositivo: se aplica la densidad de potencia para tipos de dispositivo de oleaje, y la marea media para tipos de dispositivo de corrientes. Se configura las siguientes fórmulas según los valores del raster de entrada y el parámetro *valor_máximo* configurado por el usuario.

Oleaje (capa raster de *potencia*):

- **De 0 a VALOR MÁXIMO KWh/m** aplicamos la ecuación: Valor índice= Valor del mapa raster/VALOR MÁXIMO
- **+de VALOR MÁXIMO KWh/m**: Valor constante de 1

Capa de entrada: potencia (*pw*), es una capa raster resultante de la unión de la colección de rasters de potencia en la Península, Baleares y Canarias. Se ubica en el sistema de ficheros del servidor en la carpeta Union.

El valor máximo configurado por defecto es: 0.3 KWh/m.

Corrientes (capa raster de *marea media*):

- **De 0 a VALOR MÁXIMO m/s** aplicamos la ecuación: Valor Índice= (velocidad media de mareas vivas y mareas muertas /VALOR MÁXIMO) ^3
- **+ de VALOR MÁXIMO m/s**: Valor constante de 1

Capa de entrada: Capa con la velocidad media de corriente en condiciones de marea viva (*medviv*) y capa con la velocidad media de corriente en condiciones de marea muerta (*medmue*), son capas raster pertenecientes a las capas informativas. Se ubican en el sistema de ficheros del servidor en la carpeta Union.

El valor máximo configurado por defecto es: 1.2 m/s.

2.5.4. Criterios medioambientales

- **Espacios protegidos.** De exclusión parcial. El usuario puede seleccionar varios subcriterios. Si se seleccionan más de uno se le da la opción al usuario de escoger el método de razonamiento para la combinación de los subcriterios: más que pesimista (multiplicación de valores), pesimista (mínimo), optimista (máximo) y neutro (MLP).
- ✓ **Geoparques.** Se aplican las áreas correspondientes a geoparques configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km.

El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Geoparques (*geoparques*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Reservas de la biosfera.** Se aplican las áreas correspondientes a las reservas de la biosfera configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Reservas de la biosfera (*mabs*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Patrimonio de la humanidad.** Se aplican las áreas correspondientes a patrimonio de la humanidad configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad

que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Patrimonio de la humanidad (*p_humanidad*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Red de Áreas Marinas Protegidas del Convenio OSPAR.** Se aplican las áreas correspondientes a la red de áreas marinas protegidas del convenio OSPAR configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Red de Áreas Marinas Protegidas del Convenio OSPAR (*ospar_amps*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Zonas Especialmente Protegidas de Importancia para el Mediterráneo.** Se aplican las áreas correspondientes a las zonas especialmente protegidas de importancia para el Mediterráneo configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad

que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Zonas especialmente protegidas de importancia para el Mediterráneo (*zepims*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Sitios reconocidos con el Diploma Europeo de Calidad de Espacios Protegidos.** Se aplican las áreas correspondientes a los sitios reconocidos con el Diploma Europeo de Calidad de Espacios Protegidos configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Sitios reconocidos con el Diploma Europeo de Calidad de Espacios Protegidos (*dipeuro*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Lugares de Importancia Comunitaria.** Se aplican las áreas correspondientes a los lugares de importancia comunitaria configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el

valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Lugares de Importancia Comunitaria (*lics*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Zonas de Especial Protección para las Aves.** Se aplican las áreas correspondientes a las zonas de especial protección para las aves configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Zonas de especial protección para las aves (*zepas*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Parques Nacionales.** Se aplican las áreas correspondientes a los parques nacionales configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Parques Nacionales (*ppnn*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Biotopos protegidos.** Se aplican las áreas correspondientes a los biotopos protegidos configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Biotopos protegidos (*bbppro*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Microrreservas.** Se aplican las áreas correspondientes a las microrreservas configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Microrreservas (*micror*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Monumentos Naturales.** Se aplican las áreas correspondientes a los monumentos naturales configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Monumentos naturales (*mmnat*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Parajes Naturales.** Se aplican las áreas correspondientes a los parajes naturales configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Parajes Naturales (*pparnnat*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Parques Naturales.** Se aplican las áreas correspondientes a los parques naturales configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Parques Naturales (*ppnat*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Reservas con ley de protección propia.** Se aplican las áreas correspondientes a las reservas con ley de protección propia configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Reservas con ley de protección propia (*rr_ley_prop*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Reservas Naturales.** Se aplican las áreas correspondientes a las reservas naturales configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Reservas Naturales (*rrmn*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Zonas de Especial Protección de los Valores Naturales.** Se aplican las áreas correspondientes a las zonas de especial protección de los valores naturales configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Zonas de Especial Protección de los Valores Naturales (*zepvn*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- **Biodiversidad.** De exclusión parcial. El usuario puede seleccionar varios subcriterios. Si se seleccionan más de uno se le da la opción al usuario de escoger el método de razonamiento para la combinación de los subcriterios: más que pesimista (multiplicación de valores), pesimista (mínimo), optimista (máximo) y neutro (MLP).
- ✓ **Áreas de interés para la conservación de los cetáceos.** Se aplican las áreas correspondientes a las áreas de interés para la conservación de los cetáceos configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Áreas de interés para la conservación de los cetáceos (*aicp*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Áreas compatibles con la figura de Parque Nacional.** Se aplican las áreas correspondientes a las áreas compatibles con la figura de parque nacional configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación

difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Áreas compatibles con la figura de Parque Nacional (*aopn*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Áreas importantes para las aves.** Se aplican las áreas correspondientes a las áreas importantes para las aves configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Áreas importantes para las aves (*ibas*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Zonas propuestas para realizar el Inventario de la biodiversidad marina.** Se aplican las áreas correspondientes a las zonas propuestas para realizar el Inventario de la biodiversidad marina configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas.

Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Zonas propuestas para realizar el Inventario de la biodiversidad marina (*ibm*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Áreas importantes para las aves marinas.** Se aplican las áreas correspondientes a las áreas importantes para las aves marinas configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Áreas importantes para las aves marinas (*iba_recom*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Zona marina especialmente sensible.** Se aplican las áreas correspondientes a las zonas marinas especialmente sensibles configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el

valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Zona marina especialmente sensible (*zmes*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Rutas migratorias de cetáceos.** Se aplican las áreas correspondientes a las rutas migratorias de los cetáceos configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Rutas migratorias de los cetáceos (*cetaceos*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- **Usos pesqueros.** De exclusión parcial. El usuario puede seleccionar varios subcriterios. Si se seleccionan más de uno se le da la opción al usuario de escoger el método de razonamiento para la combinación de los subcriterios: más que pesimista (multiplicación de valores), pesimista (mínimo), optimista (máximo) y neutro (MLP).
- ✓ **Instalaciones de acuicultura.** Se aplican las áreas correspondientes a las instalaciones de acuicultura configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se

aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capas de entrada: Instalaciones de acuicultura (*acuicultura*) es una capa vectorial de tipo polígono y buffer a las instalaciones de acuicultura de forma puntual (*acuicultura_buf*) es una capa vectorial de tipo polígono, ambas corresponden a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Almadrabas.** Se aplican las áreas correspondientes a las instalaciones de almadrabas configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capas de entrada: Almadrabas (*almadrabas*) es una capa vectorial de tipo polígono y buffer a las almadrabas de forma puntual (*almadrabas_buf*) es una capa vectorial de tipo polígono, ambas corresponden a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Arrecifes.** Se aplican las áreas correspondientes a los arrecifes configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le

muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Arrecifes (*arrecifes*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Caladeros.** Se aplican las áreas correspondientes a los caladeros configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Caladeros (*caladeros*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Viveros.** Se aplican las áreas correspondientes a los viveros configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km.

El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capas de entrada: Viveros (*viveros*) es una capa vectorial de tipo polígono y buffer a los viveros de forma puntual (*viveros_buf*) es una capa vectorial de tipo polígono, ambas corresponden a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Zonas protegidas de interés pesquero.** Se aplican las áreas correspondientes a las zonas protegidas de interés pesquero configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Zonas protegidas de interés pesquero (*zpips*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Piscifactorías.** Se aplican las áreas correspondientes a las piscifactorías configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los

cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capas de entrada: Piscifactorías (*piscifactoria*) es una capa vectorial de tipo polígono y buffer a las piscifactorías de forma puntual (*piscifactoria_buf*) es una capa vectorial de tipo polígono, ambas corresponden a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Reservas marinas.** Se aplican las áreas correspondientes a las reservas marinas configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Reservas marinas (*rrmm*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

- ✓ **Zonas de producción de moluscos y de otros invertebrados marinos.** Se aplican las áreas correspondientes a las zonas de producción de moluscos y de otros invertebrados marinos configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad

que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Capa de entrada: Zonas de producción de moluscos y de otros invertebrados marinos (*cria_moluscos*), es una capa vectorial de tipo polígono correspondiente a las capas informativas. Se ubica en la base de datos BBDDOceanLider en el esquema public.

Los valores configurados por defecto para todos los dispositivos son:

- Valor de idoneidad: 0.0
- Valor A: 1 Km
- Valor B: 4 Km
- Valor C: 99999 Km
- Valor D: 99999 Km

Según los valores configurados por defecto, la función de operación difusa lineal es de tipo subida.

2.6. El protocolo de la aplicación de los criterios

El protocolo de aplicación de los criterios se divide por módulos. Cada módulo genera una salida y necesita como entrada la salida de su módulo antecesor. Los módulos, los criterios y los subcriterios generan una capa raster de salida.

Los módulos del esquema están formados por una serie de criterios que se tienen que combinar entre sí, donde se permite no seleccionar ningún criterio o seleccionar N criterios. A su vez, cada criterio está formado por subcriterios que se tendrán que combinar también entre sí, donde se debe seleccionar de 1 a N subcriterios.

Dependiendo del tipo de exclusión al que pertenezca el módulo tendrá una operación u otra para su combinación. Si el módulo es de exclusión total se combinarán los criterios realizando una multiplicación de los valores de las capas resultantes de los criterios que intervengan (AND). Si el módulo es de exclusión parcial se le da la libertad al usuario para configurar el método de razonamiento para su combinación: **más que pesimista** (multiplicación de los resultados de los criterios que intervienen), **pesimista** (mínimo de los valores de los resultados de los criterios que intervienen), **optimista** (máximo de los valores de los resultados de los criterios que intervienen) o **neutro** (Media Lineal Ponderada –MLP- de los resultados de los criterios que intervienen). Para la aplicación de la Media Lineal Ponderada, se tendrá que configurar previamente los pesos de los criterios participantes.

De igual manera, si en un criterio participa más de un subcriterio, se le da la libertad al usuario de elegir el método de razonamiento para la combinación de subcriterios con estas cuatro posibilidades.

La secuencia de aplicación es como se ilustra en la *figura 6: Secuencia modular del protocolo de aplicación de criterios*. También se puede visualizar el empleo de las operaciones de

combinación entre los criterios en la *figura 7: Empleo de las operaciones de los módulos del protocolo de aplicación de los criterios.*

En primer lugar se ejecutan los criterios físicos de exclusión total del escenario combinándose con una operación AND (multiplicación de los valores), este módulo genera una salida que entra como capa raster resultante en el módulo de criterios físicos de exclusión parcial. En el módulo de criterios físicos de exclusión parcial participan los criterios del escenario de este tipo combinándose según el método de razonamiento escogido por el usuario generando una salida que se combinará con el resultado del módulo de criterios físicos de exclusión total con la operación AND_FUZZY (mínimo de los valores). Esta salida entra para cada rama del esquema: rama de criterios operacionales, rama de criterios de recurso energético y rama de criterios medio ambientales, y se combinará con una operación AND (multiplicación de los valores) con los criterios de exclusión total de cada rama. La combinación de la salida de los criterios físicos y los criterios de exclusión total genera una salida raster que será entrada del módulo de exclusión parcial de cada rama.

En el módulo de exclusión parcial de cada rama se combinan los criterios de este tipo con el método de razonamiento elegido por el usuario y la salida que se genera se combina con la salida del módulo de exclusión total de cada una de ellas con la operación AND_FUZZY (mínimo de los valores).

De esta manera, después de haberse ejecutado cada una de estas tres ramas se tendrá tres salidas generadas, a las cuales se aplicará la regla de toma de decisión final con la técnica de **Media Ponderada Ordenada (MPO u OWA)**. Para la aplicación de MPO es necesario configurar los pesos y pesos de orden de cada rama, pues permite seleccionar el grado de compromiso y de riesgo mediante la selección de los pesos de orden.

El mapa resultado procedente de la regla de decisión final (en este caso MPO) indica si las zonas son aptas, no aptas o negociables, considerando todos los criterios configurados y ponderando los criterios operacionales, medioambientales y de recurso energético. Cada rejilla de la capa tendrá asociado un número real en el intervalo [0,1], significando 0 zona no apta, 1 zona apta y valores intermedios para zonas negociables.

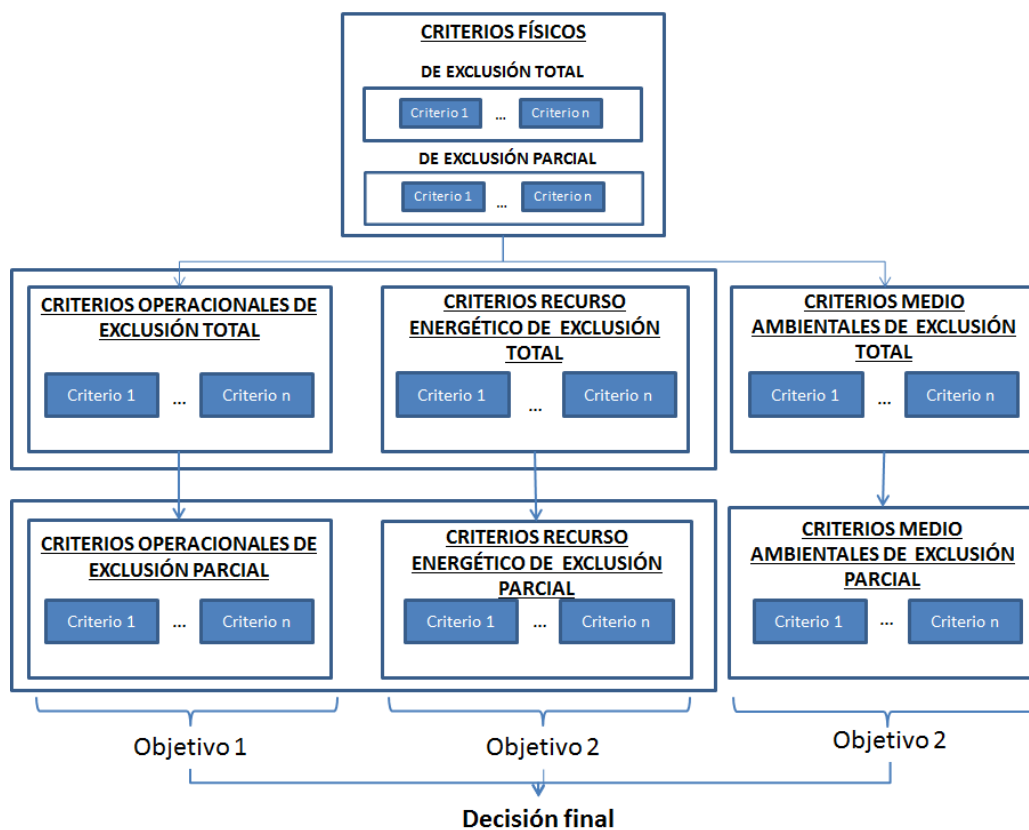


Figura 6: Secuencia modular del protocolo de aplicación de criterios

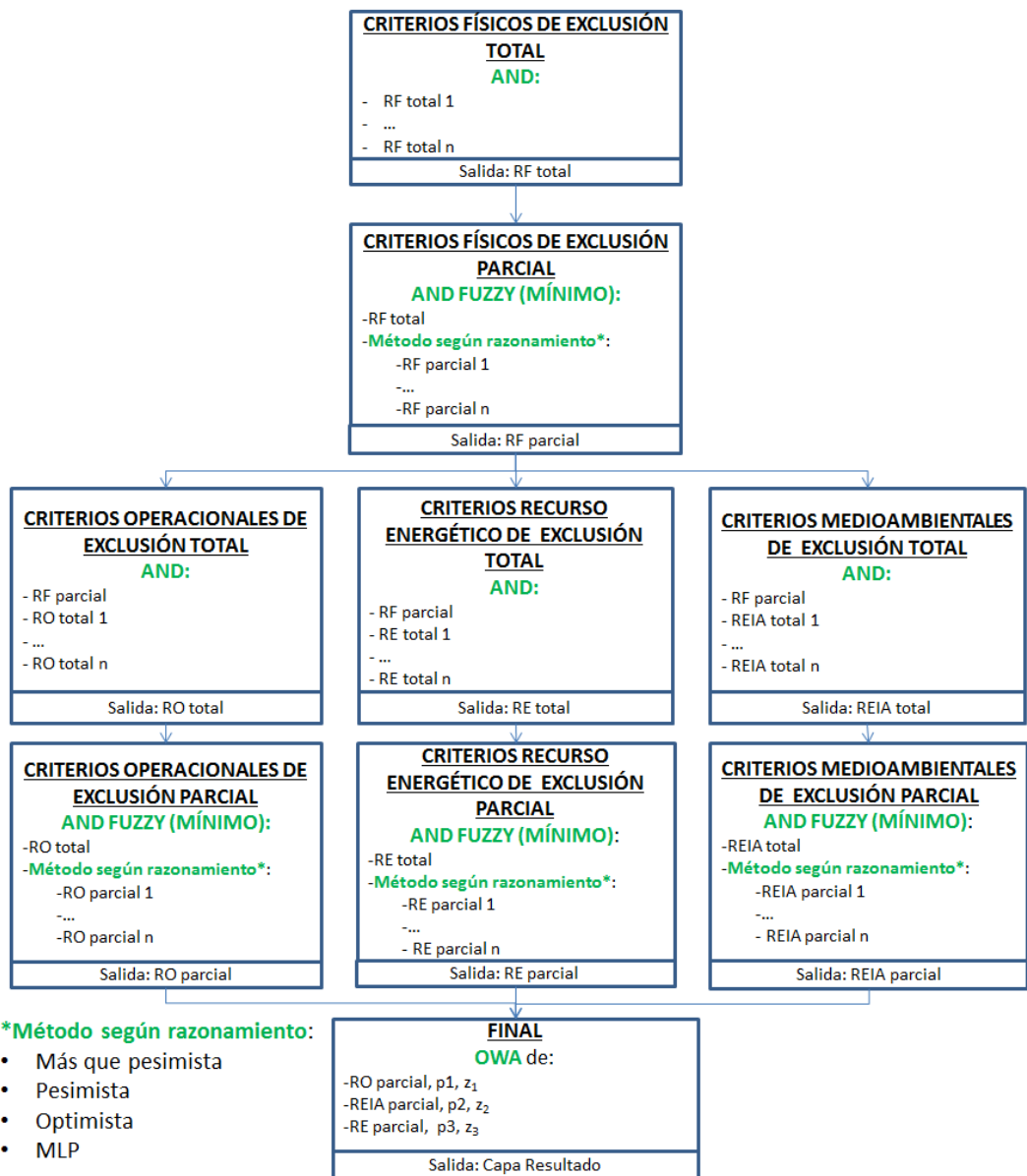


Figura 7: Empleo de las operaciones de los módulos del protocolo de aplicación de los criterios

GENERACIÓN AUTOMÁTICA DE SCRIPTS

3.1. Contexto y objetivos

En este apartado se va a describir la base donde se encuentra la implementación de los criterios (descritos en el apartado 2.5. *Datos y criterios fijados en las reuniones OceanLider*) para la programación automática de scripts.

En la herramienta DSS es necesario disponer de dos bases de datos según la procedencia y uso de los datos: la base de datos donde se almacena los datos vinculantes a la aplicación DSS y la base de datos donde se almacenan las capas vectoriales pertenecientes a la base de datos OceanLider.

Para albergar de forma persistente los datos que utiliza la aplicación DSS utilizaremos el sistema gestor de base de datos **PostgreSQL** teniendo una base de datos relacional. En esta base de datos se identifican dos partes, dependiendo de la naturaleza de los datos, según sea información estática o dinámica.

La **información estática** queda establecida desde el principio en la aplicación y será la base para parametrizar las operaciones de los criterios y toda aquella información que es fija para algunas de las funcionalidades de la herramienta DSS. La **información dinámica** se irá añadiendo, modificando y borrando en el transcurso de la aplicación DSS. Se hablará con mayor detalle en el apartado 3.3. *Modelo de datos*.

Sin embargo para almacenar los datos geográficos proporcionados por las diferentes entidades que participan en este proyecto, es necesario disponer de un sistema gestor de base de datos que ofrezca soporte geoespacial. Utilizaremos **PostGIS** siguiendo las especificaciones GDAL/OGR, siendo esta base de datos independiente de la base de datos de la herramienta DSS.

3.2. Análisis del modelo de datos

3.2.1. Requisitos

La base de datos de la herramienta DSS:

- Debe almacenar los datos e información que utiliza la aplicación para su funcionamiento.
- Las entidades y relaciones representadas deben acercarse lo máximo posible a una representación real de los sucesos y objetos existentes en la aplicación.
- Debe dar soporte para la información estática de la aplicación y la información dinámica.
- El almacenamiento físico de los datos debe ser transparente para el usuario.
- La forma de almacenar los datos no debe influir en la manipulación de los mismos.
- En la medida de lo posible se debe minimizar el impacto en la aplicación en caso de realizarse cambios en la estructura de los datos. También, debe ser posible realizar dichas operaciones sin alterar los demás datos.
- Debe mantener restricciones para la integridad de los datos asegurando una correcta introducción, modificación y borrado de los datos.
- Debe albergar de forma consistente y ordenada los datos que se introduzcan.
- Debe minimizar el tiempo de respuesta de una solicitud en la medida de lo posible.
- Minimizar las posibles inconsistencias que ocurran en el transcurso de manipulación y consulta de los datos.

La base de datos de datos geospaciales:

- Debe almacenar y dar soporte a las capas vectoriales pertenecientes al proyecto OceanLider.
- Tiene que contener operaciones espaciales para poder operar con información espacial.
- Debe almacenar objetos geográficos en diferentes sistemas de coordenadas. En el caso que nos concierne, debería soportar los diferentes sistemas de identificador de referencia espacial, *EPSG: 23030* (para la entidad de Península), *23031* (para la entidad de Baleares), *32628* (para la entidad de Canarias) y *4326* (universal, como sistema mundial para los mapas de unión).

3.2.2. Modelo entidad-relación

En este apartado se describe el esquema entidad-relación de la **base de datos de la aplicación DSS**. La base de datos se encuentra estructurada según las entidades que intervienen: Usuario, Escenario, Dispositivo, Tipo de dispositivo, Tipo de captador del dispositivo, Criterio (Subcriterio), Grupo (Criterios), Parámetros por defecto de un criterio, Parámetros instanciados de un criterio, Parámetro por defecto de un grupo, Parámetros instanciados de un grupo, Operación, tipos de razonamientos posibles para la combinación de criterios o subcriterios, y Resultados.

También ha sido necesaria la creación de algunas entidades independientes que maneja la aplicación DSS esenciales para diferentes funcionalidades. Estas entidades son: *Capas_raster*, *Dispositivo_unidades*, *Escala_longitud*, *Escala_potencia*, *Escala_velocidad*, *MapServer_Capas_Union* y *Opciones_fuzzy*.

A continuación se muestra el diagrama Entidad-Relación de la base de datos de la aplicación DSS. En el apartado 3.3.3. *Modelo estático* y 3.3.4. *Modelo dinámico* se puede visualizar las entidades y relaciones correspondientes al modelo estático y el modelo dinámico pertenecientes al diagrama Entidad-Relación.

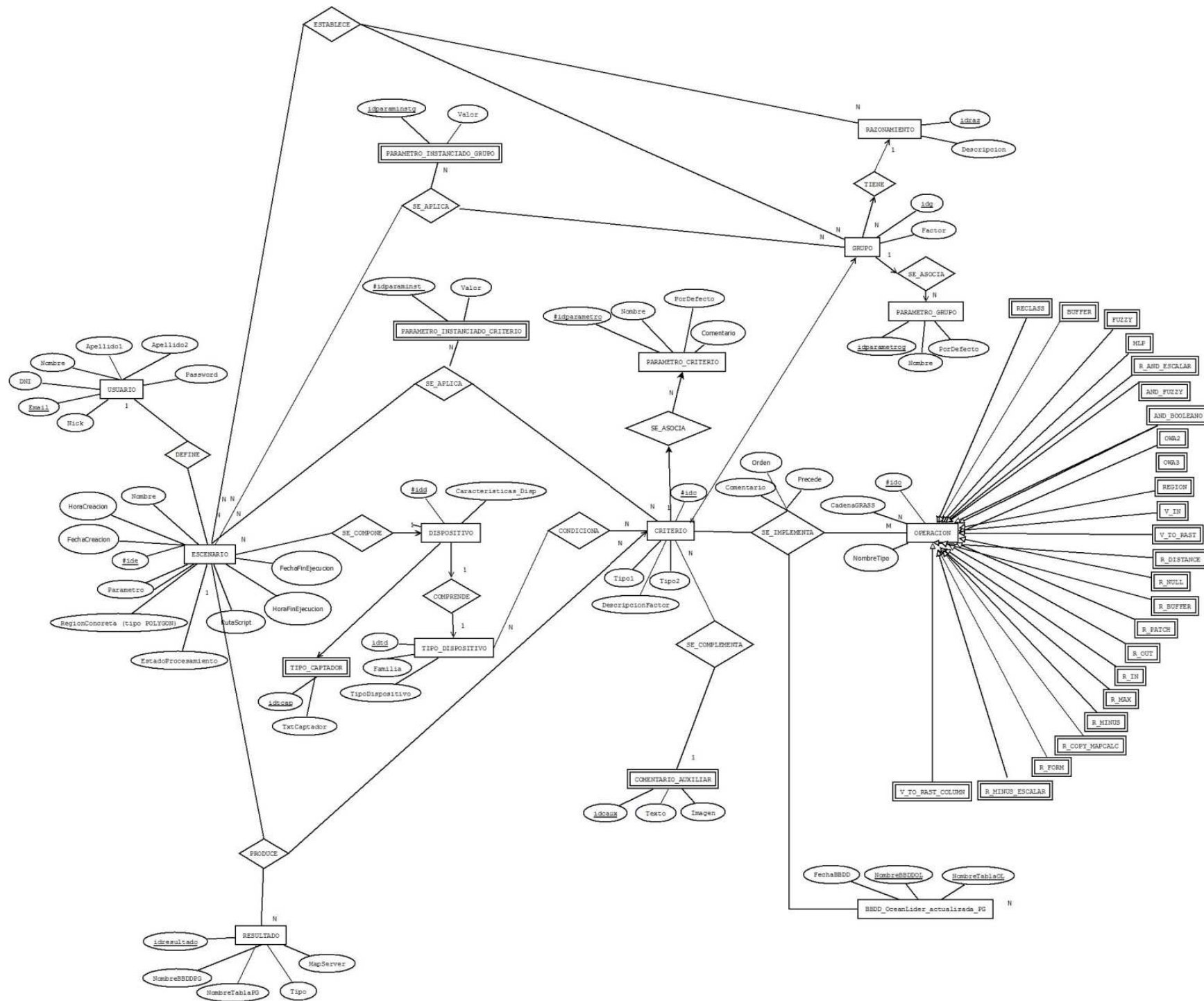


Figura 8: Diagrama Entidad-Relación de la base de datos de la herramienta DSS

3.3. Modelo de datos

3.3.1. Modelo relacional

A continuación se muestran los atributos y tipos de **PostgreSQL** aplicados para las tablas asociadas a cada entidad y relación del diagrama E-R.

Usuario

ATRIBUTO	TIPO
<u>Email</u>	character varying
Nombre	character varying
Apellido1	character varying
Apellido2	character varying
Nick	character varying
Password	character varying

Escenario

ATRIBUTO	TIPO
<u>Id</u>	Serial
Email	character varying
idd	integer
HoraCreacion	character varying
FechaCreacion	character varying
FechaFinEjecucion	character varying
RutaScript	character varying
EstadoProcesamiento	integer
Nombre	character varying
RegionConcreta	Geometry
ModoCF	Integer
ModoCO	integer
ModoCEIA	Integer
ModoCRE	integer

Resultado

ATRIBUTO	TIPO
<u>Idresultado</u>	Serial
Id	Integer
NombreBBDDPG	character varying

NombreTablaPg	character varying
Tipo	character
Idc	Integer
MapServer	character varying

Dispositivo

ATRIBUTO	TIPO
<u>idd</u>	Serial
Nombre	character varying
DisplayTC	Boolean
TxtTipoCaptador	character varying
TipoCaptador	integer
DisplayAB	Boolean
TxtAlturaBuje	character varying
AlturaBuje	character varying
UnidadesAB	character varying
DisplayAB	Boolean
TxtAlturaDisp	character varying
AlturaDispositivo	character varying
UnidadesAD	character varying
DisplayDR	Boolean
TxtDiametroRotor	character varying
DiametroRotor	character varying
UnidadesDR	character varying
DisplayDimX	Boolean
TxtDimX	character varying
DimX	character varying
UnidadesDimX	character varying
DisplayDimY	Boolean
TxtDimY	character varying
DimY	character varying
UnidadesDimY	character varying
DisplayDimZ	Boolean
TxtDimZ	character varying
DimZ	character varying
UnidadesDimZ	character varying
DisplayProfMin	Boolean
TxtProfMin	character varying
ProfMin	character varying
UnidadesProfMin	character varying

DisplayHsMax	Boolean
TxtHsMax	character varying
HsMax	character varying
UnidadesHsMax	character varying
DisplayCoMax	Boolean
TxtCoMax	character varying
CoMax	character varying
UnidadesCoMax	character varying
DisplayPotenciaUnitaria	Boolean
TxtPotenciaUnitaria	character varying
PotenciaUnitaria	character varying
UnidadesPU	character varying
DisplaySepDispLinea	Boolean
TxtSepDispLinea	character varying
SepdispLinea	character varying
UnidadesSDL	character varying
DisplaySepLineas	Boolean
TxtSepLineas	character varying
SepLineas	character varying
UnidadesSDL	character varying

Criterio

ATRIBUTO	TIPO
<u>idc</u>	Serial
DescripcionFactor	character varying
Tipo1	character varying
Tipo2	character varying
Idgrupo	Integer
Idcaux	Integer
opciones	character varying

Operación

ATRIBUTO	TIPO
<u>ido</u>	Serial
CadenaGRASS	character varying
NombreTipo	character varying

Parámetro_criterio

ATRIBUTO	TIPO
<u>Idparametro</u>	Serial
Nombre	character varying
PorDefecto	character varying
Idc	Integer
Idtd	Integer

Parámetro_instanciado_criterio

ATRIBUTO	TIPO
<u>Idparaminst</u>	Serial
Valor	character varying

Grupo

ATRIBUTO	TIPO
<u>idg</u>	Serial
Factor	character varying
TipoRazDefecto	Integer

Parámetro_grupo

ATRIBUTO	TIPO
<u>Idparametrog</u>	Serial
Nombre	character varying
PorDefecto	character varying
Idg	Integer
Idtd	Integer

Parámetro_instanciado_grupo

ATRIBUTO	TIPO
<u>Idparaminstg</u>	Serial
Valor	character varying

Razonamiento

ATRIBUTO	TIPO
<u>Idraz</u>	Serial
Descripcion	character varying

Comentario_auxiliar

ATRIBUTO	TIPO
<u>idcaux</u>	Serial
Texto	character varying
Imagen	character varying

Tipo_dispositivo

ATRIBUTO	TIPO
<u>Idtd</u>	Serial
Familia	character varying
TipoDispositivo	character varying

Tipo_captador

ATRIBUTO	TIPO
<u>Idtcap</u>	Serial
TxtCaptador	character varying

BBDD_OceanLider_PG

ATRIBUTO	TIPO
<u>NombreBBDDOL</u>	character varying
<u>NombreTablaOL</u>	character varying
FechaBBDD	character varying

Capas_raster

ATRIBUTO	TIPO
<u>NombreRaster</u>	character varying
Ubicacion	character varying

Dispositivo_unidades

ATRIBUTO	TIPO
<u>Iddu</u>	Serial
NombreParam	character varying
SistemaUnidades	character varying

Escala_longitud

ATRIBUTO	TIPO
<u>Idlong</u>	Serial
Unidad	character varying

Escala_potencia

ATRIBUTO	TIPO
<u>idpot</u>	Serial
Unidad	character varying

Escala_velocidad

ATRIBUTO	TIPO
<u>idvel</u>	Serial
Unidad	character varying

MapServer_Capas_Union

ATRIBUTO	TIPO
<u>Id</u>	Serial
NombreCapaUnion	character varying
Ruta	character varying

Opciones_fuzzy

ATRIBUTO	TIPO
<u>Id_fuzzy</u>	Serial
Texto	character varying

Escenario_criterio_parametroinstanciado (relación se_aplica1)

ATRIBUTO	TIPO
<u>Ide</u>	Integer
<u>Idc</u>	Integer
<u>Idparaminst</u>	Integer

Escenario_grupo_parametroinstanciado (relación se_aplica2)

ATRIBUTO	TIPO
<u>Id</u> e	Integer
<u>Id</u> g	Integer
<u>Id</u> paraminstg	Integer

Tipo_raz_grupo_escenario (relación establece)

ATRIBUTO	TIPO
<u>Id</u> g	Integer
<u>Id</u> e	Integer
<u>Modo</u> Raz	Integer

Dispositivo_criterio (relación condiciona)

ATRIBUTO	TIPO
<u>Id</u> t d	Integer
<u>Id</u> c	Integer

Criterio_operación (relación se_implementa)

ATRIBUTO	TIPO
<u>Id</u> c	Integer
<u>Id</u> o	Integer
<u>Nombre</u> BBDDOL	character varying
<u>Nombre</u> TablaOL	character varying
<u>Precede</u>	Integer[]
<u>Orden</u>	Integer
<u>Display</u> Comentario	Boolean
<u>Comentario</u>	character varying
<u>Display</u> ParamInst	Boolean
<u>Nombres</u> ParamInst	character varying
<u>Display</u> ParamDispositivo	Boolean
<u>Nombres</u> ParamDispositivo	character varying

3.3.2. Descripción de las entidades y relaciones

3.3.2.1 Entidades

Usuario

Se almacenan los datos pertenecientes a los usuarios registrados en la aplicación y por tanto, los únicos usuarios que pueden hacer uso del sistema.

Escenario

Se almacenan los escenarios creados por los usuarios. En esta tabla se guardarán los datos más relevantes del escenario, como son el nombre, polígono de la región sobre un mapa en proyección WGS84², información sobre la creación y ejecución del script operacional del escenario, el estado de procesamiento, dispositivo al que está ligado y los métodos de razonamiento para los módulos de exclusión parcial de cada ámbito de los criterios (criterios físicos, criterios operacionales, criterios de recurso energético y criterios medio ambientales).

Dispositivo

Se almacenan los dispositivos disponibles en la aplicación para ligar a los escenarios. Se almacenarán las características y valores propios de un dispositivo para que posteriormente se puedan realizar los cálculos según las características del dispositivo. Los escenarios pueden ligar un dispositivo ya creado o crear nuevos dispositivos para su uso, que estarán disponibles para todos los usuarios.

Tipo de dispositivo

Se almacenan los diferentes tipos de dispositivos disponibles en la aplicación, según la familia y tipo. Dependiendo del tipo de dispositivo, el dispositivo tendrá unas características u otras.

Criterio

Se almacenan los criterios disponibles en la aplicación determinado por los expertos. Los criterios estarán ligados a un grupo de criterios y tendrán asociados unos parámetros por defecto.

² *El WGS84 es un sistema de coordenadas geográficas mundial que permite localizar cualquier punto de la Tierra (sin necesitar otro de referencia) por medio de tres unidades dadas. WGS84 son las siglas en inglés de World Geodetic System 84 (que significa Sistema Geodésico Mundial 1984). Se trata de un estándar en geodesia, cartografía, y navegación, que data de 1984. Corresponde con el código EPSG: 4326*

Operación

Se almacenan las operaciones SIG disponibles en la aplicación. Se almacenará la sintaxis definida de forma genérica y el tipo de operación. Se hará uso de la herencia en PostgreSQL para poder definir nuevas operaciones en el sistema en el futuro y poder anclarlas en el código de la aplicación DSS.

Grupo

Se almacenan los datos sobre los grupos de criterios existentes en la aplicación DSS establecidos por los expertos. Cada grupo estará ligado a unos parámetros por defecto que se le mostrará al usuario y a un método de razonamiento por defecto por el que se combinarán los criterios pertenecientes a ese grupo de criterios.

Parámetro_criterio

Son los parámetros por defecto definidos para un criterio ligado a un tipo de dispositivo concreto. Se mostrarán en la aplicación en caso que el criterio requiera uno o varios parámetros para realizar los cálculos.

Parámetro_instanciado_criterio

Son los parámetros pertenecientes a un criterio que haya definido el usuario. Es el valor del parámetro o parámetros de un grupo que haya instanciado el usuario para un escenario en particular.

Parámetro_grupo

Son los parámetros por defecto definidos para un grupo de criterios ligado a un tipo de dispositivo concreto. Se mostrarán en la aplicación en caso que el grupo de criterios requiera uno o unos parámetros para realizar los cálculos.

Parámetro_instanciado_grupo

Son los parámetros pertenecientes a un grupo de criterios que haya definido el usuario. Es el valor del parámetro o parámetros de un grupo que haya instanciado el usuario para un escenario en particular.

Resultado

Son los resultados generados por la aplicación DSS. Se almacena la información perteneciente a los resultados intermedios y finales de un escenario, así como el nombre del resultado, el tipo (final o intermedio), la ruta del fichero de MapServer, la ruta de la capa raster generada para su visualización albergada en el servidor y criterio que lo ha generado.

Razonamiento

Son los métodos de razonamiento disponibles en la aplicación para la combinación de criterios dentro de un grupo de criterios o la combinación de grupos de criterios de exclusión parcial.

Tipo_dispositivo

Son los diferentes tipos de dispositivos permitidos en la aplicación. Es necesario definir el tipo de dispositivo para crear un dispositivo y poder ejecutar un escenario.

Tipo_captador

Son los diferentes tipos de captadores permitidos para el tipo de dispositivo “De corrientes”.

BBDD_OceanLider_PG

Almacena las capas vectoriales disponibles en la base de datos OceanLider. Se mantiene una correlación con la fecha de creación de la capa para mantenerla actualizada.

Comentario_auxiliar

Son los diferentes tipos de comentarios para describir un criterio en concreto en la interfaz de la aplicación. Se define mediante un texto y una imagen asociada (en caso de requerirlo).

Capas_raster

Esta entidad es necesaria para alojar las rutas de las capas raster que usa la aplicación para mostrar mapas de uso general.

Dispositivo_unidades

Esta entidad es necesaria para recopilar el sistema de unidades que utilizará cada variable utilizada en el dispositivo.

Escala_longitud

En esta entidad se albergan las diferentes unidades disponibles en la aplicación para el sistema de unidades de longitud.

Escala_potencia

En esta entidad se albergan las diferentes unidades disponibles en la aplicación para el sistema de unidades de potencia.

Escala_velocidad

En esta entidad se albergan las diferentes unidades disponibles en la aplicación para el sistema de unidades de velocidad.

MapServer_Capas_Union

Esta entidad es necesaria para alojar las rutas de las capas raster de los ficheros de MapServer que usa la aplicación para mostrar mapas de uso general.

Opciones_fuzzy

Son las diferentes opciones para definir una operación fuzzy que están disponibles en la aplicación.

3.3.2.2 Relaciones

Escenario_criterio_parametroinstanciado

Sirve para relacionar un escenario con los criterios seleccionados y parámetros instanciados del criterio introducidos por el usuario.

Escenario_grupo_parametroinstanciado

Sirve para relacionar un escenario con los grupos de criterios seleccionados y parámetros instanciados del grupo introducidos por el usuario.

Dispositivo_criterio

Sirve para determinar los criterios disponibles según el tipo de dispositivo seleccionado para ese escenario.

Criterio_operación

En esta relación recae el mayor peso de la aplicación y es configurada de antemano por los expertos. Se definen el conjunto de operaciones necesarias para ejecutar un criterio. También se configura los parámetros que entran en cada operación.

Tipo_raz_grupo_escenario

En esta relación se almacenan los tipos de razonamientos seleccionados por el usuario para la combinación de los criterios pertenecientes a un grupo de criterios.

3.3.3. Modelo estático

El modelo estático queda comprendido por aquellas entidades y relaciones que den soporte a la información prefijada para el funcionamiento de la aplicación DSS.

Es aquella información invariable en el transcurso de la aplicación. Se muestra en la siguiente figura.

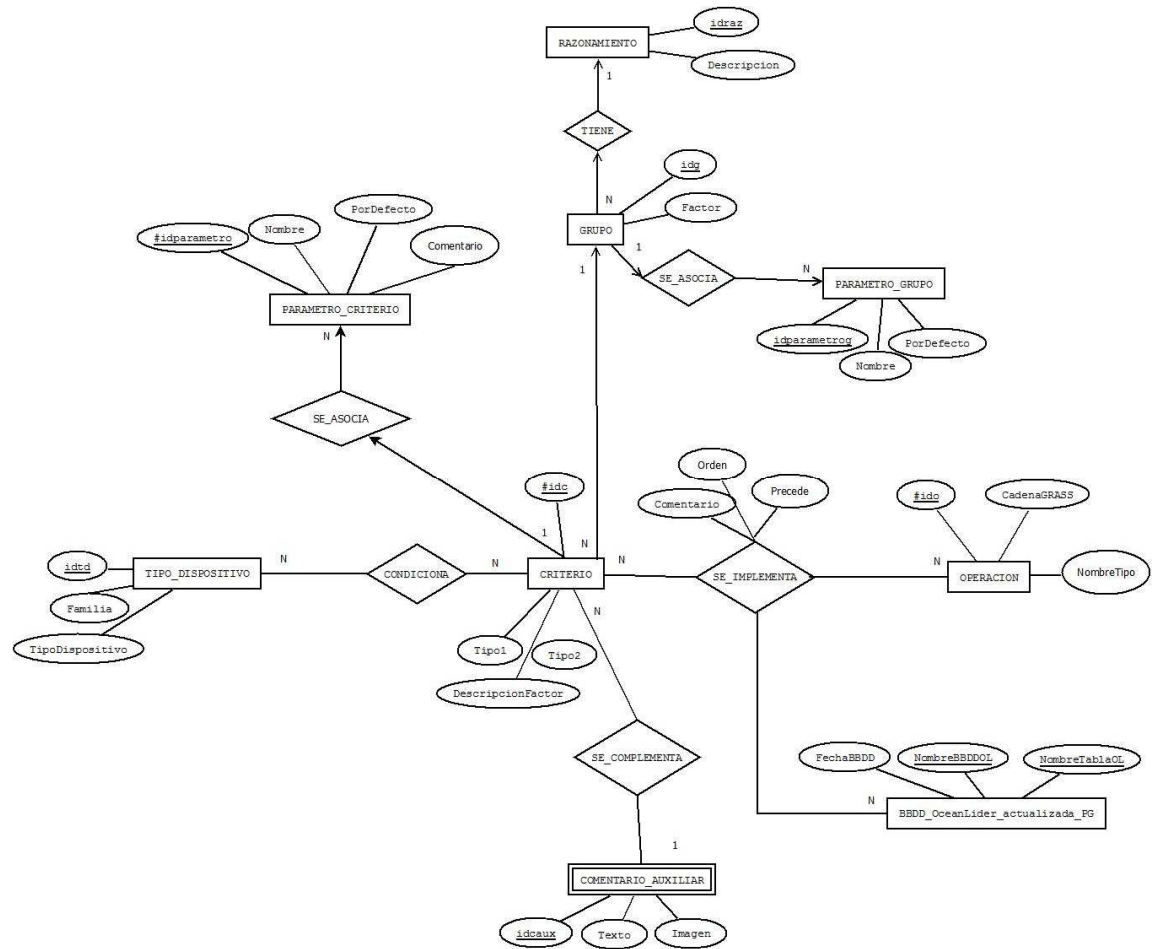


Figura 9: Modelo estático de la base de datos de la herramienta DSS

Se almacenan los **tipos de dispositivos** disponibles en la aplicación DSS. En un futuro se puede ampliar esta tabla con los *costes* asociados a cada uno de ellos. Los tipos de dispositivos disponibles en la aplicación son:

- Undimotriz: fuera de costa - flotante
- Undimotriz: fuera de costa - apoyado en el fondo
- De corrientes: fuera de costa - flotante
- De corrientes: fuera de costa - apoyado en el fondo

Nota: Los tipos de dispositivos eólicos se han desestimado por cambio en los requisitos.

Según los tipos de dispositivos disponibles los expertos determinarán los **criterios** que se pueden aplicar para cada uno de ellos y serán los criterios mostrados en la interfaz para ese escenario. En la interfaz se mostrarán los criterios agrupados según el grupo de criterios al que pertenece.

En la base de datos se almacenan todos los criterios establecidos por los expertos clasificados según el *tipo 1* (*CF* := criterio físico, *CO* := criterio operacional, *CRE* := criterio de recurso energético y *CEIA*:= criterio medioambiental) y *tipo 2* (*Total* := exclusión total o *Parcial* := exclusión parcial). Cada criterio llevará asociado la descripción del factor, el grupo al que pertenece, un comentario auxiliar para mostrar en la interfaz como descripción del criterio, los parámetros por defecto que pueda necesitar para su procesamiento y un conjunto de operaciones para poder ser ejecutado dentro de un sistema SIG.

En la **relación SE_IMPLEMENTA** recae el mayor peso de la parte estática de la base de datos. Cada tupla introducida tendrá que ser estudiada, evaluada y definida previamente por los expertos, determinando las operaciones precisas de los criterios para ser ejecutado en un SIG.

Cada criterio se implementa con una o varias operaciones y estarán asociadas a una o varias capas de entrada de la **BBDD OceanLider**. Esto permitirá la visualización gráfica (en miniatura) de las capas que intervienen a la hora de configurar el escenario en la selección del criterio y poder determinar la aplicación del criterio en caso de ausencia de datos.

En la **relación SE_IMPLEMENTA** se tendrá que definir para cada criterio: el conjunto de operaciones para ser ejecutado, las capas que intervienen para su ejecución y la/s entradas de su parametrización.

La ejecución de las operaciones se realiza de forma secuencial y cada operación puede necesitar de entrada: una capa de la base de datos, una/s salida/s de una operación antecesora o ninguna entrada. Por tanto, es necesario incluir un número de orden para cada tupla introducida y las operaciones predecesoras en caso que la operación precise de entrada los resultados de otras operaciones y no requiera de una capa de entrada de la BBDD OceanLider.

También, a cada tupla introducida se le indicará la vía para extraer los valores de las variables que utilizará para la parametrización de la operación. La procedencia de los valores de las variables puede ser: [Se especificará con mayor detalle en el apartado 3.5.4. *Programación de los criterios*]

- Campo **Comentario**. Se ha decidido flexibilizar el sistema dando cierta libertad de programación en este campo. Cuando se necesite utilizar este campo, *DisplayComentario* deberá permanecer a *true*. En el campo *Comentario* se podrá asignar cualquier variable que el experto crea necesario y se le podrá dar el valor que crea conveniente. Muchas de las operaciones están sujetas a este campo.
- Campo **NombresParamInst**. En este campo se determinan los nombres de los parámetros que se utilizan para la parametrización de la operación. Se dispondrán los nombres en una lista separados por comas. El campo *DisplayParamInst* deberá estar a *true*. El sistema extraerá los valores de los parámetros instanciados que estén sujetos a dichos nombres.
- Campo **NombresParamDispositivo**. En este campo se determinan los nombres de los parámetros del dispositivo que se utilizan para la parametrización de la operación. Se dispondrán los nombres de los parámetros del dispositivo en una lista separados por

comas. El campo *DisplayParamDispositivo* deberá estar a true. El sistema extraerá los valores de los parámetros del dispositivo que estén sujetos a dichos nombres.

Y por último cada **operación** llevará asociada una cadena definida en GRASS. Las cadenas están definidas previamente en la base de datos según las operaciones disponibles en el sistema. Se ha recurrido a la herencia de PostgreSQL para una posible ampliación de operaciones en el futuro.

Las cadenas siguen la sintaxis de GRASS utilizando nombres de variables precedidas del símbolo '\$' para su posterior parametrización. Cualquier variable precedida de '\$', el sistema interpretará como variable para parametrizar y dependiendo de la operación el sistema extraerá el valor dependiendo de la vía que se haya configurado.

No hay que olvidar que en el modelo estático también quedan incluidas las siguientes entidades esenciales para diferentes funcionalidades de la aplicación DSS: *Capas_raster*, *Dispositivo_unidades*, *Escala_longitud*, *Escala_potencia*, *Escala_velocidad*, *MapServer_Capas_Union* y *Opciones_fuzzy*.

3.3.4. Modelo dinámico

El modelo dinámico queda comprendido por aquellas entidades y relaciones que den soporte a la información variable en el transcurso de la aplicación DSS. Se muestra en la siguiente figura de forma resaltada.

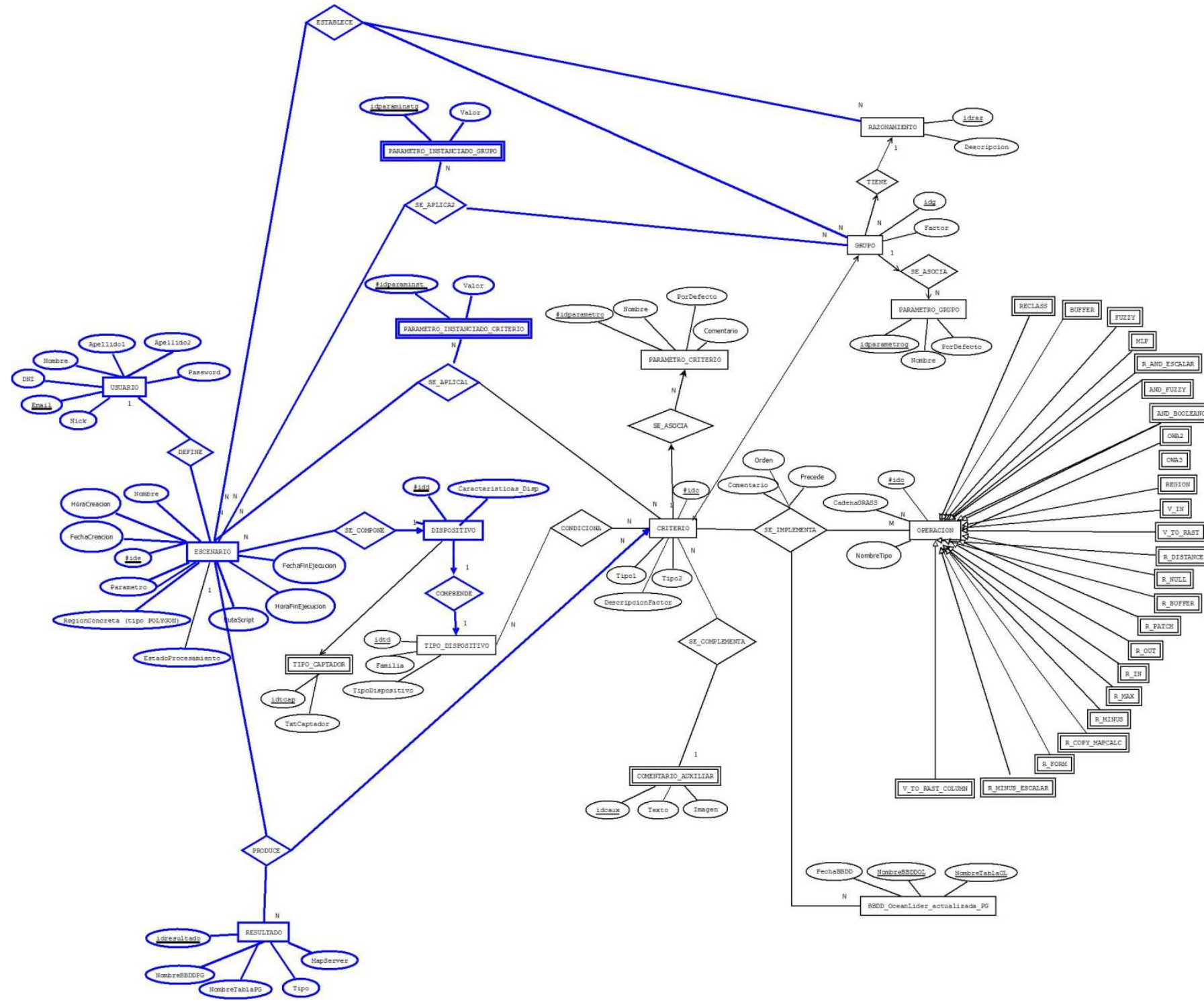


Figura 10: Diagrama E-R con el modelo dinámico resaltado

Un **usuario** define uno o varios **escenarios**. Se almacenan los datos oportunos del usuario (cumpliendo la ley 15/1999 de protección de datos). Sobre esta entidad estarán todos los usuarios que pueden acceder a la aplicación DSS.

Un **escenario** tiene unos parámetros necesarios que el usuario debe configurar y estará ligado a un dispositivo (creado o seleccionado), a unos criterios y a unos grupos de criterios.

La relación SE_APLICA1 almacena la correlación de un **escenario** con los **criterios** que ha seleccionado el usuario (entre los disponibles según el tipo de dispositivo) y los **parámetros del criterio** instanciados en el momento de configurar el escenario.

La relación SE_APLICA2 almacena la correlación de un **escenario** con los **grupos de criterios** que intervienen según los criterios seleccionados por el usuario y los **parámetros del grupo** de criterios instanciados en el momento de configurar el escenario.

La relación ESTABLECE almacena la correlación de un **escenario** con los **métodos de razonamiento** de cada **grupo** seleccionado para la combinación de los criterios del grupo. En la entidad **Resultado** se almacenarán los resultados de un escenario procesado. Cada entrada de esta tabla procede de un resultado intermedio o final ligado a un escenario y criterio en concreto.

3.4. Sistema Gestor de Base de Datos

El Sistema Gestor de Base de Datos que vamos a utilizar y nos permite definir las características comentadas en los apartados anteriores para la base de datos de la aplicación DSS es **PostgreSQL**.

PostgreSQL es un SGBD relacional orientado a objetos y libre, publicado bajo la licencia BSD.

Algunas de las características de PostgreSQL:

- Está desarrollado en C, con herramientas como Yacc y Lex.
- La API de acceso al SGBD se encuentra disponible en C, C++, Java, Perl, PHP, Python y TCL, entre otros.
- Cuenta con un rico conjunto de tipos de datos, permitiendo además su extensión mediante tipos y operadores definidos y programados por el usuario.
- Su administración se basa en usuarios y privilegios.
- Sus opciones de conectividad abarcan TCP/IP, sockets Unix y sockets NT, además de soportar completamente ODBC³.
- Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra 'ñ'.

³ **Open DataBase Connectivity (ODBC)** es un estándar de acceso a las bases de datos desarrollado por SQL Access Group en 1992. El objetivo de ODBC es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué sistema de gestión de bases de datos (DBMS) almacene los datos.

- Puede extenderse con librerías externas para soportar encriptación, búsquedas por similitud fonética (soundex), etc.
- Control de concurrencia multiversión, lo que mejora sensiblemente las operaciones de bloqueo y transacciones en sistemas multiusuario.
- Soporte para vistas, claves foráneas, integridad referencial, disparadores, procedimientos almacenados, subconsultas y casi todos los tipos y operadores soportados en SQL92 y SQL99.
- Implementación de algunas extensiones de orientación a objetos. En PostgreSQL es posible definir un nuevo tipo de tabla a partir de otra previamente definida.

Para la base de datos de las capas espaciales en formato vectorial utilizaremos **PostGIS**. PostGIS es un módulo de software libre que añade soporte para objetos geográficos a la base de datos relacional PostgreSQL para su utilización en un Sistema de Información Geográfico, esto es, para los objetos que aparecen en la especificación OpenGIS como puntos, líneas, polígonos, multilíneas, multipuntos y colecciones geométricas, así como las funciones básicas para el análisis y procesamiento de objetos SIG. Algunas de las características de PostGIS:

- *Alto rendimiento*: PostGIS utiliza un tipo de dato para la geometría y los tipos nativos para el almacenamiento de información no espacial. También hace uso de índices espaciales para aumentar la velocidad de búsqueda.
- *Consulta espacial*: PostGIS incluye un completo conjunto de operaciones para consulta espacial, optimizadas por sus índices R-Tree y su integración con PostgreSQL query planner.
- *Integridad de Datos*: permite a múltiples procesos trabajar con tablas espaciales concurrentemente y asegura la integridad de los datos.
- *Análisis espacial*: a través del conjunto de funciones proporcionadas se puede conseguir realizar análisis SIG.
- *Soporte de Estándares*: PostGIS ha sido certificado por la OGC compatible con la especificación “Simple Features for SQL 1.1”, también implementa parte de la norma ISO “SQL Multi-media” (SQL/MM) para la funcionalidad geoespacial.

3.4.1. Herencia en PostgreSQL

Herencia es un concepto de POO (Programación Orientada a Objetos), sin embargo, en la base de datos PostgreSQL es posible la herencia de manera transparente al nivel de visión de base de datos.

PostgreSQL ofrece como característica particular la herencia entre tablas, que permite definir una tabla que herede de otra previamente definida.

La herencia es a nivel de datos, no de la estructura de las tablas.

- La tabla hija puede tener más atributos que la madre
- Todas las filas insertadas en la tabla hija, pasan a estar en la tabla madre, pero no viceversa
- Los atributos heredados modificados en la tabla hija se modifican en la fila correspondiente de la tabla madre

- El OID de las filas comunes madre-hija es el mismo
- Se permite herencia múltiple
- Los procedimientos y disparadores no se heredan
- Las restricciones solo afectan a las tuplas propias de la relación, no a las que se derivan de la generalización

En PostgreSQL, una alternativa para no utilizar los OID es crear una columna de tipo serial en la tabla padre, así será heredada en la hija. El tipo serial define una secuencia de valores que se irá incrementando de forma automática, y por lo tanto constituye una buena forma de crear claves primarias, al igual que el tipo AUTO_INCREMENT en MySQL.

A continuación se describe un ejemplo de la creación de tablas con herencia de PostgreSQL con las tablas de operaciones.

```
CREATE TABLE "OPERACION"  
(  
  ido SERIAL NOT NULL,  
  "CadenaGRASS" character varying(3000) NOT NULL,  
  "NombreTipo" character varying(50) NOT NULL,  
  CONSTRAINT "PrimaryKeyOPERACION" PRIMARY KEY (ido)  
)  
WITH (  
  OIDS=FALSE  
)  
;  
ALTER TABLE "OPERACION" OWNER TO postgres;  
  
CREATE TABLE "OPERACION_RECLASS"  
(  
  CONSTRAINT "PrimaryKeyOPERACION_RECLASS" PRIMARY KEY (ido)  
)  
INHERITS ("OPERACION")  
WITH (  
  OIDS=FALSE  
)  
;  
ALTER TABLE "OPERACION_RECLASS" OWNER TO postgres;  
  
CREATE TABLE "OPERACION_BUFFER"  
(  
  CONSTRAINT "PrimaryKeyOPERACION_BUFFER" PRIMARY KEY (ido)  
)  
INHERITS ("OPERACION")  
WITH (  
  OIDS=FALSE  
)  
;  
ALTER TABLE "OPERACION_BUFFER" OWNER TO postgres;
```

3.5. Operaciones definidas en el sistema y programación de criterios

Se han definido las operaciones fundamentales de la aplicación DSS para la ejecución de los criterios. Nos hemos apoyado en la sintaxis de las operaciones del SIG GRASS para su posterior ejecución dentro de este entorno.

En este apartado se exponen las operaciones definidas en el sistema en la actualidad, el estudio y la programación de los criterios por parte de los expertos.

3.5.1. Operaciones del sistema

OPERACIÓN_ANDBOOLEANO

Está definida como una operación binaria que obtiene como resultado la multiplicación booleana de dos capas raster de entrada.

Esta operación es ampliada en la aplicación DSS en caso que se precise N operandos de entrada. Esto es posible ya que la sintaxis de GRASS lo permite.

El esqueleto de la operación:

```
"r.mapcalc "$r = $1 * $2 ""
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la primera capa de entrada y \$2 es el parámetro nombre de la segunda capa de entrada.

OPERACIÓN_ANDFUZZY

Es una operación binaria que obtiene como resultado el mínimo de dos capas raster de entrada, evaluado celda a celda.

Esta operación es ampliada en la aplicación DSS en caso que se precise de N operandos de entrada. Esto es posible ya que la parametrización de las operaciones permite la concatenación de operaciones en GRASS.

El esqueleto de la operación:

```
"r.mapcalc "$r = if(isnull( $1 ) || isnull( $2 ), null(), if($1 <= $2 , $1 , $2 ))"
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la primera capa de entrada y \$2 es el parámetro nombre de la segunda capa de entrada.

OPERACIÓN_BUFFER

Es una operación unaria aplicada sobre una capa vectorial de entrada. Se obtiene como resultado un área de influencia o buffer aplicado sobre los puntos, líneas o polígonos de la capa vectorial de entrada. Hay que especificar la distancia buffer en las unidades del mapa.

El esqueleto de la operación:

```
"v.buffer input=$1 output=$r type=point,line,area layer=1 distance=$distance "
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la primera capa de entrada y \$distance es el parámetro de la distancia buffer.

OPERACIÓN_FUZZY

Es una operación unaria aplicada sobre una capa raster de entrada. Se obtiene como resultado una operación difusa (fuzzy) sobre los valores de la capa raster de entrada asignando valores en el intervalo [0,1]. Esta operación se ha implementado como una función lineal como se muestra en la figura [Rubio, 2012].

Se deben especificar cuatro valores para determinar los puntos de control: A, B, C y D.

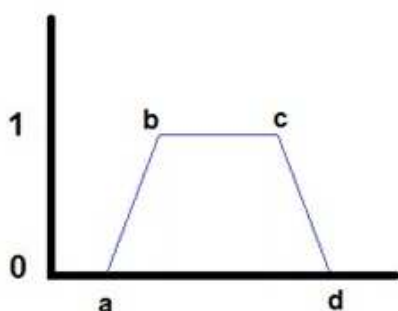


Figura 11: Aplicación difusa (fuzzy) con una función lineal

Si se establecen los puntos de control C y D con valores infinitos, se obtiene una operación difusa con rampa de subida. Si se establecen los puntos de control A y B con valores menos infinitos, se obtiene una operación difusa con rampa de bajada.

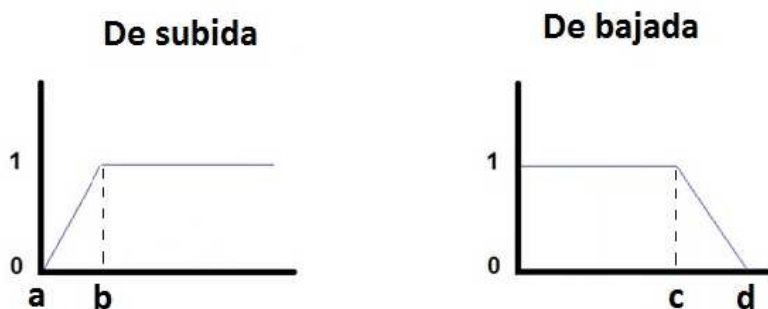


Figura 12: Aplicación difusa (fuzzy) con una función lineal con rampa de subida y de bajada

El esqueleto de la operación:

```
"r.mapcalc "$r = if(isnull( $1 ), null(), if(($1 <= $A || $1 >= $D), 0.0, if(($1 > $A && $1 < $B), eval(($1 - $A)/($B - $A)), if(($1 > $C && $1 < $D), eval(($D - $1)/($D - $C)), 1.0)))"
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la primera capa de entrada, \$A es el parámetro punto de control A, \$B es el parámetro punto de control B, \$C es el parámetro punto de control C y \$D es el parámetro punto de control D.

OPERACIÓN_MLP

Es una operación binaria que se obtiene como resultado la Media Lineal Ponderada de dos capas raster de entrada. Se tienen que especificar los pesos que lleva asociado cada capa de entrada (la suma de los pesos debe ser igual a 1).

Esta operación es ampliada en la aplicación DSS en caso que se precise una operación de N operandos de entrada. Esto es posible ya que la sintaxis de GRASS lo permite.

El esqueleto de la operación:

```
"r.mapcalc "$r = $p1 * $1 + $p2 * $2"
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la primera capa de entrada, \$2 es el parámetro nombre de la segunda capa de entrada, \$p1 es el parámetro peso de la primera capa de entrada y \$p2 es el parámetro peso de la segunda capa de entrada.

OPERACIÓN_OWA2

Es una operación binaria que se obtiene como resultado la Media Ponderada Ordenada (MPO o OWA) de dos capas raster de entrada. Se tienen que especificar los pesos y pesos de orden que lleva asociado cada capa de entrada (siendo la suma de los pesos igual a 1, y los pesos de orden también).

$$OWA = \frac{\sum_{i=1}^N x_i * w_i * z_i}{\sum_{i=1}^N w_i * z_i}$$

Figura 13: Fórmula operación OWA

Dónde:

x_i = valor de la celda de la capa i , w_i = peso de la capa i , z_i = peso de orden asociado a la celda de la capa i

Esta operación se ha implementado para dos capas raster de entrada con la ayuda del álgebra de mapas del SIG GRASS [Rubio, 2012].

El esqueleto de la operación:

```
"r.mapcalc "$r = if(isnull( $1 ) || isnull( $2 ), null(),if($1 >= $2 , eval((( $1 * $p1 * $z1) + ( $2 * $p2 * $z2 )) / ( $p1 * $z1 + $p2 * $z2 )), eval((( $1 * $z2 ) + ( $2 * $p2 * $z1 )) / ( $p1 * $z2 + $p2 * $z1))))"
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la primera capa de entrada, \$2 es el parámetro nombre de la segunda capa de entrada, \$p1 es el parámetro peso de la primera capa de entrada, \$p2 es el parámetro peso de la segunda capa de entrada, \$z1 es el parámetro peso de orden de la primera capa de entrada y \$z2 es el parámetro peso de orden de la segunda capa de entrada.

OPERACIÓN_OWA3

Es una operación ternaria que se obtiene como resultado la Media Ponderada Ordenada (MPO o OWA) de tres capas raster de entrada. Se tienen que especificar los pesos y pesos de orden que lleva asociado cada capa de entrada (siendo la suma de los pesos igual a 1, y los pesos de orden también).

Esta operación se ha implementado para tres capas raster de entrada con la ayuda del álgebra de mapas del SIG GRASS [Rubio, 2012].

El esqueleto de la operación:

```
"r.mapcalc "$r = if(isnull($1) || isnull($2) || isnull($3), null(),if($1 >= $2 && $2 >= $3, eval((( $1*$p1*$z1) + ( $2*$p2*$z2) + ( $3*$p3*$z3)) / ( $p1*$z1+$p2*$z2+$p3*$z3)),if($1 >= $3 && $3 > $2, eval((( $1*$p1*$z1) + ( $3*$p3*$z2) + ( $2*$p2*$z3)) / ( $p1*$z1+$p3*$z2+$p2*$z3)),if($2 > $1 && $1 >= $3, eval((( $2*$p2*$z1) + ( $1*$p1*$z2) + ( $3*$p3*$z3)) / ( $p2*$z1+$p1*$z2+$p3*$z3)),if($2 >= $3 && $3 > $1, eval((( $2*$p2*$z1) + ( $3*$p3*$z2) + ( $1*$p1*$z3)) / ( $p2*$z1+$p3*$z2+ $p1*$z3)),if($3 > $1 && $1 >= $2, eval((( $3*$p3*$z1) + ( $1*$p1*$z2) + ( $2*$p2*$z3)) / ( $p3*$z1+$p1*$z2+$p2*$z3)),if($3 > $2 && $2 > $1, eval((( $3*$p3*$z1) + ( $2*$p2*$z2) + ( $1*$p1*$z3)) / ( $p3*$z1+$p2*$z2+$p1*$z3)), null()))))"
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la primera capa de entrada, \$2 es el parámetro nombre de la segunda capa de entrada, \$p1 es el parámetro peso de la primera capa de entrada, \$p2 es el parámetro peso de la segunda capa de entrada, \$p3 es el parámetro peso de la tercera capa de entrada, \$z1 es el parámetro peso de orden de la primera

capa de entrada, \$z2 es el parámetro peso de orden de la segunda capa de entrada y \$z3 es el parámetro peso de orden de la tercera capa de entrada.

OPERACIÓN_R_AND_ESCALAR

Es una operación unaria aplicada sobre una capa raster de entrada. Se obtiene como resultado una capa raster con la multiplicación del valor de una capa raster de entrada y un valor escalar evaluado celda a celda.

El esqueleto de la operación:

```
"r.mapcalc "$r = $1 * $n"
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la primera capa de entrada y \$n es el parámetro del valor escalar.

OPERACIÓN_R_BUFFER

Es una operación unaria aplicada sobre una capa raster de entrada. Se obtiene como resultado un área de influencia o buffer aplicado sobre las celdas que no contienen valores nulos. Se debe especificar la distancia en metros.

El esqueleto de la operación:

```
"r.buffer input=$1 output=$r units=meters  
distances=$distance,$distance"
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la primera capa de entrada y \$distance es el parámetro de la distancia buffer.

OPERACIÓN_R_COPY_MAPCALC

Es una operación unaria aplicada sobre una capa raster de entrada. Se obtiene como resultado una clonación de la capa raster de entrada. Esta operación es necesaria cuando se requiera hacer una doble reclasificación.

El esqueleto de la operación:

```
"r.mapcalc "$r = $1"
```

Donde \$r es el parámetro nombre del resultado y \$1 es el parámetro nombre de la capa de entrada.

OPERACIÓN_R_DISTANCE

Es una operación unaria aplicada sobre una capa raster de entrada. Se obtiene como resultado una capa raster de distancias a las celdas no nulas más cercanas. Se utiliza el algoritmo de distancia Euclídea.

El esqueleto de la operación:

```
"r.grow.distance input=$1 distance=$r metric=euclidean "
```

Donde \$r es el parámetro nombre del resultado y \$1 es el parámetro nombre de la capa de entrada.

OPERACIÓN_R_FORM

Es una operación unaria aplicada sobre una capa raster de entrada. Se ha implementado esta operación para aplicar fórmulas de álgebra de mapas sobre unos rangos de valores de la capa raster de entrada.

Se ha pensado para aplicar diferentes fórmulas raster y obtener el mapa raster de resultado.

- De 0 a VALOR de la capa raster de entrada, aplicamos ecuación o asignamos cte, lo asignado a \$A.
- Resto de valores, aplicamos ecuación o asignamos cte, lo asignado a \$B.

El esqueleto de la operación:

```
"r.mapcalc "$r= if($1 >= 0 && $1 < $valor, $A, $B)""
```

Donde \$r es el parámetro del nombre del resultado, \$1 es el parámetro nombre de la primera capa de entrada y \$A es el parámetro de la fórmula para los valores de 0 a \$valor de la capa raster de entrada, \$B es el parámetro de la fórmula para el resto de valores de la capa raster de entrada y \$valor es el parámetro del valor límite para la aplicación.

OPERACIÓN_R_IN

Es una operación unaria aplicada sobre una capa raster externa de entrada. Es la operación de importación de capas raster del sistema de ficheros al SIG GRASS para poder operar con dicha capa dentro de este entorno.

El esqueleto de la operación:

```
"r.in.gdal -o input="$1" output=$r "
```

Donde \$r es el parámetro nombre del resultado y \$1 es el parámetro nombre de la capa de entrada.

OPERACIÓN_R_MAX

Es una operación binaria aplicada sobre dos capas raster de entrada. Se obtiene como resultado una capa raster con el valor máximo de la evaluación celda a celda entre las dos capas raster de entrada.

Esta operación es ampliada en la aplicación DSS en caso que se precise de N operandos de entrada. Esto es posible ya que la parametrización de las operaciones permite la concatenación de operaciones en GRASS.

El esqueleto de la operación:

```
"r.mapcalc "$r = if(isnull( $1 ) || isnull( $2 ), null(), if($1 >= $2 , $1 , $2 ))"
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la primera capa de entrada y \$2 es el parámetro nombre de la segunda capa de entrada.

OPERACIÓN_R_MINUS

Es una operación binaria aplicada sobre dos capas raster de entrada. Se obtiene como resultado una capa raster con la diferencia de los valores de las dos capas raster de entrada evaluado celda a celda.

El esqueleto de la operación:

```
"r.mapcalc "$r = $1 - $2"
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la primera capa de entrada y \$2 es el parámetro nombre de la segunda capa de entrada.

OPERACIÓN_R_MINUS_ESCALAR

Es una operación unaria aplicada sobre una capa raster de entrada. Se obtiene como resultado una capa raster con la diferencia del valor de una capa raster de entrada y un valor escalar evaluado celda a celda.

El esqueleto de la operación:

```
"r.mapcalc "$r = $1 - $n"
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la primera capa de entrada y \$n es el parámetro del valor escalar.

OPERACIÓN_R_NULL

Es una operación unaria aplicada sobre una capa raster de entrada. Esta operación se aplica sobre la misma capa raster de entrada. Se emplea para establecer un valor a los valores nulos de la capa raster.

El esqueleto de la operación:

```
"r.null map=$1 null=$null "
```


Donde \$1 es el parámetro nombre de la primera capa de entrada y \$null es el parámetro del valor para la configuración de los valores nulos.

OPERACIÓN_R_OUT

Es una operación unaria aplicada sobre una capa raster de entrada. Es la operación de exportación de capas raster a un sistema de ficheros.

El esqueleto de la operación:

```
"r.out.gdal input=$1 output="$r"
```

Donde \$r es el parámetro nombre del resultado y \$1 es el parámetro nombre de la primera capa de entrada.

OPERACIÓN_R_PATCH

Es una operación binaria aplicada sobre dos capas raster de entrada. Hace un relleno de las celdas nulas (sin datos) de la segunda capa raster con la primera capa raster.

El esqueleto de la operación:

```
"r.patch input=$1,$2 output="$r"
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la primera capa de entrada y \$2 es el parámetro nombre de la segunda capa de entrada.

OPERACIÓN_RECLASS

Es una operación unaria aplicada sobre una capa raster de entrada. Se obtiene como resultado una reclasificación de una capa raster de entrada. Es necesario pasarle la ruta del fichero de reclasificación.

En el fichero de reclasificación se configuran los niveles para la reclasificación de valores. Un ejemplo del contenido de un fichero de reclasificación:

```
1 thru 10 = 1  
11 thru 20 = 2  
21 thru 30 = 3  
30 thru 40 = NULL
```

El esqueleto de la operación:

```
"r.reclass input=$1 output="$r rules="$rules" "
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la primera capa de entrada y \$rules es el parámetro de la ruta del fichero de reclasificación.

OPERACIÓN_REGION

Es una operación de configuración de la región de trabajo en GRASS. El marco de trabajo en GRASS es la región configurada. Se le tiene que proporcionar los límites de la región en valores: norte, sur, este y oeste.

Se han fijado los valores de resolución del norte-sur (nsres) y del este-oeste (ewres), y el número de columnas (cols) y filas (rows) en la nueva región.

El esqueleto de la operación:

```
"g.region n=$n s=$s e=$e w=$w nsres=300 ewres=300 rows=3224  
cols=3453"
```

Donde \$r es el parámetro nombre del resultado, \$n es el parámetro valor del norte, \$s es el parámetro valor del sur, \$e es el parámetro valor del este, y \$w es el parámetro valor del oeste.

OPERACIÓN_V_IN

Es una operación unaria aplicada sobre una capa vectorial de entrada. Es la operación de importación de capas vectoriales ubicadas en el sistema de ficheros o PostGIS al SIG GRASS para poder operar con dicha capa dentro de este entorno.

Hay que proporcionar la capa de importación. Según los datos geográficos se pueden distribuir en diferentes capas dentro de una capa vectorial. Por defecto la información se dispone en la capa 1.

El esqueleto de la operación:

```
"v.in.ogr -o "dsn=$1" layer=$l output=$r "
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la capa de entrada y \$l es el parámetro de la capa de importación.

OPERACIÓN_V_IN_CONDITIONAL

Es una operación unaria aplicada sobre una capa vectorial de entrada. Es la operación de importación de capas vectoriales ubicadas en el sistema de ficheros o PostGIS al SIG GRASS cumpliendo una condición de entrada sobre la tabla de atributos que lleva asociada. Se importarán aquellos puntos, líneas o polígonos que cumplan dicha condición.

El esqueleto de la operación:

```
"v.in.ogr -o "dsn=$1" output=$r layer=$l where="$conditional"
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la capa de entrada, \$l es el parámetro de la capa de importación y \$conditional es la frase condicional.

Si el consecuente de la frase condicional contiene espacios en blanco debe estar comprendido por comillas simples. Un ejemplo:

Agrup='Blando cohesivo'

OPERACIÓN_V_TO_RAST

Es una operación unaria aplicada sobre una capa vectorial de entrada. Es la operación de rasterización de una capa vectorial de entrada. Se rasterizan los elementos de tipo punto, línea o polígono a una capa raster con el valor que se especifique de entrada.

El esqueleto de la operación:

```
"v.to.rast input=$1 layer=1 type=point,line,area output=$r  
use=val value=$value rows=4096 "
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la capa de entrada y \$value es el parámetro del valor para asignar en la reclasificación.

OPERACIÓN_V_TO_RAST_COLUMN

Es una operación unaria aplicada sobre una capa vectorial de entrada. Es la operación de rasterización de una capa vectorial de entrada asignando el valor de una columna en particular a los elementos tipo punto, línea y polígono.

El esqueleto de la operación:

```
"v.to.rast input=$1 layer=1 type=point,line,area output=$r  
use=attr column=$column rows=4096"
```

Donde \$r es el parámetro nombre del resultado, \$1 es el parámetro nombre de la capa de entrada y \$column es el parámetro del nombre del campo de la tabla asociada a la capa vectorial para la rasterización.

4.5.2. Estudio de las entradas de las operaciones

En este apartado se realiza un estudio sobre las entradas de las operaciones descritas en el apartado anterior. Las entradas pueden proceder de: los expertos, los parámetros del usuario y los parámetros del dispositivo.

En la base de datos, la entrada de los expertos se configura en el campo *Comentario*, la entrada de los parámetros del usuario se establece en el campo *NombresParamInst* y la entrada de los parámetros del dispositivo se establece en el campo *NombresParamDispositivo*.

Es el paso previo a la programación de los criterios fijados en la aplicación DSS.

OPERACIÓN_ANDBOOLEANO

- No tiene entrada de expertos
- No tiene entrada de parámetros instanciados
- No tiene entrada de parámetros del dispositivo

OPERACIÓN_ANDFUZZY

- No tiene entrada de expertos
- No tiene entrada de parámetros instanciados
- No tiene entrada de parámetros del dispositivo

OPERACIÓN_BUFFER

Se necesita parametrizar $\$distance$, es un único parámetro que puede proceder de los expertos, de los parámetros de usuario o de los parámetros del dispositivo.

- Puede tener entrada de expertos
- Puede tener entrada de parámetros instanciados
- Puede tener entrada de parámetros del dispositivo

OPERACIÓN_FUZZY

Se necesita parametrizar $\$A$, $\$B$, $\$C$ y $\$D$, son cuatro parámetros que pueden proceder de los expertos, de los parámetros de usuario o de los parámetros del dispositivo.

- Puede tener entrada de expertos
- Puede tener entrada de parámetros instanciados
- Puede tener entrada de parámetros del dispositivo

OPERACIÓN_MLP

Se necesita parametrizar $\$p1$ y $\$p2$, son dos parámetros que proceden de los parámetros de usuario.

- No tiene entrada de expertos (al menos que sea una operación binaria sin ampliar a N-aria)
- Puede tener entrada de parámetros instanciados
- No tiene entrada de parámetros del dispositivo.

OPERACIÓN_OWA2

Se necesita parametrizar $\$p1$, $\$p2$, $\$z1$ y $\$z2$, son cuatro parámetros que pueden proceder de los expertos o de los parámetros del usuario.

- Puede tener entrada de expertos
- Puede tener entrada de parámetros instanciados
- No tiene entrada de parámetros del dispositivo

OPERACIÓN_OWA3

Se necesita parametrizar $\$p1$, $\$p2$, $\$p3$, $\$z1$, $\$z2$ y $\$z3$, son seis parámetros que pueden proceder de los expertos o de los parámetros del usuario.

- Puede tener entrada de expertos
- Puede tener entrada de parámetros instanciados
- No tiene entrada de parámetros del dispositivo

OPERACIÓN_R_BUFFER

Se necesita parametrizar $\$distance$, es un único parámetro que puede proceder de los expertos, de los parámetros de usuario o de los parámetros del dispositivo.

- Puede tener entrada de expertos
- Puede tener entrada de parámetros instanciados
- Puede tener entrada de parámetros del dispositivo

OPERACIÓN_R_COPY_MAPCALC

- No tiene entrada de expertos
- No tiene entrada de parámetros instanciados
- No tiene entrada de parámetros del dispositivo

OPERACIÓN_R_DISTANCE

- No tiene entrada de expertos
- No tiene entrada de parámetros instanciados
- No tiene entrada de parámetros del dispositivo

OPERACIÓN_R_IN

- No tiene entrada de expertos
- No tiene entrada de parámetros instanciados
- No tiene entrada de parámetros del dispositivo

OPERACIÓN_R_MAX

- No tiene entrada de expertos
- No tiene entrada de parámetros instanciados
- No tiene entrada de parámetros del dispositivo

OPERACIÓN_R_MINUS

- No tiene entrada de expertos
- No tiene entrada de parámetros instanciados
- No tiene entrada de parámetros del dispositivo

OPERACIÓN_R_MINUS_ESCALAR

Se necesita parametrizar $\$n$, es un único parámetro que puede proceder de los expertos, de los parámetros del usuario o de los parámetros del dispositivo.

- Puede tener entrada de expertos
- Puede tener entrada de parámetros instanciados
- Puede tener entrada de parámetros del dispositivo

OPERACIÓN_RECLASS

Por una parte, se necesita introducir la cadena de niveles de reclasificación, viene definido en la entrada de los expertos, la cadena introducida vendrá de la forma “nivel/niveles, valor de nivel/niveles, ...” (Ej, "0-*\$valor*, 1, *, 0"). Esta cadena de niveles viene definida con un parámetro *\$valor* que puede proceder de los parámetros del usuario o del dispositivo.

- Tiene entrada de expertos
- Puede tener entrada de parámetros instanciados
- Puede tener entrada de parámetros del dispositivo

OPERACIÓN_REGION

- No tiene entrada de expertos
- No tiene entrada de parámetros instanciados
- No tiene entrada de parámetros del dispositivo
- Se configuran los parámetros según las coordenadas de la región del escenario

OPERACIÓN_V_IN

- No tiene entrada de expertos
- No tiene entrada de parámetros instanciados
- No tiene entrada de parámetros del dispositivo

OPERACIÓN_V_TO_RAST

Se necesita parametrizar *\$value*, es un único parámetros que procede de los expertos.

- Puede tener entrada de expertos
- Puede tener entrada de parámetros instanciados
- No tienen entrada de parámetros del dispositivo

OPERACIÓN_V_TO_RAST_COLUMN

Se necesita parametrizar *\$column*, es un único parámetro que puede proceder de los expertos o de los parámetros del dispositivo. En este caso, prevalece la entrada de los expertos, si no existe procedencia de expertos (se pone el nombre de la columna), se rasteriza por la columna del tipo de dispositivo.

- Puede tener entrada de expertos
- No tiene entrada de parámetros instanciados
- Puede tener entrada de parámetros del dispositivo

OPERACIÓN_R_PATCH

- No tiene entrada de expertos
- No tiene entrada de parámetros instanciados
- No tiene entrada de parámetros del dispositivo

OPERACIÓN_R_OUT

- No tiene entrada de expertos
- No tiene entrada de parámetros instanciados
- No tiene entrada de parámetros del dispositivo

OPERACIÓN_R_NULL

Se necesita parametrizar $\$null$, es un único parámetro que puede proceder de los expertos, de los parámetros de usuario o de los parámetros del dispositivo.

- Puede tener entrada de expertos
- Puede tener entrada de parámetros instanciados
- Puede tener entrada de parámetros del dispositivo

OPERACIÓN_R_FORM

Se necesita parametrizar $\$valor$, $\$A$ y $\$B$. $\$valor$ puede proceder de los expertos o de los parámetros de usuario, y $\$A$ y $\$B$ proceden de los expertos.

- Tiene entrada de expertos
- Puede tener entrada de parámetros instanciados
- No tiene entrada de parámetros del dispositivo

OPERACIÓN_R_AND_ESCALAR

Se necesita parametrizar $\$n$, es un único parámetro que puede proceder de los expertos, de los parámetros de usuario o de los parámetros del dispositivo.

- Puede tener entrada de expertos
- Puede tener entrada de parámetros instanciados
- Puede tener entrada de parámetros del dispositivo

OPERACIÓN_V_IN_CONDITIONAL

Se necesita parametrizar $\$conditional$, es un único parámetro que procede de los expertos.

- Tiene entrada de expertos
- No tiene entrada de parámetros instanciados
- No tiene entrada de parámetros del dispositivo

3.5.3. Tabla Operaciones en la base de datos

A continuación se muestra la tabla de las operaciones configuradas en el sistema. Cada una de ellas lleva asociado un *ido* (número de identificación), una *cadena de GRASS* y un *nombre de tipo*. El número *ido* es necesario para la programación de los criterios.

ido [PK] integer	CadenaGRASS character varying(3000)	NombreTipo character varying(50)
1	g.region n=\$n s=\$s e=\$e w=\$w nres=300 ewres=300 rows=3224 cols=3453	OPERACION_REGION
2	v.in.ogr -o "dsn=\$1" layer=\$1 output=\$r	OPERACION_V_IN
3	v.to.rast input=\$1 layer=1 type=point,line,area output=\$r use=val value=\$value rows=4096	OPERACION_V_TO_RAST
4	r.grow.distance input=\$1 distance=\$r metric=euclidean	OPERACION_R_DISTANCE
5	r.null map=\$1 null=\$null	OPERACION_R_NULL
6	r.reclass input=\$1 output=\$r rules=\$rules	OPERACION_RECLASS
7	v.buffer input=\$1 output=\$r type=point,line,area layer=1 distance=\$distance	OPERACION_BUFFER
8	r.mapcalc "\$r = if(isnull(\$1) , null(), if((\$1 <= \$A \$1 >= \$D), 0.0, if((\$1 > \$A && \$1 < \$B), eval((\$1 - \$	OPERACION_FUZZY
9	r.mapcalc "\$r = \$p1 * \$1 + \$p2 * \$2"	OPERACION_MLP
10	r.mapcalc "\$r = if(isnull(\$1) isnull(\$2) , null(), if(\$1 <= \$2 , \$1 , \$2))"	OPERACION_ANDFUZZY
11	r.mapcalc "\$r = \$1 * \$2 "	OPERACION_ANDBOOLENO
12	r.mapcalc "\$r = if(isnull(\$1) isnull(\$2) , null(),if(\$1 >= \$2 , eval(((\$1 * \$p1 * \$z1) + (\$2 * \$p2 * \$z2	OPERACION_OWA2
13	r.mapcalc "\$r = if(isnull(\$1) isnull(\$2) isnull(\$3), null(),if(\$1 >= \$2 && \$2 >= \$3, eval((\$1*\$p1*\$z1) +	OPERACION_OWA3
14	r.in.gdal -o input="\$1" output=\$r	OPERACION_R_IN
15	r.mapcalc "\$r = if(isnull(\$1) isnull(\$2) , null(), if(\$1 >= \$2 , \$1 , \$2))"	OPERACION_R_MAX
16	r.out.gdal input=\$1 output=\$r"	OPERACION_R_OUT
17	r.mapcalc "\$r = \$1 * \$n"	OPERACION_R_AND_ESCALAR
18	r.buffer input=\$1 output=\$r units=meters distances=\$distance,\$distance	OPERACION_R_BUFFER
19	r.mapcalc "\$r = \$1"	OPERACION_R_COPY_MPCALC
20	v.to.rast input=\$1 layer=1 type=point,line,area output=\$r use=attr column=\$column rows=4096	OPERACION_V_TO_RAST_COLUMN
21	r.mapcalc "\$r = \$1 - \$2"	OPERACION_R_MINUS
22	r.mapcalc "\$r = \$1 - \$n"	OPERACION_R_MINUS_ESCALAR
23	r.patch input=\$1,\$2 output=\$r	OPERACION_R_PATCH
24	r.mapcalc "\$r = if(\$1 >= 0 && \$1 < \$valor, \$A, \$B)"	OPERACION_R_FORM
25	v.in.ogr -o "dsn=\$1" output=\$r layer=\$1 where="\$sconditional"	OPERACION_V_IN_CONDITIONAL

Figura 14: Tabla de operaciones configuradas en el sistema

3.5.4. Programación de criterios

Una vez definidos los criterios de la aplicación por los expertos, tienen que definir las operaciones que intervienen en la ejecución del criterio. En la base de datos corresponde con el relleno de la tabla *Criterio_Operacion*.

Por cada operación que interviene en el criterio se configura:

- El orden de aplicación de la operación.
- Las capas que intervienen de entrada, ya sea procedente de la BBDD OceanLider o un resultado de una operación anterior. Si es una capa de entrada de la BBDD OceanLider, se configurará los campos *NombreBBDDOL* y *NombreTablaOL*. Si es una/s capa/s de entrada de un/os resultado/os de una operación anterior, se debe configurar el campo *Predece*, como puede tener de uno a varias entradas se configura como conjunto de enteros, que corresponden al número de orden de la operación del resultado.
- La entrada de los parámetros procedentes de los usuarios, del dispositivo o de los expertos. Si se quiere hacer uso del campo *Comentario*, *NombresParamInst* o *NombresParamDispositivo*, se tiene que activar el campo *Display* de cada uno de ellos.

Cada criterio necesita la operación de importación de una capa (para raster es OPERACIÓN_V_IN o para vectorial es OPERACIÓN_R_IN) dentro del entorno SIG GRASS para poder operar con dicha capa.

Si se necesita importar una capa vectorial, ésta tendrá que ser rasterizada para posteriores ejecuciones. Los resultados finales e intermedios serán capas raster.

3.5.4.1. Programación de los criterios definidos en OceanLider

3.5.4.1.1. Criterios físicos

- **Criterio distancia a la costa**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_V_IN"
5. "OPERACION_V_TO_RAST"
6. "OPERACION_R_NULL"
7. "OPERACION_ANDBOOLEANO"
8. "OPERACION_R_AND_ESCALAR"
9. "OPERACION_FUZZY"
10. "OPERACION_R_NULL"

Se calcula la distancia a la línea de costa y sobre este mapa se suprimen las porciones de tierra procedente de la capa de Comunidades Autónomas. Sobre el mapa de distancias se realiza la operación de fuzzificación.

idc integer	ido integer	NombreBDDOL character varying(50)	NombreTablaOl character varying(50)	Precede integer[]	Orden integer	DisplayComentario boolean	Comentario character varying(200)	DisplayParamInst boolean	NombresParamInst character varying(50)	DisplayParamDispositivo boolean	NombresParamDispositivo character varying(50)
2	2	BBDDOceanLider	lin costa		1	f		f		f	
2	3	-	-	{1}	2	t	value=1	f		f	
2	4	-	-	{2}	3	f		f		f	
2	2	BBDDOceanLider	ccaa		4	f		f		f	
2	3	-	-	{4}	5	t	value=0	f		f	
2	5	-	-	{5}	6	t	null=1	f		f	
2	11	-	-	{3,5}	7	f		f		f	
2	17	-	-	{7}	8	t	\$n=100	f		f	
2	8	-	-	{8}	9	f		t	Valor A,Valor B, Va	f	
2	5	-	-	{9}	10	t	value=0	f		f	

Figura 15: Tabla de programación del criterio distancia a la costa

- **Criterio profundidad**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_R_IN"
2. "OPERACION_FUZZY"

Se calcula la operación fuzzy sobre la capa raster de Profundidad importado en el entorno SIG GRASS.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaOL character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character var	DisplayParamIns boolean	NombresParamInst character varying(5)	DisplayParamDis boolean	NombresParamDispositivo character varying(50)
3	14	R	profundidad		1	f		f		f	
3	8	-	-	{1}	2	f		f		t	ProfMax, ProfMin

Figura 16: Tabla de programación del criterio profundidad

- **Grupo supervivencia del dispositivo**
 - *Criterio altura de ola significativa*

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_R_IN"
2. "OPERACION_RECLASS"
3. "OPERACION_R_COPY_MAPCALC"
4. "OPERACION_R_NULL"

Se hace la reclasificación sobre la capa raster de altura media de ola significativa y es asigna 0 a los valores nulos.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaOL character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character varying	DisplayParamIns boolean	NombresParamInst character varying(5)	DisplayParamDis boolean	NombresParamDispc character varying(5)
4	14	R	unionhsmedia		1	f		f		f	
4	6	-	-	{1}	2	t	0-valor,1,*,0	f		t	HsMax
4	19	-	-	{2}	3	f		f		f	
4	5	-	-	{3}	4	t	null=0	f		f	

Figura 17: Tabla de programación del criterio Altura de ola significativa

- *Criterio velocidad de corriente*

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_R_IN"
2. "OPERACION_RECLASS"
3. "OPERACION_R_COPY_MAPCALC"
4. "OPERACION_R_NULL"

Se hace la reclasificación sobre el mapa de entrada raster de marea viva y se asigna 0 a los valores nulos.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaOL character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character varying	DisplayParamIns boolean	NombresParamInst character varying(5)	DisplayParamDis boolean	NombresParamDis character varying(5)
5	14	R	medviv		1	f		f		f	
5	6	-	-	{1}	2	t	0=\$valor,1,*,0	f		t	CoMax
5	19	-	-	{2}	3	f		f		f	
5	5	-	-	{3}	4	t	null=0	f		f	

Figura 18: Tabla de programación del criterio Velocidad de corriente

4.5.4.1.2. Criterios operacionales

- Grupo mantenimiento de la posición
 - Criterio facilidad de anclaje al fondo (composición del sustrato marino)

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN_CONDITIONAL"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_V_IN_CONDITIONAL"
4. "OPERACION_V_TO_RAST"
5. "OPERACION_V_IN_CONDITIONAL"
6. "OPERACION_V_TO_RAST"
7. "OPERACION_V_IN_CONDITIONAL"
8. "OPERACION_V_TO_RAST"
9. "OPERACION_R_PATCH"
10. "OPERACION_R_PATCH"
11. "OPERACION_R_PATCH"
12. "OPERACION_R_NULL"

Se importa los polígonos, líneas o puntos de la capa vectorial de tipos de sustratos de fondos marinos (*Geofis*) pertenecientes a roca, vegetado, blando cohesivo y blando no cohesivo de forma independiente, y se rasteriza cada una de las capas generadas con el valor acordado en la interfaz. Se hace una unión de todas las capas y se asigna valor 1 a los valores nulos de la capa resultante.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaOL character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character varying(200)	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
6	25	BBDDOceanLide:geofis			1	t	Agrup='Roca'	f		f	
6	3	-	-	{1}	2	f		t	Roca	f	
6	25	BBDDOceanLide:geofis			3	t	Agrup='Vegetado'	f		f	
6	3	-	-	{3}	4	f		t	Vegetado	f	
6	25	BBDDOceanLide:geofis			5	t	Agrup='Blando cohesivo'	f		f	
6	3	-	-	{5}	6	f		t	Blando cohesivo	f	
6	25	BBDDOceanLide:geofis			7	t	Agrup='Blando no cohesivo'	f		f	
6	3	-	-	{7}	8	f		t	Blando no cohesivo	f	
6	23	-	-	{2,4}	9	f		f		f	
6	23	-	-	{6,8}	10	f		f		f	
6	23	-	-	{9,10}	11	f		f		f	
6	5	-	-	{11}	12	t	null=1	f		f	

Figura 19: Tabla de programación del criterio Facilidad de anclaje al fondo (composición del sustrato marino)

o Criterio facilidad de anclaje al fondo (morfología del fondo marino)

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN_CONDITIONAL"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_V_IN_CONDITIONAL"
4. "OPERACION_V_TO_RAST"
5. "OPERACION_R_PATCH"
6. "OPERACION_R_NULL"

Se importan los polígonos, líneas o puntos asociados al primer grupo (denominado como Grupo0) de la capa de los accidentes geográficos (*Formas*) y se rasteriza con el valor asignado en la interfaz. Posteriormente se hace las mismas operaciones para el segundo grupo (denominado como Grupo1). Se hace una unión de las dos capas raster y se asigna valor 1 a los valores nulos de la capa resultante.

Los tipos de accidentes de la capa Formas que pertenecen a cada grupo se refleja en la siguiente tabla.

NOMBRE	NOMBRE CAPA	METADATOS	NOMBRE DE GRUPO	TIPO DE ACCIDENTE GEOGRÁFICO	CORRIENTES		OLEAJE	
					FUERA DE COSTA (Apoyados sobre base de gravedad o pilotados)	FUERA DE COSTA (Fondeados)	FUERA DE COSTA (Apoyados sobre base de gravedad o pilotados)	FUERA DE COSTA (Flotantes Fondeados)
Accidentes geográficos submarinos	Formas_P.shp	Cartografía vectorial lineal que se corresponde con los accidentes geográficos submarinos más reseñables de la península Ibérica, Ceuta y Melilla. Se encuentra en ETRS89 huso 30.	Grupo1	Afloramiento rocoso	1	1	1	1
			Grupo1	Marcas de arrastre	1	1	1	1
			Grupo1	Plataforma continental	1	1	1	1
			Grupo1	Playas sumergidas	1	1	1	1
			Grupo1	Ripples	1	1	1	1
			Grupo1	Talud continental	1	1	1	1
			Grupo0	Abanico deltaico	0	0	0	0
			Grupo0	Abanico deltaico profundo	0	0	0	0
			Grupo0	Cañon submarino	0	0	0	0
			Grupo0	Cañon submarino	0	0	0	0
			Grupo0	Llanuras abisales	0	0	0	0
			Grupo0	Megaripples	0	0	0	0
			Grupo0	Megaripples	0	0	0	0
			Grupo0	Morfología de crestas y valles	0	0	0	0
			Grupo0	Ondas de arena	0	0	0	0
Grupo0	Relieve volcanico	0	0	0	0			

Figura 20: Tabla de agrupaciones de la capa vectorial de accidentes geográficos (Formas)

idc integ	ido integ	NombreBDDOL character varying	NombreTablaOL character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character varying	DisplayParamIns boolean	NombresParamIns character varying(DisplayParamDis boolean	NombresParamDis character varying
7	25	BBDDOceanLider	formas		1	t	Agrup='Grupo0'	f		f	
7	3	-	-	{1}	2	f		t	Grupo0	f	
7	25	BBDDOceanLider	formas		3	t	Agrup='Grupo1'	f		f	
7	3	-	-	{3}	4	f		t	Grupo1	f	
7	23	-	-	{2,4}	5	f		f		f	
7	5	-	-	{5}	6	t	null=1	f		f	

Figura 21: Tabla de programación del criterio Facilidad de anclaje al fondo (morfología del fondo marino)

- **Grupo viabilidad de instalación en zonas ocupadas**
 - *Criterio viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español*

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"

Se importa la capa vectorial perteneciente a las ayudas a la navegación del sistema portuario español y se rasteriza asignando valor 0 a aquellos elementos tipo punto, línea o polígono. Sobre esta capa raster se calcula el mapa de distancias y se aplica la fuzzificación con los valores asignados en la interfaz.

idc	ido	NombreBDDOL	NombreTablaOL	Precede	Orden	DisplayComentar	Comentario	DisplayParamIns	NombresParamInst	DisplayParamDis	NombresParamDis
integi	integi	character varying	character varying	integer[]	integer	boolean	character va	boolean	character varying(50)	boolean	character varying
8	2	BBDDOceanLider	aton		1	f		f		f	
8	3	-	-	{1}	2	t	value=0	f		f	
8	4	-	-	{2}	3	f		f		f	
8	17	-	-	{3}	4	t	\$n=100	f		f	
8	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	

Figura 22: Tabla de programación del criterio Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español

- *Criterio viabilidad de instalación en zonas designadas como Dominio Público Portuario*

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"

Se importa la capa vectorial de la delimitación del Dominio Público Portuario y se rasteriza asignando valor 0 a aquellos elementos tipo punto, línea o polígono. Sobre esta capa raster se calcula el mapa de distancias y se aplica la fuzzificación con los valores asignados en la interfaz.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaOL character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
9	2	BDDOceanLider	dpp		1	f		f		f	
9	3	-	-	{1}	2	t	value=0	f		f	
9	4	-	-	{2}	3	f		f		f	
9	17	-	-	{3}	4	t	\$n=100	f		f	
9	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	

Figura 23: Tabla de programación del criterio Viabilidad de instalación en zonas designadas como Dominio Público Portuario

- **Criterio viabilidad de instalación en zonas preseleccionadas para la extracción de áridos**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"

Se importa la capa vectorial de las zonas recomendadas para la extracción de áridos y se rasteriza asignando valor 0 a aquellos elementos tipo punto, línea o polígono. Sobre esta capa raster se calcula el mapa de distancias y se aplica la fuzzificación con los valores asignados en la interfaz.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaOL character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
10	2	BDDOceanLider	ext aridos		1	f		f		f	
10	3	-	-	{1}	2	t	value=0	f		f	
10	4	-	-	{2}	3	f		f		f	
10	17	-	-	{3}	4	t	\$n=100	f		f	
10	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	

Figura 24: Tabla de programación del criterio Viabilidad de instalación en zonas preseleccionadas para la extracción de áridos

- **Criterio viabilidad de instalación en zonas ocupadas por conducciones submarinas**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"

Se importa la capa vectorial de cables eléctricos submarinos, emisarios o tuberías y se rasteriza asignando valor 0 a aquellos elementos tipo punto, línea o polígono. Sobre esta capa raster se calcula el mapa de distancias y se aplica la fuzzificación con los valores asignados en la interfaz.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaOL character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
11	2	BBDDOceanLider	conducciones		1	f		f		f	
11	3	-	-	{1}	2	t	value=0	f		f	
11	4	-	-	{2}	3	f		f		f	
11	17	-	-	{3}	4	t	\$n=100	f		f	
11	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	

Figura 25: Tabla de programación del criterio Viabilidad de instalación en zonas ocupadas por conducciones submarinas

- **Criterio viabilidad de instalación en zonas preseleccionadas para el vertido de material de dragado**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"

Se importa la capa vectorial de vertidos y se rasteriza asignando valor 0 a aquellos elementos tipo punto, línea o polígono. Sobre esta capa raster se calcula el mapa de distancias y se aplica la fuzzificación con los valores asignados en la interfaz.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaOL character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
12	2	BBDDOceanLider	vertido dragado		1	f		f		f	
12	3	-	-	{1}	2	t	value=0	f		f	
12	4	-	-	{2}	3	f		f		f	
12	17	-	-	{3}	4	t	\$n=100	f		f	
12	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	

Figura 26: Tabla de programación del criterio Viabilidad de instalación en zonas preseleccionadas para el vertido de material de dragado

- **Criterio viabilidad de instalación en zonas ocupadas por sensores instrumentales de medición de parámetros oceanográficos**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"

Se importa la capa vectorial de los sensores instrumentales de medición de parámetros oceanográficos y se rasteriza asignando valor 0 a aquellos elementos tipo punto, línea o polígono. Sobre esta capa raster se calcula el mapa de distancias y se aplica la fuzzificación con los valores asignados en la interfaz.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaOL character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
13	2	BDDOceanLider	rmo		1	f		f		f	
13	3	-	-	{1}	2	t	value=0	f		f	
13	4	-	-	{2}	3	f		f		f	
13	17	-	-	{3}	4	t	\$n=100	f		f	
13	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	

Figura 27: Tabla de programación del criterio Viabilidad de instalación en zonas ocupadas por sensores instrumentales de medición de parámetros oceanográficos

- o Criterio viabilidad de instalación en zonas ocupadas por dispositivos de separación del tráfico marítimo

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"

Se importa la capa vectorial de los dispositivos de separación del tráfico marítimo y se rasteriza asignando valor 0 a aquellos elementos tipo punto, línea o polígono. Sobre esta capa raster se calcula el mapa de distancias y se aplica la fuzzificación con los valores asignados en la interfaz.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaOL character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
14	2	BDDOceanLider	dat		1	f		f		f	
14	3	-	-	{1}	2	t	value=0	f		f	
14	4	-	-	{2}	3	f		f		f	
14	17	-	-	{3}	4	t	\$n=100	f		f	
14	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	

Figura 28: Tabla de programación del criterio Viabilidad de instalación en zonas ocupadas por dispositivos de separación del tráfico marítimo

- Grupo viabilidad de instalación en zonas restringidas
 - o Criterio viabilidad de instalación en zonas restringidas para uso militar

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"

Se importa la capa vectorial de la zonas de uso militar y se rasteriza asignando valor 0 a aquellos elementos tipo punto, línea o polígono. Sobre esta capa raster se calcula el mapa de distancias y se aplica la fuzzificación con los valores asignados en la interfaz.

idc integi	ido integi	NombreBBDDOL character varying	NombreTablaO character varyi	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character vai	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
15	2	BBDDOceanLider	militar		1	f		f		f	
15	3	-	-	{1}	2	t	value=0	f		f	
15	4	-	-	{2}	3	f		f		f	
15	17	-	-	{3}	4	t	\$n=100	f		f	
15	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	

Figura 29: Tabla de programación del criterio Viabilidad de instalación en zonas restringidas para uso militar

- **Criterio viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"

Se importa la capa vectorial de la zonas de las zonas restringidas por la presencia de cables y se rasteriza asignando valor 0 a aquellos elementos tipo punto, línea o polígono. Sobre esta capa raster se calcula el mapa de distancias y se aplica la fuzzificación con los valores asignados en la interfaz.

idc integi	ido integi	NombreBBDDOL character varying	NombreTablaO character varyi	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character vai	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
16	2	BBDDOceanLider	zona cables		1	f		f		f	
16	3	-	-	{1}	2	t	value=0	f		f	
16	4	-	-	{2}	3	f		f		f	
16	17	-	-	{3}	4	t	\$n=100	f		f	
16	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	

Figura 30: Tabla de programación del criterio Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías

- **Criterio viabilidad de instalación en zonas restringidas por la presencia de vertidos.**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"

Se importa la capa vectorial de las zonas de vertido de material de dragado y se rasteriza asignando valor 0 a aquellos elementos tipo punto, línea o polígono. Sobre esta capa raster se calcula el mapa de distancias y se aplica la fuzzificación con los valores asignados en la interfaz.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaOL character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
17	2	BBDDOceanLider	vertidos		1	f		f		f	
17	3	-	-	{1}	2	t	value=0	f		f	
17	4	-	-	{2}	3	f		f		f	
17	17	-	-	{3}	4	t	\$n=100	f		f	
17	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	

Figura 31: Tabla de programación del criterio Viabilidad de instalación en zonas restringidas por la presencia de vertidos

- **Criterio viabilidad de instalación en zonas restringidas por la existencia de vertidos de material militar**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"

Se importa la capa vectorial de las zonas vertidos de material militar y se rasteriza asignando valor 0 a aquellos elementos tipo punto, línea o polígono. Sobre esta capa raster se calcula el mapa de distancias y se aplica la fuzzificación con los valores asignados en la interfaz.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaOL character varying(50)	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
18	2	BBDDOceanLider	vertido material militar		1	f		f		f	
18	3	-	-	{1}	2	t	value=0	f		f	
18	4	-	-	{2}	3	f		f		f	
18	17	-	-	{3}	4	t	\$n=100	f		f	
18	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	

Figura 32: Tabla de programación del criterio Viabilidad de instalación en zonas restringidas por la existencia de vertidos de material militar

4.5.4.1.3. Criterios de recurso energético

Es dependiente del tipo de dispositivo: se aplica la densidad de potencia para tipos de dispositivo de oleaje y la marea media para tipos de dispositivo de corrientes.

- **Criterio mareas vivas y muertas en condiciones de marea media**

Se aplican las siguientes fórmulas:

- De 0 a VALOR_MÁXIMO m/s aplicamos ecuación:

$$\text{Valor Índice} = \left(\frac{\text{valor marea media}}{\text{VALOR_MAXIMO}} \right)^3$$

- Mayor de VALOR_MÁXIMO m/s:

$$\text{Valor Índice} = 1$$

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_R_IN"
2. "OPERACION_R_IN"
3. "OPERACION_MLP"
4. "OPERACION_R_FORM"
5. "OPERACION_R_NULL"

Se hace la importación de las capas raster de mareas vivas y mareas muertas, se hace una media lineal sobre estas dos y se aplican las fórmulas en los intervalos definidos por el *valor_máximo* que ha introducido el usuario. Por último se aplica valor 0 a los valores nulos de la capa resultante.

idc	ido	NombreBDDOL	NombreTab	Precede	Orden	DisplayComentar	Comentario	DisplayParamIns	NombresParamIn:	DisplayParamDis	NombresParamDis
integ	integ	character varying	character v	integer[]	integer	boolean	character varying(200)	boolean	character varying	boolean	character varying
19	14	R	medviv		1	f		f		f	
19	14	R	medmue		2	f		f		f	
19	9	-	-	{1,2}	3	t	\$p1=0.5,\$p2=0.5	f		f	
19	24	-	-	{3}	4	t	\$A=eval((\$1/\$valor)^3),\$B=1	t	Valor Máximo	f	
19	5	-	-	{4}	5	t	null=0	f		f	

Figura 33: Tabla de programación del criterio mareas vivas y muertas en condiciones de marea media

- Criterio Densidad (para dispositivos de oleaje)

Se aplican las siguientes fórmulas:

- De 0 a VALOR_MÁXIMO KWh/m aplicamos ecuación:

$$\text{Valor Índice} = \frac{\text{valor potencia}}{\text{VALOR_MAXIMO}}$$

- Mayor de VALOR_MÁXIMO KWh/m:

$$\text{Valor Índice} = 1$$

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_R_IN"
2. "OPERACION_R_FORM"
3. "OPERACION_R_NULL"

Se importa la capa raster de potencia y se aplican las fórmulas en los intervalos definidos por el *valor_máximo* que ha introducido el usuario. Por último se aplica valor 0 a los valores nulos de la capa resultante.

idc integ	ido integ	NombreBDDOL character varying	NombreTab character v	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character varying(200)	DisplayParamIns boolean	NombresParamIn: character varying	DisplayParamDis boolean	NombresParamDi: character varying
20	14	R	pw		1	f		f		f	
20	24	-	-	{1}	2	t	\$A=eval (\$1/\$valor), \$B=1	t	Valor Máximo	f	
20	5	-	-	{2}	3	t	null=0	f		f	

Figura 34: Tabla de programación del criterio densidad

4.5.4.1.4. Criterios medio ambientales

- Grupo espacios protegidos
 - Criterio geoparques

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de geoparques y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integi	ido integi	NombreBDDOL character varying	NombreTabla character var	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
21	2	BDDOceanLider	geoparques		1	f		f		f	
21	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
21	4	-	-	{2}	3	f		f		f	
21	17	-	-	{3}	4	t	\$n=100	f		f	
21	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
21	23	-	-	{2,5}	6	f		f		f	

Figura 35: Tabla de programación del criterio Geoparques

o **Criterio reservas de la biosfera**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de reservas de la biosfera y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integi	ido integi	NombreBDDOL character varying	NombreTabla character var	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
22	2	BDDOceanLider	mabs		1	f		f		f	
22	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
22	4	-	-	{2}	3	f		f		f	
22	17	-	-	{3}	4	t	\$n=100	f		f	
22	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
22	23	-	-	{2,5}	6	f		f		f	

Figura 36: Tabla de programación del criterio Reservas de la biosfera

o **Criterio patrimonio de la humanidad**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"

3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de patrimonio de la humanidad y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integer	ido integer	NombreBDDOL character varying	NombreTabla character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character varying	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
23	2	BDDOceanLider	p_humanidad		1	f		f		f	
23	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
23	4	-	-	{2}	3	f		f		f	
23	17	-	-	{3}	4	t	\$n=100	f		f	
23	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
23	23	-	-	{2,5}	6	f		f		f	

Figura 37: Tabla de programación del criterio Patrimonio de la humanidad

- o **Criterio Red de Áreas Marinas Protegidas del Convenio OSPAR**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de los lugares incluidos en la Red de Áreas Marinas Protegidas del Convenio OSPAR y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integer	ido integer	NombreBDDOL character varying	NombreTabla character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character varying	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
24	2	BDDOceanLider	ospar amps		1	f		f		f	
24	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
24	4	-	-	{2}	3	f		f		f	
24	17	-	-	{3}	4	t	\$n=100	f		f	
24	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
24	23	-	-	{2,5}	6	f		f		f	

Figura 38: Tabla de programación del criterio Red de Áreas Marinas Protegidas del Convenio OSPAR

○ **Criterio Zonas Especialmente Protegidas de Importancia para el Mediterráneo**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de las zonas especialmente protegidas de importancia para el Mediterráneo y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc	ido	NombreBDDOL	NombreTabla	Precede	Orden	DisplayComentar	Comentario	DisplayParamIns	NombresParamIns	DisplayParamDis	NombresParamDis
integer	integer	character varying	character varying	integer[]	integer	boolean	character varying	boolean	character varying(50)	boolean	character varying
25	2	BBDDOceanLider	zepims		1	f		f		f	
25	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
25	4	-	-	{2}	3	f		f		f	
25	17	-	-	{3}	4	t	\$n=100	f		f	
25	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
25	23	-	-	{2,5}	6	f		f		f	

Figura 39: Tabla de programación del criterio Zonas Especialmente Protegidas de Importancia para el Mediterráneo

○ **Criterio sitios reconocidos con el Diploma Europeo de Calidad de Espacios Protegidos**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de los sitios reconocidos con el Diploma Europeo de Calidad de Espacios Protegidos y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integer	ido integer	NombreBDDOL character varying	NombreTabla character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
26	2	BBDDOceanLider	dipeuro		1	f		f		f	
26	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
26	4	-	-	{2}	3	f		f		f	
26	17	-	-	{3}	4	t	\$n=100	f		f	
26	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
26	23	-	-	{2,5}	6	f		f		f	

Figura 40: Tabla de programación del criterio Sitios reconocidos con el Diploma Europeo de Calidad de Espacios Protegidos

o **Criterio Lugares de Importancia Comunitaria**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de los Lugares de Importancia Comunitaria y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integer	ido integer	NombreBDDOL character varying	NombreTabla character var	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
27	2	BBDDOceanLider	lics		1	f		f		f	
27	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
27	4	-	-	{2}	3	f		f		f	
27	17	-	-	{3}	4	t	\$n=100	f		f	
27	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
27	23	-	-	{2,5}	6	f		f		f	

Figura 41: Tabla de programación del criterio Lugares de Importancia Comunitaria

o **Criterio Zonas de Especial Protección para las Aves**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de las zonas de especial protección para las aves y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integer	ido integer	NombreBDDOL character varying	NombreTabla character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character varying	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
28	2	BBDDOceanLider	zepas		1	f		f		f	
28	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
28	4	-	-	{2}	3	f		f		f	
28	17	-	-	{3}	4	t	\$n=100	f		f	
28	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
28	23	-	-	{2,5}	6	f		f		f	

Figura 42: Tabla de programación del criterio Zonas de Especial Protección para las Aves

o **Criterio Parques Nacionales**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de los parques naturales y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integer	ido integer	NombreBDDOL character varying	NombreTabla character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character varying	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
29	2	BBDDOceanLider	ppnn		1	f		f		f	
29	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
29	4	-	-	{2}	3	f		f		f	
29	17	-	-	{3}	4	t	\$n=100	f		f	
29	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
29	23	-	-	{2,5}	6	f		f		f	

Figura 43: Tabla de programación del criterio Parques Nacionales

o **Criterio Biotopos protegidos**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"

3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de los biotopos protegidos y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integer	ido integer	NombreBDDOL character varying	NombreTabla character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character varying	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
30	2	BBDDOceanLider	bbppro		1	f		f		f	
30	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
30	4	-	-	{2}	3	f		f		f	
30	17	-	-	{3}	4	t	\$n=100	f		f	
30	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
30	23	-	-	{2,5}	6	f		f		f	

Figura 44: Tabla de programación del criterio Biotopos Protegidos

o **Criterio Microrreservas**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de las microrreservas y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integer	ido integer	NombreBDDOL character varying	NombreTabla character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character varying	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
31	2	BBDDOceanLider	micror		1	f		f		f	
31	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
31	4	-	-	{2}	3	f		f		f	
31	17	-	-	{3}	4	t	\$n=100	f		f	
31	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
31	23	-	-	{2,5}	6	f		f		f	

Figura 45: Tabla de programación del criterio Microrreservas

○ **Criterio Monumentos Naturales**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de los monumentos naturales y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integ	ido integ	NombreBDDOL character varying	NombreTabla character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
32	2	BDDOceanLider	mmnat		1	f		f		f	
32	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
32	4	-	-	{2}	3	f		f		f	
32	17	-	-	{3}	4	t	\$n=100	f		f	
32	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
32	23	-	-	{2,5}	6	f		f		f	

Figura 46: Tabla de programación para el criterio Monumentos Naturales

○ **Criterio Parajes Naturales**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de parajes naturales y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integ	ido integ	NombreBDDOL character varying	NombreTabla character var	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character vai	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
33	2	BDDOceanLider	pparnnat		1	f		f		f	
33	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
33	4	-	-	{2}	3	f		f		f	
33	17	-	-	{3}	4	t	\$n=100	f		f	
33	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
33	23	-	-	{2,5}	6	f		f		f	

Figura 47: Tabla de programación para el criterio Parajes Naturales

o **Criterio Parques Naturales**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de parques naturales y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integ	ido integ	NombreBDDOL character varying	NombreTabla character var	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character vai	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
34	2	BDDOceanLider	ppnat		1	f		f		f	
34	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
34	4	-	-	{2}	3	f		f		f	
34	17	-	-	{3}	4	t	\$n=100	f		f	
34	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
34	23	-	-	{2,5}	6	f		f		f	

Figura 48: Tabla de programación para el criterio Parques Naturales

o **Criterio Reservas con ley de protección propia**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de las reservas con ley de protección propia y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de

idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaC character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character varying	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
35	2	BBDDOceanLider	rr ley prop		1	f		f		f	
35	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
35	4	-	-	{2}	3	f		f		f	
35	17	-	-	{3}	4	t	\$n=100	f		f	
35	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
35	23	-	-	{2,5}	6	f		f		f	

Figura 49: Tabla de programación para el criterio Reservas con ley de protección propia

o **Criterio Reservas Naturales**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de las reservas con ley de protección propia y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaC character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character varying	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
36	2	BBDDOceanLider	rrnn		1	f		f		f	
36	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
36	4	-	-	{2}	3	f		f		f	
36	17	-	-	{3}	4	t	\$n=100	f		f	
36	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
36	23	-	-	{2,5}	6	f		f		f	

Figura 50: Tabla de programación para el criterio Reservas Naturales

o **Criterio Zonas de Especial Protección de los Valores Naturales**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"

5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de zonas de especial protección de los valores naturales y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaC character var	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
37	2	BBDDOceanLider	zepvn		1	f		f		f	
37	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
37	4	-	-	{2}	3	f		f		f	
37	17	-	-	{3}	4	t	\$n=100	f		f	
37	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
37	23	-	-	{2,5}	6	f		f		f	

Figura 51: Tabla de programación para el criterio Zonas de Especial Protección de los Valores Naturales

- **Grupo Biodiversidad**
 - *Criterio Áreas de interés para la conservación de los cetáceos*

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de las áreas de interés para la conservación de los cetáceos y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaC character var	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
38	2	BBDDOceanLider	aicp		1	f		f		f	
38	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
38	4	-	-	{2}	3	f		f		f	
38	17	-	-	{3}	4	t	\$n=100	f		f	
38	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
38	23	-	-	{2,5}	6	f		f		f	

Figura 52: Tabla de programación para el criterio Áreas de interés para la conservación de los cetáceos

○ **Criterio Áreas compatibles con la figura de Parque Nacional**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de las áreas compatibles con la figura de Parque Nacional y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integ:	ido integ:	NombreBDDOL character varying	NombreTablaC character vary	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
39	2	BBDDOceanLider	aopn		1	f		f		f	
39	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
39	4	-	-	{2}	3	f		f		f	
39	17	-	-	{3}	4	t	\$n=100	f		f	
39	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
39	23	-	-	{2,5}	6	f		f		f	

Figura 53: Tabla de programación para el criterio Áreas compatibles con la figura de Parque Nacional

○ **Criterio Áreas importantes para las aves**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de las áreas importantes para las aves y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaC character var	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
40	2	BDDOceanLider	ibas		1	f		f		f	
40	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
40	4	-	-	{2}	3	f		f		f	
40	17	-	-	{3}	4	t	\$n=100	f		f	
40	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
40	23	-	-	{2,5}	6	f		f		f	

Figura 54: Tabla de programación para el criterio Áreas importantes para las aves

- Criterio Zonas propuestas para realizar el Inventario de la biodiversidad marina

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de las zonas propuestas para realizar el inventario de la biodiversidad marina y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaC character var	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
41	2	BDDOceanLider	ibm		1	f		f		f	
41	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
41	4	-	-	{2}	3	f		f		f	
41	17	-	-	{3}	4	t	\$n=100	f		f	
41	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
41	23	-	-	{2,5}	6	f		f		f	

Figura 55: Tabla de programación para el criterio Zonas propuestas para realizar el Inventario de la biodiversidad marina

- Criterio Áreas importantes para las aves marinas

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de las áreas importantes para las aves marinas y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integi	ido integi	NombreBDDOL character varying	NombreTablaC character vary	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
42	2	BBDDOceanLider	iba recom		1	f		f		f	
42	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
42	4	-	-	{2}	3	f		f		f	
42	17	-	-	{3}	4	t	\$n=100	f		f	
42	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
42	23	-	-	{2,5}	6	f		f		f	

Figura 56: Tabla de programación para el criterio Áreas importantes para las aves marinas

o **Criterio Zona marina especialmente sensible**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de las zonas marinas especialmente sensibles y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integi	ido integi	NombreBDDOL character varying	NombreTablaC character vary	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
43	2	BBDDOceanLider	zmes		1	f		f		f	
43	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
43	4	-	-	{2}	3	f		f		f	
43	17	-	-	{3}	4	t	\$n=100	f		f	
43	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
43	23	-	-	{2,5}	6	f		f		f	

Figura 57: Tabla de programación para el criterio Zona marina especialmente sensible

o **Criterio Rutas migratorias de cetáceos**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"

3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de las rutas migratorias de cetáceos y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaC character var	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
44	2	BDDOceanLider	cetaceos		1	f		f		f	
44	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
44	4	-	-	{2}	3	f		f		f	
44	17	-	-	{3}	4	t	\$n=100	f		f	
44	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
44	23	-	-	{2,5}	6	f		f		f	

Figura 58: Tabla de programación para el criterio Rutas migratorias de cetáceos

- **Grupo usos pesqueros**
 - *Criterio Instalaciones de acuicultura*

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_V_IN"
4. "OPERACION_V_TO_RAST"
5. "OPERACION_R_PATCH"
6. "OPERACION_R_DISTANCE"
7. "OPERACION_R_AND_ESCALAR"
8. "OPERACION_FUZZY"
9. "OPERACION_R_PATCH"

Se importan las capas vectoriales de las áreas designadas como instalaciones de acuicultura marina y de buffer a las instalaciones de acuicultura (procedencia de origen de datos diferente). Se rasterizan de forma independiente asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Se combinan y se aplica una operación de distancias sobre la capa raster de combinación. Posteriormente se aplica una operación fuzzy sobre el mapa de distancias con los valores configurados en la interfaz (valor A, valor B, valor C y valor D) y se combina con la capa raster de combinación, siendo la capa resultante.

idc integi	ido integi	NombreBDDOL character varying	NombreTablaOL character varying(5)	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
45	2	BBDDOceanLider	acuicultura		1	f		f		f	
45	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
45	2	BBDDOceanLider	acuicultura buf		3	f		f		f	
45	3	-	-	{3}	4	f		t	Valor de idoneidad	f	
45	23	-	-	{2,4}	5	f		f		f	
45	4	-	-	{5}	6	f		f		f	
45	17	-	-	{6}	7	t	\$n=100	f		f	
45	8	-	-	{7}	8	f		t	Valor A,Valor B,Valor C,Valor D	f	
45	23	-	-	{5,8}	9	f		f		f	

Figura 59: Tabla de programación para el criterio de Instalaciones de acuicultura

o **Criterio Almadrabas**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_V_IN"
4. "OPERACION_V_TO_RAST"
5. "OPERACION_R_PATCH"
6. "OPERACION_R_DISTANCE"
7. "OPERACION_R_AND_ESCALAR"
8. "OPERACION_FUZZY"
9. "OPERACION_R_PATCH"

Se importan las capas vectoriales de almadrabas y de buffer a las almadrabas (procedencia de origen de datos diferente). Se rasterizan de forma independiente asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Se combinan y se aplica una operación de distancias sobre la capa raster de combinación. Posteriormente se aplica una operación fuzzy sobre el mapa de distancias con los valores configurados en la interfaz (valor A, valor B, valor C y valor D) y se combina con la capa raster de combinación, siendo la capa resultante.

idc integi	ido integi	NombreBDDOL character varying	NombreTablaOL character varying(5)	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
46	2	BBDDOceanLider	almadrabas		1	f		f		f	
46	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
46	2	BBDDOceanLider	almadrabas buf		3	f		f		f	
46	3	-	-	{3}	4	f		t	Valor de idoneidad	f	
46	23	-	-	{2,4}	5	f		f		f	
46	4	-	-	{5}	6	f		f		f	
46	17	-	-	{6}	7	t	\$n=100	f		f	
46	8	-	-	{7}	8	f		t	Valor A,Valor B,Valor C,Valor D	f	
46	23	-	-	{5,8}	9	f		f		f	

Figura 60: Tabla de programación para el criterio de Almadrabas

o **Criterio Arrecifes**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"

3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de los arrecifes y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaOL character varying(5)	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
47	2	BDDOceanLider	arrecifes		1	f		f		f	
47	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
47	4	-	-	{2}	3	f		f		f	
47	17	-	-	{3}	4	t	\$n=100	f		f	
47	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
47	23	-	-	{2,5}	6	f		f		f	

Figura 61: Tabla de programación para el criterio de Arrecifes

o **Criterio Caladeros**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de los caladeros y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integer	ido integer	NombreBDDOL character varying	NombreTablaOL character varying(5)	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
48	2	BDDOceanLider	caladeros		1	f		f		f	
48	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
48	4	-	-	{2}	3	f		f		f	
48	17	-	-	{3}	4	t	\$n=100	f		f	
48	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
48	23	-	-	{2,5}	6	f		f		f	

Figura 62: Tabla de programación para el criterio Caladeros

o **Criterio Viveros**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_V_IN"
4. "OPERACION_V_TO_RAST"
5. "OPERACION_R_PATCH"
6. "OPERACION_R_DISTANCE"
7. "OPERACION_R_AND_ESCALAR"
8. "OPERACION_FUZZY"
9. "OPERACION_R_PATCH"

Se importan las capas vectoriales de los viveros y de buffer a los viveros (procedencia de origen de datos diferente). Se rasterizan de forma independiente asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Se combinan y se aplica una operación de distancias sobre la capa raster de combinación. Posteriormente se aplica una operación fuzzy sobre el mapa de distancias con los valores configurados en la interfaz (valor A, valor B, valor C y valor D) y se combina con la capa raster de combinación, siendo la capa resultante.

idc	ido	NombreBDDOL	NombreTablaOL	Precede	Orden	DisplayComentar	Comentario	DisplayParamIns	NombresParamInst	DisplayParamDis	NombresParamDis
integ	integ	character varying	character varying	integer[]	integer	boolean	character va	boolean	character varying(50)	boolean	character varying
49	2	BBDDOceanLider	viveros		1	f		f		f	
49	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
49	2	BBDDOceanLider	viveros buf		3	f		f		f	
49	3	-	-	{3}	4	f		t	Valor de idoneidad	f	
49	23	-	-	{2,4}	5	f		f		f	
49	4	-	-	{5}	6	f		f		f	
49	17	-	-	{6}	7	t	\$n=100	f		f	
49	8	-	-	{7}	8	f		t	Valor A,Valor B,Valor C,Valor D	f	
49	23	-	-	{5,8}	9	f		f		f	

Figura 63: Tabla de programación para el criterio Viveros

o **Criterio Zonas protegidas de interés pesquero**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de las zonas protegidas de interés pesquero y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D).

Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integ	ido integ	NombreBDDOL character varying	NombreTablaOL character varying(50)	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDe character varying
50	2	BDDOceanLider	zpipes		1	f		f		f	
50	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
50	4	-	-	{2}	3	f		f		f	
50	17	-	-	{3}	4	t	\$n=100	f		f	
50	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
50	23	-	-	{2,5}	6	f		f		f	

Figura 64: Tabla de programación para el criterio Zonas protegidas de interés pesquero

○ **Criterio Piscifactorías**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_V_IN"
4. "OPERACION_V_TO_RAST"
5. "OPERACION_R_PATCH"
6. "OPERACION_R_DISTANCE"
7. "OPERACION_R_AND_ESCALAR"
8. "OPERACION_FUZZY"
9. "OPERACION_R_PATCH"

Se importan las capas vectoriales de las piscifactorías y de buffer a las piscifactorías (procedencia de origen de datos diferente). Se rasterizan de forma independiente asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Se combinan y se aplica una operación de distancias sobre la capa raster de combinación. Posteriormente se aplica una operación fuzzy sobre el mapa de distancias con los valores configurados en la interfaz (valor A, valor B, valor C y valor D) y se combina con la capa raster de combinación, siendo la capa resultante.

idc integ	ido integ	NombreBDDOL character varying	NombreTablaOL character varying(50)	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character va	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDe character varying
S1	2	BDDOceanLider	piscifactoria		1	f		f		f	
S1	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
S1	2	BDDOceanLider	piscifactoria buf		3	f		f		f	
S1	3	-	-	{3}	4	f		t	Valor de idoneidad	f	
S1	23	-	-	{2,4}	5	f		f		f	
S1	4	-	-	{5}	6	f		f		f	
S1	17	-	-	{6}	7	t	\$n=100	f		f	
S1	8	-	-	{7}	8	f		t	Valor A,Valor B,Valor C,Valor D	f	
S1	23	-	-	{5,8}	9	f		f		f	

Figura 65: Tabla de programación para el criterio Piscifactorías

○ **Criterio Reservas marinas**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"

2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de las reservas marinas y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integi	ido integi	NombreBDDOL character varying	NombreTablaOL character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character vai	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
52	2	BBDDOceanLider	rrmm		1	f		f		f	
52	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
52	4	-	-	{2}	3	f		f		f	
52	17	-	-	{3}	4	t	\$n=100	f		f	
52	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
52	23	-	-	{2,5}	6	f		f		f	

Figura 66: Tabla de programación para el criterio Reservas marinas

- o **Criterio Zonas de producción de moluscos y de otros invertebrados marinos**

Este criterio necesita las siguientes operaciones en orden:

1. "OPERACION_V_IN"
2. "OPERACION_V_TO_RAST"
3. "OPERACION_R_DISTANCE"
4. "OPERACION_R_AND_ESCALAR"
5. "OPERACION_FUZZY"
6. "OPERACION_R_PATCH"

Se importa la capa vectorial de las zonas de producción de moluscos y se rasteriza asignando a los polígonos, líneas o puntos el valor dado por el usuario en la interfaz (parámetro valor de idoneidad). Posteriormente se aplica un mapa de distancias sobre esta capa raster y se aplica la operación fuzzy con los valores configurados en la interfaz (valor A, valor B, valor C y valor D). Por último se combina la capa raster con los valores de idoneidad sobre los polígonos, líneas o puntos y la capa raster en la que se ha aplicado la operación fuzzy.

idc integi	ido integi	NombreBDDOL character varying	NombreTablaOL character varying	Precede integer[]	Orden integer	DisplayComentar boolean	Comentario character vai	DisplayParamIns boolean	NombresParamInst character varying(50)	DisplayParamDis boolean	NombresParamDis character varying
53	2	BBDDOceanLider	cria moluscos		1	f		f		f	
53	3	-	-	{1}	2	f		t	Valor de idoneidad	f	
53	4	-	-	{2}	3	f		f		f	
53	17	-	-	{3}	4	t	\$n=100	f		f	
53	8	-	-	{4}	5	f		t	Valor A,Valor B,Valor C,Valor D	f	
53	23	-	-	{2,5}	6	f		f		f	

Figura 67: Tabla de programación para el criterio Zonas de producción de moluscos y de otros invertebrados marinos

3.6. Métodos de razonamiento

Para la combinación de diferentes grupos de criterios parciales dentro de una rama (físicos, operacional, medioambiental o de recurso energético), y criterios dentro del grupo de criterios se utiliza un método de razonamiento, para cada caso.

Hay cuatro métodos de razonamiento disponibles en la herramienta DSS:

- **Más que pesimista:** se aplica una operación de multiplicación a aquellas capas que intervengan. Está implementado mediante la operación **OPERACIÓN_ANDBOOLEANO**.
- **Pesimista:** se aplica una operación mínimo de las capas que intervienen. Está implementado mediante la operación **OPERACIÓN_ANDFUZZY**.
- **Optimista:** se aplica una operación máximo de las capas que intervienen. Está implementado mediante la operación **OPERACIÓN_R_MAX**.
- **Neutro:** se aplica una operación MLP de las que capas que intervienen. Está implementado mediante la operación **OPERACIÓN_MLP**. Es necesario establecer los pesos de las capas y comprobar que la suma de pesos sea igual a 1.

Si es el método escogido para la combinación de los grupos de criterios parciales dentro de una rama, el término capa se refiere a las capas resultado de la aplicación de cada grupo de criterios; y si es para la combinación de los criterios dentro de un grupo, son las capas resultado de la aplicación de cada criterio.

El método de razonamiento puede ser elegido por el usuario o determinado por los expertos. En el caso que sea para la combinación de los diferentes grupos de criterios parciales dentro de una rama el método es configurable por el usuario. Si es para la combinación de criterios dentro de un grupo, el método de razonamiento puede ser elegido por el usuario o determinado por los expertos.

Los expertos en la programación de la herramienta pueden fijar el método de razonamiento que se aplicará para la combinación de los diferentes criterios dentro de un grupo de criterios, dicho método de razonamiento prevalecerá en la herramienta DSS y no se le dará la opción de elección al usuario.

Los métodos de razonamiento se almacenan en las diferentes tablas en caso que proceda:

- Si es para la combinación de los grupos de criterios parciales dentro de una rama se almacena en el campo *ModoCF*, *ModoCO*, *ModoCEIA* y *ModoCRE* en la tabla **Escenario**.
- Si es para la combinación de los criterios dentro de un grupo y es prefijado por los expertos se almacena en el campo *TipoRazDefecto* en la tabla **Grupo**.
- Si es para la combinación de los criterios dentro de un grupo y es fijado por el usuario se almacena en la tabla **Tipo_raz_grupo_escenario**.

DESARROLLO DE LA HERRAMIENTA

4.1. Plan de desarrollo y seguimiento

4.1.1. Introducción

4.1.1.1. Propósito

El propósito de este Plan de Desarrollo de Software es ofrecer toda la información necesaria para controlar el desarrollo de la herramienta DSS. Es una visión al más alto nivel de abstracción que facilita al equipo de trabajo la organización del desarrollo.

Los usuarios potenciales de este Plan de Desarrollo de Software son los siguientes:

- **Jefe de proyecto:** planificación temporal y de recursos, así como control de progresos.
- **Miembros del equipo del proyecto:** entender lo que necesitan hacer, cuándo tienen que hacerlo y de qué otras actividades dependen.

Este proyecto va a contar con diferentes jefes de proyecto, que en el apartado *3.1 Estructura Organizativa* se especifica, según la estancia del contrato de Esperanza González de la Mora, encargada de realizar las tareas de análisis, diseño, implementación, puesta en marcha y pruebas de la herramienta DSS.

4.1.1.2. Alcance

Este Plan de Desarrollo de Software describe el plan general que será utilizado por el equipo de desarrollo para llevar a cabo el presente proyecto. Los detalles de las iteraciones individuales se describirán en los apartados correspondientes a cada plan de iteración.

Los planes detallados en este documento están basados en los requisitos del producto recogidos en el apartado *4.2.1.1. Requisitos*.

Dicho documento podrá sufrir modificaciones a lo largo de todo el proyecto, si fuera necesario, y la planificación del mismo cambiará de forma considerable si así fuese preciso.

4.1.1.3. Definiciones, Acrónimos y Abreviaturas

Ver el *anexo B Diccionario*.

4.1.1.4. Perspectiva general

Este Plan de Desarrollo de Software contiene la siguiente información:

- **Perspectiva General del Proyecto.**- Proporciona una descripción del propósito, alcance y objetivos del proyecto. Asimismo define los entregables que se esperan del proyecto.
- **Organización del Proyecto.**- Describe la estructura organizativa del equipo de desarrollo.
- **Gestión de Proceso.**- Explica el coste estimado y la planificación temporal, define las fases principales e hitos del proyecto, y describe cómo será el seguimiento del proyecto.

4.1.2. Perspectiva General del Proyecto

4.1.2.1. Propósito, Alcance y Objetivos del Proyecto

El proyecto persigue varios objetivos principales:

1. **La herramienta de Apoyo a la toma de decisión (DSS)**, debe ser una aplicación que proporcione al usuario los mejores emplazamientos para la ubicación de distintos tipos de Unidades Integradas de Generación de Energía eléctrica en medio Oceánico (UNIGEOs) teniendo en cuenta las restricciones estructurales, geográficas, físicas, legales, sociales y económicas que se disponga en el momento de realizar la toma de decisión.
2. **Estudio y análisis de diferentes tecnologías** para el desarrollo del proyecto por parte del Equipo de Desarrollo.
3. **Analizar, diseñar y construir una herramienta** de tales características aplicando técnicas de evaluación multicriterio (EMC) dentro del entorno de los SIG y del tratamiento digital de imágenes.

4.1.2.2. Suposiciones y Restricciones

Se consideran las siguientes suposiciones y restricciones que condicionan el desarrollo del proyecto:

- **Duración del proyecto.** La fecha de inicio del proyecto será el 1 de Octubre de 2011, siendo la fecha de entrega del mismo el 31 Marzo 2013.
- **Horario de trabajo.** El horario está condicionado a los contratos disfrutados por Esperanza González de la Mora y la estancia de trabajo vinculada. Este trabajo se ha repartido entre: prácticas en empresa, becaria asociada al proyecto y contrato asociado a Proyecto de investigación.

Los diferentes contratos que han tenido lugar para su desarrollo:

- **Octubre 2011- Julio 2012:** beca con un total de 780 horas.
- **Julio 2012- Enero 2013:** contrato con un total de 900 horas.
- **Enero 2013- Abril 2013:** ampliación de contrato con un total de 450 horas.

Las diferentes estancias para su desarrollo:

- **Octubre 2011- 24 Abril 2012:** estancia en el Grupo de Sistemas Inteligentes (GSI) de la Escuela de Informática de la Universidad de Valladolid. Horario flexible ajustado a las horas del contrato.
 - **24 Abril 2012- 15 Febrero 2012:** estancia en GMV Sistemas de Valladolid, realizado como contrato de prácticas en empresa. Horario seguido por la empresa ajustado a las horas del contrato.
 - **15 Febrero 2012- Abril 2012:** estancia en el Grupo de Sistemas Inteligentes (GSI) de la Escuela de Informática de la Universidad de Valladolid. Horario flexible ajustado a las horas del contrato.
- **Aprendizaje del equipo.** Una parte del tiempo total dedicado al proyecto será destinada al aprendizaje del equipo en las tecnologías necesarias para la elaboración del mismo. Parte de los conocimientos sobre las tecnologías fueron adquiridas en la elaboración del prototipo o prueba de concepto realizado en el proyecto fin de carrera de Esperanza González de la Mora con título “*Estudio de herramientas SIG de libre distribución para la selección de emplazamientos de unidades de generación de energía oceánica*”. Se tendrá en cuenta en la planificación de la iteración de inicio pero no en las siguientes iteraciones.
 - **Calendario de trabajo.** En la planificación no se han contabilizado los fines de semana ni festivos como días laborales. En un principio no se van a tener en cuenta los periodos de vacaciones de los miembros del proyecto en el plan de desarrollo por su desconocimiento anticipado, posteriormente tendrá que reajustarse la planificación.
 - **Monitorización del proyecto.** Se monitorizará el proyecto en todo momento, llevando un conteo del tiempo dedicado a cada actividad.
 - **Recursos del proyecto.** Los recursos software y hardware dependerán de los requisitos que se fijarán al principio y podrán variar a lo largo del tiempo. Se tendrá que hacer las modificaciones oportunas cuando se precise. Están detallados en el apartado 4.1.4.2.5. *Recursos del Proyecto*.
 - **Modelo de Proceso.** El modelo de proceso para construir el sistema será UPEDU (Unified Process for EDUcation) : Es un proceso de desarrollo software especializado para la educación [UPEDU, 2013]
 - **Análisis y seguimiento de riesgos.** No se va hacer un seguimiento de riesgos de forma precisa aunque éstos se tengan en cuenta en todo momento.

4.1.2.3. Entregables del Proyecto

Los artefactos entregables se muestran agrupados por flujos de trabajo.

- Gestión del Proyecto.
 - Plan de Desarrollo de Software. Apartado 4.1.
 - Planes de Iteraciones (uno por cada iteración definida). Apartado 4.1.5.
 - Informes de Seguimientos de Iteraciones (uno por cada iteración). Apartado 4.1.6.
 - Glosario del Proyecto. *Anexo B*.
- Requisitos.
 - Especificación de Requisitos de Software (SRS). Apartado 4.2.1.1.
 - Modelo de Casos de Uso. Apartado 4.2.1.2 (análisis) y 4.2.2.1 (diseño).
- Análisis y Diseño.
 - Modelo de Análisis. Apartado 4.2.1.
 - Modelo de Diseño. Apartado 4.2.2.
 - Modelo de Datos. Apartado 3.3.
- Implementación.
 - Modelo de Implementación. Apartado 4.3.
 - Código fuente y versiones ejecutables (Véase el apartado 4.1.4.2.3. *Versiones*).
- Pruebas.
 - Casos de Pruebas. *Capítulo 5*.
- Operación.
 - Manual de instalación. Apartado 6.1.
 - Manual de usuarios. Apartado 6.2.

4.1.2.4. Evolución del Plan de Desarrollo de Software

El *Plan de Desarrollo de Software* será revisado antes del comienzo de cada Iteración, comenzando por la primera iteración de la fase de Elaboración. El plan de cada iteración será realizado durante la iteración anterior, a excepción del primero que se llevará a cabo durante la fase de inicio, junto con este *Plan de Desarrollo de Software*.

Los sucesivos planes de iteración servirán, además de para concretar las tareas correspondientes a cada una, para corregir cualquier desviación respecto al plan inicial.

4.1.3. Organización del proyecto

4.1.3.1. Estructura Organizativa

Los tutores *José Belarmino Pulido Junquera* y *Carlos J. Alonso González* serán las personas responsables cuya función es supervisar y evaluar el contenido del proyecto durante el periodo de vida del proyecto, así como revisar los requisitos del proyecto software que aquí se afronta. La tarea de **Jefe de Proyecto**, será desempeñada por un miembro diferente según la estancia de desarrollo:

- En el Grupo de Sistemas Inteligentes (GSI) llevada a cabo por *José Belarmino Pulido Junquera*.
- En GMV Sistemas de Valladolid llevada a cabo por *Marcos López Cabeceira*.

El **Equipo de Desarrollo del Proyecto** estará constituido por *Esperanza González de la Mora* que desempeñará su función como analista, diseñador, programador y probador.

4.1.3.2. Interfaces Externas

José Belarmino Pulido Junquera será el principal representante del presente proyecto ante las diferentes entidades vinculadas para cualquier aspecto que pueda aparecer en el desarrollo. Las tareas de organización y coordinación serán llevadas a cabo por los jefes de proyecto en las diferentes etapas.

Se dará libertad al equipo de desarrollo del proyecto para ponerse en comunicación con las diferentes entidades si fuese necesario.

4.1.3.3. Roles y Responsabilidades del Equipo de Desarrollo

Se especifican los roles dentro del Equipo de Desarrollo: *Jefe de proyecto, Analista, Diseñador, Programador, Evaluador y Probador*. Sin embargo, al ser un equipo de pocos miembros no se especifican los roles específicos de UPEDU como ingeniero de componentes, analista de sistemas, especificador de casos de uso, etc., y se señalan roles de carácter más general.

Persona	Rol
José Belarmino Pulido Junquera Marcos López Cabeceira	Jefe de proyecto
Esperanza González de la Mora	Analista
Esperanza González de la Mora	Diseñador
Esperanza González de la Mora	Programador
Esperanza González de la Mora	Probador
José Belarmino Pulido Junquera Carlos J. Alonso González Juan Corzo Delibes	Evaluador

Figura 68: Tabla de roles y responsabilidades del equipo de desarrollo

4.1.4. Gestión de Proceso

4.1.4.1. Estimaciones del Proyecto

Al contar con una fuente limitada de recursos, se han realizado la planificación temporal con una estimación de días suficientes para afrontar las tareas que son necesarias en el proyecto.

En el apartado 4.1.4.2.1 de este mismo documento se muestra la especificación de la misma.

4.1.4.2. Plan de Proyecto

4.1.4.2.1. Plan de Fases

El modelo de proceso del proyecto será **UPEDU**.

Se han considerado como días laborales de lunes a viernes a la hora de realizar la planificación del proyecto. No se han considerado periodos vacacionales ni festivos en la planificación inicial.

A continuación se muestra la planificación del desarrollo al más alto nivel, en términos de fases e iteraciones, especificando la duración de cada una de ellas.

	Duración	Comienzo	Fin
Fase de Inicio	67 días	1 Octubre 2011	31 Diciembre 2011
1ª iteración	67 días	1 Octubre 2011	31 Diciembre 2011
Fase de Elaboración	82 días	1 Enero 2012	23 Abril 2012
1ª iteración	82 días	1 Enero 2012	23 Abril 2012
Fase de Construcción	179 días	24 Abril 2012	31 Diciembre 2012
1ª iteración	71 días	24 Abril 2012	31 Julio 2012
2ª iteración	65 días	1 Agosto 2012	30 Octubre 2012

3ª iteración	43 días	1 Noviembre 2012	31 Diciembre 2012
Fase de Transición	65 días	1 Enero 2013	31 Marzo 2013
1ª iteración	65 días	1 Enero 2013	31 Marzo 2013

Figura 69: Tabla del plan de fases

En el siguiente Diagrama de Gantt se aprecia gráficamente la distribución temporal de las distintas fases e iteraciones, y las normas de precedencia entre ellas.

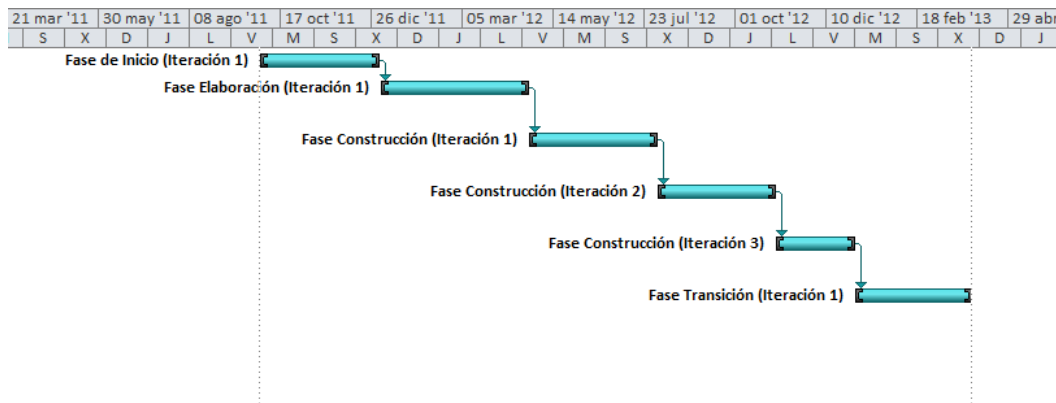


Figura 70: Diagrama de Gantt del plan de fases

En la siguiente tabla se especifican los hitos de cada una de las iteraciones. Véase el apartado 4.1.4.2.3. Versiones para comprender la naturaleza de cada una de las versiones intermedias del sistema ejecutable.

Iteración	Hito
Inicio	Plan de Desarrollo Software. Prototipo de comunicación.
Elaboración	Análisis del sistema y Arquitectura. Prototipo A.
Construcción 1	Versión Alfa.
Construcción 2	Versión Beta.
Construcción 3	Versión Gamma.
Transición	Producto final.

Figura 71: Tabla de versiones

4.1.4.2.2. Objetivos de las Iteraciones

En la siguiente tabla se especifican los objetivos de cada una de las fases del proyecto.

Fase	Objetivos
Inicio	<ul style="list-style-type: none"> • Delimitar el alcance y objetivos del proyecto • Definir la funcionalidad y capacidades del producto • Obtener una arquitectura candidata • Realizar estimaciones iniciales de planificación y costes. • Reducir los riesgos cuanto antes.
Elaboración	<ul style="list-style-type: none"> • Estudio profundo de la funcionalidad y el dominio. • Definición de la arquitectura básica. • Diseño del sistema.
Construcción	<ul style="list-style-type: none"> • Construcción del sistema
Transición	<ul style="list-style-type: none"> • Producto final. • Elaboración de los manuales de usuario e instalación

4.1.4.2.3. Versiones

Se distinguen cinco versiones de la aplicación:

- **Prototipo de comunicación:** Un prototipo muy simple y desechable obtenido tras la fase de Inicio, cuyo objetivo es eliminar el riesgo tecnológico de comunicación entre la aplicación, el sistema gestor de bases de datos y los sistemas que intervienen. Además servirá para el adiestramiento del equipo de desarrollo. Uso interno.
- **Prototipo A:** Primera versión ejecutable obtenida tras la primera iteración de Elaboración, con una funcionalidad mínima, no desechable, con las interfaces de usuario creadas y cuya finalidad es validar la arquitectura y funcionalidad mínima.
- **Versión Alfa:** Una evolución del Prototipo A obtenido tras la iteración Construcción 1, con los errores detectados corregidos y con la mayor parte de la funcionalidad y las interfaces añadidas.
- **Versión Beta:** Una evolución de la versión Alfa obtenido tras la iteración Construcción 2, con los errores detectados corregidos y con la funcionalidad ampliada y las interfaces añadidas.
- **Versión Gamma:** Una evolución de la versión Beta obtenido tras la iteración Construcción 3, con los errores detectados corregidos y con toda la funcionalidad y las interfaces añadidas.
- **Versión Definitiva:** Última versión ejecutable obtenida tras la iteración Transición, que consiste en la versión Gamma con los errores corregidos y sin añadir más funcionalidad. Si no se detectan errores, esta versión posiblemente coincidirá con la versión Gamma.

4.1.4.2.4. Planificación Temporal del Proyecto

La planificación temporal al más alto nivel de abstracción está descrita en el apartado 4.1.4.2.1. *Plan de Fases*. En cada iteración se incluye el *Plan de Iteración*, en el que se muestra la planificación temporal de forma detallada.

4.1.4.2.5. Recursos del Proyecto

4.1.4.2.5.1. Recursos humanos

En la ejecución de dicho proyecto, participarán todos los miembros del equipo de desarrollo del Proyecto, descritos con anterioridad:

- *Esperanza González de la Mora*

El trabajo será supervisado por los tutores *José Belarmino Pulido Junquera* y *Carlos J. Alonso González*. Además, los jefes de proyectos participantes orientarán al Equipo de Desarrollo y junto con los supervisores coordinarán el trabajo de las demás entidades participantes.

4.1.4.2.5.2. Recursos software

El sistema operativo sobre el que se va a realizar el desarrollo es Microsoft Windows 7. A continuación, se incluye una lista de las herramientas que se van a utilizar a lo largo del proyecto:

- Microsoft Word 2010. Editor de textos.
- Microsoft Project 2010. Gestor de proyectos.
- StarUML 5.0.2. Programa de creación de diagramas UML.
- Dia 0.97.2. Programa de creación de diagramas entidad-relación.
- Google Chrome 27. Explorador web.
- Mozilla Firefox 21 Explorador web junto con el complemento de inspector de elementos FireBug 1.11.
- NetBeans 7. Entorno de desarrollo.
- Visual Studio 2010. Entorno de desarrollo.
- TortoiseSVN 1.6.6. Herramienta para sistema operativo Windows, gratuita, para el acceso al repositorio de GMV Sistemas.
- FileZilla 3.7. Cliente FTP
- PostgreSQL 9.1 + PostGIS 2.0. Sistema gestor de base de datos geoespacial.
- GvSIG 1.11. Sistema de Información geográfico.

- GRASS 6.4.2. Sistema de Información geográfico.
- QuatumGIS Lisboa. Sistema de Información geográfico.

4.1.4.2.5.3. Recursos hardware

- Se facilitará un ordenador de trabajo en el Grupo de Sistemas Inteligentes (GSI) y GMV Sistemas para el equipo de desarrollo.
- Servidor de GMV Sistemas para la publicación de la herramienta DSS.

4.1.4.3. Seguimiento y Control del Proyecto

El seguimiento y control del proyecto se recoge en el apartado *4.1.6. Seguimiento*.

4.1.4.4. Gestión de Requisitos

Los requisitos iniciales para este sistema son capturados en el apartado *4.2.1.1.1*. Estos requisitos se consideran cerrados para la etapa de inicio, aunque están sujetos a posibles modificaciones durante el resto del proyecto. No considerándose en caso de que estos tengan lugar en la fase de Transición.

4.1.4.5. Control de Calidad

Todos los entregables serán sometidos a un proceso de revisión por los miembros del equipo de desarrollo, los evaluadores y las entidades participantes vinculadas al proyecto. De esta forma se asegurará un nivel de calidad aceptable en cada artefacto.

4.1.4.6. Informes y Medidas

Al final de cada iteración se llevará a cabo una actualización de la planificación temporal si fuera necesario.

4.1.4.7. Gestión de Riesgos

No se ha realizado una gestión de Riesgos exhaustiva debido al ámbito del proyecto relacionado con actividades de investigación y su reducida plantilla, por tanto se ha decidido desestimar la realización de una documentación sobre riesgos. Sin embargo, se tendrán en cuenta en todo momento y se hará una actualización de los requisitos a medida que puedan variar y una revisión de las actividades relacionadas.

4.1.4.8. Gestión de configuraciones

Para el control de versiones se tendrá acceso a un repositorio de la empresa GMV Sistemas.

4.1.5. Plan de Iteraciones

4.1.5.1. Plan de fase de Inicio

4.1.5.1.1. Introducción

4.1.5.1.1.1. Objetivos

El objetivo es presentar el plan de fase de la primera iteración correspondiente a la fase de inicio del proyecto. En esta fase se tienen como objetivos establecer el ámbito y el alcance del proyecto, identificar los requisitos, casos de uso críticos y escenarios principales, sintetizar la arquitectura a utilizar, realizar la planificación temporal del proyecto completo e identificar posibles riesgos que pueden surgir a lo largo de su desarrollo.

4.1.5.1.1.2. Alcance

Se utilizará como referencia a lo largo de la realización de la fase de inicio. Se intentará cumplir en todo momento con la planificación realizada. Los resultados de esta fase se pueden consultar en el *apartado de seguimiento fase de Inicio*.

Los artefactos obtenidos al finalizar esta fase son los siguientes:

- Apartado de visión general:
 - Requisitos generales del proyecto
 - Características principales
 - Restricciones
- Modelo inicial de casos de uso (10% a 20% listos)
- Glosario inicial: terminología clave del dominio
- Caso de negocio:
 - Contexto
 - Criterios de éxito
- Identificación inicial de posibles riesgos a tener en cuenta
- Plan de proyecto, mostrando las fases e iteraciones
- Modelo de negocio
- Documento inicial sobre la arquitectura a utilizar
- Plan de fase de la iteración
- Documento de seguimiento de la iteración
- Plan de la segunda iteración

4.1.5.1.2. Plan de la Iteración

En este apartado se detallarán las actividades implicadas en la fase de inicio y se mostrarán las figuras correspondientes a los archivos de Project obtenidos.

4.1.5.1.2.1. Descripción de tareas

- **Ámbito y alcance del proyecto:** esta tarea consiste en la definición del alcance y límites del proyecto. Su realización requiere un rol de analista.
- **Definición de Actores:** esta tarea consiste en la identificación de los actores presentes en a lo largo del proyecto. Su realización requiere un rol de analista.
- **Elicitación de requisitos:** esta tarea consiste en la identificación de los requisitos funcionales del sistema. Su realización requiere un rol de analista.
- **Casos de uso críticos:** esta tarea consiste en la identificación y descripción de los casos de uso principales presentes en el sistema (entre un 10% y 20% del total). Su realización requiere un rol de analista.
- **Escenarios principales:** esta tarea consiste en la identificación y descripción de los escenarios principales que se van a dar en el sistema. Su realización requiere un rol de analista.
- **Glosario inicial de términos:** esta tarea consiste en la definición de los principales términos que se van a manejar a lo largo del proyecto. Su realización requiere un rol de analista.
- **Arquitectura candidata a utilizar:** esta tarea consiste en el análisis de las posibles arquitecturas que se puedan emplear en el proyecto. Su realización requiere un rol de diseñador.
- **Planificación temporal del proyecto:** esta tarea consiste en la calendarización de las actividades del proyecto implicadas en la fase de inicio.
- **Estimación de riesgos potenciales:** esta tarea consiste en la identificación de los posibles riesgos que pueden darse a lo largo del desarrollo del proyecto. Su realización requiere un rol de analista.
- **Plan de la primera iteración:** esta tarea consiste en la calendarización de la primera iteración, descripción de actividades implicadas y artefactos obtenidos al finalizar la iteración. Su realización requiere un rol de jefe de proyecto.
- **Seguimiento de la primera iteración:** esta tarea consiste en recoger la evolución de la primera iteración, especificando si se han cumplido los objetivos, si se han obtenido

todos los artefactos necesarios y si se ha producido algún riesgo. Su realización requiere un rol de jefe de proyecto.

- **Plan de la segunda iteración:** esta tarea consiste en la planificación de la iteración siguiente (indicando actividades y artefactos implicados), en este caso en la planificación de la fase de elaboración. Su realización requiere un rol de jefe de proyecto.

4.1.5.1.2.2. Calendarización de tareas

En la *figura 72* se pueden observar las actividades que pertenecen a la primera iteración, su calendarización y los recursos necesarios para llevarlas a cabo. También se presenta el diagrama de Gantt (*figura 73*) para visualizar de una manera más gráfica la evolución de la fase.

Se ha querido alargar esta fase para consensuar entre todas las entidades vinculadas al proyecto los términos que engloban la construcción de la herramienta y estudiar la adaptación del prototipo del proyecto anterior construido a los nuevos requisitos.

	Nombre de la tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
	Fase de Inicio	67 días	1 Octubre 2011	31 Diciembre 2011		
1	Ámbito y alcance del proyecto	6 días	sáb 01/10/11	vie 07/10/11		Analista
2	Definición de actores	6 días	sáb 08/10/11	vie 14/10/11		Analista
3	Elicitación de requisitos	11 días	sáb 15/10/11	vie 28/10/11		Analista
4	Casos de uso críticos	10 días	lun 31/10/11	vie 11/11/11	1;2;3	Analista
5	Escenarios principales	5 días	lun 14/11/11	vie 18/11/11	4	Analista
6	Glosario inicial de términos	3 días	sáb 19/11/11	mar 22/11/11		Analista
7	Arquitectura candidata a utilizar	28 días	mié 23/11/11	sáb 31/12/11	1	Diseñador
8	Planificación temporal del proyecto	2 días	lun 10/10/11	mar 11/10/11	1	Jefe de proyecto
9	Estimación de riesgos potenciales	2 días	lun 31/10/11	mar 01/11/11	1;3	Analista/ Jefe de proyecto
10	Plan de la primera	2 días	mié 02/11/11	jue 03/11/11	9	Jefe de proyecto

	iteración					
11	Plan de desarrollo software	3 días	vie 04/11/11	mar 08/11/11	10	Jefe de proyecto
12	Seguimiento de la primera iteración	1 día	lun 05/12/11	lun 05/12/11	11	Jefe de proyecto
13	Plan de la segunda iteración	2 días	mar 06/12/11	mié 07/12/11	12	Jefe de proyecto

Figura 72: Plan calerandizado de la primera iteración

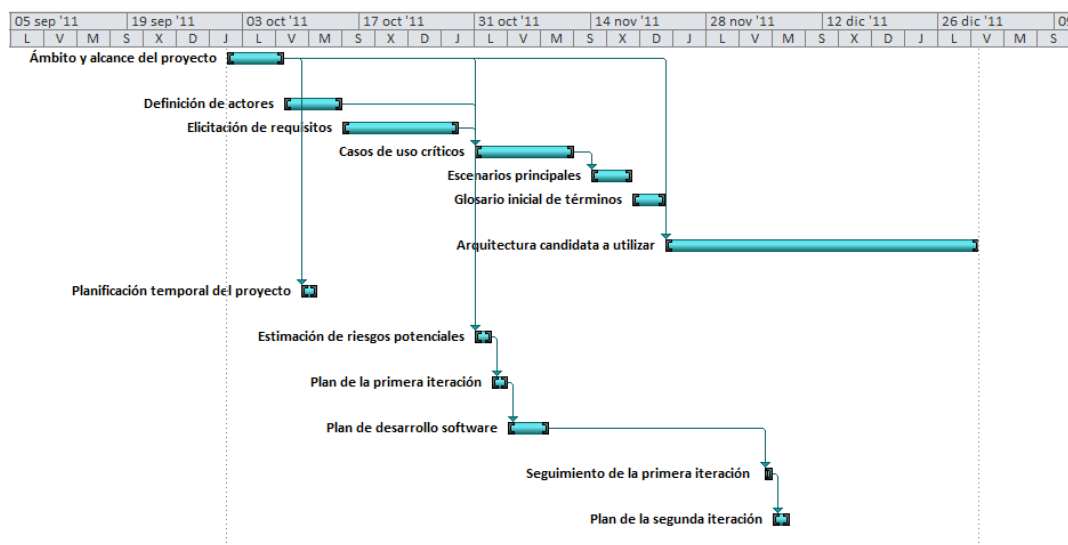


Figura 73: Diagrama de Gantt de la fase de inicio

4.1.5.1.3. Criterios de evaluación

Para comprobar que se han cumplido los objetivos de la fase de inicio se puede dar respuesta a las siguientes preguntas que sirven como criterios de evaluación. La respuesta a las mismas se puede consultar en el apartado de seguimiento.

- ¿Se ha definido correctamente el alcance el proyecto, incluyendo los conceptos operativos y los criterios de aceptación?
- ¿Se han entendido totalmente los requisitos?
- ¿Se ha realizado una planificación realista?
- ¿Son adecuadas las estimaciones del proceso de desarrollo?

- ¿Se han identificado los casos de uso críticos y los escenarios principales?

4.1.5.2. Plan de fase de Elaboración

4.1.5.2.1. Introducción

4.1.5.2.1.1. Objetivos

El objetivo es presentar el plan de fase de la primera iteración correspondiente a la fase de elaboración del proyecto. En esta fase se tienen como objetivos analizar el dominio del proyecto, establecer una arquitectura base sólida, desarrollar un plan de proyecto adecuado para la siguiente fase (de construcción) y eliminar los elementos que conformen un riesgo elevado para el desarrollo exitoso del proyecto.

4.1.5.2.1.2. Alcance

Se utilizará como referencia a lo largo de la realización de la fase de elaboración. Se intentará cumplir en todo momento con la planificación realizada. Los resultados de esta fase se pueden consultar en el *apartado de seguimiento fase de Elaboración*.

Los artefactos obtenidos al finalizar esta fase son los siguientes:

- Apartado de arquitectura del sistema
- Apartado de requisitos final
- Plan de desarrollo software estable
- Plan de desarrollo para el resto del proyecto
- Arquitectura ejecutable que contemple los casos de uso críticos
- Modelo de casos de uso (80% completo) con descripciones detalladas
- Modelo entidad-relación de la base de datos
- Requisitos no funcionales o no asociados a casos de uso
- Descripción de la Arquitectura del Software
- Un prototipo ejecutable de la arquitectura

- Un manual de usuario preliminar
- Documento de seguimiento de la iteración
- Plan de la tercera iteración

4.1.5.2.2. Plan de la Iteración

En este apartado se detallarán las actividades implicadas en la fase de elaboración y se mostrarán las figuras correspondientes a los archivos de Project obtenidos.

4.1.5.2.2.1. Descripción de tareas

- **Formación desarrolladores en el empleo de un SIG adecuado para la herramienta DSS y aprendizaje en las tecnologías que se van a usar:** Se partirá del conocimiento adquirido del prototipo creado del proyecto fin de carrera anterior. Ha sido necesaria incluir esta tarea como parte de la fase de elaboración debido a que fue imprescindible conocer el empleo de operaciones SIG fuera del propio entorno para su uso en la herramienta DSS. Su realización requiere un rol de desarrollador.
- **Ingeniería de requisitos:** esta tarea consiste en la revisión y actualización de los requisitos obtenidos en la fase de elaboración. Su realización requiere un rol de analista.
- **Modelo casos de uso análisis:** esta tarea consiste en la revisión y descripción de los casos de uso de la fase de análisis. Su realización requiere un rol de analista.
- **Modelo clases análisis:** esta tarea consiste en la realización del diagrama de clases de la fase de análisis. Su realización requiere un rol de analista.
- **Modelo casos de uso diseño:** esta fase consiste en realizar en detalle los casos de uso de la fase de análisis para conseguir el modelo de casos de uso de la fase de diseño. Su realización requiere un rol de analista.
- **Modelo clases diseño:** esta tarea consiste en la realización del diagrama de clases de la fase de diseño. Su realización requiere un rol de analista.
- **Modelo entidad-relación de datos análisis:** esta tarea consiste en la realización del diagrama entidad-relación de la base de datos de la fase de análisis. Su realización requiere un rol de analista.
- **Modelo entidad-relación de datos diseño:** esta tarea consiste en la realización del diagrama entidad-relación de la base de datos de la fase de diseño. Su realización requiere un rol de analista.

- **Arquitectura sistema:** esta tarea consiste en establecer una arquitectura base sólida que se mantenga a lo largo de todo el proyecto. Detallando los elementos implicados y las interrelaciones entre los mismos. Su realización requiere un rol de diseñador.
- **Primer prototipo:** en esta tarea se realiza un primer prototipo ejecutable de la arquitectura, en el que se recojan los casos de uso más críticos y los riesgos identificados. Su realización requiere un rol de programador.
- **Seguimiento de la primera iteración de la fase de elaboración:** esta tarea consiste en recoger la evolución de la segunda iteración, especificando si se han cumplido los objetivos, si se han obtenido todos los artefactos necesarios y si se ha producido algún riesgo. Su realización requiere un rol de jefe de proyecto.
- **Plan de fases de la siguiente iteración:** esta tarea consiste en la planificación de la iteración siguiente (indicando actividades y artefactos implicados), en este caso en la planificación de la iteración de la fase de construcción. Su realización requiere un rol de jefe de proyecto.

4.1.5.2.2.2. Calendarización de tareas

En la *figura 74* se pueden observar las actividades que pertenecen a la segunda iteración del proyecto, su calendarización y los recursos necesarios para llevarlas a cabo. También se presenta el diagrama de Gantt (*figura 75*) para visualizar de una manera más gráfica la evolución de la fase.

	Nombre de la tarea	Duración	Comienzo	Fin	Predecesoras	Nombre de los recursos
	Fase de elaboración	82 días	1 Enero 2012	23 Abril 2012		
1	Formación desarrolladores en el empleo de un SIG adecuado para la herramienta DSS y aprendizaje en las tecnologías	16 días	dom 01/01/12	sáb 21/01/12		Desarrollador
2	Ingeniería de requisitos	3 días	lun 23/01/12	mié 25/01/12		Analista
3	Modelo casos de	3 días	jue 26/01/12	lun 30/01/12	2	Analista

	uso análisis					
4	Modelo clases análisis	3 días	mar 31/01/12	jue 02/02/12	2	Analista
5	Modelo casos de uso diseño	5 días	vie 03/02/12	jue 09/02/12	3	Analista
6	Modelo clases diseño	5 días	vie 10/02/12	jue 16/02/12	4,5	Analista
7	Modelo entidad-relación de datos análisis	8 días	vie 17/02/12	mar 28/02/12	3,4	Analista
8	Modelo entidad-relación de datos diseño	13 días	mié 29/02/12	vie 16/03/12	6,7	Analista
9	Arquitectura sistema	15 días	lun 19/03/12	vie 06/04/12	6,8	Diseñador
10	Primer prototipo	11 días	lun 09/04/12	lun 23/04/12	9	Programador
11	Seguimiento de la primera iteración de la fase de elaboración	3 días	lun 09/04/12	mié 11/04/12	9	Jefe de proyecto
12	Plan de fases de la siguiente iteración	3 días	jue 12/04/12	lun 16/04/12	11	Jefe de proyecto

Figura 74: Plan calerandizado de la primera iteración de la fase de elaboración

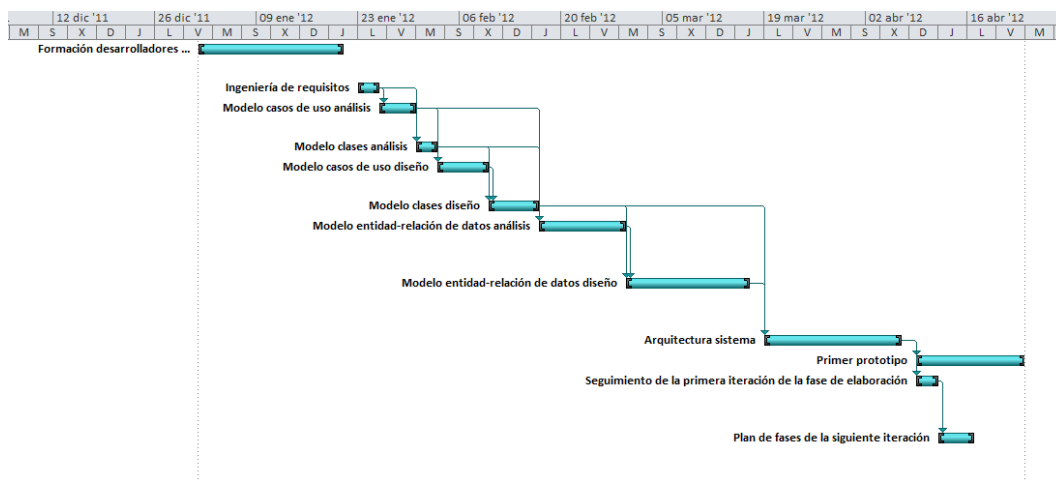


Figura 75: Diagrama de Gantt de la primera iteración de la fase de elaboración

4.1.5.2.3. Criterios de evaluación

Para comprobar que se han cumplido los objetivos de la fase de elaboración se puede dar respuesta a las siguientes preguntas que sirven como criterios de evaluación. La respuesta a las mismas se puede consultar en el documento de seguimiento de la fase de elaboración.

- ¿Es estable la visión del producto?
- ¿Es estable la arquitectura?
- ¿El plan del proyecto es realista?

4.1.5.3. Plan de fase de Construcción

4.1.5.3.1. Iteración 1

4.1.5.3.1.1. Introducción

4.1.5.3.1.1.1. Objetivos

El objetivo de este apartado es presentar el plan de fase de la primera iteración correspondiente a la fase de construcción del proyecto. Esta fase consta de tres iteraciones. En este apartado se expone el plan de la primera iteración de la fase de construcción. En esta fase se tienen como objetivos conseguir una calidad adecuada del producto y conseguir versiones utilizables del proyecto.

Las actividades que se realizarán en esta fase para cumplir con los objetivos son las siguientes:

- Desarrollo de componentes completos y prueba de los mismos frente a criterios de evaluación.
- Valoración de las versiones de producto.

4.1.5.3.1.1.2. Alcance

Se utilizará como referencia a lo largo de la realización de la fase de construcción en su primera iteración. Se intentará cumplir en todo momento con la planificación realizada. Los resultados de esta fase se pueden consultar en el *documento de seguimiento fase de Construcción primera iteración*.

Los artefactos obtenidos al finalizar esta fase son los siguientes:

- Versión alfa del producto
- Documento de seguimiento de la iteración
- Plan de la siguiente iteración

4.1.5.3.1.2. Plan de la Iteración

En este apartado se detallarán las actividades implicadas en la fase de construcción (primera iteración) y se mostrarán las figuras correspondientes a los archivos de Project obtenidos.

4.1.5.3.1.2.1. Descripción de tareas

- ***Acondicionamiento de la interfaz***: esta tarea consiste en ampliar y modificar la interfaz realizada por GMV Sistemas con contenido dinámico de la base de datos. Su realización requiere un rol de programador.
- ***Prueba de estudio de la evaluación multicriterio***: esta tarea consiste en extraer criterios de otros sistemas existentes (WindSpeed) para la prueba de evaluación multicriterio sobre el sistema. Su realización requiere un rol de analista y evaluador.
- ***Programación de los criterios de prueba***: esta tarea consiste en la programación de los criterios de prueba para la versión alfa del producto. Su realización requiere un rol de programador.
- ***Versión alfa del producto***: esta tarea refleja el producto final obtenido en la primera iteración de la fase de construcción (versión alfa del producto). Su realización requiere un rol de programador.

- **Seguimiento de la iteración actual:** esta tarea consiste en recoger la evolución de la tercera iteración, especificando si se han cumplido los objetivos, si se han obtenido todos los artefactos necesarios y si se ha producido algún riesgo. Su realización requiere un rol de jefe de proyecto.
- **Plan de fases de la siguiente iteración:** esta tarea consiste en la planificación de la iteración siguiente (indicando actividades y artefactos implicados), en este caso en la planificación de la fase de construcción (segunda iteración de esta fase). Su realización requiere un rol de jefe de proyecto.

4.1.5.3.1.2.2. Calendarización de tareas

En la *figura 76* se pueden observar las actividades que pertenecen a la tercera iteración del proyecto, su calendarización y los recursos necesarios para llevarlas a cabo. También se presenta el diagrama de Gantt (*figura 77*) para visualizar de una manera más gráfica la evolución de la fase.

	Nombre de la tarea	Duración	Comienzo	Fin	Predecesoras	Nombre de los recursos
	Fase de Construcción (iteración 1)	71 días	24 Abril 2012	31 Julio 2012		
1	Acondicionamiento de la interfaz	10 días	mar 24/04/12	lun 07/05/12		Programador
2	Prueba de estudio de la evaluación multicriterio	10 días	mar 08/05/12	lun 21/05/12		Analista y evaluador
3	Programación de los criterios de prueba	8 días	mar 22/05/12	jue 31/05/12	2	Programador
4	Versión alfa del producto	21 días	vie 01/06/12	vie 29/06/12	3	Programador
5	Seguimiento de la iteración actual	3 días	vie 01/06/12	mar 05/06/12	3	Jefe de proyecto
6	Plan de fases de la siguiente iteración	3 días	mié 06/06/12	vie 08/06/12	5	Jefe de proyecto

Figura 76: Plan calendarizado de la primera iteración de la fase de construcción

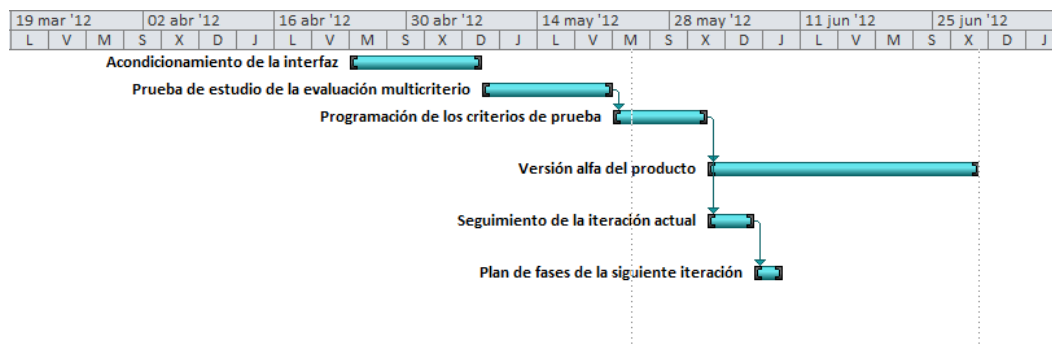


Figura 77: Diagrama de Gantt de la primera iteración de la fase de construcción

4.1.5.3.1.3. Criterios de evaluación

Para comprobar que se han cumplido los objetivos de la fase de construcción (primera iteración) se puede dar respuesta a las siguientes preguntas que sirven como criterios de evaluación. La respuesta a las mismas se puede consultar en el *documento de seguimiento de la fase de Construcción primera iteración*.

- ¿El producto es lo suficientemente estable?
- ¿Existen defectos o cambios pendientes que suponen un obstáculo para conseguir el propósito de la entrega siguiente?
- ¿Están los usuarios dispuestos para la fase de transición?
- ¿El esfuerzo horas/hombre de los recursos son aceptables todavía frente a los planeados?
- ¿El producto se encuentra en un estado lo suficientemente maduro y estable para instalarlo en el entorno del cliente?

4.1.5.3.2. Iteración 2

4.1.5.3.2.1. Introducción

4.1.5.3.2.1.1. Objetivos

El objetivo de este apartado es presentar el plan de fase de la segunda iteración correspondiente a la fase de construcción del proyecto. Como se ha comentado en el plan de la iteración anterior esta fase consta de tres iteraciones. En este apartado se expone el plan de la segunda iteración de la fase de construcción. En esta fase se tienen como objetivos conseguir una calidad adecuada del producto y conseguir versiones utilizables del proyecto.

Las actividades que se realizarán en esta fase para cumplir con los objetivos son las siguientes:

- Desarrollo de componentes completos y prueba de los mismos frente a criterios de evaluación.

- Valoración de las versiones de producto frente a los criterios de aceptación.

4.1.5.3.2.1.2. Alcance

Se utilizará como referencia a lo largo de la realización de la fase de construcción en su segunda iteración. Se intentará cumplir en todo momento con la planificación realizada. Los resultados de esta fase se pueden consultar en el *documento de seguimiento fase de Construcción segunda iteración*.

Los artefactos obtenidos al finalizar esta fase son los siguientes:

- Versión beta del producto
- Documento de seguimiento de la iteración
- Plan de la siguiente iteración

4.1.5.3.2.2. Plan de la Iteración

En este apartado se detallarán las actividades implicadas en la fase de construcción (segunda iteración) y se mostrarán las figuras correspondientes a los archivos de Project obtenidos.

4.1.5.3.2.2.1. Descripción de tareas

- **Consensuar los criterios finales del DSS:** esta tarea consiste en definir los criterios finales del DSS entre todas las entidades vinculadas al proyecto. Su realización requiere un rol de analista y evaluador.
- **Importación de capas geoespaciales a la base de datos:** esta tarea consiste en importar a la base de datos las capas geoespaciales pertenecientes al proyecto. Su realización requiere un rol de programador.
- **Unión de capas a una única proyección:** esta tarea consiste en unir las capas de las entidades Península, Baleares y Canarias a una única proyección. Su realización requiere un rol de programador.
- **Versión beta del producto:** esta tarea refleja el producto final obtenido en la segunda iteración de la fase de construcción (versión beta del producto). Su realización requiere un rol de programador.
- **Seguimiento de la iteración actual:** esta tarea consiste recoger la evolución de la cuarta iteración, especificando si se han cumplido los objetivos, si se han obtenido todos los artefactos necesarios y si se ha producido algún. Su realización requiere un rol de jefe de proyecto.

- **Plan de fases de la siguiente iteración:** esta tarea consiste en la planificación de la iteración siguiente (indicando actividades y artefactos implicados), en este caso en la fase de construcción (tercera iteración de esta fase). Su realización requiere un rol de jefe de proyecto.

4.1.5.3.2.2.2. Calendarización de tareas

En la *figura 78* se pueden observar las actividades que pertenecen a la cuarta iteración del proyecto, su calendarización y los recursos necesarios para llevarlas a cabo. También se presenta el diagrama de Gantt (*figura 79*) para visualizar de una manera más gráfica la evolución de la fase.

	Nombre de la tarea	Duración	Comienzo	Fin	Predecesoras	Nombre de los recursos
	Fase de Construcción (iteración 2)	65 días	1 Agosto 2012	30 Septiembre 2012		
1	Consensuar los criterios finales del DSS	10 días	mié 01/08/12	mar 14/08/12		Analista y evaluador
2	Importación de capas geoespaciales a la base de datos	15 días	mié 15/08/12	mar 04/09/12		Programador
3	Unión de capas a una única proyección	15 días	mié 05/09/12	mar 25/09/12	2	Programador
4	Versión beta del producto	26 días	mié 26/09/12	mié 31/10/12	1;3	Programador
5	Seguimiento de la iteración actual	3 días	mié 26/09/12	vie 28/09/12	3	Jefe de proyecto
6	Plan de fases de la siguiente iteración	3 días	lun 01/10/12	mié 03/10/12	5	Jefe de proyecto

Figura 78: Plan calendarizado de la segunda iteración de la fase de construcción

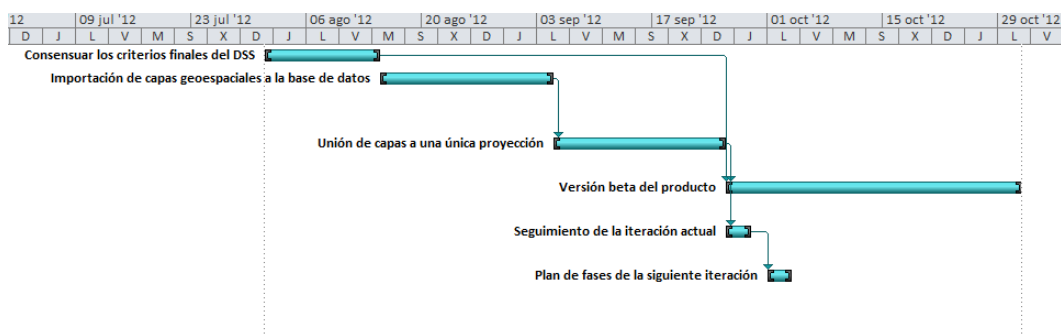


Figura 79: Diagrama de Gantt de la segunda iteración de la fase de construcción

4.1.5.3.2.3. Criterios de evaluación

Para comprobar que se han cumplido los objetivos de la fase de construcción (segunda iteración) se puede dar respuesta a las siguientes preguntas que sirven como criterios de evaluación. La respuesta a las mismas se puede consultar en el *documento de seguimiento de la fase de Construcción segunda iteración*.

- ¿El producto es lo suficientemente estable y maduro como para ser entregado al usuario?
- ¿Existen defectos o cambios pendientes que suponen un obstáculo para conseguir el propósito de la entrega siguiente?
- ¿Están los usuarios dispuestos para la fase de transición?
- ¿El producto se encuentra en un estado lo suficientemente maduro y estable para instalarlo en el entorno del cliente?

4.1.5.3.3. Iteración 3

4.1.5.3.3.1. Introducción

4.1.5.3.3.1.1. Objetivos

El objetivo de este apartado es presentar el plan de fase de la tercera iteración correspondiente a la fase de construcción del proyecto. Como se ha comentado en el plan de la iteración anterior esta fase consta de tres iteraciones. En este apartado se expone el plan de la tercera iteración de la fase de construcción. En esta fase se tienen como objetivos conseguir una calidad adecuada del producto y conseguir versiones utilizables del proyecto.

Las actividades que se realizarán en esta fase para cumplir con los objetivos son las siguientes:

- Desarrollo de componentes completos y prueba de los mismos frente a criterios de evaluación.
- Valoración de las versiones de producto frente a los criterios de aceptación.

4.1.5.3.3.1.2. Alcance

Se utilizará como referencia a lo largo de la realización de la fase de construcción en su tercera iteración. Se intentará cumplir en todo momento con la planificación realizada. Los resultados de esta fase se pueden consultar en el *documento de seguimiento fase de Construcción segunda iteración*.

Los artefactos obtenidos al finalizar esta fase son los siguientes:

- Versión gamma del producto
- Documento de seguimiento de la iteración
- Plan de la siguiente iteración

4.1.5.3.3.2. Plan de la Iteración

En este apartado se detallarán las actividades implicadas en la fase de construcción (tercera iteración) y se mostrarán las figuras correspondientes a los archivos de Project obtenidos.

4.1.5.3.3.2.1. Descripción de tareas

- **Implementación para la generación dinámica de ficheros .MAP:** esta tarea consiste en la implementación para la creación de ficheros .MAP para la visualización de capas raster y vectorial. Su realización requiere un rol de programador.
- **Crear visualizador de mapas:** esta tarea consiste en la implementación del visor de los mapas de resultados en OpenLayers. Su realización requiere un rol de programador.
- **Ampliación y modificación de interfaz:** esta tarea consiste en la ampliación y modificación de la interfaz para las nuevas funcionalidades. Su realización requiere un rol de programador.
- **Versión gamma del producto:** esta tarea refleja el producto final obtenido en la tercera iteración de la fase de construcción (versión gamma del producto). Su realización requiere un rol de programador.
- **Generar primer documento de pruebas:** esta tarea consiste en general la primera versión del documento de pruebas de los criterios fijados entre las entidades vinculadas al proyecto. Su realización requiere un rol de programador y evaluador.
- **Seguimiento de la iteración actual:** esta tarea consiste en recoger la evolución de la quinta iteración, especificando si se han cumplido los objetivos, si se han obtenido todos los artefactos necesarios y si se ha producido algún. Su realización requiere un rol de jefe de proyecto.

- **Plan de fases de la siguiente iteración:** esta tarea consiste en la planificación de la iteración siguiente (indicando actividades y artefactos implicados), en este caso en la planificación de la fase de transición. Su realización requiere un rol de jefe de proyecto.

4.1.5.3.3.2.2. Calendarización de tareas

En la *figura 80* se pueden observar las actividades que pertenecen a la quinta iteración del proyecto, su calendarización y los recursos necesarios para llevarlas a cabo. También se presenta el diagrama de Gantt (*figura 81*) para visualizar de una manera más gráfica la evolución de la fase.

	Nombre de la tarea	Duración	Comienzo	Fin	Predecesoras	Nombre de los recursos
	Fase de Construcción (iteración 3)	43 días	1 Noviembre 2012	31 Diciembre 2012		
1	Implementación para la generación dinámica de ficheros .MAP	5 días	jue 01/11/12	mié 07/11/12		Programador
2	Crear visualizador de mapas	5 días	jue 08/11/12	mié 14/11/12	1	Programador
3	Ampliación y modificación de la interfaz	3 días	jue 15/11/12	lun 19/11/12	2	Programador
4	Versión gamma del producto	17 días	mar 20/11/12	mié 12/12/12	3	Programador
5	Generar primer documento de pruebas	13 días	jue 13/12/12	lun 31/12/12	4	Programador y evaluador
6	Seguimiento de la iteración actual	3 días	jue 13/12/12	lun 17/12/12	4	Jefe de proyecto
7	Plan de fases de la siguiente iteración	3 días	mar 18/12/12	jue 20/12/12	6	Jefe de proyecto

Figura 80: Plan calendarizado de la tercera iteración de la fase de construcción

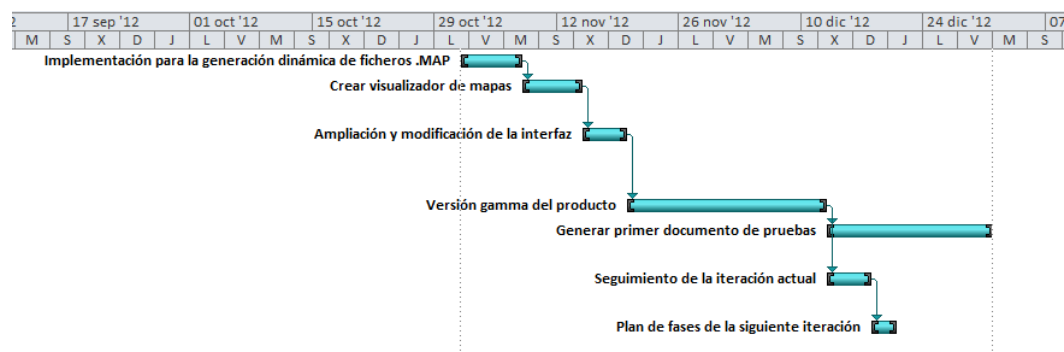


Figura 81: Diagrama de Gantt de la tercera iteración de la fase de construcción

4.1.5.3.3.3. Criterios de evaluación

Para comprobar que se han cumplido los objetivos de la fase de construcción (tercera iteración) se puede dar respuesta a las siguientes preguntas que sirven como criterios de evaluación. La respuesta a las mismas se puede consultar en el *documento de seguimiento de la fase de Construcción segunda iteración*.

- ¿El producto es lo suficientemente estable y maduro como para ser entregado al usuario?
- ¿Existen defectos o cambios pendientes que suponen un obstáculo para conseguir el propósito de la entrega siguiente?
- ¿Están los usuarios dispuestos para la fase de transición?
- ¿El producto se encuentra en un estado lo suficientemente maduro y estable para instalarlo en el entorno del cliente?

4.1.5.4. Plan de fase de Transición

4.1.5.4.1. Introducción

4.1.5.4.1.1. Objetivos

El objetivo de este apartado es presentar el plan de fase de la primera iteración correspondiente a la fase de transición del proyecto. En esta fase se tienen como objetivos conseguir que el usuario sea capaz de mantener el producto, conseguir la aceptación por el usuario de que lo entregado es completo y consistente con el criterio de evaluación fijado en la visión inicial del proyecto y por último conseguir un producto final sólido y eficiente.

Las actividades que se realizarán en esta fase para cumplir con los objetivos son las siguientes:

- Sincronización e integración de los incrementos de construcción en líneas base de entrega consistente.
- Realización de ingeniería específica de entrega

- Valoración de entrega frente a la visión global y los criterios de aceptación en el conjunto de requisitos

4.1.5.4.1.2. Alcance

Se utilizará como referencia a lo largo de la realización de la fase de transición en su primera iteración. Se intentará cumplir en todo momento con la planificación realizada. Los resultados de esta fase se pueden consultar en el *documento de seguimiento fase de Transición primera iteración*.

Los artefactos obtenidos al finalizar esta fase son los siguientes:

- Seguimiento de la iteración actual
- Plan de pruebas
- Producto final

4.1.5.4.2. Plan de la Iteración

En este apartado se detallarán las actividades implicadas en la fase de transición y se mostrarán las figuras correspondientes a los archivos de Project obtenidos.

4.1.5.4.2.1. Descripción de tareas

- **Seguimiento de la iteración actual:** esta tarea consiste en la emisión de un documento en el que se recoja la evolución de la última iteración, especificando si se han cumplido los objetivos, si se han obtenido todos los artefactos necesarios y si se ha producido algún riesgo (en caso de que se haya producido indicar la fecha en la que se produjo, la causa y en plan de actuación). Su realización requiere un rol jefe de proyecto.
- **Prueba y evaluación de resultados:** en esta tarea se realizan las pruebas y evaluación de los resultados obtenidos en continuación con lo establecido en la anterior iteración. Se estudian los mapas resultado obtenidos de la toma de decisión con la combinación de diferentes criterios, se detectan los fallos de la aplicación y se procede a su modificación. Se pretende detectar el mayor número de errores posible con el fin de que la batería de pruebas sea un éxito y se puedan realizar los cambios oportunos para eliminar el mayor número de errores. Su realización requiere un rol de probador y evaluador
- **Producto final:** esta tarea refleja el resultado de la obtención del producto final que se entrega al cliente. Este producto corresponde a la versión gamma creada en la segunda iteración de la fase de construcción después de aplicar las modificaciones oportunas al realizar las pruebas que se pueden consultar en el apartado de pruebas que se adjunta en la documentación. Su realización requiere un rol programador.

- **Publicación de la aplicación:** esta tarea consiste en la publicación en un servidor externo de la aplicación. Su realización requiere un rol de programador.

4.1.5.4.2. Calendarización de tareas

En la 82 se pueden observar las actividades que pertenecen a la última iteración del proyecto, su calendarización y los recursos necesarios para llevarlas a cabo. También se presenta el diagrama de Gantt (figura 83) para visualizar de una manera más gráfica la evolución de la fase. Su realización requiere un rol de probador.

	Nombre de la tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
	Fase de Transición	65 días	1 Enero 2013	1 Abril 2013		
1	Seguimiento de la iteración actual	3 días	mar 01/01/13	jue 03/01/13		Jefe de proyecto
2	Prueba y evaluación de resultados	25 días	vie 04/01/13	jue 07/02/13		Probador y evaluador
3	Producto final	25 días	vie 08/02/13	jue 14/03/13	2	Programador
4	Publicación de la aplicación	12 días	vie 15/03/13	dom 31/03/13	3	Programador

Figura 82: Plan calendarizado de la primera iteración de la fase de transición

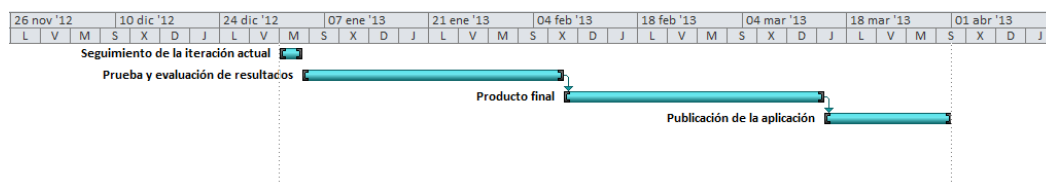


Figura 83: Diagrama de Gantt de la primera iteración de la fase de transición

4.1.5.4.3. Criterios de evaluación

Para comprobar que se han cumplido los objetivos de la fase de transición (primera iteración) se puede dar respuesta a las siguientes preguntas que sirven como criterios de evaluación. La

respuesta a las mismas se puede consultar en el *documento de seguimiento de la fase de Transición primera iteración*.

- ¿El usuario está satisfecho con el producto entregado?
- ¿Han sido cubiertos todos los requisitos del proyecto?
- ¿Se han cumplido los objetivos?

4.1.6. Seguimiento

4.1.6.1. Introducción

4.1.6.1.1. Objetivo

El objetivo de este apartado es presentar el seguimiento de cada una de las iteraciones del proyecto para reflejar de manera detallada si se han cumplido con los objetivos de cada fase analizando los criterios de evaluación que se pueden consultar en los apartados correspondientes a cada uno de los planes de iteración de las seis iteraciones realizadas en el proyecto: 1 iteración en la fase de inicio, 1 iteración en la fase de elaboración, 3 iteraciones en la fase de construcción y 1 iteración en la fase de transición.

4.1.6.2. Fase de Inicio

4.1.6.2.1. Seguimiento 1ª iteración

La fase de inicio tiene como fecha de comienzo el día 1/10/11 y como fecha de finalización el día 31/12/11. En esta primera iteración del proyecto se han cumplidos con los objetivos de la misma.

Se han obtenido los artefactos previstos de esta fase (requisitos generales, casos de uso iniciales, arquitectura candidata, etc.) sin ningún contratiempo.

Al contar con una planificación holgada y con la experiencia del anterior prototipo realizado no se han obtenido percances o retrasos.

Los criterios de evaluación descritos para la fase de inicio son los siguientes (a continuación de cada criterio se exponen los resultados obtenidos al valorar cada uno de los puntos):

- ¿Se ha definido correctamente el alcance el proyecto, incluyendo los conceptos operativos y los criterios de aceptación?

Sí, los conceptos que engloban el presente trabajo son una ampliación del proyecto anterior realizado. Se tiene en cuenta en las próximas planificaciones una aclaración de conceptos a las entidades vinculantes.

- ¿Coinciden los “usuarios” con la definición del ámbito y las estimaciones en función del tiempo?

Sí, los usuarios potenciales de la aplicación identificados durante esta fase corresponden a los usuarios reales del sistema.

- ¿Se han entendido totalmente los requisitos?

Sí, los requisitos se han fijado con las entidades vinculantes mediante reuniones telefónicas, aunque se acuerda alguna posible modificación futura que se tendrá en cuenta para la planificación de las siguientes iteraciones.

- ¿Se ha realizado una planificación realista?

Sí, se ha hecho una planificación realista y holgada para desempeñar las actividades de esta fase.

- ¿Se han identificado los casos de uso críticos y los escenarios principales?

Sí, una vez identificados los requisitos funcionales se han obtenido los casos de uso críticos y los escenarios principales.

4.1.6.3. Fase de Elaboración

4.1.6.3.1. Seguimiento 1ª iteración

La fase de elaboración tiene como fecha de comienzo el día 01/01/12 y como fecha de finalización el día 23/04/12. En esta segunda iteración del proyecto también se han cumplido con todos los objetivos de la misma aunque ha sido necesaria la ampliación de nuevas actividades.

Al precisar de un mapa de profundidad y éste no estar dentro de la base de datos interna en el ámbito del proyecto, ha sido necesario incluir un estudio para generar el mapa profundidad a partir de un mapa de batimetría.

Los criterios de evaluación descritos para la fase de elaboración son los siguientes (a continuación de cada criterio se exponen los resultados obtenidos al valorar cada uno de los puntos):

- ¿Es estable la visión del producto?

Sí, la visión que se tiene del producto en esta fase es estable, se tienen muy clara la funcionalidad que debe cumplir la herramienta DSS.

- ¿Es estable la arquitectura?

Sí, se tiene una arquitectura sólida fijada aunque no se ha completado el prototipo ejecutable de la misma.

- ¿El plan del proyecto es realista?

Sí, se ha contado con una planificación holgada para abarcar todas las tareas descritas en la iteración.

4.1.6.4. Fase de Construcción

4.1.6.4.1. Seguimiento 1ª iteración

La 1ª iteración de la fase de construcción tiene como fecha de comienzo el día 24/04/2012 y como fecha de finalización el día 31/07/2012. En esta tercera iteración del proyecto no se han cumplidos con todos los objetivos planteados.

El motivo por el que se expone que no se han cumplido los objetivos de esta iteración es porque no se ha completado en esta versión Alfa toda la funcionalidad que se había planificado por un cambio en la tecnología de programación web. El prototipo A de la fase de elaboración estaba escrito en PHP y ha sido necesario reescribir la implementación a JSP.

Los criterios de evaluación descritos para la fase de construcción son los siguientes (a continuación de cada criterio se exponen los resultados obtenidos al valorar cada uno de los puntos):

- ¿El producto es lo suficientemente estable y maduro como para ser entregado al usuario?

Se ha obtenido el prototipo Alfa del producto, pero no se ha conseguido llegar a implementar toda la funcionalidad requerida. No obstante, la parte de funcionalidad que está implementada se encuentra en un estado estable y maduro como para ser entregado al cliente.

- ¿Existen defectos o cambios pendientes que suponen un obstáculo para conseguir el propósito de la entrega siguiente?

No existen defectos ni cambios pendientes en esta iteración. Los propósitos fijados para la siguiente entrega se pueden obtener sin problemas.

- ¿Están los usuarios dispuestos para la fase de transición?

No, aún quedan dos iteraciones para finalizar la fase de construcción.

- ¿El producto se encuentra en un estado lo suficientemente maduro y estable para instalarlo en el entorno del cliente?

Sí, aunque queda funcionalidad por completar el producto se puede instalar para hacer la publicación en un servidor y ver su funcionamiento con los criterios de pruebas.

4.1.6.4.2. Seguimiento 2ª iteración

La segunda iteración de la fase de construcción tiene como fecha de comienzo el día 01/08/2012 y como fecha de finalización el día 30/10/2012. En esta cuarta iteración del proyecto se han

cumplido con todos los objetivos de la misma aunque han aparecido actividades que no estaban en la planificación inicial.

En el estudio de los criterios finales consensuados por las entidades vinculadas, se ha incluido una actividad para el estudio de los documentos proporcionados con los valores de idoneidad para los criterios medioambientales y su importación a la base de datos.

También ha sido necesario fijar algunas reuniones internas para la aclaración de conceptos del ámbito del proyecto por parte de algunas entidades vinculadas al proyecto; y reuniones para la fijación de los criterios finales de la herramienta DSS.

Se ha obtenido una versión Beta del producto y el plan de la tercera iteración de la fase de construcción.

Los criterios de evaluación descritos para la fase de construcción son los siguientes (a continuación de cada criterio se exponen los resultados obtenidos al valorar cada uno de los puntos):

- ¿El producto es lo suficientemente estable y maduro como para ser entregado al usuario?

Sí, se ha completado parte de la funcionalidad requerida con los criterios finales fijados. Aunque no se ha completado toda la funcionalidad, el producto se encuentra en un estado estable y maduro.

- ¿Existen defectos o cambios pendientes que suponen un obstáculo para conseguir el propósito de la entrega siguiente?

No existen defectos ni cambios pendientes en esta iteración.

- ¿Están los usuarios dispuestos para la fase de transición?

No, aún queda una iteración para finalizar la fase de construcción.

- ¿El producto se encuentra en un estado lo suficientemente maduro y estable para instalarlo en el entorno del cliente?

Sí, el producto se encuentra en un estado maduro y estable para actualizar la publicación en el servidor y esté visible a los clientes.

4.1.6.4.3. Seguimiento 3ª iteración

La tercera iteración de la fase de construcción tiene como fecha de comienzo el día 01/11/2012 y como fecha de finalización el día 31/12/2012. En esta quinta iteración del proyecto se han cumplido con todos los objetivos de la misma aunque han aparecido actividades que no estaban en la planificación inicial con motivo de cambio en los requisitos.

Los requisitos que han cambiado son referentes a las características (parámetros) de los dispositivos vinculados a los criterios y la introducción de escala de unidades.

Se ha obtenido una versión Gamma del producto y el plan de iteración de la fase de transición.

Los criterios de evaluación descritos para la fase de construcción son los siguientes (a continuación de cada criterio se exponen los resultados obtenidos al valorar cada uno de los puntos):

- ¿El producto es lo suficientemente estable y maduro como para ser entregado al usuario?

Sí, se ha completado la funcionalidad aunque queda pendiente el cambio de la paleta de colores para los mapas de resultados.

- ¿Existen defectos o cambios pendientes que suponen un obstáculo para conseguir el propósito de la entrega siguiente?

No existen defectos ni cambios pendientes en esta iteración.

- ¿Están los usuarios dispuestos para la fase de transición?

Sí, los usuarios están dispuestos para realizar la fase de transición. En esta fase se obtiene un producto estable con toda la funcionalidad requerida.

- ¿El producto se encuentra en un estado lo suficientemente maduro y estable para instalarlo en el entorno del cliente?

Sí, el producto se encuentra en un estado maduro y estable.

4.1.6.5. Fase de Transición

4.1.6.5.1. Seguimiento 1ª iteración

La fase de transición tiene como fecha de comienzo el día 01/01/2013 y como fecha de finalización el día 31/03/2013. En esta sexta iteración del proyecto se han cumplido todos los objetivos fijados. Se ha obtenido la versión final del producto.

Se han realizado las pruebas identificadas y detalladas en el capítulo de *validación y pruebas* que se adjunta en la memoria. Tras realizar las pruebas y modificaciones correspondientes se ha obtenido la versión final del producto, versión Gamma mejorada que será entregada al cliente.

Los criterios de evaluación descritos para la fase de transición son los siguientes (a continuación de cada criterio se exponen los resultados obtenidos al valorar cada uno de los puntos):

- ¿El usuario está satisfecho con el producto entregado?

Sí, aunque ha habido cambios en los requisitos en la etapa de transición, se entiende que el contrato ha sido finalizado y los cambios requeridos pasarán a cargo de GMV Sistemas.

- ¿Han sido cubiertos todos los requisitos del proyecto?

Sí, se ha realizado un análisis final de comprobación de requisitos fijados inicialmente y se han cubierto todos los requisitos del proyecto.

- ¿Se han cumplido los objetivos?

Sí, se ha realizado un análisis final de comprobación de objetivos y se han cumplido todos los del proyecto.

4.2. Ingeniería del software

4.2.1. Análisis

4.2.1.1. Requisitos

En este apartado se pretende dar una descripción de los diversos requisitos que definen el sistema (SRS- Software Requirements Specification o Especificación de Requisitos Software), obteniéndose mediante un estudio exhaustivo de la especificación del sistema propuesto y siguiendo la siguiente fuente de referencia:

ANSI/IEEE Std 830-1984. IEEE Guide to Software Requirements Specification.

La descripción de dichos requisitos servirá para proporcionar la ayuda necesaria para la fase de diseño y un apoyo fundamental para las pruebas del sistema ya que si cumple los requisitos aquí especificados, significará que se ha realizado con éxito. Además, se identificarán los distintos elementos del sistema, casos de uso, actores, así como la localización del mismo dentro de un contexto y el diagrama de clases del sistema.

4.2.1.1.1. Requisitos generales

- La aplicación deberá funcionar en los navegadores más usados actualmente.
- Debe ser accesible para cualquier persona una vez registrada.
- Debe ser capaz de presentar una evaluación en un máximo periodo de tiempo.
- Ha de ser escalable manteniendo la habilidad en la capacidad de acción y adaptación.

4.2.1.1.2. Requisitos de base de datos

- Requerirá dos bases de datos según la naturaleza de los datos: relacional para almacenar los datos de la aplicación y espacial para almacenar los datos geoespaciales.
- La aplicación almacenará los datos de cada cuenta de usuario registrado.
- Almacenará la configuración de los distintos escenarios creados por cada uno de los usuarios.

- Almacenará los resultados de cada uno de los escenarios mandados a procesar por los usuarios.
- Almacenará el estado en que se encuentra cada uno de los distintos escenarios creados por el usuario.
- Almacenará la sintaxis de las operaciones procedentes de GRASS para la generación automática de scripts operacionales.
- Almacenará la secuencia de operaciones introducida por los expertos para cada criterio.
- Almacenará la información de las rutas de las capas raster alojadas en el servidor.
- Almacenará información geoespacial de tipo vectorial procedente de la base de datos OceanLider.

4.2.1.1.3. Requisitos de apariencia

- La aplicación constará de diferentes apartados claramente diferenciados correspondientes a los pasos de creación y configuración de un nuevo escenario, consulta de un escenario creado y resultados.
- La primera página mostrará los principales objetivos de la aplicación.
- La segunda página permitirá crear nuevos escenarios.
- Las sucesivas páginas serán pertenecientes a la configuración de los criterios que tendrá ligados el escenario.
- Habrá una página en la que se pueda consultar un escenario ya creado, así como los parámetros y criterios que intervienen, y el estado en el que se encuentra (creado, en espera de procesar, procesando o finalizado).
- También contendrá una página de visualización de resultados en caso que el escenario se haya procesado.
- La aplicación constará de una sección en la que existirá un manual de uso de la aplicación.
- Contendrá un apartado en el que se proporcionará una vía de contacto con los administradores de la aplicación.
- La aplicación mostrará los logotipos de las entidades involucradas en el desarrollo de la aplicación.
- La aplicación debe contener el mismo estilo visual e iconos en sus distintas ventanas. Debe tener una consistencia visual.

4.2.1.1.4. Requisitos de autenticación en la aplicación

- Antes de poder utilizar la funcionalidad de creación y procesamiento de escenarios, el usuario ha de tener una cuenta en la aplicación.
- La aplicación requerirá la autenticación de usuario y contraseña para uso de las funcionalidades.
- Si el usuario no tiene una cuenta en la aplicación, se mostrará un formulario con los datos necesarios para crear una cuenta.
- Para la navegación entre las páginas de la aplicación web, se deberá reconocer y comprobar la autenticación del usuario.
- El formulario contendrá los siguientes datos:

- Nombre
 - Apellidos
 - Nombre de usuario
 - Contraseña encriptada para protección de datos
 - Email
 - DNI/NIF
- La aplicación permitirá salir de una cuenta de usuario activa.
 - En la creación de una nueva cuenta, la aplicación deberá hacer la comprobación repetitiva de correo electrónico y contraseña.

4.2.1.1.5. Requisitos de presentación de objetivos

- La aplicación mostrará de forma clara y concisa los principales objetivos de la aplicación, así como los resultados que se obtendrá al final de todo el proceso.
- Se le indicará de igual modo, los pasos pertinentes para la configuración de un escenario, consulta y ejecución.
- Se mostrará una figura que ilustre un posible resultado del análisis y procesamiento de un escenario.

4.2.1.1.6. Requisitos de creación de escenarios

- La aplicación mostrará en un primer momento los parámetros que atañen de forma exclusiva a las características del escenario: nombre, dispositivo, y zona geográfica donde se quiere hacer dicho estudio.
- La aplicación permitirá seleccionar cualquier zona marítima española independiente del tamaño y ubicación.
- La aplicación mostrará varias pestañas con las distintas opciones para configurar los criterios que estarán ligados al escenario: criterios físicos, criterios operacionales, criterios medio ambientales y criterios de recurso energético.
- También la aplicación proporcionará el último paso para configurar los parámetros pertenecientes a la toma de decisión final para la ejecución y obtención de los resultados.
- La aplicación mostrará esas opciones en formas de formulario.
- Al lado de cada uno de los campos del formulario se habilitará un botón que proporcionará una descripción del campo, así como un posible valor que pueda tomar.
- La aplicación no permitirá introducir parámetros incorrectos.
- La aplicación permitirá (en caso que el escenario no haya sido procesado y no se haya mandado a cola de ejecución) poder cambiar cualquier parámetro ligado a él, así también como los criterios configurados.

4.2.1.1.7. Requisitos de consulta y ejecución de escenarios

La aplicación mostrará todos los escenarios creados que tiene el usuario, así como el estado y configuración que tiene cada uno de ellos.

La aplicación dará la opción de iniciar el procesamiento de un escenario que no ha sido procesado todavía.

La aplicación no dejará ejecutar de nuevo un escenario una vez se haya mandado a ejecutar y esté en cola de procesamiento.

4.2.1.1.8. Requisitos de presentación de resultados

La aplicación mostrará cada uno de los resultados intermedios y finales del escenario una vez procesado.

Mostrará la información vinculada al resultado y proporcionará acceso a visualizar el resultado en un sistema de cartografía web.

4.2.1.2. Casos de Uso

A continuación se muestran los casos de uso del sistema: *Autenticación de usuario*, *Crear escenario*, *Modificar parámetros y criterios del escenario*, *Cargar escenario*, *Realizar cálculos* y *Visualizar resultados*.

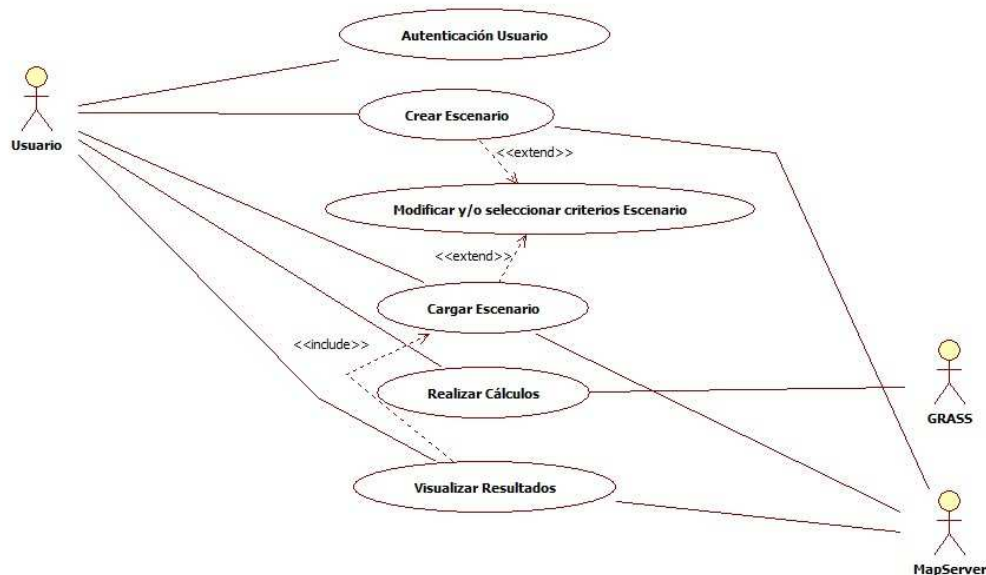


Figura 84: Diagrama casos de uso de análisis

4.2.1.2.1. Descripción de casos de uso

En este apartado se dará una breve descripción de los casos de uso reflejados en los diagrama anterior.

CU-01: Autenticación usuario

Este caso de uso permitirá al usuario acceder a todas las funcionalidades del sistema, registrarse en la aplicación y salir de la sesión una vez logado.

CU-02: Crear escenario

Este caso de uso permitirá al usuario realizar la configuración de los escenarios. Podrá seleccionar la naturaleza del dispositivo (y los parámetros que le caracterizan), seleccionar la zona geográfica donde se quiere realizar el estudio, seleccionar los criterios que intervengan y finalmente guardar el escenario creado.

CU-03: Cargar escenario

Este caso de uso permitirá al usuario acceder a los datos de un escenario, los criterios ligados y modificar los datos.

CU-04: Realizar cálculos

Este caso de uso pondrá en cola de ejecución a un escenario una vez dada la orden de ejecutar, el sistema extrae la información perteneciente al escenario, traduce e implementa las operaciones de los criterios con los datos configurados, crea el script operacional y lo ejecuta de forma automática generando los resultados intermedios y finales, y finalmente cambiará el estado del escenario a 'procesado'.

CU-05: Visualizar resultados

Este caso de uso mostrará la información de los resultados intermedios y finales de un escenario concreto ya procesado, y dispondrá al usuario la opción de visualizarlo.

CU-06: Modificar y/o seleccionar criterios del escenario

Este caso de uso permite al usuario modificar y/o seleccionar los criterios de un escenario una vez creado o cargado.

4.2.1.3. Diagrama de clases

En este apartado se muestra el diagrama de clases de análisis una vez estudiados los requisitos que se describieron en la captura de requisitos, refinándolos y estructurándolos. El objetivo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema.

Varios son los objetivos que se plantean llevando a cabo un estudio de los requisitos del sistema a través de un Modelo de Análisis [BAR09]:

- Ofrecer una especificación más precisa de los requisitos que la obtenida en el proceso de captura de los mismos.
- Introducir un mayor formalismo sobre la especificación del funcionamiento del sistema, ya que el Modelo de Análisis se describe mediante el lenguaje de los desarrolladores.
- Estructurar los requisitos de modo que se facilite su comprensión, su preparación, su modificación y su mantenimiento.
- Establecer una primera aproximación al modelo de diseño, utilizada como entrada fundamental cuando se de forma al sistema durante el diseño y la implementación.

Posteriormente, se procederá a la descripción de las clases en detalle.

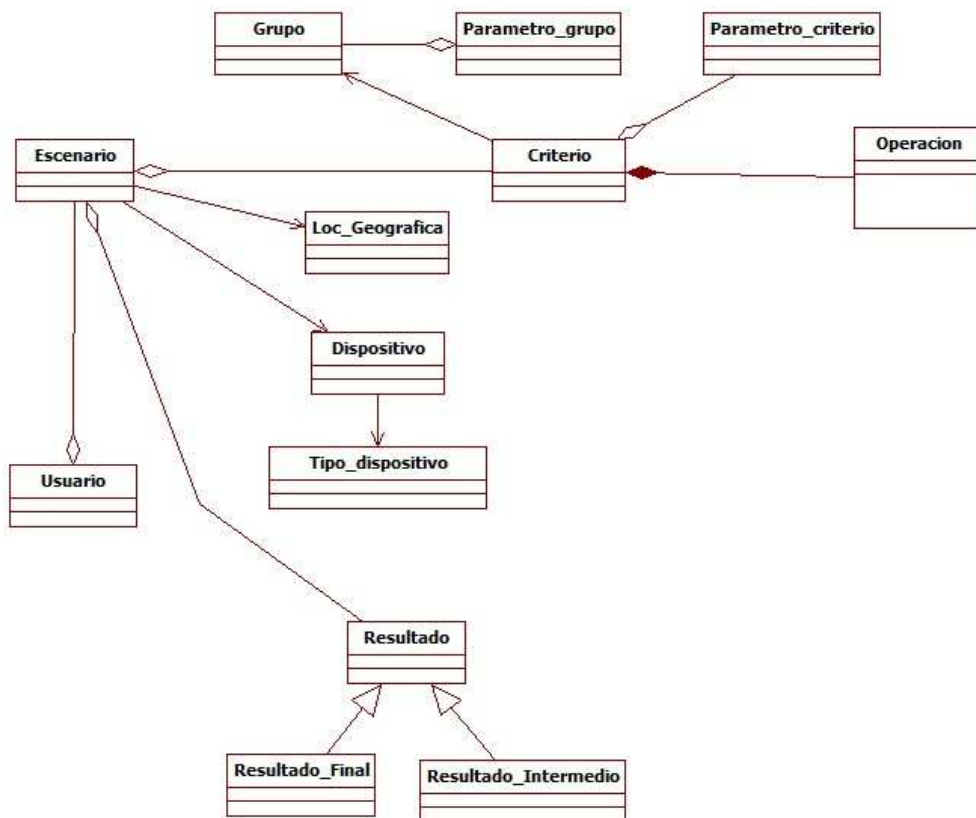


Figura 85: Diagrama de clases de análisis de la herramienta DSS

4.2.1.3.1. Descripción de las clases

- Un **usuario** define uno o varios **escenarios**. Se almacenan los datos oportunos del usuario (cumpliendo la ley 15/1999 de protección de datos), entre ellos el *correo electrónico* para el aviso de la obtención de resultados.
- Un **escenario** tiene una *región concreta (localización geográfica)* que ha sido seleccionada por el usuario, un dispositivo asociado, los resultados (intermedios y finales) en caso que se haya ejecutado y los criterios seleccionados para ese escenario.
- Un **criterio** se implementa con una o varias **operaciones** y estarán asociadas a una o varias capas de entrada de la **BBDD OceanLider**. Lo que permitirá la visualización gráfica (en miniatura) al configurar el escenario y la determinación para la aplicación del criterio en caso de ausencia de datos.
- Se instanciarán las **operaciones** a aplicar con los parámetros concretos que se hayan introducido para cada criterio.
- Cada **operación** llevará asociado una *cadena* definida en GRASS
- Las *cadenas* en GRASS están definidas previamente en la BBDD (instanciadas con parámetros estándar o iniciales)
- Se almacenan **los tipos de dispositivos** (eólico, mareomotriz, undimotriz).
- Dependiendo del **tipo de dispositivo** seleccionado condiciona a los **criterios** que se pueden aplicar en el escenario. Estos criterios son los que se activan en el controlador de la interfaz gráfica.
- Los **criterios** pueden ser de *tipo1* (física, operacional, de recurso energético y medio ambiental) y *tipo2* (total o parcial, según el tipo de exclusión). Cada **criterio** lleva asociado la *descripción del factor* y unos **parámetros**.
- Cada **criterio** pertenece a un **grupo** de criterios.
- Los **grupos** de criterios tendrán asociado unos **parámetros**.

4.2.2. Diseño

4.2.2.1. Casos de Uso

4.2.2.1.1. General

A continuación se muestran los casos de uso generales del sistema web: *Autenticación de usuario*, *Crear escenario*, *Modificar parámetros y criterios del escenario*, *Cargar escenario*, *Realizar cálculos* y *Visualizar resultados*.

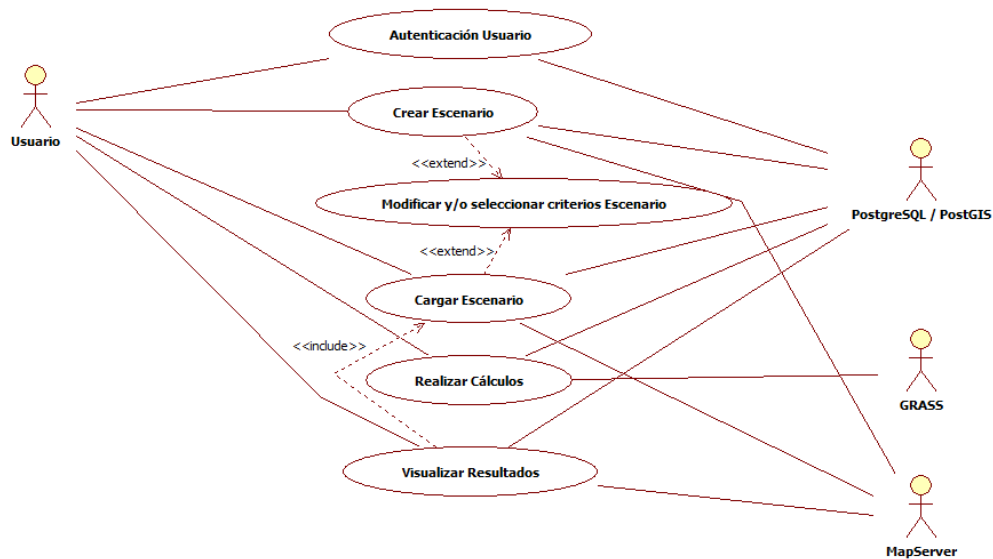


Figura 86: Casos de uso generales de diseño

4.2.2.1.2. Autenticación usuario

El caso de uso de autenticación de usuario se compone de: *hacer login, registrarse y salir de la sesión.*



Figura 87: Caso de uso de diseño - Autenticación usuario-

4.2.2.1.3. Crear escenario

El caso de uso crear escenario se compone de: *selección del dispositivo, seleccionar zona geográfica del estudio, introducir parámetros del escenario, seleccionar o modificar criterios que se van a ligar al escenario y guardar escenario.*

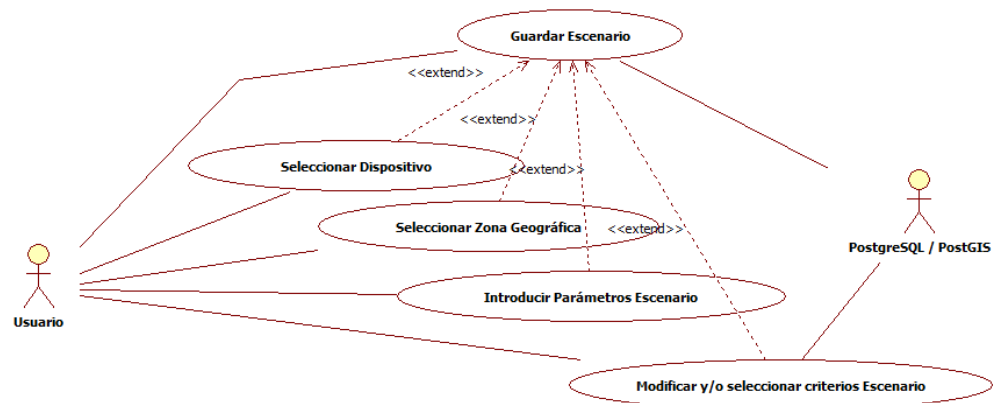


Figura 88: Caso de uso de diseño - Crear escenario-

4.2.2.1.4. Cargar escenario

El caso de uso cargar escenario se compone de: *selección de escenario, modificar dispositivo, modificar la zona geográfica del estudio, modificar los parámetros pertenecientes al escenario, modificar y/o seleccionar los criterios ligados al escenario y guardar escenario.*

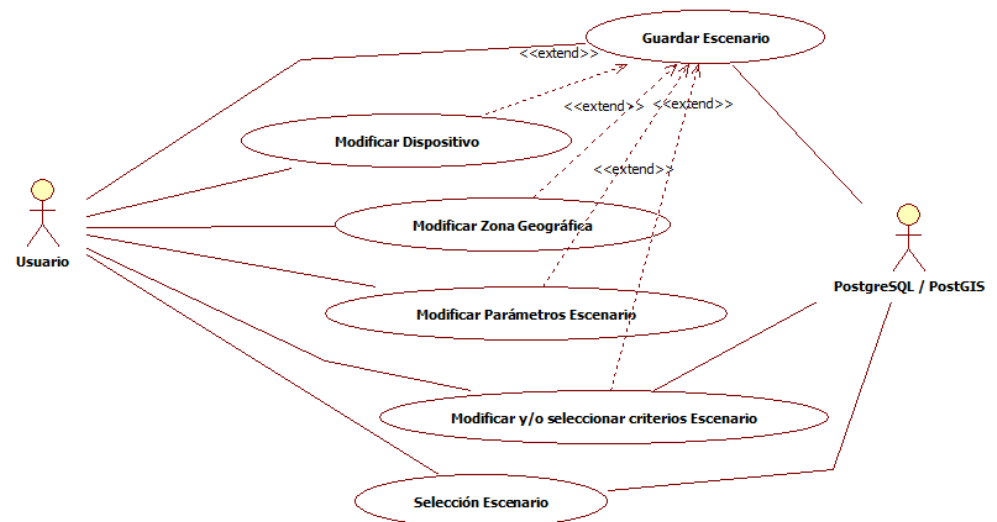


Figura 89: Caso de uso de diseño- Cargar escenario

4.2.2.1.5. Realizar cálculos

El caso de uso realizar cálculos se compone de: *extraer información del escenario, traducir/implementar los criterios a operaciones de GRASS, realización del script ejecutable, ejecución del script de forma automática y modificación del estado del escenario.*

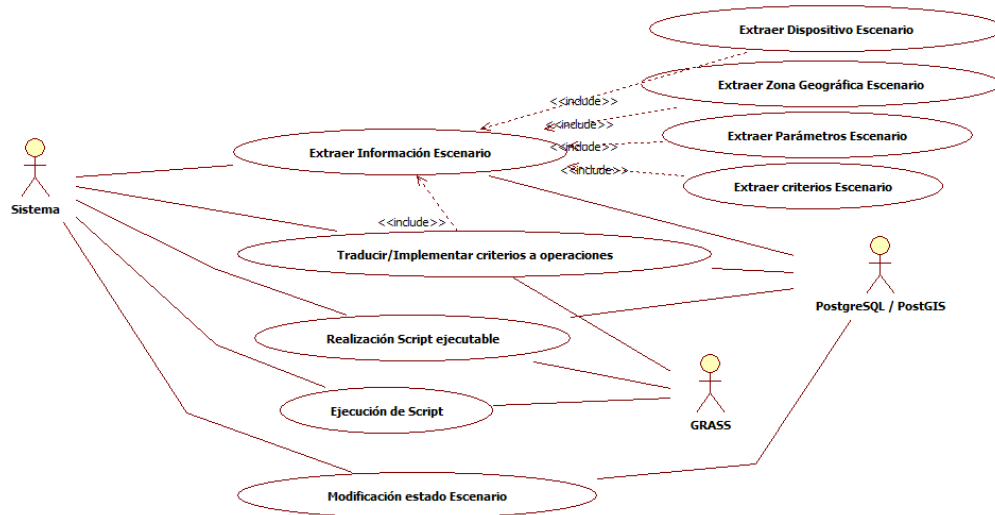


Figura 90: Caso de uso de diseño- Realizar Cálculos-

4.2.2.1.6. Visualizar resultados

El caso de uso visualizar resultados se compone de: *mostrar información de los resultados intermedios y finales del escenario una vez procesado y visualización de los resultados en un visor web.*

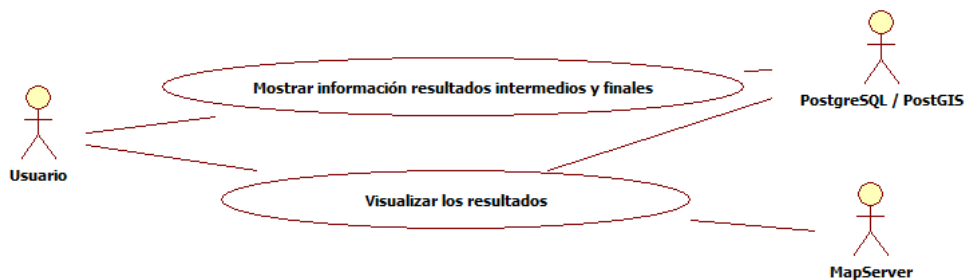


Figura 91: Caso de uso de diseño- Visualizar resultados-

4.2.2.2. Descripción de casos de uso

En este apartado se dará una breve descripción de todos los casos de uso reflejados en los diagramas anteriores.

4.2.2.2.1. General

CU-01: Autenticación usuario

Este caso de uso permitirá al usuario acceder a todas las funcionalidades del sistema, registrarse en la aplicación y salir de la sesión una vez logado.

CU-02: Crear escenario

Este caso de uso permitirá al usuario realizar la configuración de los escenarios. Podrá seleccionar la naturaleza del dispositivo (y los parámetros que le caracterizan), seleccionar la zona geográfica donde se quiere realizar el estudio, seleccionar los criterios que intervengan y finalmente guardar el escenario creado.

CU-03: Cargar escenario

Este caso de uso permitirá al usuario acceder a los datos de un escenario, los criterios ligados y modificar los datos.

CU-04: Realizar cálculos

Este caso de uso pondrá en cola de ejecución a un escenario una vez dada la orden de ejecutar, el sistema extrae la información perteneciente al escenario, traduce e implementa las operaciones de los criterios con los datos configurados, crea el script operacional y lo ejecuta de forma automática generando los resultados intermedios y finales, y finalmente cambiará el estado del escenario a 'procesado'.

CU-05: Visualizar resultados

Este caso de uso mostrará la información de los resultados intermedios y finales de un escenario concreto ya procesado, y dispondrá al usuario la opción de visualizarlo.

CU-06: Modificar y/o seleccionar criterios del escenario

Este caso de uso permite al usuario modificar y/o seleccionar los criterios de un escenario una vez creado o cargado.

4.2.2.2.2. Autenticación de usuario

CU-07: Hacer login

Este caso de uso permitirá que un usuario con una cuenta dada de alta en el sistema, pueda acceder a la aplicación.

CU-08: Registrarse

Este caso de uso permitirá que un usuario que no tiene una cuenta en el sistema, pueda generar una para poder acceder a la aplicación.

CU-09: Salir de la sesión

Mediante este caso de uso el usuario podrá cerrar su sesión en la aplicación.

4.2.2.2.3. Crear escenario

CU-10: Seleccionar Dispositivo

Este caso de uso permite al usuario cambiar el dispositivo asociado al escenario. Podrá visualizar los parámetros e información vinculante o crear un nuevo dispositivo si lo desea.

CU-11: Seleccionar zona geográfica

Este caso de uso permite al usuario seleccionar la zona geográfica del estudio.

CU-12: Introducir los parámetros del escenario

Este caso de uso permite al usuario introducir los parámetros específicos del escenario.

CU-13: Modificar y/o seleccionar los criterios del escenario

Este caso de uso permite al usuario seleccionar o deseleccionar los diferentes criterios que se ligarán al escenario.

CU-14: Guardar escenario

Este caso de uso permite al usuario almacenar los datos una vez configurado el escenario.

4.2.2.2.4. Cargar escenario

CU-15: Selección escenario

Este caso de uso permite al usuario seleccionar un escenario ya creado para poder visualizar la información vinculada o proceder a su ejecución.

CU-16: Modificar dispositivo

Este caso de uso permite al usuario modificar el dispositivo asociado de un escenario cargado, siempre y cuando no se haya mandado ejecutar.

CU-17: Modificar zona geográfica

Este caso de uso permite al usuario modificar la zona geográfica de estudio de un escenario cargado, siempre y cuando no se haya mandado ejecutar.

CU-18: Modificar parámetros escenario

Este caso de uso permite al usuario modificar los datos de un escenario cargado, siempre y cuando no se haya mandado ejecutar.

CU-19: Modificar y/o seleccionar criterios del escenario

Este caso de uso permite al usuario cambiar la lista de criterios de un escenario cargado, siempre y cuando no se haya mandado ejecutar.

4.2.2.2.5. Realizar cálculos

CU-20: Extraer información del escenario

En este caso de uso el sistema extrae la información necesaria procedente del escenario: los datos del dispositivo, los datos de la zona geográfica, los datos de los parámetros del escenario y los datos de los criterios ligados al escenario.

CU-21: Traducir/Implementar los criterios a operaciones

En este caso de uso el sistema traduce los datos del escenario a operaciones pertenecientes a un sistema SIG para su ejecución.

CU-22: Realización del script ejecutable

En este caso de uso el sistema crea el fichero ejecutable con las operaciones SIG de un escenario concreto.

CU-23: Ejecución de script

En este caso de uso el sistema ejecuta de forma automática el script operacional del escenario sobre un SIG.

CU-24: Modificación del estado del escenario

En este caso de uso el sistema modifica el estado del escenario a 'procesado' una vez haya finalizado la ejecución.

4.2.2.2.6. Visualizar resultados

CU-25: Mostrar información de los resultados intermedios y finales

Este caso de uso permite al usuario visualizar la información de los resultados intermedios y finales que genera la herramienta de un escenario ya procesado.

CU-26: Visualizar los resultados

Este caso de uso permite al usuario visualizar el mapa raster generado de un resultado final o intermedio de un escenario procesado.

4.2.2.3. Diagrama de clases

El diagrama de clases de diseño del sistema completo se divide en cuatro componentes: *Interfaz*, *DAOPattern*, *Entities* y *Motor*. Se va a realizar esta división por componentes con la intención de aplicar el concepto de entidades EJBs (donde una entidad es el reflejo de la estructura almacenada en la base de datos) y la distribución del sistema en dos partes.

La distribución del sistema se realiza en dos partes, según la funcionalidad Web y la parte operacional del sistema. La parte de la funcionalidad Web utiliza las componentes: *DAOPattern*, *Entities* e *Interfaz*. La parte operacional del sistema utiliza las componentes: *DAOPattern*, *Entities* y *Motor*.

No es un modelo en tres capas propiamente dicho ya que las interrelaciones entre las capas de interfaz, dominio y datos no es como debería corresponder. Se refleja en la siguiente figura para cada sistema.

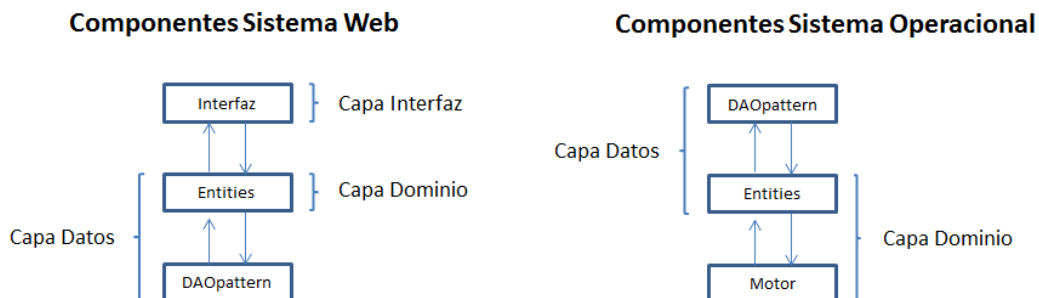


Figura 92: Componentes del sistema web y el sistema operacional

A continuación se procede a la descripción por componentes y posteriormente se mostrará la visión conjunta.

4.2.2.3.1. Componente datos

Esta componente es la aplicación del patrón DAO (Data Access Object) en la que se suministra una interfaz común para el acceso a los datos de la aplicación y la implementación en un sistema gestor determinado. En el caso que nos concierne, el sistema gestor de base de datos utilizado es PostgreSQL.

El patrón DAO se completa con el patrón Factoría: una clase a la que al pedirle la InterfaceDAO decide cuál de las implementaciones instanciar y la devuelve.

De esta manera se consigue un aislamiento de la tecnología de persistencia subyacente en la aplicación.

IDA0: Es la interfaz de las operaciones permitidas para el acceso a la base de datos.

PostgresDAO: Es la implementación de la interfaz para el sistema gestor de base de datos PostgreSQL.

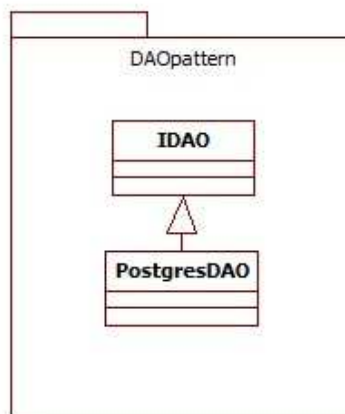


Figura 93: Componente DAOpattern del diagrama de clases de diseño

4.2.2.3.2. Componente entities

Esta componente incluye las entities del modelo, las interfaces de las operaciones específicas y las clases con los métodos específicos de selección e inserción a la base de datos.

Se van a utilizar las entities creadas por el framework de Java para Entities EJBs. Facilitan la estructura para albergar los datos de las entidades de la base de datos.

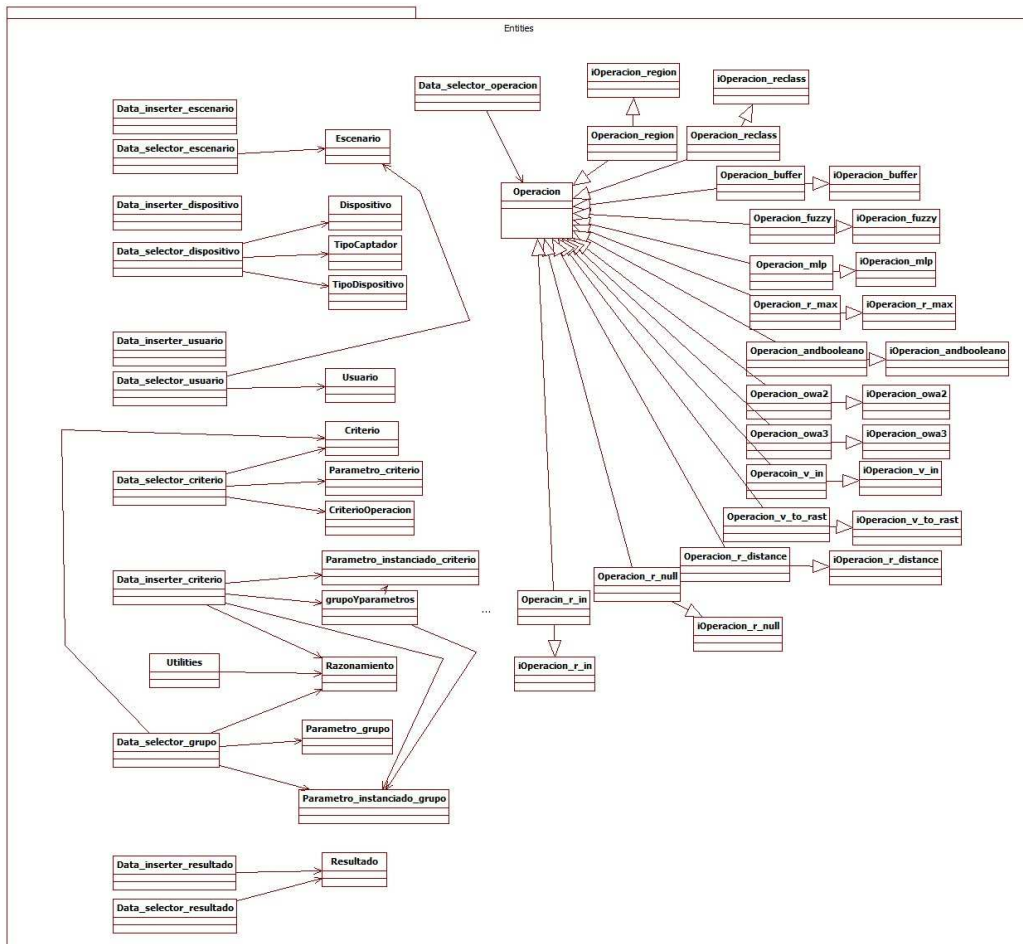


Figura 94: Componente Entities del diagrama de clases de diseño

Se ha utilizado el patrón de diseño **Abstract Factory** (Fábrica Abstracta) para las operaciones específicas. Se utiliza una interfaz que tienen que cumplir para cada una de ellas e implementan la clase Operación.

- **Clases Entities:**

Clase Usuario

En la clase Usuario se albergan los datos de cada usuario: DNI, nombre, primer apellido, segundo apellido, nick o alias, dirección de correo electrónico y contraseña.

Clase Escenario

En la clase Escenario se albergan los datos: nombre del escenario, dispositivo asociado, fecha y hora de creación del escenario, fecha y hora fin de ejecución del escenario, estado de

procesamiento en el que se encuentra, localización geográfica y métodos de razonamiento para los criterios parciales procedentes de los diferentes ámbitos (criterios físicos, criterios operacionales, criterios medioambientales y criterios de recurso energético).

Clase Criterio

En la clase Criterio se albergan los datos: descripción del factor asociado, tipo de criterio, tipo de exclusión, grupo asociado al criterio y comentario auxiliar vinculado al criterio.

Clase Dispositivo

En la clase Dispositivo se albergan los datos asociados a cada dispositivo: nombre dispositivo, tipo de dispositivo (undimotriz, maremotriz o eólico), tipo de captador, altura de buje, altura de dispositivo, diámetro rotor, dimensión X, Y y Z, profundidad máxima y mínima, altura de ola máxima permitida, velocidad de corriente máxima permitida, potencia, separación entre dispositivos de una misma línea y separación entre líneas de dispositivos.

Clase Operación

La clase padre de todas las operaciones del sistema. Tiene los atributos: el esqueleto de la operación para la generación del script operacional con operaciones pertenecientes al SIG GRASS y el nombre del tipo de operación.

Clase Resultado

La clase Resultado alberga los datos de los resultados de un escenario: escenario al que pertenece, la ruta del fichero asociado al resultado (será tipo raster y estará albergado en el sistema de ficheros del servidor), nombre del resultado, y ruta y nombre del fichero .MAP procedente de MapServer que se creará para su visualización web.

Clase Parametro_criterio

La clase Parametro_criterio contiene los datos de un parámetro definido por defecto perteneciente a un criterio. Está condicionado al tipo de dispositivo elegido en el escenario. Los atributos que contiene son: identificador del parámetro, nombre del parámetro, valor por defecto, identificador del criterio asociado, identificador del tipo de dispositivo asociado.

Clase Parametro_instanciado_criterio

La clase Parametro_instanciado_criterio contiene los datos de un parámetro definido por el usuario que sustituye al valor definido por defecto.

Los atributos que contiene son: identificador del parámetro instanciado, valor del parámetro instanciado e identificador del parámetro por defecto al que se vincula.

Clase Parametro_grupo

La clase Parametro_grupo contiene los datos de un parámetro definido por defecto perteneciente a un grupo de criterios. Está condicionado al tipo de dispositivo elegido en el escenario.

Los atributos que contiene son: : identificador del parámetro, nombre del parámetro, valor por defecto, identificador del grupo de criterios asociado, identificador del tipo de dispositivo asociado.

Clase Parametro_instanciado_grupo

La clase Parametro_instanciado_grupo contiene los datos de un parámetro definido por el usuario que sustituye al valor definido por defecto.

Los atributos que contiene son: Los atributos que contiene son: identificador del parámetro instanciado, valor del parámetro instanciado e identificador del parámetro por defecto al que se vincula.

Clase TipoCaptador

La clase TipoCaptador contiene los datos pertenecientes a un tipo de captador en concreto.

Clase TipoDispositivo

La clase TipoDispositivo contiene los datos pertenecientes a un tipo de dispositivo en concreto.

Clase CriterioOperacion

Contiene la información almacenada de una operación en la programación de un criterio: identificador del criterio que se está programando, identificador de la operación que interviene, NombreBBDDOL y NombreTablaOL son los nombres de la ubicación de la capa de datos de entrada que interviene en la operación, Prece es el atributo que almacena la/s operaciones/s que la preceden para la entrada de datos, Orden es el número de orden de ejecución de la operación, DisplayComentario es una bandera para la utilización del campo Comentario, Comentario es la entrada procedente de los expertos, DisplayParamInst es la bandera para la utilización del campo NombresParamInst, NombresParamInst es la entrada de los nombres de los de diferentes parámetros procedentes del usuario, DisplayParamDispositivo es la bandera para la utilización del campo NombresParamDispositivo y NombresParamDispositivo es la entrada de los nombres de los diferentes parámetros procedentes del dispositivo.

Clases de las operaciones específicas

Son las clases: *Operación_region*, *Operación_reclass*, *Operación_buffer*, *Operación_fuzzy*, *Operación_mlp*, *Operación_r_max*, *Operación_andbooleano*, *Operación_owa2*, *Operación_owa3*, *Operación_v_in*, *Operación_v_to_rast*, *Operación_r_distance*, *Operación_r_null*, *Operación_r_in*, etc.

Cada operación tendrá unos atributos concretos especificados por su interfaz.

- **Clases de apoyo:**

grupoYparametros

La clase grupoYparametros alberga los parámetros que intervienen en un grupo de criterios: los parámetros específicos del grupo de criterios y los parámetros de los criterios que pertenecen a ese grupo.

- **Interfaces para las operaciones específicas:**

Son las interfaces de las operaciones SIG específicas del sistema: *iOperacion_region*, *iOperacion_reclass*, *iOperacion_buffer*, *iOperacion_fuzzy*, *iOperacion_mlp*, *iOperacion_r_max*, *iOperacion_andbooleano*, *iOperacion_owa2*, *iOperacion_owa3*, *iOperacion_v_in*, *iOperacion_v_to_rast*, *iOperacion_r_distance*, *iOperacion_r_null*, *iOperacion_r_in*, etc.

- **Clases con los métodos específicos de la base de datos:**

Data_inserter_escenario

Clase estática que contiene los métodos de inserción a la base de datos relacionados con un escenario. Los métodos que debe contener son:

insertEscenario(String nombre, String mail, Integer tp, String multipoint):int -> Inserta un escenario nuevo devolviendo su identificador.

InsertPathScriptEscenario(int de, String ruta) -> Alberga la ruta y nombre del script ejecutado

updateEstadoProcesamientoEscenarioEnCola(int ide) -> Cambia el estado de un escenario cuando pasa a estar en cola de ejecución.

updateEstadoProcesamientoEscenarioProcesado(int ide) -> Cambia el estado de un escenario cuando ya se ha procesado.

updateEstadoProcesamientoEscenarioProcesando(int ide) -> Cambia el estado de un escenario cuando comienza a procesarse.

updateFechaYhoraCreacionEscenario(int ide) -> Actualiza los campos de los datos de creación de un escenario "HoraCreacion" y "FechaCreacion".

updateFechaYhoraEjecucionEscenario(int ide) -> Actualiza los campos de los datos de fin de ejecución de un escenario "FechaFinEjecucion" y "HoraFinEjecucion".

updateModoRazonamientoCEIAEscenario(int ide, int modorazonamiento) -> Actualiza el método de razonamiento para la rama de los criterios medioambientales en el escenario.

updateModoRazonamientoCFEscenario(int ide, int modorazonamiento) -> Actualiza el método de razonamiento para la rama de los criterios físicos de un escenario.

updateModoRazonamientoCOEscenario(int ide, int modorazonamiento) -> Actualiza el método de razonamiento para la rama de criterios operacionales de un escenario.

updateModoRazonamientoCREEscenario(int de, int modorazonamiento) -> Actualiza el método de razonamiento para la rama de criterios de recurso energético de un escenario.

Data_selector_escenario

Clase estática que contiene los métodos de selección a la base de datos relacionados con un escenario. Los métodos que debe contener son:

getEscenario(int ide): Entities.Escenario -> Devuelve un escenario según su identificador

getEscenarioEnCola(): Entities.Escenario -> Devuelve el primer escenario sin procesar y le cambia el estado a cola de ejecución. Sigue una política FIFO.

getEscenariosUsuario(String email): List<Entities.Escenario> -> Devuelve una lista de los escenarios pertenecientes a un usuario.

getEstadoProcesamientoEscenario(int ide): int -> Consigue el estado de procesamiento de un escenario.

getModoRazonamientoCEIA(int ide): int -> Devuelve el método de razonamiento de los criterios parciales de la rama de criterios medioambientales.

getModoRazonamientoCF(int ide): int -> Devuelve el método de razonamiento de los criterios parciales de la rama de criterios físicos.

getModoRazonamientoCO(int ide): int -> Devuelve el método de razonamiento de los criterios parciales de la rama de criterios operacionales.

getModoRazonamientoCRE(int ide): int -> Devuelve el método de razonamiento de los criterios parciales de la rama de criterios de recurso energético.

getNumeroEscenariosEnProcesamiento(): int -> Devuelve el número de escenarios en estado de ejecución.

getParametrosRegionUsuario(int ide): ArrayList<Double> -> Devuelve los parámetros norte, sur, este y oeste de la región configurada de un escenario.

getPesosRTDFEscenario(int ide): ArrayList<Double> -> Devuelve los pesos y pesos de orden de la toma de decisión final de un escenario.

Data_inserter_dispositivo

Clase estática que contiene los métodos de inserción a la base de datos relacionados con un dispositivo. Los métodos que debe contener son:

crearNuevoDispositivo(String Nombre, String TipoDispositivo, String TipoCaptador, String AlturaBuje, String UnidadesAB, String DiametroRotor, String UnidadesDR, String AlturaDispositivo, String UnidadesAD, String DimX, String UnidadesDimX, String DimY, String UnidadesDimY, String DimZ, String UnidadesDimZ, String ProfMax, String UnidadesProfMax, String ProfMin, String UnidadesProfMin, String HsMax, String UnidadesHsMax, String CoMax, String UnidadesCoMax, String PotenciaUnitaria, String UnidadesPU, String SepDispLinea, String UnidadesSDL, String SepLineas, String UnidadesSL): int -> Crea un dispositivo nuevo con las características pasados por parámetros. Devuelve el identificador asignado en la inserción de la base de datos.

Data_selector_dispositivo

Clase estática que contiene los métodos de selección a la base de datos relacionados con un dispositivo. Los métodos que debe contener son:

getCaracteristicasDispositivo(int idd): Entities.Dispositivo -> Devuelve el dispositivo concreto asociado al identificador pasado por parámetros.

getDispositivos(): List<Entities.Dispositivo> -> Devuelve la lista de dispositivos existentes en el sistema.

getNombreDisp(int idtd): String-> Devuelve el nombre del tipo de dispositivo a partir de su identificador.

getTipoCaptadores(): List<Entities.TipoCaptador> -> Devuelve la lista de tipos de captadores existentes en el sistema.

getTipoDispDispositivo(int idd): int -> Devuelve el tipo de dispositivo de un dispositivo concreto a partir de su identificador.

getTipoDispositivos(): List<Entities.TipoDispositivo> -> Devuelve una lista de tipos de dispositivos existentes en el sistema.

getUnitsParamDisp(String nombreParam):ArrayList<String> -> Devuelve la lista de las unidades permitidas para un parámetro del dispositivo.

Data_inserter_usuario

Clase estática que contiene los métodos de inserción a la base de datos relacionados con un usuario. Los métodos que debe contener son:

createNewUser(String nombre, String apellido1, String apellido2, String nick, String dni, String pass, String mail): boolean -> Es el método de creación de un usuario nuevo en la base de datos. Devuelve un booleano con la comprobación previa de la existencia del usuario.

Data_selector_usuario

Clase estática que contiene los métodos de selección a la base de datos relacionados con un usuario. Los métodos que debe contener son:

getUsuario(String email): Entities.Usuario -> Devuelve un usuario según el correo pasado por parámetro.

Data_selector_criterio

Clase estática que contiene los métodos de selección a la base de datos relacionados con un criterio. Los métodos que debe contener son:

getCriteriosEscenario(int ide): List<Entities.Criterio> -> Devuelve los criterios seleccionados de un escenario creado.

criterioEstaInstanciado(Entities.Criterio criterio, Entities.Escenario escenario): boolean -> Consulta si un criterio está seleccionado para un escenario concreto.

getCriterioId(Integer id): Entities.Criterio -> Devuelve un criterio según el identificador pasado por parámetro.

getCriterioOperaciones(int id): ArrayList<Entities.CriterioOperacion> -> Devuelve la programación de cada criterio.

getCriteriosAmbParcialDisp(Entities.Escenario escenario): List<Entities.Criterio> -> Devuelve una lista de criterios medioambientales de exclusión parcial disponibles a partir del tipo de dispositivo asociado al escenario.

getCriteriosAmbTotalDisp(Entities.Escenario escenario): List<Entities.Criterio> -> Devuelve una lista de criterios medioambientales de exclusión total disponibles a partir del tipo de dispositivo asociado al escenario.

getCriteriosFisParcialDisp(Entities.Escenario escenario): List<Entities.Criterio> -> Devuelve una lista de criterios físicos de exclusión parcial disponibles a partir del tipo de dispositivo asociado al escenario.

getCriteriosFisTotalDisp(Entities.Escenario escenario): List<Entities.Criterio> -> Devuelve una lista de criterios físicos de exclusión total disponibles a partir del tipo de dispositivo asociado al escenario.

getCriteriosOperacParcialDisp(Entities.Escenario escenario): List<Entities.Criterio> -> Devuelve una lista de criterios operacionales de exclusión parcial disponibles a partir del tipo de dispositivo asociado al escenario.

getCriteriosOperacTotalDisp(Entities.Escenario escenario): List<Entities.Criterio> -> Devuelve una lista de criterios operacionales de exclusión total disponibles a partir del tipo de dispositivo asociado al escenario.

getCriteriosRecEnergeticoParcialDisp(Entities.Escenario escenario): List<Entities.Criterio> -> Devuelve una lista de criterios de recurso energético de exclusión parcial disponibles a partir del tipo de dispositivo asociado al escenario.

getCriteriosRecEnergeticoTotalDisp(Entities.Escenario escenario): List<Entities.Criterio> -> Devuelve una lista de criterios de recurso energético de exclusión total disponibles a partir del tipo de dispositivo asociado al escenario.

getDescripcionCriterio(int idc): String -> Devuelve la descripción del factor de un criterio.

getDescripcionParametroInstanciado(int idparaminst): String -> Devuelve la descripción de un parámetro instanciado.

getNombreTipo1(int idc): String -> Devuelve el tipo de ámbito de un criterio según el identificador pasado por parámetro.

getNombreTipo2(int idc): String -> Devuelve el tipo de exclusión de un criterio según el identificador pasado por parámetro.

getParametros(int idc, Entities.Escenario escenario): List<Entities.ParametroCriterio> -> Devuelve la lista de parámetros por defecto de un criterio según el tipo de dispositivo seleccionado asociado al escenario.

getParametrosInstanciados(Entities.Criterio criterio, Entities.Escenario escenario): List<Entities.ParametroInstanciadoCriterio> -> Devuelve la lista de parámetros instanciados de un criterio configurados en un escenario.

getCriterios(): List<Entities.Criterio> -> Devuelve una lista de criterios existentes en el sistema.

Data_inserter_criterio

Clase estática que contiene los métodos de inserción/borrado a la base de datos relacionados con los criterios. Los métodos que debe contener son:

deleteCEIAEscenarioNoSeleccionadas(int ide, List<Entities.grupoYparametros> grupos) -> Borra los criterios medioambientales de exclusión total y parcial del escenario que anteriormente estaban seleccionadas y actualmente no se han seleccionado.

deleteCFEEscenarioNoSeleccionadas(int ide, List<Entities.grupoYparametros> grupos) -> Borra los criterios físicos de exclusión total y parcial del escenario que anteriormente estaban seleccionadas y actualmente no se han seleccionado.

deleteCOEscenarioNoSeleccionadas(int ide, List<Entities.grupoYparametros> grupos) -> Borra los criterios operacionales de exclusión total y parcial del escenario que anteriormente estaban seleccionadas y actualmente no se han seleccionado.

deleteCREEscenarioNoSeleccionadas(int ide, List<Entities.grupoYparametros> grupos) -> Borra los criterios de recurso energético de exclusión total y parcial del escenario que anteriormente estaban seleccionadas y actualmente no se han seleccionado.

deleteRTDFEscenarioNoSeleccionadas(int ide) -> Borra la regla de la toma de decisión final.

Insert_Grupo_CriteriosGrupo_y_parametros(List<Integer> criterios_grupo, List<Entities.ParametroInstanciadoCriterio> parametrosnuevoscriterio, List<Entities.ParametroInstanciadoGrupo> parametrosnuevosgrupo, Entities.Escenario escenario, int idgrupo, String tiporazonamientoparciales, int tiporazgrupo) -> Liga a un escenario los grupos de criterios seleccionados junto con sus criterios y parámetros asociados.

Data_selector_grupo

Clase estática que contiene los métodos de selección a la base de datos relacionados con los grupos de criterios. Los métodos que debe contener son:

esGrupoParcial(int idg): boolean -> Verifica si un grupo de criterios es de exclusión parcial.

getDescripcionGrupo(int idg): String -> Devuelve la descripción del grupo pasado como identificador por parámetro.

getGrupoDeCriterio(int idc): int -> Devuelve el identificador de grupo al que pertenece un criterio.

getGruposDeCriterios(ArrayList<Entities.Criterio> criterios): ArrayList<Integer> -> Devuelve una lista de identificador de grupos de criterios a los que pertenecen los criterios pasados por parámetro.

getMetRazGrupo(int idg, int ide):int -> Devuelve el método de razonamiento por el que se van a combinar los criterios del grupo de criterios. Si el método de razonamiento está fijado por los

expertos para un grupo de criterios devuelve el método fijado por los expertos, sino devuelve el método de razonamiento seleccionado para un grupo de criterios y un escenario.

getMetRazGrupoEscenario(int ide, int idg): int -> Devuelve el método de razonamiento para un grupo de criterios y un escenario.

getMetRazGrupoExpertos(int idg): int -> devuelve el método de razonamiento fijado por los expertos para un grupo de criterios.

getNumeroCriteriosGrupo(int idg, ArrayList<Entities.Criterio> criterios): int -> Devuelve el número de criterios que tiene un grupo de criterios.

getParametrosDefectoGrupo(int idg, int idd): List<Entities.ParametroGrupo> -> Devuelve la lista de parámetros por defecto de un grupo de criterios.

getParametrosInstanciadosGrupo(int ide, int idg): List<Entities.ParametroInstanciadoGrupo> -> Devuelve la lista de parámetros instanciados para un grupo de criterios en un escenario.

grupoEstaInstanciado(int idg, int ide): boolean -> Verifica si un grupo ha sido seleccionado en un escenario.

metodoRazGrupoEstaFijado(int idg): boolean -> Verifica si el método de razonamiento para un grupo de criterios está fijado por los expertos.

Data_inserter_resultado

Clase estática que contiene los métodos de inserción a la base de datos relacionados con los resultados. Los métodos que debe contener son:

deleteAllResultadosIntermedios(int ide) -> Borra los resultados intermedios que están ligados a un escenario.

insertResultadoFinal(int ide, Entities.Resultado resultadoFinal) -> Método de inserción del resultado final de un escenario.

insertResultadoIntermedio(int ide, Entities.Resultado resultadoIntermedio) -> Método de inserción de un resultado intermedio de un escenario.

setRutaMapResultado(int idresultado, String rutamap) -> Almacena la ruta del fichero .MAP del resultado para su visualización.

Data_selector_resultado

Clase estática que contiene los métodos de selección a la base de datos relacionados con los resultados. Los métodos que debe contener son:

getResultado(int idresultado): Entities.Resultado -> Devuelve un resultado a partir de un identificador pasado por parámetro.

getResultadoFinalEscenario(int ide): Entity.Resultado -> Devuelve el resultado final de un escenario.

getResultadosIntermedios(int ide): List<Entities.Resultado> -> Devuelve la lista de resultados intermedios de un escenario.

getRutaMap(int idresultado): String -> Devuelve la ruta del fichero .MAP de un resultado.

Data_selector_operacion

Clase estática que contiene los métodos de selección a la base de datos relacionados con las operaciones. Los métodos que debe contener son:

getDatosResultado(): ArrayList<String> -> Devuelve, en forma de vector de cadenas, el nombre de la base de datos donde se encuentra el resultado (en caso que sea un resultado vectorial) y el nombre del resultado que se tiene que adjudicar a un nuevo resultado de una operación del sistema.

getOperacionAndbooleano(): Entities.OperacionAndbooleano -> Devuelve una operación ANDBOOLEANO

getOperacionAndbooleano(int id, String nombre): Entities.OperacionAndbooleano -> Devuelve una operación ANDBOOLEANO pasando el identificador y nombre de la operación.

getOperacionAndfuzzy(): Entities.OperacionAndfuzzy -> Devuelve una operación ANDFUZZY

getOperacionAndfuzzy(int id, String nombre): Entities.OperacionAndfuzzy -> Devuelve una operación ANDFUZZY pasando el identificador y nombre de la operación.

getOperacionBuffer(int id, String nombretipo): Entities.OperacionBuffer -> Devuelve una operación BUFFER pasando el identificador y nombre de la operación.

getOperacionFuzzy(int id, String nombretipo): Entities.OperacionFuzzy -> Devuelve una operación FUZZY pasando el identificador y nombre de la operación.

getOperacionId(int id): Entities.Operacion -> Devuelve una operación según su identificador.

getOperacionMlp(): Entities.OperacionMlp -> Devuelve una operación MLP

getOperacionMlp(int id, String nombretipo): Entities.OperacionMlp -> Devuelve una operación MLP pasando el identificador y nombre de la operación.

getOperacionOwa2(): Entities.OperacionOwa2 -> Devuelve una operación OWA de dos operandos.

getOperacionOwa2(int id, String nombretipo): Entities.OperacionOwa2 -> Devuelve una operación OWA de dos operandos pasando el identificador y nombre de la operación.

getOperacionOwa3(): Entities.OperacionOwa3 -> Devuelve una operación OWA de tres operandos .

getOperacionOwa3(int id, String nombretipo): Entities.OperacionOwa3 -> Devuelve una operación OWA de tres okperandos pasando el identificador y nombre de la operación.

getOperacionRAndEscalar(int id, String nombretipo): Entities.OperacionRAndEscalar -> Devuelve una operación R_AND_ESCALAR pasando el identificador y nombre de la operación.

getOperacionRBuffer(int id, String nombretipo): Entities.OperacionRBuffer -> Devuelve una operación R_BUFFER pasando el identificador y nombre de la operación.

getOperacionRCopyMapcalc (int id, String nombretipo): Entities.OperacionRCopyMapcalc -> Devuelve una operación R_COPY_MAPCALC pasando el identificador y nombre de la operación.

getOperacionRDistance(int id, String nombretipo): Entities.OperacionRDistance -> Devuelve una operación R_DISTANCE pasando su identificador y nombre de la operación.

getOperacionRForm(int id, String nombretipo): Entities.OperacionRForm -> Devuelve una operación R_FORM pasando su identificador y nombre de la operación.

getOperacionRIn(int id, String nombretipo): Entities.OperacionRIn -> Devuelve una operación R_IN pasando su identificador y nombre de la operación.

getOperacionRMax(): Entities.OperacionRMax -> Devuelve una operación R_MAX

getOperacionRMax(int id, String nombretipo): Entities.OperacionRMax -> Devuelve una operación R_MAX pasando su identificador y nombre de la operación.

getOperacionRMinus(int id, String nombretipo): Entities.OperacionRMinus -> Devuelve una operación R_MINUS pasando su identificador y nombre de la operación.

getOperacionRMinusEscalar(int id, String nombretripo): Entities.OperacionRMinusEscalar -> Devuelve una operación R_MINUS_ESCALAR pasando su identificador y nombre de la operación.

getOperacionRNull(int id, String nombretipo): Entities.OperacionRNull -> Devuelve una operación R_NULL pasando su identificador y nombre de la operación.

getOperacionROut(): Entities.OperacionROut -> Devuelve una operación R_OUT pasando su identificador y nombre de la operación.

getOperacionRPatch(int id, String nombretipo): Entities.OperacionRPatch -> Devuelve una operación R_PATCH pasando su identificador y nombre de la operación.

getOperacionReclass(int id, String nombretipo): Entities.OperacionReclass -> Devuelve una operación RECLASS pasando su identificador y nombre de la operación.

getOperacionRegion(): Entities.OperacionRegion -> Devuelve una operación REGION

getOperacionRegion(int id, String nombretipo): Entities.OperacionRegion -> Devuelve una operación REGION pasando su identificador y nombre de la operación.

getOperacionVIn(int id, String nombretipo): Entities.OperacionVIn -> Devuelve una operación V_IN pasando su identificador y nombre de la operación.

getOperacionVToRast(int id, String nombretipo): Entities.OperacionVToRast -> Devuelve una operación V_TO_RAST pasando su identificador y nombre de la operación.

getOperacionVToRastColumn(int id, String nombretipo): Entities.OperacionVToRastColumn -> Devuelve una operación V_TO_RAST_COLUMN pasando su identificador y nombre de la operación.

Utilities

Clase estática que contiene diferentes métodos relacionados con la creación de ficheros .MAP, conversores, etc. Los métodos que debe contener son:

diccionarioUnidades(String units): String -> Pasar unidades de inglés a español

generateClass(string color, String outlinecolor, String width, String type): String -> Genera una clase en cadena de texto de un fichero .MAP

generateFooter(): String -> Devuelve el pie del fichero .MAP en cadena de texto.

generateHeaderMapaRaster(String mapName, String units, boolean debug, ArrayList<Double> extendNSEW): String -> Devuelve la cabecera de un fichero .MAP de un mapa raster.

generateLayer(String layerName, String type, String connectiontype, String connection, String data): String -> Genera la parte LAYER de un fichero .MAP para un mapa vectorial

generateLayerRasterEscalaColores(String layerName, String type, String ruta fichero): String -> Genera la parte LAYER de un fichero .MAP de un mapa raster

generateWeb(int epsg, String ruta): String -> Genera la parte WEB de un fichero .MAP de un mapa raster o vectorial

saveToFile(String rutamap, String content) -> Genera el fichero en el sistema de ficheros del servidor pasándole la ruta y el contenido del fichero.

convierteLongitud(String unidadesorigen, String unidadesdestino, Double valorentrada): Double -> Conversor de unidades de la escala de longitud

convierteVelocidad(String unidadesorigen, String unidadesdestino, Double valorentrada): Double -> Conversor de unidades de la escala velocidad

createMAPResultado(int idresultado, String type, Double norte, Double sur, Double este, Double oeste): String -> Generación del fichero .MAP de MapServer para visualización de mapas.

extraeCamposCadena(String cadena, char separador): ArrayList<String> -> Devuelve un vector de cadenas a partir de una cadena con campos separados por un carácter separador.

generateMapaUnionVectorial(String vect, String tipo): String -> Genera el fichero .MAP de MapServer para visualización de mapas vectoriales almacenados en PostGIS y proyectados en EPSG: 4326

getCoordenadas(String multipoint): ArrayList<Double> -> Devuelve en un vector de reales los parámetros norte, sur, este y oeste a partir de un objeto MULTIPOINT de PostGIS.

getMetodosRazonamiento(): List<Entities.Razonamiento> -> Devuelve la lista de tipos de razonamiento posibles fijados por los expertos para la combinación de criterios parciales y/o para la combinación de subcriterios de un mismo criterio.

getMultipointsDeParametros(Double norte, Double sur, Double este, Double oeste): String -> Devuelve un objeto MULTIPOINT de PostGIS a partir de los parámetros norte, sur, este y oeste de la región configurada de un escenario.

getRutaMapCapaUnion(String nombrecapaunion): String -> Devuelve la ruta del fichero .MAP de una capa de unión del sistema

insertRutaMapServerCapaUnion(String nombrecapaunion, String rutamapserver) -> Inserta la ruta del fichero .MAP de una capa de unión del sistema

getRutaCapaRaster(String nombrecapa): String -> Devuelve la ruta de una capa raster ubicada en el sistema de ficheros del servidor.

4.2.2.3.3. Componente motor

Esta componente incluye las clases pertenecientes a la generación de script automáticos y lanzamiento de operaciones SIG en background.

Las clases del motor heredarán de las entidades que pertenecen a la componente *Entities* y en ellas se implementará los métodos de la lógica para la generación de scripts automáticos.

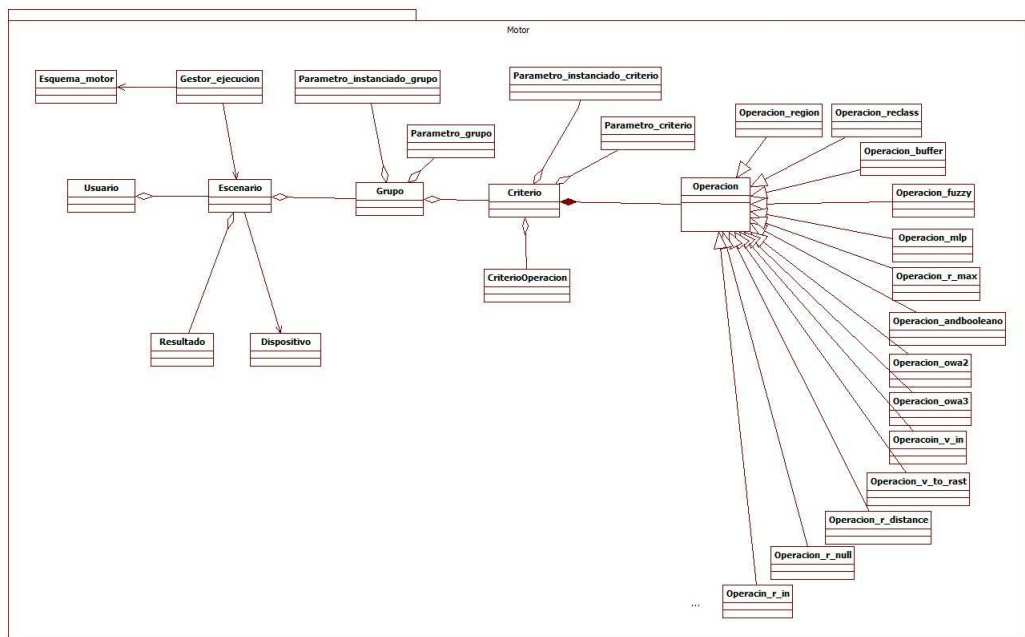


Figura 95: Componente Motor del diagrama de clases de diseño

Clase GestorEjecucion

Es la clase controladora encargada de cargar los datos de un escenario en espera de ejecución, extraer las operaciones SIG pertenecientes a los grupos de criterios y criterios seleccionados en el escenario siguiendo el esquema propuesto para la toma de decisión final (ver apartado 2.6. *El protocolo de la aplicación de los criterios*), crear el script operacional con las operaciones SIG y ejecutar las operaciones escritas en el script operacional en background sobre el SIG GRASS.

Clase EsquemaMotor

Es la clase que implementa el esquema del motor para la toma de decisión final según los grupos de criterios y criterios seleccionados de un escenario pasado como parámetro. También implementa métodos para la ampliación de operaciones N-arias como son: ANDBOOLEANO (multiplicación booleana), ANDFUZZY (mínimo), RMAX (máximo), y MLP (Media Lineal Ponderada), que según la sintaxis siguen una implementación binaria.

El esquema del motor se puede consultar en el 2.6. *El protocolo de la aplicación de los criterios*.

Clase Usuario

La clase Usuario hereda de la clase Usuario de la componente *Entitie*. Alberga los datos del usuario: DNI, nombre, primer apellido, segundo apellido, nick o alias, dirección de correo electrónico y contraseña. Contiene el agregado de Escenarios creados por el Usuario.

Clase Escenario

La clase Escenario hereda de la clase Escenario de la componente *Entitie*. En la clase Escenario se alberga los datos y funcionalidades asociados a la creación del script operacional del escenario.

Los atributos que contiene: nombre del escenario, dispositivo asociado, fecha y hora de creación del escenario, fecha y hora fin de ejecución del escenario, estado de procesamiento en el que se encuentra, localización geográfica, el agregado de grupos de criterios seleccionados y los métodos de razonamiento (más que pesimista, pesimista, optimista y neutro) para la combinación de grupo de criterios parciales de cada ámbito (físico, operacional, medioambiental o de recurso energético).

Los métodos más relevantes que contiene: agrupación de los criterios seleccionados en los grupos de criterios que les corresponde, creación del script operacional (archivo .BAT) en el sistema de archivos del servidor y configuración de la operación región a partir de los parámetros configurados del escenario.

Clase Dispositivo

La clase Dispositivo hereda de la clase Dispositivo de la componente *Entitie*. En la clase Dispositivo se alberga los datos asociados al dispositivo elegido en el escenario: nombre dispositivo, identificador del tipo de dispositivo (según sea undimotriz, maremotriz o eólico), identificador del tipo de captador (según sea de eje horizontal o vertical), altura de buje, unidades de altura de buje, altura de dispositivo, unidades de altura de dispositivo, diámetro rotor, unidades de diámetro rotor, dimensión X, Y y Z, unidades de la dimensión X, Y y Z, profundidad máxima y mínima, unidades para la profundidad máxima y mínima, altura de ola máxima permitida, unidades de la altura de ola máxima permitida, velocidad de corriente máxima permitida, unidades de la velocidad de corriente máxima permitida, potencia, unidades de potencia, separación entre dispositivos de una misma línea, unidades de separación entre dispositivos de una misma línea, separación entre líneas de dispositivos, unidades de separación entre líneas de dispositivos.

Clase Resultado

La clase Resultado hereda de la clase Resultado de la componente *Entitie*. La clase Resultado alberga los datos de los resultados de un escenario: la ruta del fichero asociado al resultado (será tipo raster y estará albergado en el sistema de ficheros del servidor), nombre del resultado, ruta y nombre del fichero .MAP procedente de MapServer que se creará para su visualización web y el tipo de resultado (intermedio o final).

Clase Grupo

La clase Grupo hace referencia a un grupo de criterios. Cada grupo contiene aquellos criterios que han sido seleccionados en el escenario pertenecientes a ese grupo de criterios.

Los atributos que contiene: el método de razonamiento para la combinación de criterios dentro de ese grupo de criterios (más que pesimista, pesimista, optimista o neutro), un identificador de grupo, el tipo de ámbito del grupo de criterios (físico, operacional, medioambiental o de recurso energético), el tipo de exclusión (total o parcial), el nombre del resultado de la aplicación y combinación de los criterios seleccionados para ese grupo de criterios, el agregado de criterios seleccionados y los parámetros asociados al grupo de criterios (por defecto e introducidos por el usuario).

Clase Criterio

La clase Criterio hereda de la clase Criterio de la componente *Entitie*. En la clase Criterio se alberga los datos y funcionalidades asociados a un criterio, así como los métodos para la instanciación de las operaciones vinculadas al criterio.

Contiene los métodos para instanciar las operaciones que se han definido para ese criterio según los parámetros del usuario, los parámetros del dispositivo y los datos proporcionados por los expertos.

Los atributos que contiene: descripción del factor asociado, tipo de ámbito del criterio (físico, operacional, medioambiental, o de recurso energético), tipo de exclusión (total o parcial), agregado de parámetros, operaciones vinculadas al criterio e instancias de la clase CriterioOperacion.

Los métodos más relevantes que contiene:

estableceCapasEntradaOperaciones() -> Establece las capas de entrada de las operaciones.

getCadenasOperacionesGRASS(): ArrayList<String> -> Método que devuelve las cadenas de GRASS de las operaciones instanciadas.

getNombreResultadoUltimaOperacion(): String -> Método que devuelve el nombre del resultado de la última operación vinculada al criterio.

Clase CriterioOperacion

La clase CriterioOperacion hereda de la clase CriterioOperacion de la componente *Entitie*.

Contiene la información almacenada de una operación en la programación de un criterio: identificador del criterio que se está programando, identificador de la operación que interviene, NombreBDDOL y NombreTablaOL que son los nombres de la ubicación de la capa de datos de entrada que interviene en la operación, Prece es el atributo que almacena la/s operación/s que la preceden para la entrada de datos, Orden es el número de orden de ejecución de la operación, DisplayComentario es una bandera para la utilización del campo Comentario, Comentario es la entrada procedente de los expertos, DisplayParamInst es la bandera para la utilización del campo NombresParamInst, NombresParamInst es la entrada de los nombres de los de diferentes parámetros procedentes del usuario, DisplayParamDispositivo es la bandera para la utilización

del campo NombresParamDispositivo y NombresParamDispositivo es la entrada de los nombres de los diferentes parámetros procedentes del dispositivo.

Clase Parametro_criterio

La clase Parametro_criterio hereda de la clase Parametro_criterio de la componente *Entitie*. Contiene los datos de un parámetro definido por defecto perteneciente a un criterio. Está condicionado al tipo de dispositivo elegido en el escenario.

Los atributos que contiene son: identificador del parametro, nombre del parámetro, valor por defecto, identificador del criterio asociado, identificador del tipo de dispositivo asociado.

Clase Parametro_instanciado_criterio

La clase Parametro_instanciado_criterio hereda de la clase Parametro_instanciado_criterio de la componente *Entitie*. Contiene los datos de un parámetro definido por el usuario que sustituye al valor definido por defecto.

Los atributos que contiene son: identificador del parámetro instanciado, valor del parámetro instanciado e identificador del parámetro por defecto al que se vincula.

Clase Parametro_grupo

La clase Parametro_grupo hereda de la clase Parametro_grupo de la componente *Entitie*. Contiene los datos de un parámetro definido por defecto perteneciente a un grupo de criterios. Está condicionado al tipo de dispositivo elegido en el escenario.

Los atributos que contiene son: identificador del parametro, nombre del parámetro, valor por defecto, identificador del grupo de criterios asociado, identificador del tipo de dispositivo asociado.

Clase Parametro_instanciado_grupo

La clase Parametro_instanciado_grupo hereda de la clase Parametro_instanciado_grupo de la componente *Entitie*. Contiene los datos de un parámetro definido por el usuario que sustituye al valor definido por defecto.

Los atributos que contiene son: Los atributos que contiene son: identificador del parámetro instanciado, valor del parámetro instanciado e identificador del parámetro por defecto al que se vincula.

Clase Operación

Todas las operaciones definidas en el sistema heredarán de la clase Operación. La clase Operación hereda de la clase Operación de la componente *Entitie*.

Los atributos que contiene: el esqueleto de la operación definido dentro del entorno SIG GRASS, el nombre del tipo de operación, el agregado de los nombres de las capas de entrada a la operación y el nombre del resultado final de la operación.

El método más relevante que contiene es *instanciaCadenaGRASS* que se sobrescribirá para cada operación específica del sistema.

instanciaCadenaGRASS(ArrayList<ArrayList<String>> nombreentradas, String nombreresultado, ArrayList<Entities.ParametroInstanciadoCriterio> parametrosInstanciados, ArrayList<ParametroDispositivo> parametrosDispositivo, boolean displayComentario, String comentario, boolean displayParamInst, ArrayList<String> paramInstUtilizados, boolean displayParamDispositivo, ArrayList<String> paramDispUtilizados)-> Función que instancia el esqueleto de la cadena de la operación con los parámetros instanciados por el usuario u/o del dispositivo u/o de los expertos.

Clases de las operaciones específicas

Son las clases de las operaciones disponibles en el sistema. Cada una implementará el método *instanciaCadenaGRASS* según el estudio realizado para cada una (Ver apartado 3.5. *Operaciones definidas en el sistema y programación de criterios*).

Son las clases: *Operación_region*, *Operación_reclass*, *Operación_buffer*, *Operación_fuzzy*, *Operación_mlp*, *Operación_r_max*, *Operación_andbooleano*, *Operación_owa2*, *Operación_owa3*, *Operación_v_in*, *Operación_v_to_rast*, *Operación_r_distance*, *Operación_r_null*, *Operación_r_in*, etc.

Corrección en el diseño del componente motor

En principio se diseñó una clase de diseño adicional, llamada *ModuloRegla*, que contenía la lógica de procesamiento de los criterios para que la clase *Criterio* no estuviera sobrecargada. Posteriormente se pensó volver a realojar los métodos en la clase *Criterio* para no perder el sentido de significado de las clases participantes en la *componente del motor*, que es como actualmente se encuentra en la implementación.

El diagrama de clases que seguía anteriormente la componente Motor se muestra a continuación.

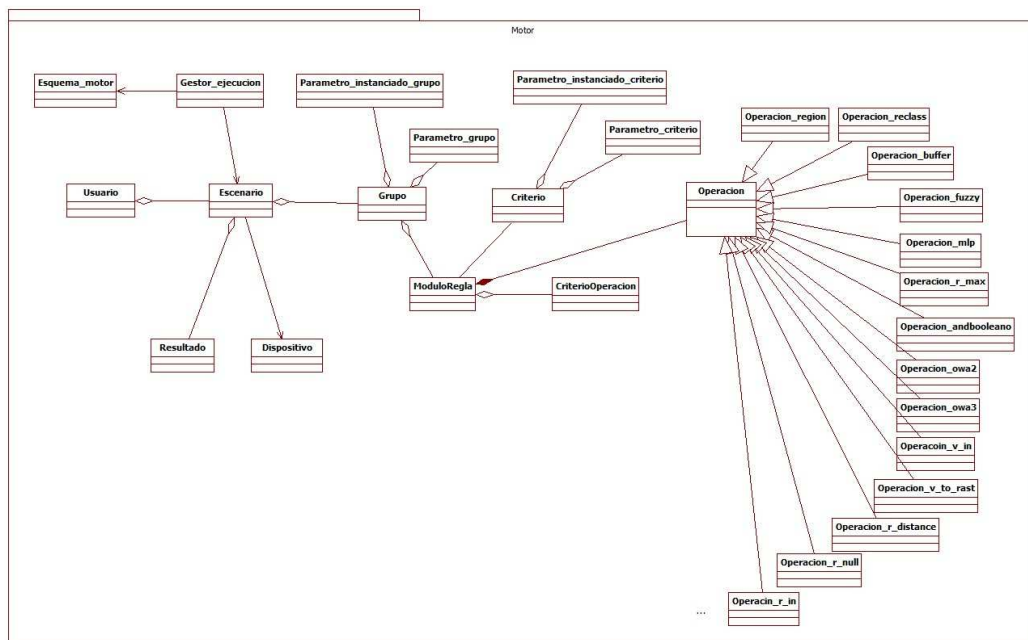


Figura 96: Componente Motor del diagrama de clases de diseño antes de la corrección

4.2.2.3.4. Componente interfaz

En esta componente intervendrán las clases de la interfaz, así como los servlet que dan servicio para la consultas en el lado del servidor.

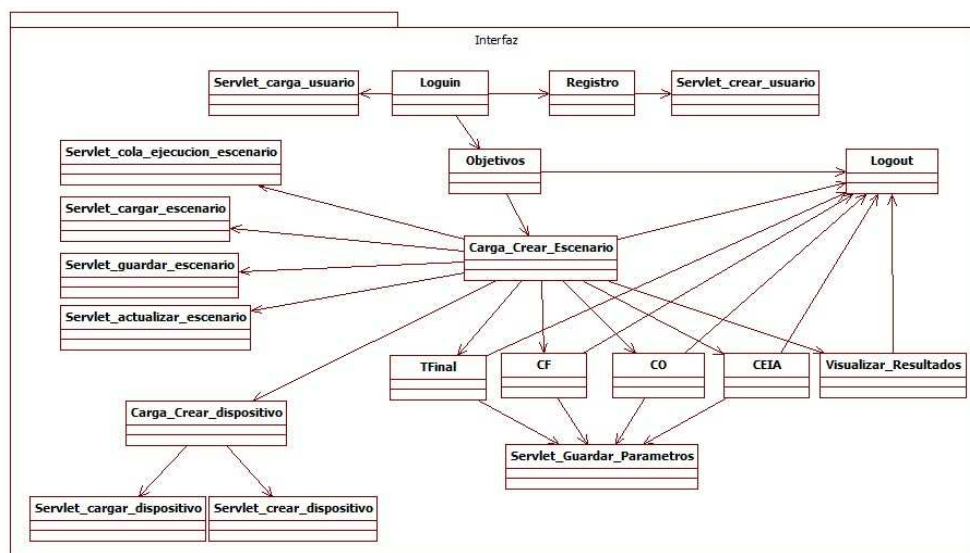


Figura 97: Componente interfaz del diagrama de clases de diseño

Las siguientes clases se implementan con JSP (JavaServer Pages) que embeben código Java junto con código HTML (y tecnologías web: jQuery, JavaScript, etc) permitiendo generar código dinámico compilando el código en el lado del servidor y entregándose en el lado del cliente.

Loguin

Es la página de inicio que da servicio de autenticación de usuario para el uso de la herramienta. En él se encuentra un enlace de registro en caso que el usuario no esté dado de alta en el sistema.

Registro

Es la página de registro para dar de alta a un usuario nuevo.

Objetivos

Es la página de presentación de la herramienta una vez se haya logueado el usuario. En esta página se hace una pequeña presentación de la herramienta, la explicación de cómo realizar los pasos para crear y ejecutar un escenario, y un ejemplo de un resultado final.

Carga_Crear_Escenario

Es la página para la creación y carga de escenarios. Se le muestra al usuario la lista de escenarios creados por él y la posibilidad de crear un nuevo escenario.

Carga_Crear_dispositivo

Es la página para la carga y creación de dispositivos. Se le muestra al usuario la lista de dispositivos creados en el sistema y la posibilidad de crear un nuevo dispositivo.

CF

Es la página para mostrar los criterios y grupo de criterios físicos disponibles para ese escenario según el tipo de dispositivo elegido.

CO

Es la página para mostrar los criterios y grupo de criterios operacionales disponibles para ese escenario según el tipo de dispositivo elegido.

CEIA

Es la página para mostrar los criterios y grupo de criterios medioambientales disponibles para ese escenario según el tipo de dispositivo elegido.

TFinal

Es la página para configurar los parámetros de la toma de decisión final según los criterios seleccionados en el escenario.

Visualizar_resultados

Es la página para la visualización de resultados una vez ejecutado un escenario.

Logout

Es la página con la funcionalidad para salirse de la aplicación.

Los siguientes servlets contienen los métodos llamados desde las páginas JSP mediante POST. Incluyen llamadas a los servicios de la base de datos implementados en la componente *Entitie*.

Servlet_carga_usuario

Contiene las llamadas específicas para la carga de un usuario, se extrae la instancia del Usuario que se ha logueado.

Servlet_crear_usuario

Contiene las llamadas para la creación de un usuario nuevo.

Servlet_cola_ejecucion_escenario

Contiene las llamadas para cambiar de estado a un escenario pasándole a cola de espera de ejecución.

Servlet_carga_escenario

Contiene las llamadas para cargar un escenario seleccionado por el usuario.

Servlet_guardar_escenario

Contiene las llamadas para guardar un escenario nuevo.

Servlet_actualizar_escenario

Contiene las llamadas para actualizar las características de un escenario ya creado.

Servlet_cargar_dispositivo

Contiene las llamadas para cargar un dispositivo.

Servlet_crear_dispositivo

Contiene las llamadas para crear un nuevo dispositivo.

Servlet_Guardar_Parametros

Contiene las llamadas para guardar los grupos de criterios y criterios seleccionados para ese escenario, y almacenar los parámetros ligados a ellos.

Los JSPs son una forma alternativa de crear servlets ya que el código JSP se traduce a código de servlet Java la primera vez que se le invoca y en adelante es el código del nuevo servlet el que se ejecuta produciendo como salida el código HTML que compone la página web de respuesta.

Los JSPs y servlets se ejecutan en una máquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de ordenador (servidor), siempre exista una máquina virtual Java.

4.2.2.4. Diagramas de secuencia

A continuación se presentan los diagramas de secuencia de análisis para cada uno de los casos de uso identificados.

4.2.2.4.1. CU-01: Autenticación usuario

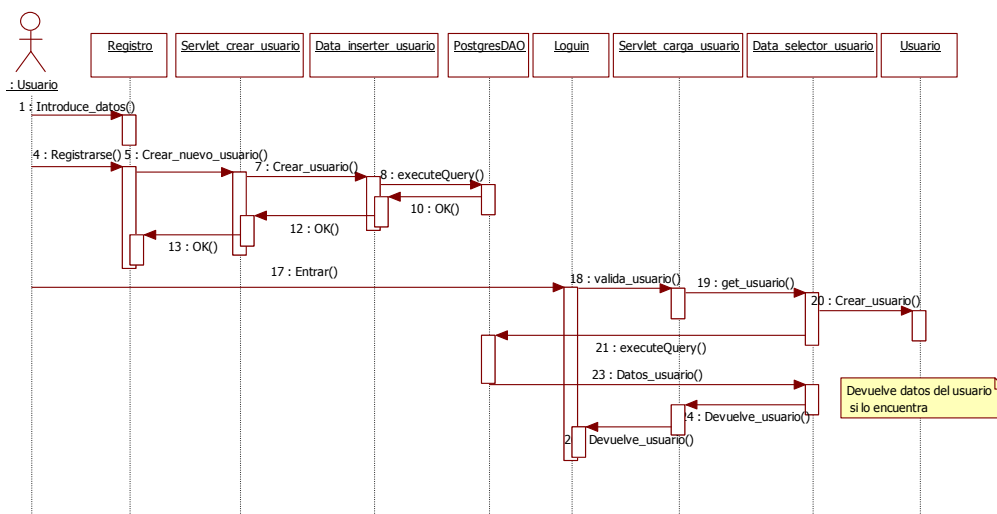


Figura 98: Diagrama de secuencia de diseño del CU Autenticación usuario

4.2.2.4.2. CU-02: Crear escenario

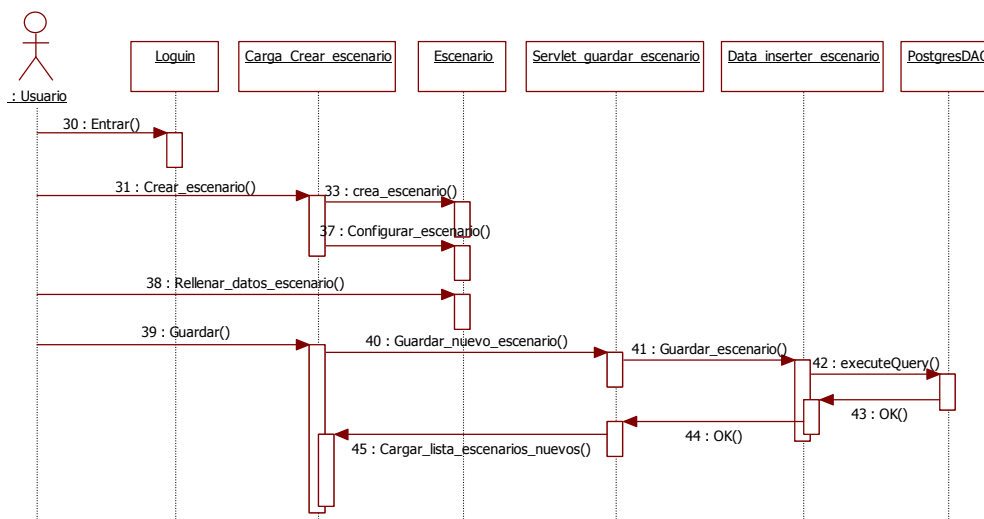


Figura 99: Diagrama de secuencia de diseño del CU Crear escenario

4.2.2.4.3. CU-03: Cargar escenario

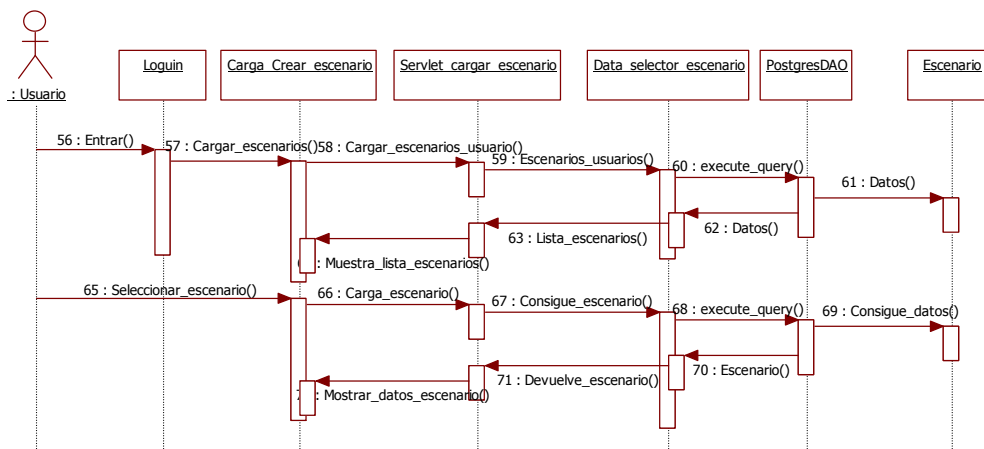


Figura 100: Diagrama de secuencia de diseño del CU Cargar escenario

4.2.2.4.4. CU-04: Realizar cálculos

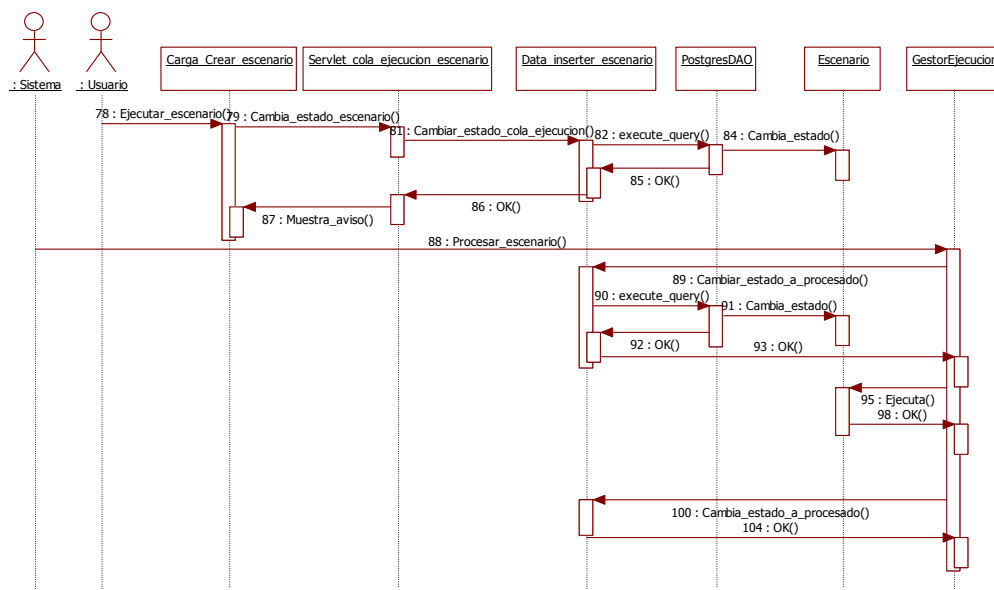


Figura 101: Diagrama de secuencia de diseño del CU Realizar cálculos

4.2.2.4.5. CU-05: Visualizar resultados

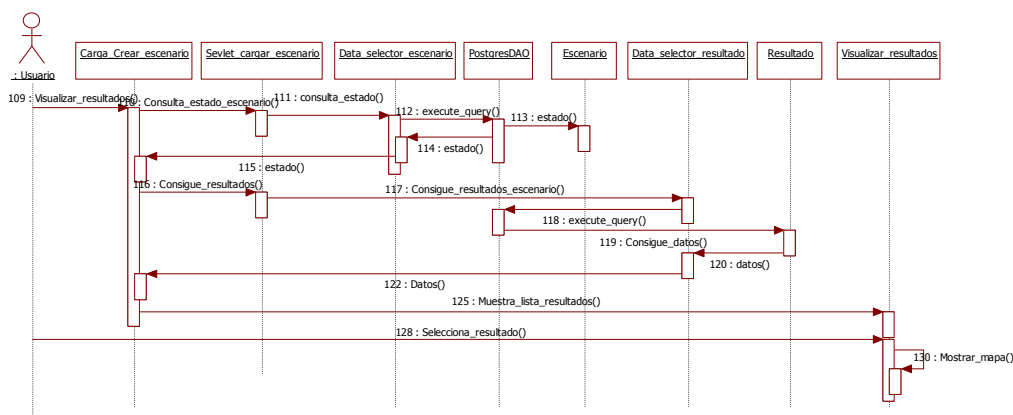


Figura 102 Diagrama de secuencia de diseño del CU Visualizar resultados

4.2.2.4.6. CU-6: Modificar y/o seleccionar criterios escenario

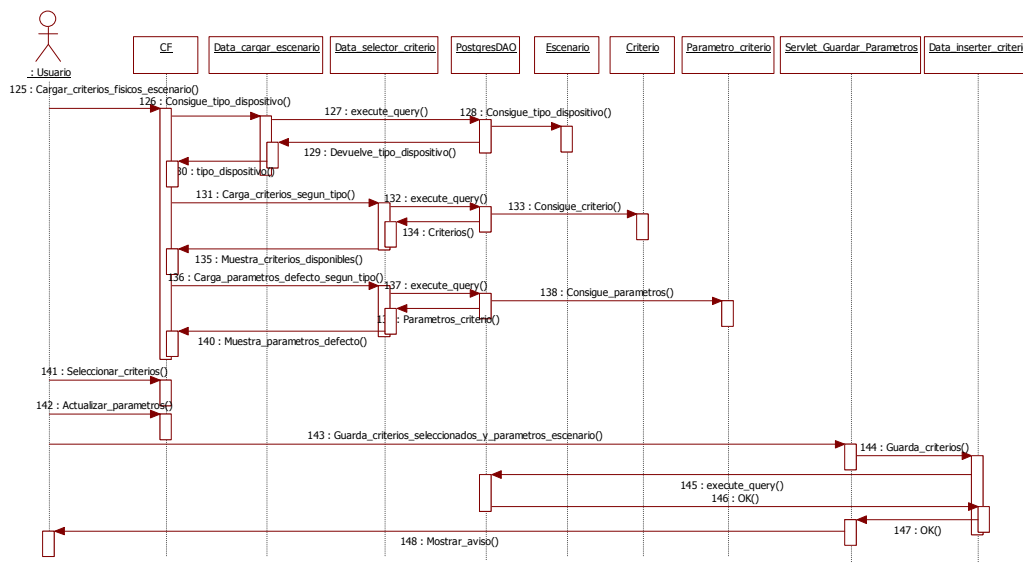


Figura 103: Diagrama de secuencia de diseño del CU Modificar y/o seleccionar criterios escenario

4.2.2.4.7. CU-7: Hacer Loguin

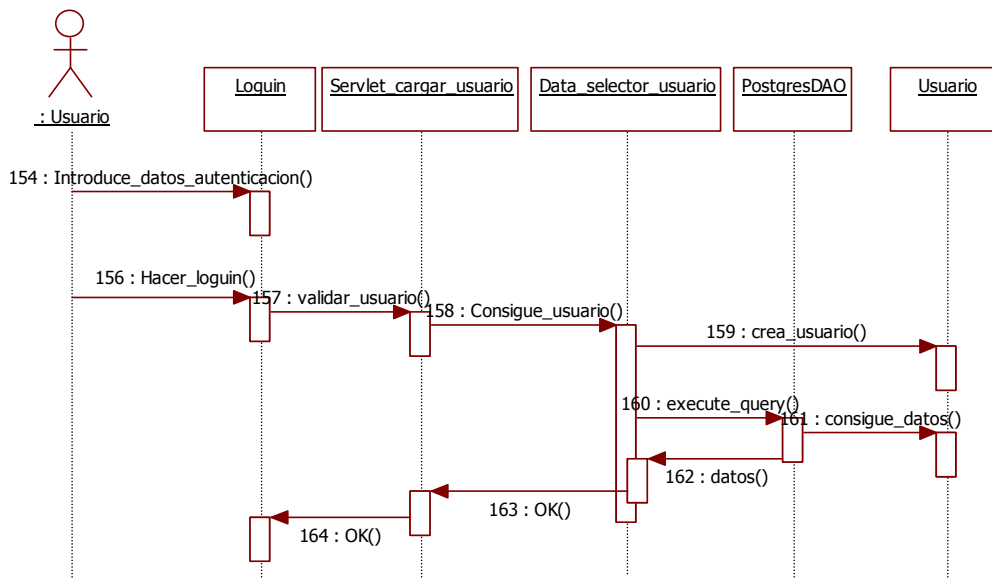


Figura 104: Diagrama de secuencia de diseño del CU Hacer loguin

4.2.2.4.8. CU-8: Registrarse

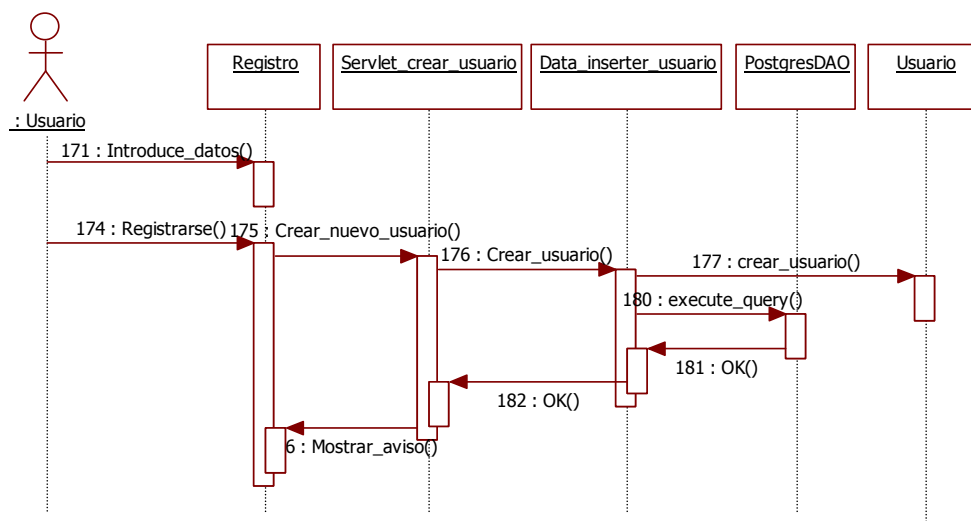


Figura 105: Diagrama de secuencia de diseño del CU Registrarse

4.2.2.4.9. CU-9: Salir de sesión

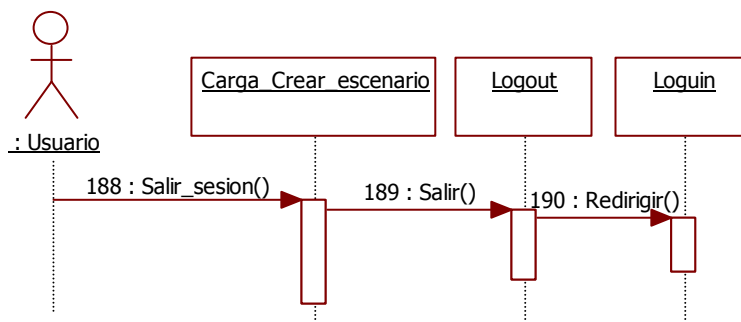


Figura 106: Diagrama de secuencia de diseño del CU Salir de sesión

4.2.2.4.10. CU-10: Seleccionar Dispositivo

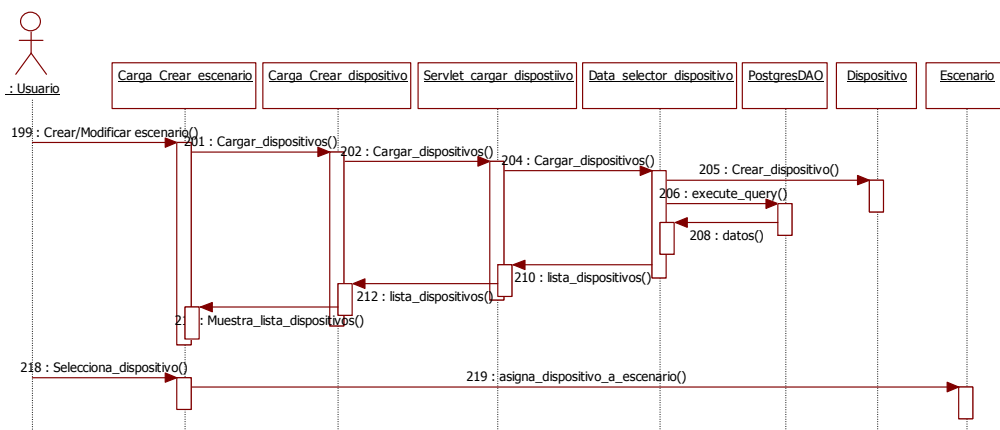


Figura 107: Diagrama de secuencia de diseño del CU Seleccionar dispositivo

4.2.2.4.11. CU-11: Seleccionar zona geográfica

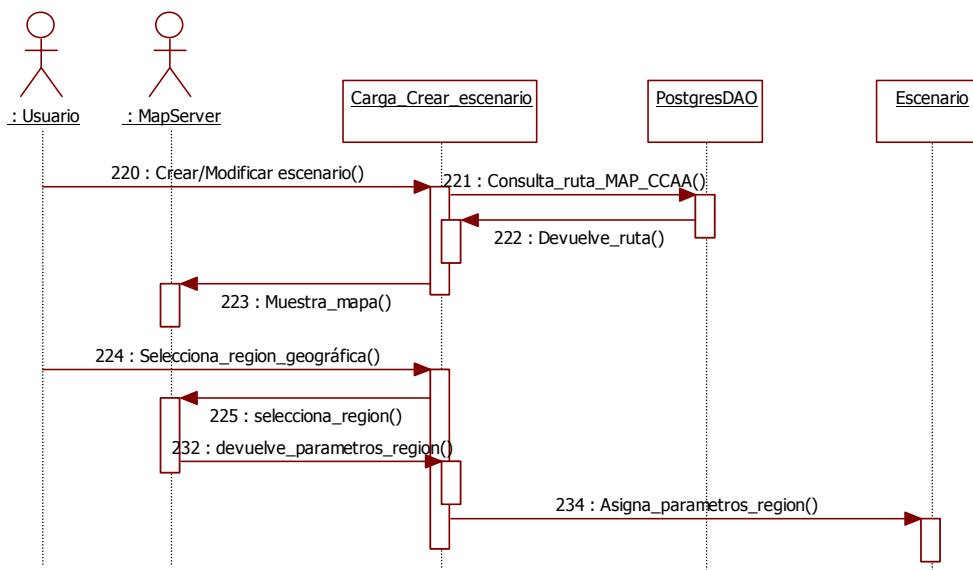


Figura 108: Diagrama de secuencia de diseño del CU Seleccionar zona geográfica

4.2.2.4.12. CU-12: Introducir parámetros escenario

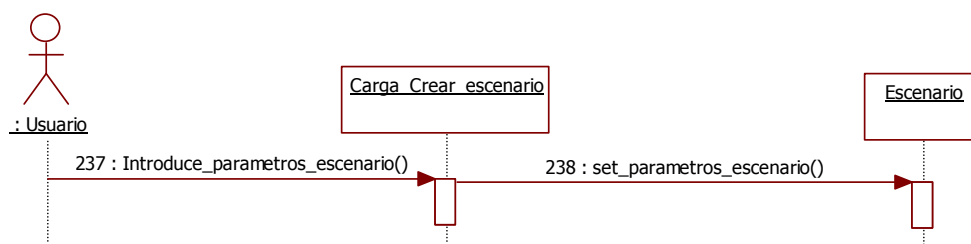


Figura 109: Diagrama de secuencia de diseño del CU Introducir parámetros escenario

4.2.2.4.13. CU-13: Modificar y/o seleccionar criterios escenario

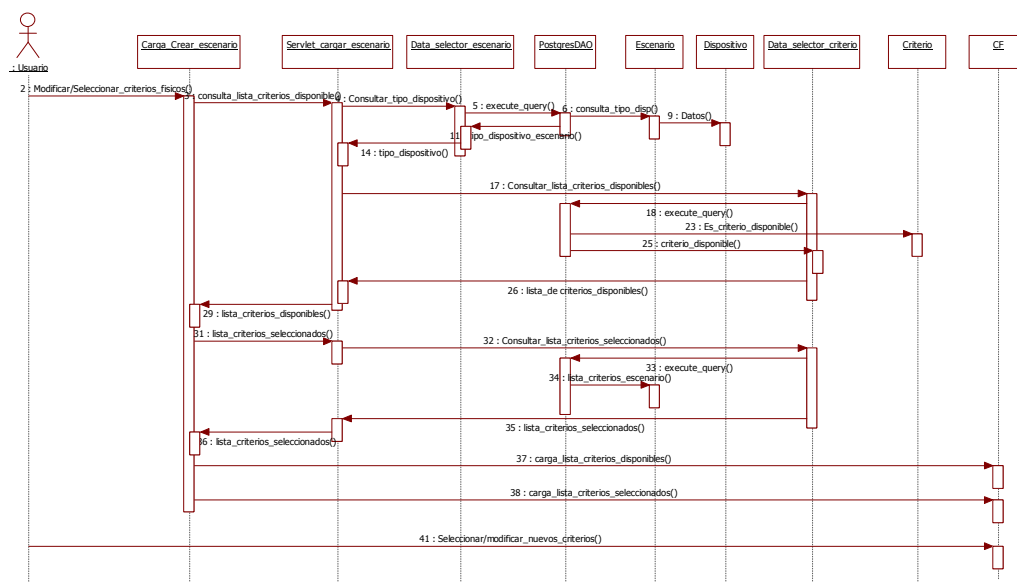


Figura 110: Diagrama de secuencia de diseño del CU Modificar y/o seleccionar criterios escenario

4.2.2.4.14. CU-14: Guardar escenario

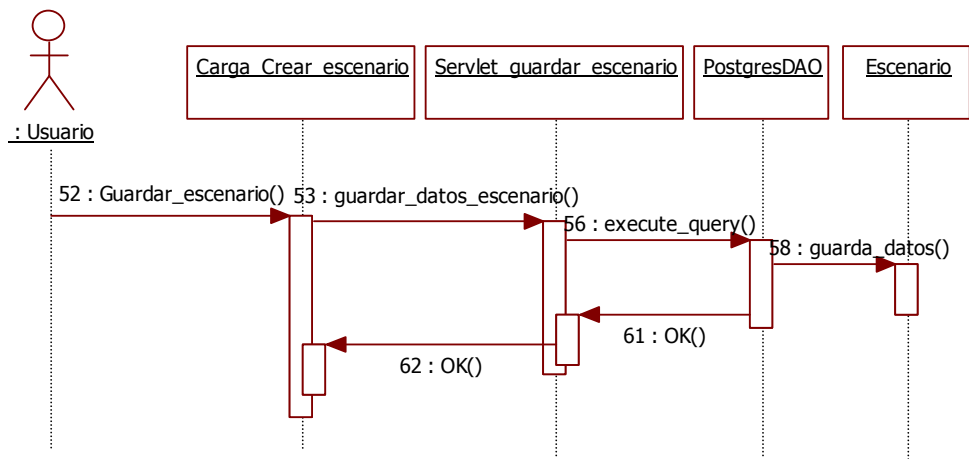


Figura 111: Diagrama de secuencia de diseño Guardar escenario

4.2.2.4.15. CU-15: Selección escenario

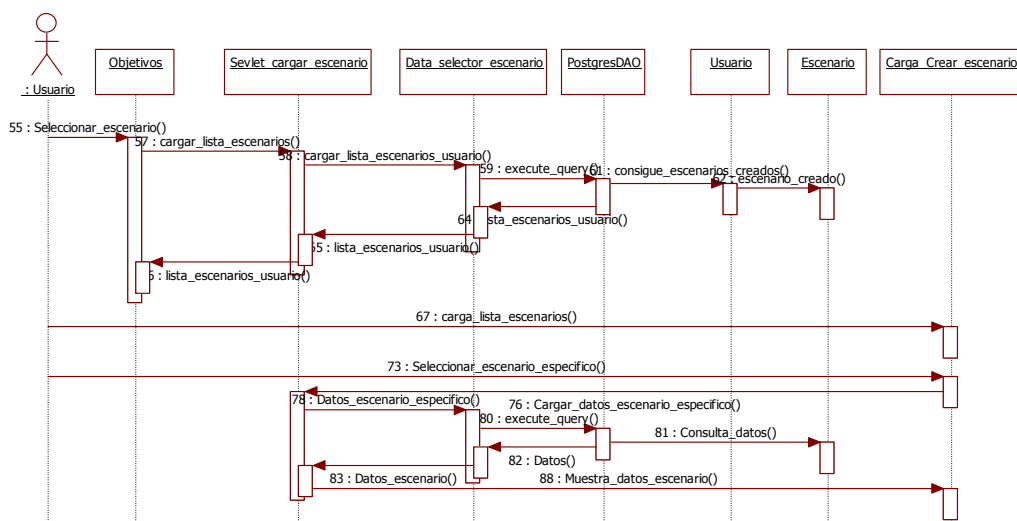


Figura 112: Diagrama de secuencia de diseño del CU Selección escenario

4.2.2.4.16. CU-16: Modificar dispositivo

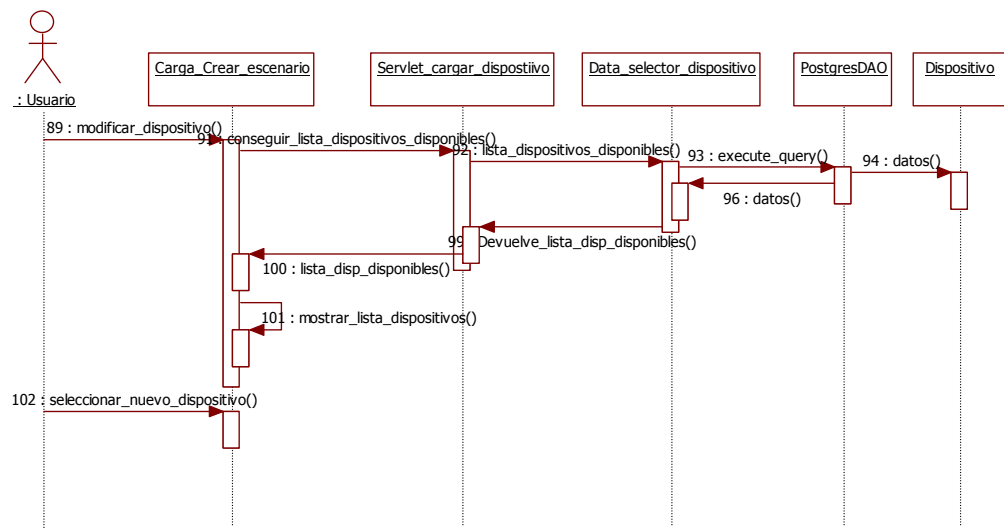


Figura 113: Diagrama de secuencia de diseño del CU Modificar dispositivo

4.2.2.4.17. CU-17: Modificar zona geográfica

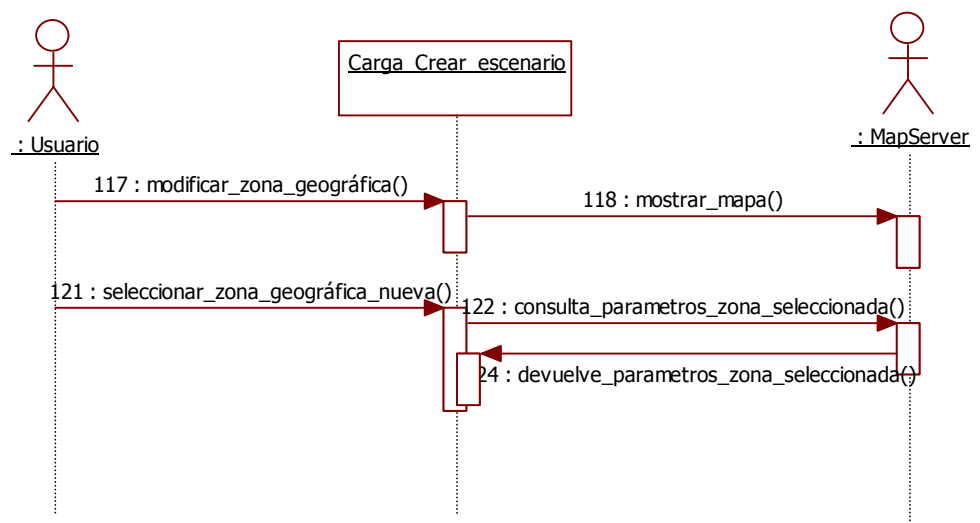


Figura 114: Diagrama de secuencia de diseño del CU Modificar zona geográfica

4.2.2.4.18. CU-18: Modificar parámetros escenario

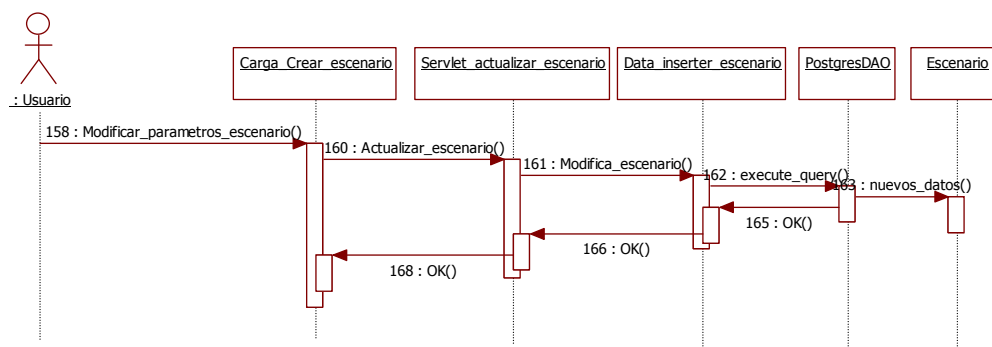


Figura 115: Diagrama de secuencia de diseño del CU Modificar parámetros escenario

4.2.2.4.19. CU-19: Modificar y/o seleccionar criterios escenario

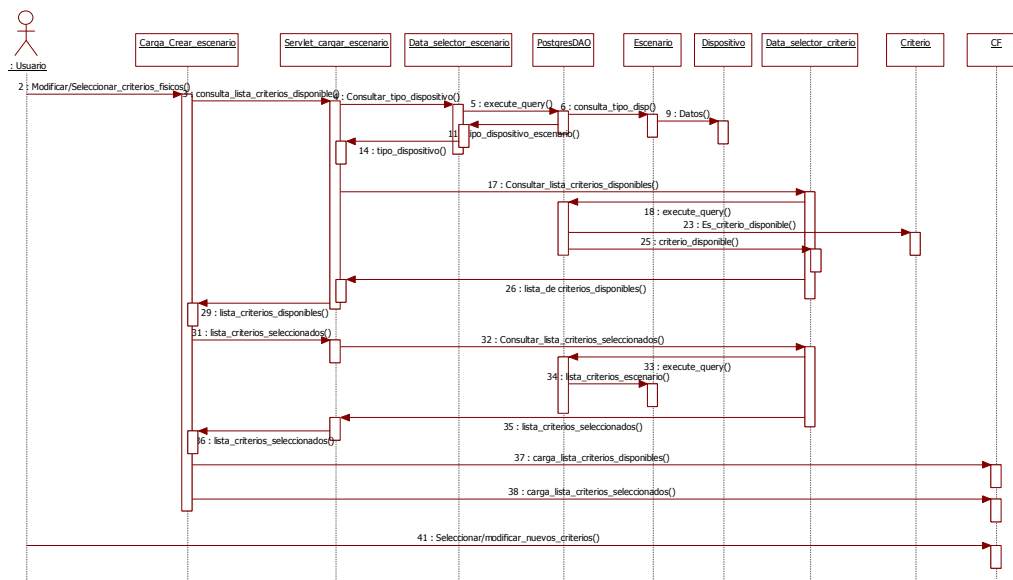


Figura 116: Diagrama de secuencia de diseño del CU Modificar y/o seleccionar criterios escenario

4.2.2.4.20. CU-20: Extraer información escenario

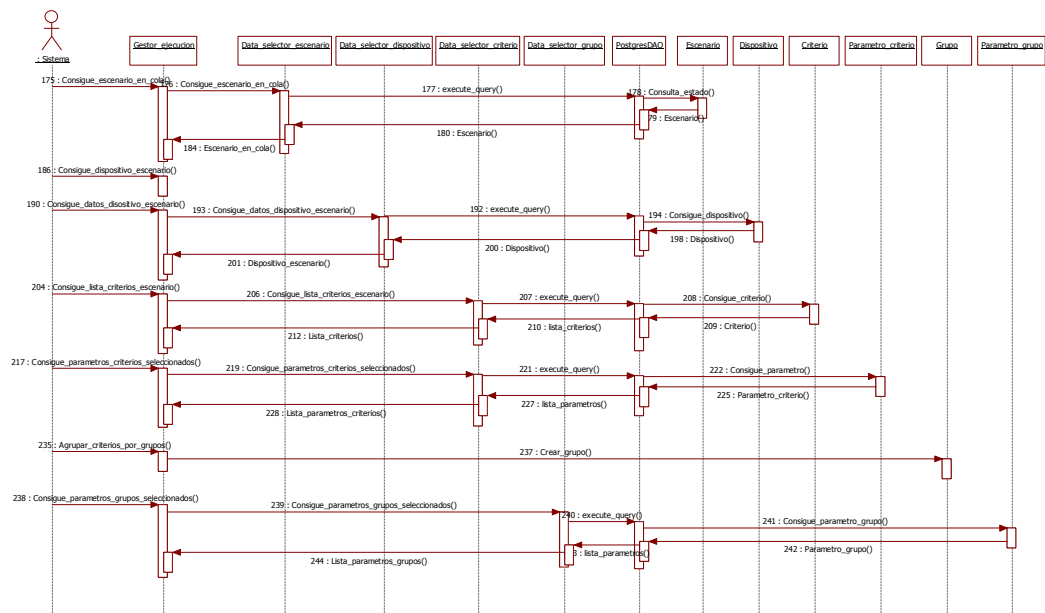


Figura 117: Diagrama de secuencia de diseño del CU Extraer información escenario

4.2.2.4.21. CU-21: Traducir/Implementar criterios a operaciones

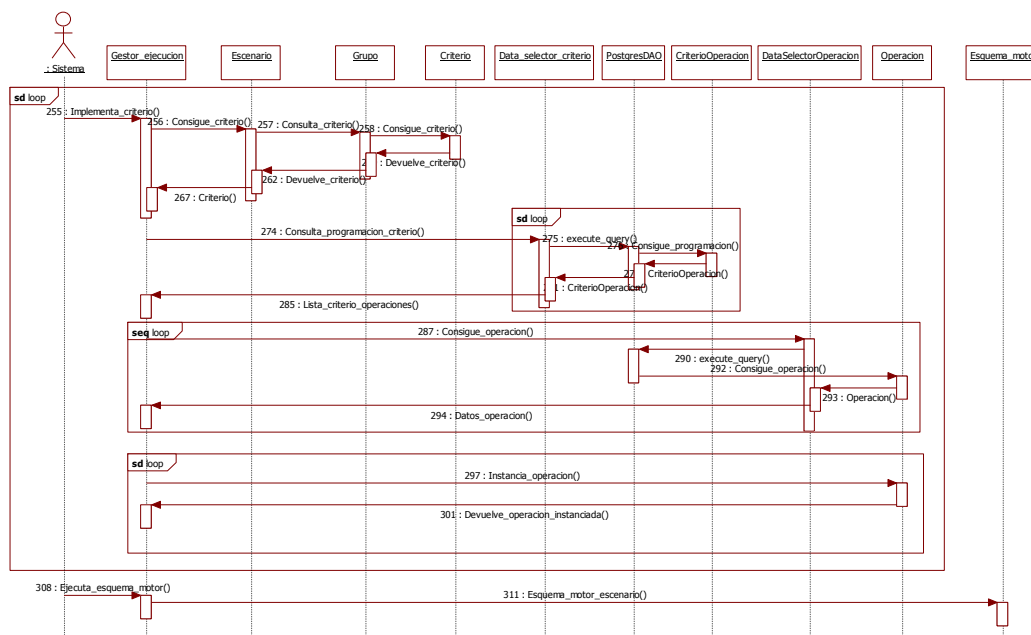


Figura 118: Diagrama de secuencia de diseño del CU Traducir/Implementar criterios a operaciones

4.2.2.4.22. CU-22: Realización script ejecutable

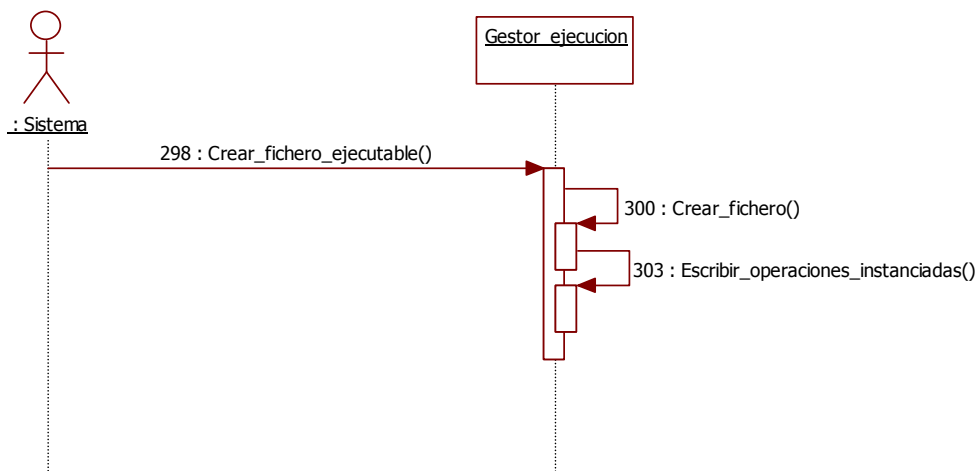


Figura 119: Diagrama de secuencia de diseño del CU Realización script ejecutable

4.2.2.4.23. CU-23: Ejecución de script

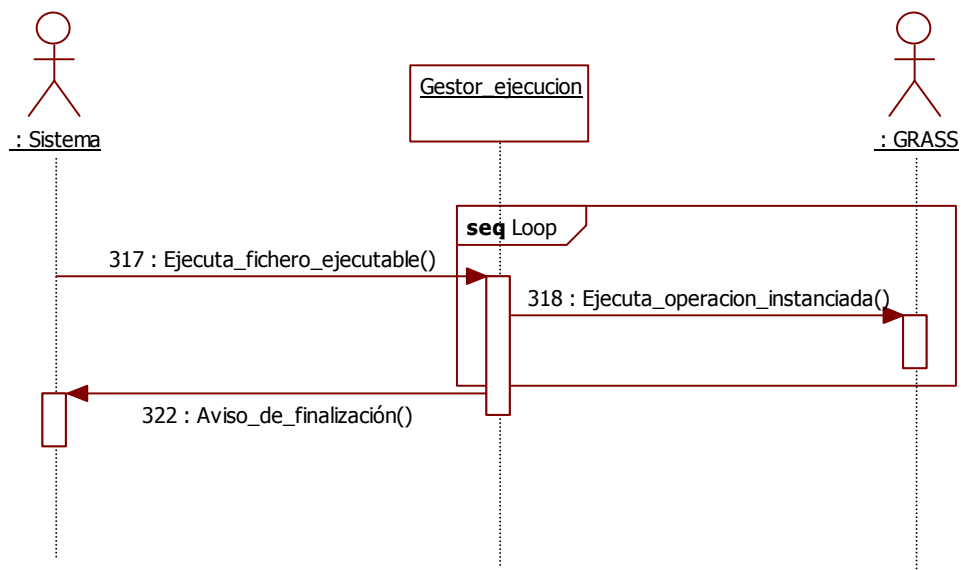


Figura 120: Diagrama de secuencia de diseño del CU Ejecución de script

4.2.2.4.24. CU-24: Modificación estado escenario

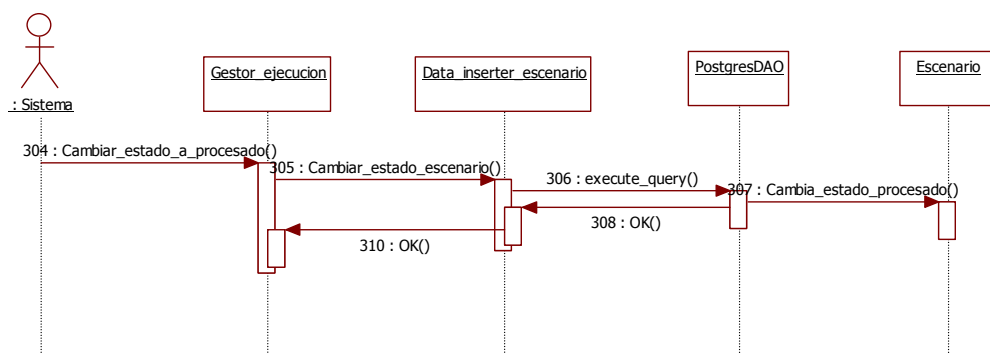


Figura 121: Diagrama de secuencia de diseño del CU Modificación estado escenario

4.2.2.4.25. CU-25: Mostrar información resultados intermedios y finales

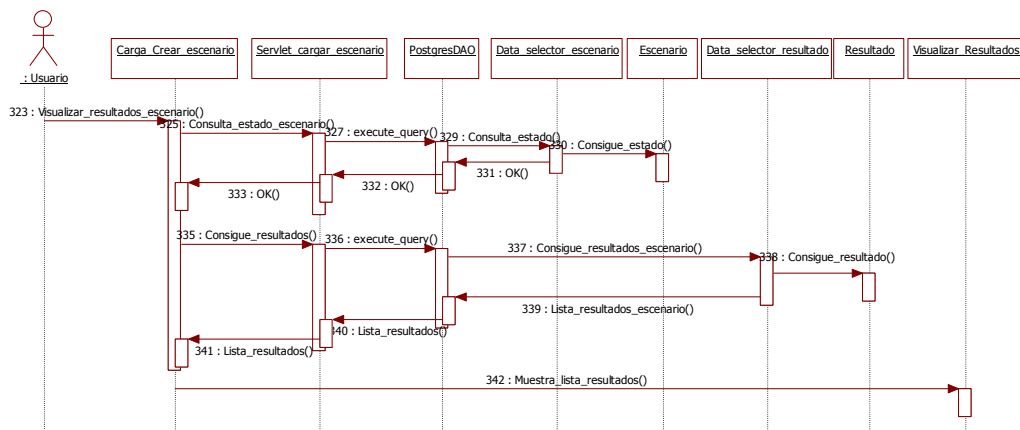


Figura 122: Diagrama de secuencia de diseño del CU Mostrar información resultados intermedios y finales

4.2.2.4.26. CU-26: Visualizar resultados

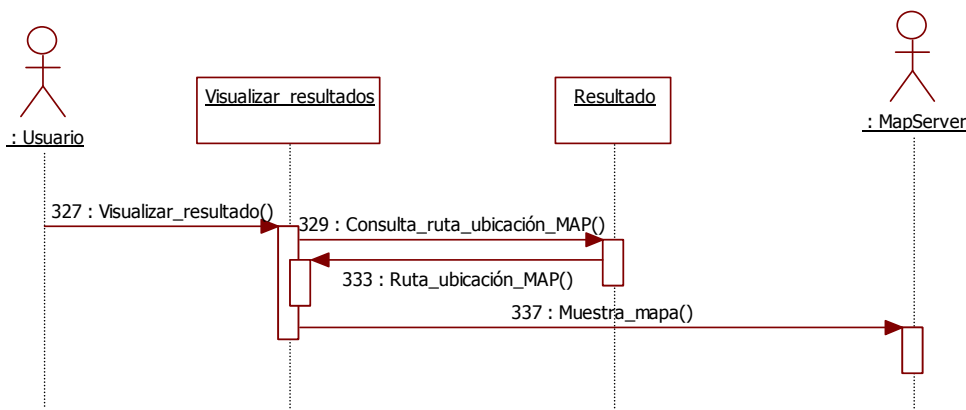


Figura 123: Diagrama de secuencia de diseño del CU Visualizar resultados

4.3. Tecnologías más relevantes

4.3.1. Visión global y tecnología

Se puede dividir el procesamiento de la aplicación DSS en tres fases: la primera fase corresponde con la **configuración del escenario (parte web)**, la segunda fase corresponde con la **parte operacional del sistema (parte operacional)** y la tercera fase sería finalmente la **visualización**

de los resultados (parte web). Cada una de estas fases precisa de un seguimiento secuencial y requiere la finalización de su antecedente.

Para la parte web se implementa con tecnología **JSP**⁴, **jQuery**⁵ y propiamente JavaScript. Por tanto, tiene componentes que se ejecutan en el servidor y otros en el cliente, en este caso sobre el navegador web.

Para la parte operacional (creación de scripts operacionales y ejecución de operaciones SIG de forma automática) se realiza de forma independiente en la parte del servidor. Se ha implementado en **Java** y se utiliza el concepto de **Entity EJBs** para las entidades ubicadas en la base de datos que utiliza la aplicación.

El objetivo de una entity EJBs es encapsular los objetos del lado del servidor que almacenan los datos y mantener la persistencia de los datos correspondientes en la base de datos. Nosotros, sólo utilizaremos el concepto de entitites EJBs para el uso de las entidades almacenadas en la base de datos y contenedoras de dichos atributos, a través del API que ofrece Java.

Además, se ha implementado un scheduler como planificador de procesos para la creación automatizada de scripts operacionales utilizando la librería *Quartz*. Y se ha hecho uso de librerías externas de *ApacheCommons*.

Para la ejecución de operaciones SIG, nos apoyamos en el SIG **GRASS**. Por tanto, las operaciones SIG que procesará el sistema se implementarán con la sintaxis de los comandos de GRASS y posteriormente se instanciarán con los parámetros configurados en el escenario, pudiéndose ejecutar automáticamente en el lado del servidor.

En la parte de visualización de resultados se usará **OpenLayers**⁶ y **MapServer**⁷ para poder visualizar, consultar y analizar la información geográfica a través de la red mediante la tecnología IMS⁸. El sistema tendrá que crear un fichero .MAP de MapServer para cada mapa que se quiera visualizar, y OpenLayers lo mostrará en la Web.

⁴**JSP (JavaServer Pages)** es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

⁵**jQuery** es una biblioteca de JavaScript de software libre y de código abierto, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

⁶**OpenLayers** es una biblioteca de JavaScript de código abierto para mostrar mapas interactivos en los navegadores web.

⁷**MapServer** es un entorno de desarrollo en código abierto para la creación de aplicaciones SIG en internet. Su funcionamiento básico está configurado en un fichero de texto, que tiene la extensión .MAP.

⁸**IMS (Internet Map Server o servidor de cartografía digital)** provee cartografía a través de la red tanto en modo vectorial como con imágenes.

A pesar de ser una aplicación web y que ciertas funcionalidades se ejecuten en el cliente, el peso de esta aplicación recae en funcionalidades en el lado del servidor como son: la creación y ejecución de scripts operacionales, donde su productividad y tiempos de respuesta se verán afectados sustancialmente por el rendimiento del servidor.

A continuación se muestra el *Esquema de la arquitectura DSS*, donde se expone de forma gráfica los cometidos en el lado del cliente y del servidor.

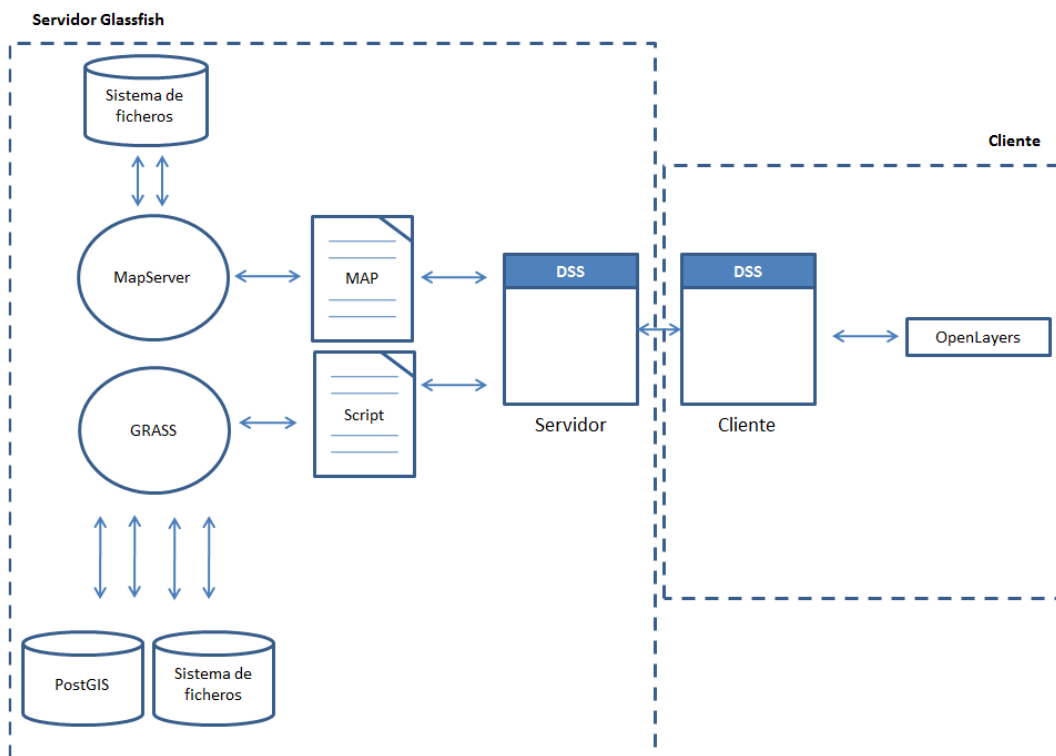


Figura 124: Esquema de la arquitectura del DSS

4.3.1.1. Partes del sistema

Configuración de escenario

El entorno de realización será una aplicación web escrito en lenguaje HTML, JavaScript, jQuery y CSS que muestre una interfaz de usuario para definir el escenario objeto de análisis. La página de creación de un escenario debe contener una serie de formularios para definir el escenario: la zona geográfica donde se realizará el estudio, el dispositivo (UNIGEO) para el cual se va a aplicar y algunos parámetros propios del escenario.

Una vez se haya creado el escenario, se podrá configurar los criterios disponibles según el tipo de dispositivo seleccionado y también los parámetros asociados a cada uno de los criterios y subcriterios.

Algunos criterios tienen ligados parámetros del dispositivo elegido, puede no tener parámetros o puede introducir el valor de un parámetro que previamente se cargará con un valor por defecto.

Parte operacional del sistema

Será un proceso off-line. Una vez el usuario haya dado la orden de ejecutar el escenario, se pondrá en cola de ejecución para ser procesado.

Se hará uso de scripts que serán creados a través de Java en el lado del servidor para la ejecución de operaciones SIG de forma automática. Dichos scripts realizan las operaciones de cada criterio y subcriterio instanciando para ese escenario apoyándose en el SIG GRASS.

Para el almacenaje de capas de entrada se hará uso de PostGIS un módulo que añade soporte espacial para la base de datos PostgreSQL. En el caso de las capas raster se almacenarán en el sistema de ficheros en el servidor.

Visualización de resultados

Se ha instalado MapServer sobre un servidor Glassfish para la publicación de mapas (capas) que nos proporcionará un acceso a las capas raster de resultados y capas vectoriales almacenadas en PostGIS para su visualización a través de la red por medio del servicio WMS.

Una vez GRASS realice los cálculos, los resultados serán publicados a través de MapServer creando un fichero .MAP correspondiente para dar servicio WMS a dichos mapas. Procesado el escenario, se cambiará el estado a 'procesado', momento en el cual el usuario podrá consultar la información asociada a los resultados y visualizar los mapas de resultados intermedios y final.

Para la visualización de resultados se utilizará una aplicación web con código HTML, JavaScript, jQuery y CSS. Se ha hecho uso de OpenLayers que es una biblioteca de JavaScript de código abierto para mostrar mapas interactivos en navegadores web. Será el encargado de acceder a los servicios WMS y obtener las capas resultado previamente publicadas.

4.3.1.2. Aplicación MVC

El patrón más adecuado para este tipo de aplicaciones es el "Modelo Vista Controlador". Es apropiado para **la parte Web** donde la **vista** es la página HTML y el código provee de datos dinámicos a la página. El **modelo** es el Sistema de gestión de base de datos y la lógica de negocio, y el **controlador** es el responsable de recibir eventos de entrada desde la vista.

Características principales

Este patrón separa el modelado del dominio, la presentación y las acciones basadas en los datos introducidos por el usuario en tres bloques bien diferenciados:

- **Modelo:** administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar de estado (habitualmente desde el controlador).
- **Vista:** maneja la visualización de la información. Existen tantas como formas diferentes se pueda presentar la información al usuario.
- **Controlador:** gestiona las entradas del usuario, interpretando las acciones del ratón y el teclado e informando al modelo o a la vista para que cambien según resulte apropiado.

Las ventajas que nos presenta este patrón son:

- Soporte para múltiples vistas: esto se debe a que la vista se encuentra separada del modelo y no hay dependencia directa del modelo respecto de las vistas. La interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente, por ejemplo, múltiples páginas web pueden utilizar el mismo modelo de objetos mostrado de manera diferente.
- Si se cambia la información, los datos se actualizarán en todas las representaciones.
- Adaptación al cambio: los requisitos de interfaz tienden a cambiar con mayor rapidez que las reglas de negocio, como el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

El siguiente esquema representa este patrón:

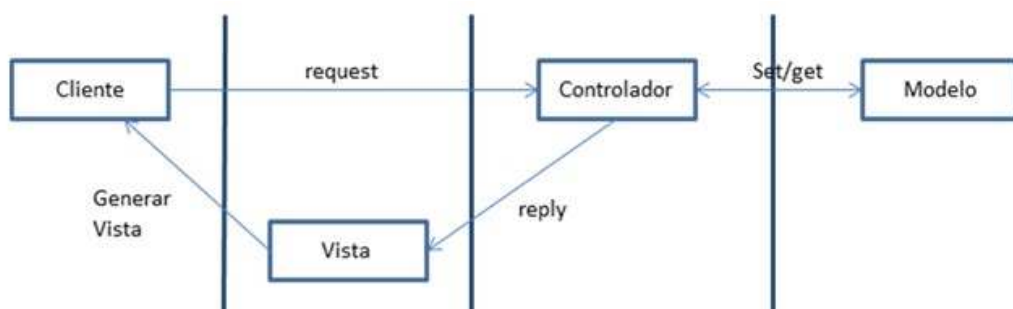


Figura 125: Esquema de aplicación del patrón Modelo Vista Controlador

En el caso que se nos presenta se ha aplicado este modelo, puesto que se ha separado:

- **Vista:** diferentes páginas web que muestran la información, tanto para la configuración del escenario como para la visualización de resultados
- **Controlador:** scripts JavaScript y JSP que procesan la información introducida por los usuarios y las acciones realizadas por éstos.

- **Modelo:** incumbe a todas las clases Java que se han identificado como entidades y servlets que dan servicio a lo pedido por el controlador

A continuación se muestra un esquema de lo explicado hasta ahora:

Configuración del escenario

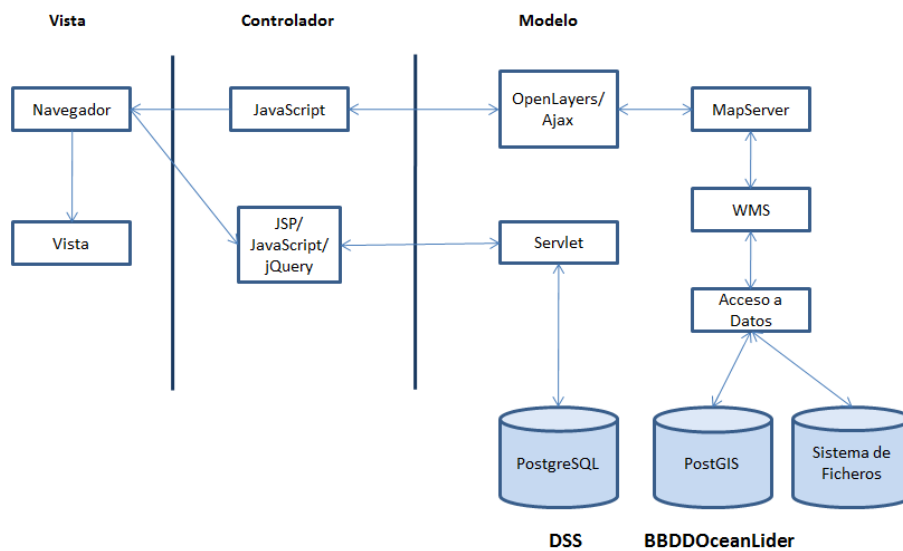


Figura 126: Esquema de aplicación MVC al DSS para la configuración de escenario

Visualización de resultados

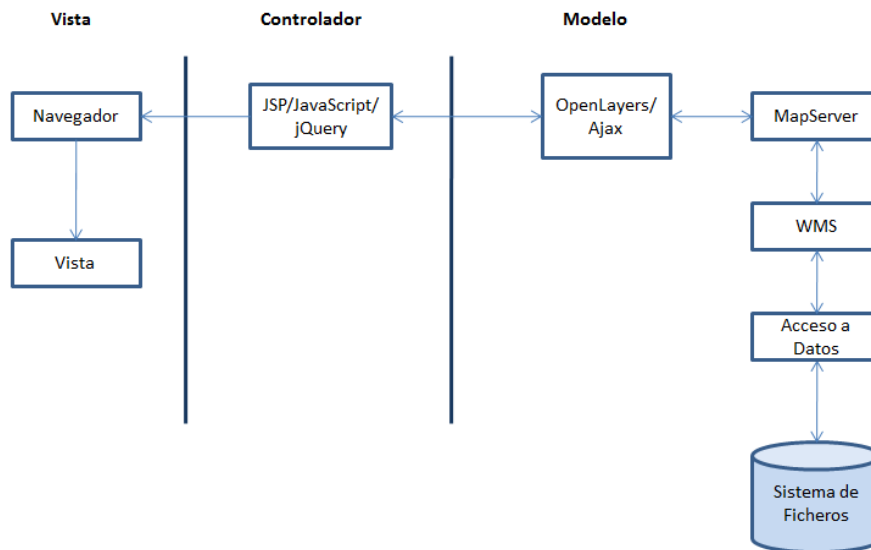


Figura 127: Esquema de aplicación MVC al DSS para la visualización de resultados

4.4. Interfaz del sistema

La aplicación aportará las interfaces adecuadas para la realización de los distintos pasos para la configuración del escenario y la visualización de resultados, así como el soporte para las partes que integran la operabilidad del sistema y el acceso a la base de datos.

4.4.1. Objetivos generales del Diseño

A continuación se muestran los principales objetivos que se persiguen en este documento:

- Adquirir una comprensión en profundidad de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con lenguajes de programación, componentes reutilizables, etc.
- Crear una entrada adecuada para la actividad de implementación.
- Descomponer las tareas de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo, teniendo en cuenta la posible concurrencia.
- Capturar las interfaces entre los sistemas lo antes posible.
- Ser capaces de visualizar y reflexionar sobre el diseño utilizando una notación común.
- Plantear el paso a la implementación como un refinamiento directo del diseño que rellena lo existente sin modificar la estructura.

4.4.2. Diseño de la Interfaz de Usuario

A continuación se describe la interfaz de usuario que permitirá a los diferentes usuarios del sistema realizar todos los escenarios que se han descrito en los apartados anteriores.

4.4.2.1. Bocetos

En este apartado se muestran los bocetos que describirán las diferentes interfaces mediante las cuales los usuarios interactuarán con el sistema recordando que se utilizará un navegador web como interfaz de usuario.

En el diseño de la aplicación se han utilizado los siguientes elementos de estructura:

- **Window:** área del interfaz de usuario en la que se muestran los objetos de presentación, es decir, una ventana de navegador en la que se muestran páginas web.
- **Formulario:** elemento de una página web cliente que permite solicitar información al usuario a través de una serie de campos de entrada (Radio Buttons, Text Boxes, etc.). Una vez introducidos los valores en estos campos serán almacenados y posteriormente procesados para la creación del script que hará uso de las funcionalidades de GRASS para la parte operacional: generación, almacenaje y publicación de resultados.
- **Contenedor:** recurso que permite dividir un navegador (window) en diversas secciones y presentar en cada sección una funcionalidad diferente.

Por otro lado están los elementos que permiten al usuario interactuar con el sistema:

- **Tabs:** son elementos de pestañas de una página web por los que el usuario puede acceder a completar los formularios presentados en cada una de ellas para definir el escenario. Se hace la clasificación del proceso en forma de pestañas para dar claridad al usuario. Se puede proceder a hacer un seguimiento secuencial de las pestañas para configurar el escenario (sería lo recomendado), aunque puede hacerse de forma alternativa siempre y cuando no se omita ningún paso.
- **Button:** tipo de campo de entrada de un formulario que indica que se realice una acción como enviar el formulario.
- **Text Boxes:** elemento de un formulario que permite introducir texto simple.
- **Radio Buttons:** bloque de botones dentro de un formulario que permiten una selección exclusiva entre varias opciones.
- **Check Boxes:** bloque de botones dentro de un formulario que permiten la selección múltiple entre varias opciones.

Estructura

A continuación se pasa a describir la estructura de la interfaz de usuario.

La estructura general de las páginas del sistema está compuesta por tres secciones contendoras:

- **Contenedor superior,** en el que se mostrará una cabecera.

- **Contenedor principal**, en el que se muestra el contenido principal de la funcionalidad de la página. Dentro de este contenedor aparecerá un menú de navegación que muestra lo que esté disponible en ese momento, según el paso o sección en el que se encuentra el usuario.
- **Contenedor inferior**, en el que se mostrará el pie de página.

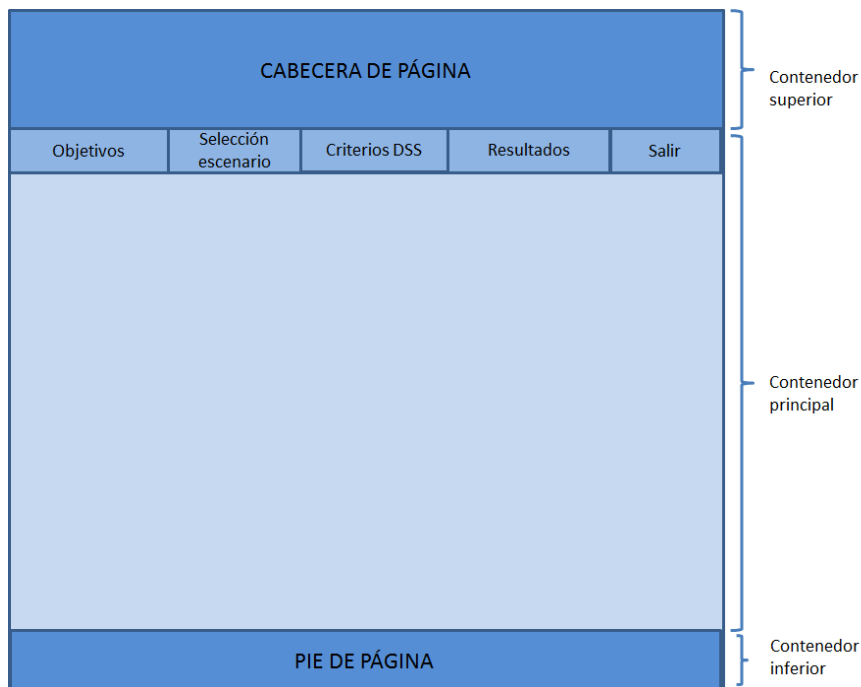


Figura 128: Vista de la interfaz DSS dividida en contenedores

a. Página principal

Es la página de acceso al sistema para que el usuario pueda logarse o registrarse.

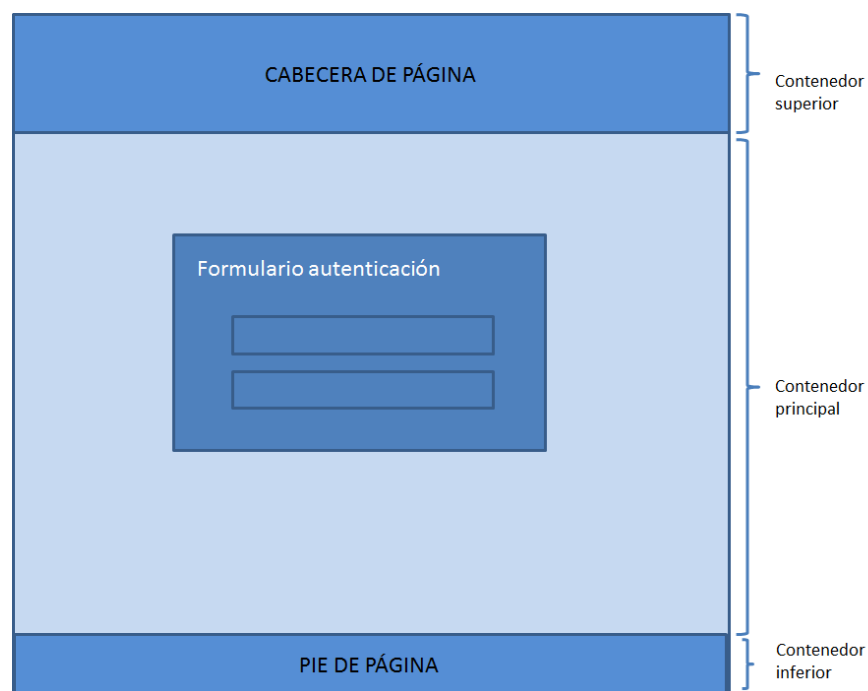


Figura 129: Diseño de la página de entrada de la interfaz DSS

b. Página de presentación de Objetivos

Es la primera página que se muestra al logarse el usuario. Es una página de bienvenida a modo presentación de la herramienta. Se muestra en unas breves líneas la descripción de su uso y un ejemplo de resultado final.

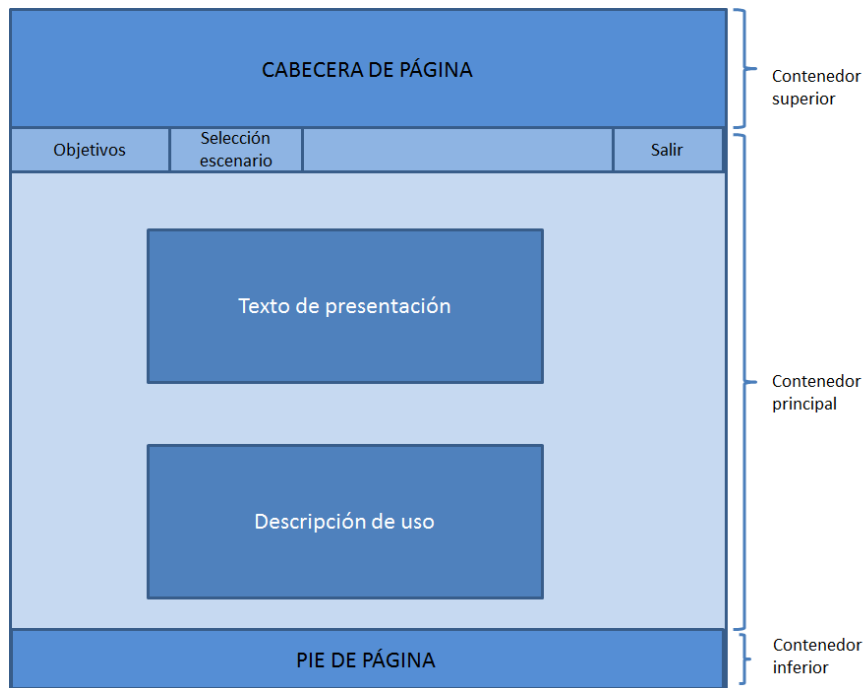


Figura 130: Diseño de la página de bienvenida de la interfaz DSS

c. Página de selección del escenario

En esta página de selección del escenario se presentan dos alternativas mostradas en pestañas según la acción a realizar: cargar un escenario creado o crear un escenario nuevo.

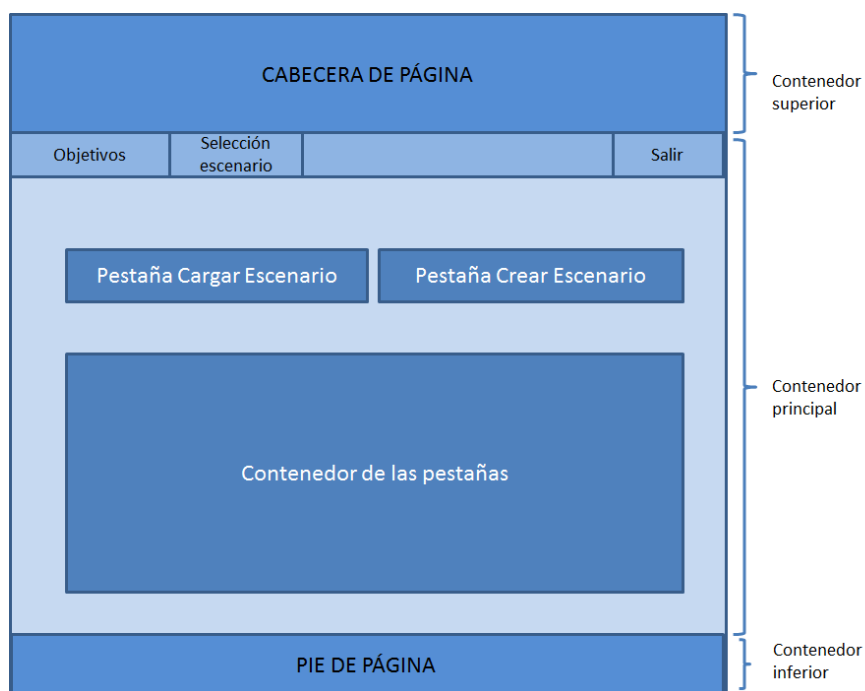


Figura 131: Diseño de la página para la selección del escenario de la interfaz DSS

d. Página de configuración de criterios

En esta página se muestra la lista de criterios disponibles para ese escenario una vez cargado o creado. El usuario podrá seleccionar y deseleccionar aquellos que considere adecuados y proporcionar el valor del parámetro que se requiera para cada criterio o subcriterio.

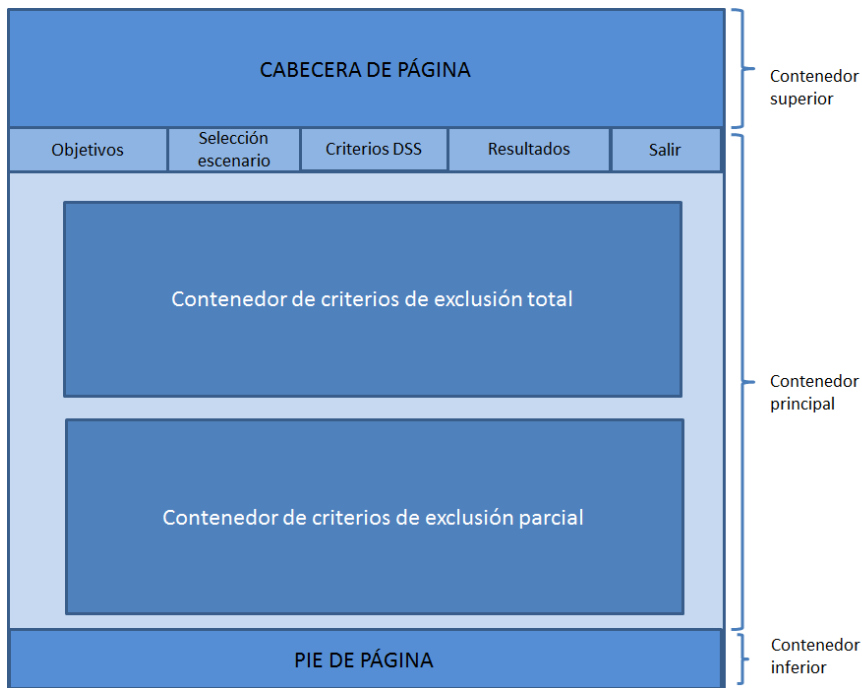


Figura 132: Diseño de la página para la configuración de criterios del escenario de la interfaz DSS

e. Página de visualización de resultados

Esta es la página de consulta de la información de los resultados intermedios y el resultado final una vez se haya procesado el escenario. A través de esta página tendrá acceso al visor del resultado raster.



Figura 133: Diseño de la página para la consulta de información de los resultados de un escenario de la interfaz DSS

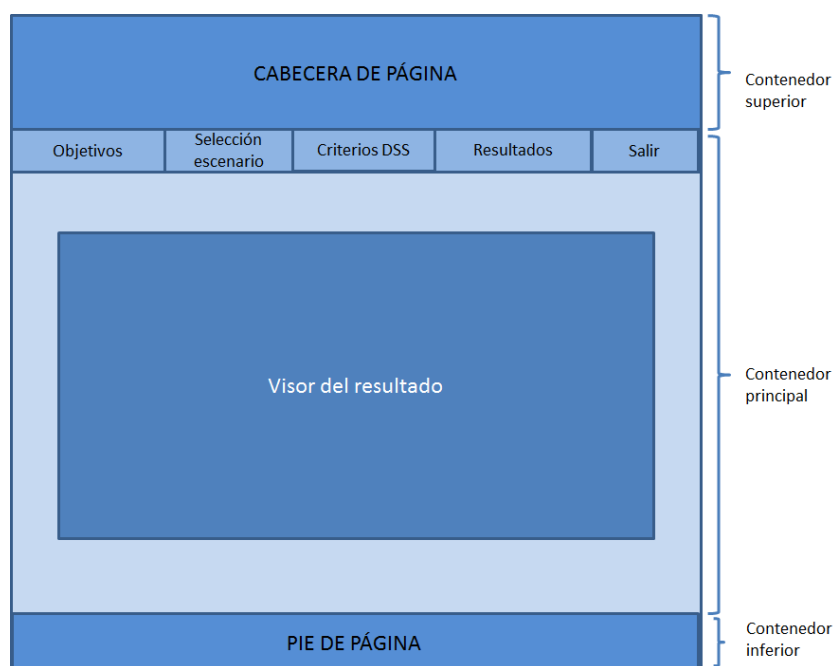


Figura 134: Diseño de la página para la visualización del resultado de un escenario de la interfaz DSS

4.5. Funcionalidades relevantes

4.5.1. Generación de scripts operacionales

Como se ha comentado en el apartado 4.3.1.1. *Partes del sistema*, la generación de scripts operacionales es un proceso off-line. Una vez el usuario haya dado la orden de ejecutar el escenario, se pondrá en cola de ejecución para ser procesado.

En el código de la herramienta DSS se ha implementado un **planificador de procesos** (scheduler) para la creación automatizada de scripts operacionales de los escenarios en cola de procesamiento y su posterior ejecución, utilizando la biblioteca *Quartz* [Quartz, 2013]. Quartz es una biblioteca de código abierto que da servicio a la planificación de procesos y puede ser integrada en aplicaciones Java.

El planificador está a la escucha continua y cada dos minutos consultará a la base de datos los escenarios en cola de ejecución. Si un escenario se está procesando el planificador pasará al siguiente ciclo a la espera de que haya terminado. En caso de que haya terminado el del anterior ciclo cogerá el siguiente escenario en cola de ejecución. Se da prioridad de paso a los escenarios que estén en la cola de ejecución que hayan sido creados con anterioridad.

En el proceso de ejecución de un escenario, la herramienta DSS extrae toda la información relacionada con el escenario, así como los criterios y subcriterios seleccionados y los parámetros relacionados con los mismos. Posteriormente procesa esta información y pasa por el **motor del protocolo de aplicación de los criterios** (reflejado en el apartado 2.6. *El protocolo de la aplicación de los criterios*).

En el motor del protocolo de aplicación de los criterios, se instancian las **operaciones de los criterios y subcriterios** pertenecientes al escenario. Estas operaciones son las fijadas por los expertos en la programación de la herramienta DSS para los criterios y subcriterios, y posteriormente se instancian acorde con los parámetros configurados en el escenario.

También en el motor a medida que se van procesando los ramales de los criterios físicos, operacionales, medioambientales y de recurso energético, se van instanciando otras **operaciones intermedias** para la combinación de resultados de criterios y subcriterios, reflejadas en la *figura 7 Empleo de las operaciones de los módulos del protocolo de aplicación de los criterios*. Y a continuación esta secuencia de operaciones se escribe en los scripts (ficheros .BAT) para ser ejecutados.

Una vez creado el .BAT, la herramienta DSS lanza a ejecución las operaciones que alberga para que sean ejecutadas en el **SIG GRASS** de forma secuencial. Al terminar la ejecución del script en su totalidad, la herramienta cambia el estado del escenario a procesado e importa los resultados al sistema de ficheros en el servidor para posteriormente poder ser visualizados.

4.5.1.1. Mecanismos de depuración en la ejecución de operaciones SIG

Se ha provisto de un mecanismo de depuración en la ejecución de operaciones SIG a través de código Java. Se crea un fichero adicional en cada escenario en el que se refleja la salida de las

operaciones ejecutadas en el SIG GRASS. De esta manera, si ha habido algún error en el procesamiento de estas operaciones se puede hacer un seguimiento de lo que ha sucedido y así detectar el punto en concreto donde ha ocurrido.

Se lanza un proceso para ejecutar el script, éste escribe la salida del comando de GRASS que se está ejecutando en el fichero de depuración. El código es el siguiente:

```
Process p=Runtime.getRuntime().exec ("cmd /c \" "+
nombreyrutafichero+ "\" 2>
"+bd.DatosBBDD.pathResultadosRaster+escenario.getId()+ "\\Resulta
dos_grass.txt" );
```

4.5.1.2. Ejemplo de script operacional

Se va a describir un ejemplo de script operacional generado por la herramienta DSS. A continuación se muestra para el criterio físico de profundidad.

- **Profundidad.** De exclusión parcial. Se aplica una operación difusa sobre la profundidad extrayendo los parámetros del dispositivo (parámetros: Profundidad mínima, ProfMin, y profundidad máxima, ProfMax). Las medidas se aplican en metros. Se consideran los siguientes valores que procesa internamente:

valor A:= ProfMin-5m, valor B:= ProfMin, valor C:= ProfMax, valor D:= ProfMax+5m

El dispositivo configurado en el escenario con las características que le definen para aplicar el criterio es el que se muestra a continuación:

Características del dispositivo

Nombre:

Tipo de dispositivo:

Tipo de captador:

Altura de buje: metros

Diámetro rotor: metros

Profundidad máxima: metros

Profundidad mínima: metros

Altura de ola máxima admitida: metros

Velocidad de la corriente máxima admitida: m/s

Potencia unitaria: vatios

Separación entre dispositivos de una misma línea: kilómetros

Separación entre líneas de dispositivos: kilómetros

Se muestra el script operacional con comentarios añadidos para entender las operaciones realizadas en el criterio de profundidad.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */  
  
//Configuración de la localización y Mapset del proyecto en el entorno  
de trabajo de GRASS  
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84  
//Configuración de la región que engloba a Península, Canarias y  
Baleares  
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224  
cols=3453  
  
/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */  
  
//Importación en GRASS de una capa raster ubicada en el sistema de  
ficheros
```

```

r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_281
//Operación difusa
r.mapcalc "R_282 = if(isnull( R_281 ) , null(), if((R_281 <= -55.0 ||
R_281 >= -10.0), 0.0, if((R_281 > -55.0 && R_281 < -50.0), eval((R_281 -
-55.0)/(-50.0 --55.0 )), if((R_281 > -15.0 && R_281 < -10.0 ), eval((-
10.0 - R_281 )/(-10.0 - -15.0)), 1.0)))"

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_283 = if(isnull(ccaa),R_282, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.716053259111 s=35.243885290361 e=-3.939985078125 w=-
7.345746796875 nsres=300 ewres=300 cols=3224 rows=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_282
output="C:\Users\GSI\OceanLider\Resultados\3\RI_R_282.tif"
r.out.gdal input=R_283
output="C:\Users\GSI\OceanLider\Resultados\3\RF_R_283.tif"

```

Se puede visualizar los resultados en el apartado 5.1.1.1.

4.5.2. Visualización de mapas en la web

4.5.2.1. Introducción

Para la visualización de mapas en la herramienta DSS usamos **MapServer** y **OpenLayers**. El cometido de su uso es para:

- La visualización de capas vectoriales alojadas en PostGIS. Son los mapas de referencia para la selección de la región geográfica del escenario (en este caso se muestra la línea de costa).
- La visualización de las capas raster alojadas en el sistema de ficheros. Son los mapas de los resultados intermedios y finales de los escenarios

MapServer es un motor de procesamiento de datos geográficos de código abierto escrito en C. Proporciona navegación de datos SIG tanto para capas raster y vectoriales. Actualmente MapServer es un proyecto de OSGeo⁹ y es mantenido por desarrolladores de todo el mundo.

Características de MapServer:

- **Salidas cartográficas avanzadas:** Ejecución de la aplicación y dibujo de elementos según la escala, etiquetado de elementos, salidas basadas en plantillas personalizables, automatización de los elementos del mapa (barra de escala, mapa de referencia y leyenda), mapas temáticos usando clases basadas en expresiones lógicas o expresiones regulares.
- **Soporte a los lenguajes de scripting y ambientes de desarrollo más populares:** PHP, Python, Perl, Ruby, Java y .NET.
- **Soporte multi-plataforma:** Linux, Windows, Mac OS X, Solaris, etc.
- **Soporte a estándares del Open Geospatial Consortium (OGC¹⁰):** WMS (cliente/servidor), WFS no-transaccional (cliente/servidor), WMC, WCS, etc.
- **Múltiples formatos de datos vector y raster por medio de GDAL¹¹ y OGR¹²:** TIFF/GeoTIFF, EPPL1, archivos Shapefile de ESRI, PostGIS, ESRI ArcSDE, Oracle Spatial, MySQL, etc.
- **Soporte de proyecciones cartográficas:** proyección de mapas al vuelo con diferentes proyecciones disponibles a través de la librería *Proj4*.

Para la interfaz de visualización de mapas, MapServer implementa tres estándares OGR que son **WMS** (Web Map Service), **WFS** (Web Feature Service) y **WCS** (Web Coverage Service). Estos estándares permiten cambiar de servidor sin quedarse atado a una sola tecnología.

WMS es un formato estándar para hacer requests de imágenes a servidores de mapas. Los parámetros del request se encuentran estandarizados por OGC y determinan la proyección que se quiere, los límites de la imagen en grados o metros, el tamaño de la imagen e incluso el layer que se quiere mostrar. Además el resultado puede ser una imagen jpeg, png, gif e incluso algún formato vectorizado.

Otro protocolo que soporta MapServer es WFS. El mismo se encuentra especificado por OGC y pretende transmitir información geográfica sobre un mapa. El protocolo suele ser SOAP, aunque también permite agregar los parámetros en la url. El resultado que devuelve este servicio se encuentra en formato GML (Geography Markup Language) que representa información geográfica en un XML.

Por último, el tercer protocolo estándar que soporta MapServer es WCF. Este protocolo intenta dar información geográfica de zonas. En el request SOAP o en los parámetros, se solicita

⁹**OSGeo** es una comunidad para del desarrollo colaborativo de software geoespacial de código abierto y promover su uso generalizado. <http://www.osgeo.org/>

¹⁰**OGC** (Open Geospatial Consortium es una organización de estándares en el dominio de los SIG.

¹¹**GDAL** (Geospatial Data Abstraction Library) es una librería de escritura y lectura de datos raster. SE utiliza como mecanismo principal para la lectura de datos en MapServer. <http://www.gdal.org/>

¹²**OGR** es una librería escrita en C++ para el acceso a datos vectoriales. Proporciona acceso a múltiples formatos de datos y actualmente forma parte de GDAL.

cobertura sobre que se quiere y el protocolo devolverá información sobre esa zona en formato GML.

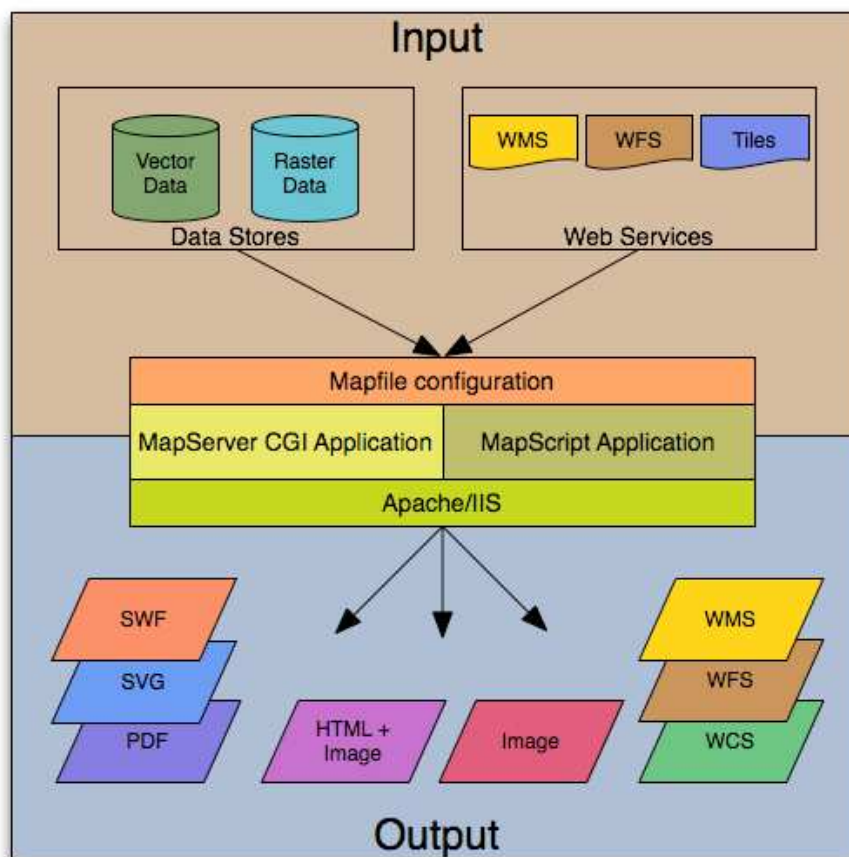


Figura 135: Diagrama de la arquitectura de MapServer [Hexacta, 2013]

OpenLayers es una biblioteca de JavaScript de código abierto para mostrar mapas interactivos en los navegadores web. OpenLayers ofrece un API para acceder a diferentes fuentes de información cartográfica en la red: WMS, WFS, mapas comerciales (tipo Google maps, Bing, Yahoo), distintos formatos vectoriales, etc. Actualmente forma parte de los proyectos de OSGeo y el desarrollo y soporte es a través de los colaboradores.

Nosotros haremos uso de **MapServer** como motor de procesamiento de los mapas y servidor de mapas, y **OpenLayers** como interfaz de visualización de mapas en el navegador utilizando el servicio WMS.

4.5.2.2. Generación .MAP para mapas raster y vectoriales

El **MapFile (.MAP)** es el archivo que utiliza MapServer para configurar los layers (capas) del mapa. Un mapa puede estar compuesto por varios layers (capas) de diferente procedencia y/o tipo. MapServer requiere el archivo .MAP que se va utilizar en cada petición WMS para visualizar el mapa con los diferentes layers (capas).

En este archivo se configura la forma en que MapServer se conectará a la base de datos (en nuestro caso PostGIS) o sistema de ficheros en el servidor. En cada layer se debe indicar las propiedades gráficas que tendrán los datos geométricos (nombre, tipo de geometría, estilo, etc). Estos .MAP pueden ser creados con programas de escritorio como QuantumGIS o GvSIG (únicamente en la versión 1.1), de forma manual con un editor de textos o con módulos implementados en Java para la creación automática.

En la herramienta DSS se ha implementado unos módulos para la creación de ficheros .MAP de mapas vectoriales y raster para que se generen cuando la herramienta lo requiera. Por simplificación en la herramienta DSS, únicamente hay un layer (capa) por fichero .MAP. Lo que se debe tener claro a la hora de crear un archivo de este tipo es la ubicación de los datos (por ejemplo, base de datos o sistema de ficheros) y el tipo de datos (raster o vectorial), entre otros.

Un .MAP está compuesto por varios apartados. No se considera adecuada la explicación detallada de todos los apartados de los que se compone un MapFile. Se puede consultar en la referencia [MapServer g, 2013].

Se muestra a continuación un ejemplo de fichero .MAP. En él se puede ver las partes principales de las que se compone teniendo variables generales, variables específicas de la interfaz web y variables de las capas que lo componen. Este ejemplo es un fichero de un .MAP de un mapa con una única capa raster.

```
MAP
  NAME "sample"
  STATUS ON
  SIZE 600 400
  SYMBOLSET "../etc/symbols.txt"
  EXTENT -180 -90 180 90
  UNITS DD
  SHAPEPATH "../data"
  IMAGECOLOR 255 255 255
  FONTSET "../etc/fonts.txt"

  #
  # Start of web interface definition
  #
  WEB
    IMAGEPATH "/ms4w/tmp/ms_tmp/"
    IMAGEURL "/ms_tmp/"
  END # WEB

  #
  # Start of layer definitions
  #
  LAYER
    NAME 'global-raster'
    TYPE RASTER
    STATUS DEFAULT
    DATA bluemarble.gif
  END # LAYER
END # MAP
```

Figura 136: Ejemplo de la estructura de un MapFile (.MAP) [MapServer d, 2013]

4.5.2.2.1. Creación .MAP para mapas vectoriales

En la herramienta DSS generamos ficheros .MAP para la visualización de capas vectoriales almacenadas en PostGIS. En la herramienta DSS necesitamos visualizar mapas de referencia para la selección de la región geográfica del escenario (por ejemplo, la línea de costa). Los mapas de referencia están almacenados en la base de datos *BBDDOceanLider* estando en la proyección con código EPSG 4326.

La creación de este fichero para las capas vectoriales ubicadas en PostGIS es de forma similar que para las capas vectoriales que están ubicados en el sistema de ficheros.

Un ejemplo de layer de un MapFile para una capa vectorial almacenada en una carpeta del sistema de ficheros:

```
LAYER
  NAME "world_poly"
  DATA 'shapefile/countries_area.shp'
  STATUS ON
  TYPE POLYGON
  CLASS
    NAME 'The World'
    STYLE
      OUTLINECOLOR 0 0 0
    END # STYLE
  END # CLASS
END # LAYER
```

Figura 137: Parte Layer de un MapFile (.MAP) para definir una capa vectorial ubicada en el sistema de ficheros [MapServer d, 2013]

Para definir una capa vectorial que está almacenada en PostGIS, debemos configurar los parámetros CONNECTIONTYPE, CONNECTION y DATA dentro de la etiqueta LAYER. Un ejemplo de layer de un MapFile para una capa vectorial ubicada en PostGIS:

```
LAYER
  NAME "ccaa"
  STATUS ON
  TYPE polygon

  DUMP false
  CONNECTIONTYPE POSTGIS
  CONNECTION "host=localhost port=5432 user=postgres password=968XCdPh dbname=BBDDOceanLider"
  DATA "geom from ccaa using unique gid using srid=4326"

  CLASS
  STYLE
  COLOR 135 125 73
  OUTLINECOLOR 118 54 94
  WIDTH 4
  END
  END

END #layer
```

Figura 138: Parte Layer de un MapFile (.MAP) para definir una capa vectorial ubicada en PostGIS

En CONNECTIONTYPE se indica el tipo de conexión, en el parámetro CONNECTION se especifica la información para la conexión a la base de datos y en el parámetro DATA se determina la columna geométrica de la tabla. Para más información consultar [MapServer e, 2013].

4.5.2.2.2. Creación .MAP para mapas raster

En la herramienta DSS generamos ficheros .MAP para la visualización de capas raster almacenadas en el sistema de ficheros en el servidor. Necesitamos visualizar los mapas de los

resultados intermedios y finales de los escenarios ejecutados que estarán alojados en el servidor con el código de proyección EPSG 4326.

La creación de este fichero con capas raster es similar al que se crea para capas vectoriales ubicados en el sistema de ficheros. Se tiene que configurar expresamente el parámetro TYPE indicando que es de tipo raster.

Un ejemplo de layer de un MapFile para una capa raster:

```
LAYER
  NAME "JacksonvilleNC_CIB"
  DATA "Jacksonville.tif"
  TYPE RASTER
  STATUS ON
END
```

Figura 139: Parte Layer de un MapFile (.MAP) para definir una capa raster ubicada en el sistema de ficheros

A través del .MAP podemos hacer una clasificación de los valores del raster. Cada clasificación le podemos asignar un estilo concreto, lo que nos permite definir una paleta de colores a los resultados de los escenarios.


```
LAYER
NAME "JacksonvilleNC_CIB"
DATA "Jacksonville.tif"
TYPE RASTER
STATUS ON
CLASSITEM "[pixel]"
# class using simple string comparison, equivalent to ([pixel] = 0)
CLASS
  EXPRESSION "0"
  STYLE
  COLOR 0 0 0
  END
END
# class using an EXPRESSION using only [pixel].
CLASS
  EXPRESSION ([pixel] >= 64 AND [pixel] < 128)
  STYLE
  COLOR 255 0 0
  END
END
# class using the red/green/blue values from the palette
CLASS
  NAME "near white"
  EXPRESSION ([red] > 200 AND [green] > 200 AND [blue] > 200)
  STYLE
  COLOR 0 255 0
  END
END
# Class using a regular expression to capture only pixel values ending in 1
CLASS
  EXPRESSION /*1/
  STYLE
  COLOR 0 0 255
  END
END
END
```

Figura 140: Parte Layer de un MapFile (.MAP) para definir una capa raster con reclasificación de valores

Las clases (CLASS) son evaluadas en orden de primero a último, y se queda con la primera coincidencia.

A continuación se muestra la parte LAYER de un MapFile sobre un resultado final de un escenario. En este ejemplo se puede apreciar la paleta de colores 2 (figura 188) utilizada para la visualización y la clasificación de los valores del raster:

```
LAYER
NAME R_349
TYPE RASTER
STATUS default
DATA 'C:\Users\GSI\OceanLider\Resultados\11\RF_R_349.tif'
CLASS
  EXPRESSION ([pixel] = -1.0)
  STYLE
```

```
        COLOR 204 102 0
    END
END
CLASS
    EXPRESSION ([pixel] = 0.0)
    STYLE
        COLOR 204 0 0
    END
END
CLASS
    EXPRESSION ([pixel] > 0.0 and [pixel]< 0.1)
    STYLE
        COLOR 204 255 102
    END
END
CLASS
    EXPRESSION ([pixel] >= 0.1 and [pixel]< 0.2)
    STYLE
        COLOR 204 255 0
    END
END
CLASS
    EXPRESSION ([pixel] >= 0.2 and [pixel]< 0.3)
    STYLE
        COLOR 153 255 51
    END
END
CLASS
    EXPRESSION ([pixel] >= 0.3 and [pixel]< 0.4)
    STYLE
        COLOR 102 255 0
    END
END
CLASS
    EXPRESSION ([pixel] >= 0.4 and [pixel]< 0.5)
    STYLE
        COLOR 0 204 0
    END
END
CLASS
    EXPRESSION ([pixel] >= 0.5 and [pixel]< 0.6)
    STYLE
        COLOR 102 204 0
    END
END
CLASS
    EXPRESSION ([pixel] >= 0.6 and [pixel]< 0.7)
    STYLE
        COLOR 102 153 0
    END
END
CLASS
    EXPRESSION ([pixel] >= 0.7 and [pixel]< 0.8)
    STYLE
        COLOR 51 102 0
    END
END
```

```
        END
    END
    CLASS
        EXPRESSION ([pixel] >= 0.8 and [pixel]< 0.9)
        STYLE
            COLOR 0 102 0
        END
    END
    CLASS
        EXPRESSION ([pixel] >= 0.9 and [pixel]< 1.0)
        STYLE
            COLOR 0 51 0
        END
    END
    CLASS
        EXPRESSION ([pixel] = 1.0)
        STYLE
            COLOR 23 34 0
        END
    END
    CLASS
        EXPRESSION ([pixel] < 0.0 and [pixel]> 1.0)
        STYLE
            COLOR 255 255 255
        END
    END
END
```

Para mayor información consultar la referencia [MapServer a, 2013].

4.5.2.2.3. Mecanismos de debug en MapServer

Se puede activar el mecanismo de depuración de MapServer con el parámetro DEBUG y enviar la salida bien a un archivo de salida, salida estándar de errores o salida estándar. En la herramienta DSS se ha hecho uso de este mecanismo para hacer control de los errores ocurridos en MapServer cuando se precisa visualizar los mapas.

Para activar la depuración de nuestros MapFile, tenemos que indicar en la sección WEB las siguientes sentencias:

```
CONFIG "MS_ERRORFILE" "ms_error.txt"
DEBUG ON
```

Esto redirige la salida a un archivo ubicado en el mismo directorio que el MapFile y muestra la información correspondiente al nivel 1 de depuración (correspondiente a DEBUG ON ó DEBUG 1). El nivel 1 corresponde con la información sobre los errores y noticias.

Hay 5 niveles de depuración, cada nivel corresponde a un nivel de detalle en la depuración, cuanto mayor sea el nivel más alto será el detalle de la depuración. Si queremos asignar otro nivel tenemos que indicar la palabra DEBUG seguido del número del nivel. La información sobre los niveles se puede encontrar en la siguiente referencia [MapServer f, 2013]

4.5.2.2.4. Generación automática de .MAP en la herramienta DSS

En la clase *Utilities* de la componente *Entities* de la herramienta DSS (Ver apartado 4.2.2.3.2. *Componente entities*), se encuentran los diferentes métodos para la creación de los ficheros .MAP para la visualización de capas raster y vectoriales.

Como se ha mencionado anteriormente, por simplificación en la herramienta DSS, únicamente hay un apartado LAYER (capa) en cada fichero .MAP, lo que significa que se creará un fichero .MAP por cada capa que se quiera visualizar.

El método de creación del fichero .MAP de los resultados intermedios y finales de un escenario es:

```
createMAPResultado(int idresultado, String type, Double norte,
Double sur, Double este, Double oeste): String
```

Este método genera un fichero .MAP pasándole como argumento el identificador del resultado, el tipo (en este caso será raster, pero se ha ampliado la función en caso que los resultados fueran vectoriales), y las coordenadas en las que se va a mostrar el mapa (si es en la *vista global* se le indicará las coordenadas que incumbe a la Península, Baleares y Canarias, y si es en *vista concreta* se le pasará las coordenadas del escenario). Devuelve la ruta donde se ha creado el fichero.

El método de creación del fichero .MAP de las capas vectoriales de referencia que se utilizan en la herramienta DSS alojadas en PostGIS es:

```
generateMapaUnionVectorial(String vect, String tipo): String
```

Este método genera un fichero .MAP pasándole como argumento el nombre de la capa vectorial alojado en PostGIS y el tipo (la capa vectorial puede ser de tipo POLYGON, LINE o POINT). Devuelve la ruta donde se ha creado el fichero.

4.5.2.2.5. Problemática de visualización de mapas con MapServer y PostGIS 2.0

En la herramienta DSS hemos utilizado la versión 6.1 de MapServer que viene integrado en el paquete MS4W 3.0.4. Esta versión de MapServer utiliza las funciones geométricas de PostGIS anteriores a la versión 2.0 donde el nombre de las funciones vienen precedidas del prefijo “ST_”. Por tanto, es necesario crear la función sin prefijo en PostGIS para que MapServer funcione correctamente.

El error encontrado viene referente a la función `geomfromtext(text, integer)` de PostGIS.

Para solventarlo abrimos la consola de sentencias SQL dentro de pgAdminIII y escribimos:

```
CREATE OR REPLACE FUNCTION geomfromtext(text, integer)
```

```
RETURNS geometry AS
$BODY$
        SELECT st_geomfromtext($1, $2);
$BODY$
LANGUAGE SQL
COST 1;
ALTER FUNCTION geomfromtext(text, integer)
OWNER TO postgres;
```

4.5.2.3. Visualización utilizando la biblioteca OpenLayers

Para hacer uso de la biblioteca OpenLayers pondremos en la cabecera de los JSP de la visualización de los resultados en vista global y concreta el siguiente código:

```
<script type="text/javascript"
src="http://dev.openlayers.org/releases/OpenLayers-
2.12/OpenLayers.js"></script>
```

Esto hará que se carguen todas las funciones de la biblioteca de OpenLayers de forma online. Se puede consultar ejemplos y las funciones de la biblioteca en [OpenLayers, 2013].

Lo primero que hace la página al cargarse es llamar a la función **init()** que está en la parte de script de los JSP. A continuación se pondrá unos ejemplos de su funcionamiento e implementación para la visualización de los resultados, que será diferente para la vista global y concreta.

Para la **visualización de los resultados en vista global** el código es el siguiente:

```
function init() {
    map = new OpenLayers.Map('map',
        new OpenLayers.Bounds(<%=oeste%>, <%=sur%>, <%=este%>, <%=norte%>),
    maxResolution: 'auto',
        units: "m",
        projection: "EPSG:4326"
    });

    layerlineacosta=new OpenLayers.Layer.WMS("Linea de Costa",
        "cgi-bin/mapserv.exe?mode=map&map=<%=rutamapabase%>"
        , { layers: "lin_costa", format: "image/png" }
        , { singleTile: true
        , isBaseLayer: false
        , transparent: true
        , ratio: 1
        , transitionEffect: 'resize'
        , displayOutsideMaxExtent: true
        });

    layer = new OpenLayers.Layer.WMS("<%= nombreFicheroResultado
%>",
        "cgi-bin/mapserv.exe?mode=map&map=<%= rutamap %>"
```

```

    , { layers: "<%= nombreFicheroResultado %>", format:
"image/png" }
    , { singleTile: true
    , isBaseLayer: true
    , transparent: true
    , ratio: 1
    , transitionEffect: 'resize'
    });

    map.addLayers([layerlineacosta,layer]);
    map.zoomToMaxExtent();

    //Control de Pestañas
    map.addControl(new OpenLayers.Control.LayerSwitcher());
    //Control de posición de ratón
    map.addControl(new OpenLayers.Control.MousePosition());
}

```

Donde las variables *norte*, *sur*, *este* y *oeste* son las coordenadas del mapa en la proyección 4326 que incumbe a las tres entidades Península, Baleares y Canarias; *rutamabase* es la ruta al mapa de la línea de costa; *rutamap* es la ruta al .MAP del resultado y *nombreFicheroResultado* es el nombre del resultado.

Utilizamos el servicio WMS pasando la ruta al fichero .MAP de cada capa que queremos mostrar y ponemos como mapa base a la línea de costa. Posteriormente se configuran algunos controles sobre la ventana de visualización de OpenLayers sobre la que se mostrarán los mapas.

Para la **visualización de los resultados en la vista concreta** se muestra únicamente la capa resultado que ha seleccionado el usuario con las coordenadas específicas del escenario. El código es parecido al mostrado anteriormente:

```

function init() {
    map = new OpenLayers.Map('map', {
        maxExtent: new
OpenLayers.Bounds(<%=oeste%>,<%=sur%>,<%=este%>,<%=norte%>),
        maxResolution: 'auto',
        units: "m",
        projection: "EPSG:4326"
    });
    layer = new OpenLayers.Layer.WMS("<%= nombreFichero %>",
    "cgi-bin/mapserv.exe?mode=map&map=<%= rutamap %>"
    , { layers: "<%= nombreFichero %>", format:
"image/png" }
    , { singleTile: true
    , ratio: 1
    , transitionEffect: 'resize'
    });

    map.addLayers([layer]);
    map.zoomToMaxExtent();
}

```

```
//Control de Pestañas  
map.addControl(new OpenLayers.Control.LayerSwitcher());  
//Control de posición de ratón  
map.addControl(new OpenLayers.Control.MousePosition());  
}
```

Donde en este caso los parámetros *norte*, *sur*, *este* y *oeste* apuntan a las coordenadas del escenario en la proyección 4326; *rutamap* es la ruta al .MAP del resultado y *nombreFicheroResultado* es el nombre del resultado.

4.6. Operaciones intermedias

4.6.1. Generación del mapa profundidad como MDE

El objetivo de este apartado es presentar la información para generar un mapa de profundidad a partir de un mapa de batimetría como un Modelo Digital de Elevaciones (MDE). Se van a mencionar las operaciones existentes en diferentes SIG para el cálculo de un MDE a partir de una capa de líneas de contorno. Para cada una de estas operaciones se describirán los pasos para obtener el MDE con los SIG: GvSIG y GRASS.

Se partirá de una capa vectorial de Batimetría de la Península que contiene líneas de contorno con valores (en metros) de profundidad: -500, -200, -100, -50, -30, -20, -10.



Figura 141: Imagen de la capa vectorial de Batimetría perteneciente a la Península

4.6.1.1. Cálculo de un MDE en GvSIG

Existen diferentes módulos para generar el MDE a partir de una capa con líneas de contorno en la biblioteca **Sextante** dentro de GvSIG:

- Rellenar celdas sin datos
- Rellena celdas sin datos por vecindad

Para estas dos operaciones es necesario hacer la rasterización de la capa vectorial de partida. Para ello, utilizamos la operación “*Rasterizar capa vectorial*” dentro del apartado “*Rasterización e interpolación*” de Sextante.

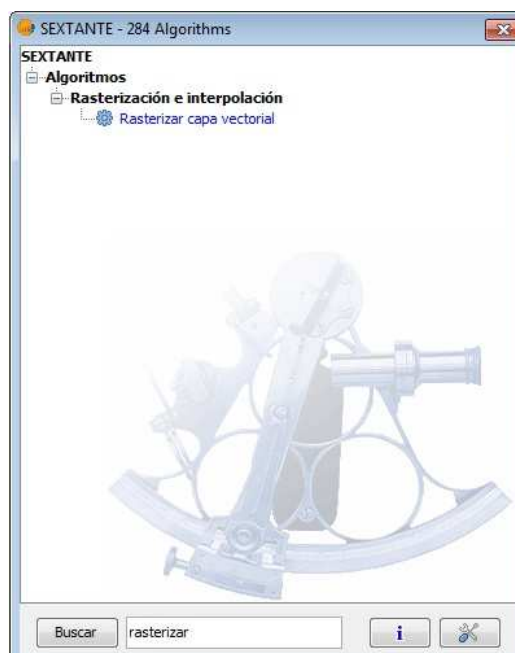


Figura 142: Ventana de dialogo de Sextante para la operación de rasterización

Realizamos la rasterización de la capa vectorial “Batimetría.shp” a través del campo “PROFUNDIDA” (para hacer la rasterización se debe indicar un campo numérico, en este caso el que contiene los valores de la profundidad). En la pestaña de “Salida Raster” configuramos adecuadamente el tamaño de celda a 250 m. A continuación se muestra la salida obtenida.



Figura 143: Resultado de la rasterización de la operación de rasterización de Sextante en GvSIG

4.6.1.1.1. Rellenar celdas sin datos

Para acceder a la operación “**Rellenar celdas sin datos**” debemos ubicarnos en el apartado “**Herramientas básicas para capas raster**” de Sextante dentro de GvSIG.

Esta operación completa la capa raster que contiene las líneas de contorno con su valor de profundidad a una capa raster con los valores intermedios. Este algoritmo rellena la información de una capa raster que presenta celdas sin datos. La información para completar las celdas sin datos se toma de la propia capa, mediante métodos de interpolación.

Según se indica en la ayuda de Sextante, la interpolación se realiza mediante splines (curvas adaptativas) con tensión. El parámetro de tensión controla el comportamiento de la superficie. Haciendo un símil físico, el proceso de interpolación es similar al ajuste de una lámina de un material dado sobre una serie de puntos fijos. La tensión establece las propiedades físicas del material con el que está construida esa lámina, tales como su rigidez, lo que condiciona la forma resultante.

Generamos el MDE indicando el valor de tensión que viene por defecto 0.1.



Figura 144: Resultado de la operación de rellenar celdas sin datos de Sextante a partir de Batimetría con tensión 0.1

4.6.1.1.2. Rellenar celdas sin datos (por vecindad)

Accedemos a la operación “**Rellenar celdas sin datos (por vecindad)**” dentro del apartado “**Herramientas básicas para capas raster**” de Sextante dentro de GvSIG.

Este algoritmo completa la información de una capa raster que presenta celdas sin datos. La información para completar las celdas sin datos se toma de la propia capa utilizando el método de interpolación por vecindad. Cada celda sin datos coge el valor de la celda con datos más cercana. Tan sólo tendremos que indicar la capa raster de batimetría como capa de entrada a la operación.



Figura 145: Resultado de la operación rellenar celdas sin datos por vecindad de Sextante a partir de la Batimetría

4.6.1.2. Calculo de un MDE en GRASS

Existen diferentes métodos dentro de GRASS que nos permiten hacer diferentes métodos de interpolación para generar el MDE. [OSGEO, 2012]

En un primer paso, debemos importar la capa de Batimetría en GRASS y configurar la región acorde a este mapa para que los posteriores cálculos se hagan dentro de esta región.

```
g.region n=489330.9 s=3886115.36 e=1051971.5 w=-35571.62 nres=250  
wres=250  
v.in.ogr dsn="E:/OceanLider/bd GMV/OCEAN LIDER_25062012/CAPAS  
INFORMATIVAS/01_IB/01_CB/Batimetría_P.shp" output=Batimetria_P
```

La mayoría de los módulos para la conversión a un MDE dentro de GRASS utiliza una capa raster como capa de entrada. Por tanto, es necesario hacer la rasterización de la capa vectorial de

Batimetría. Es importante mencionar que la resolución y el tamaño de las capas raster quedan determinadas por la resolución definida en la región.

```
v.to.rast in=Batimetria_P out=Batimetria_P type=line  
column=PROFUNDIDA
```

4.6.1.2.1. Módulo *r.surf.idw*

Realiza la interpolación ponderando la distancia inversa de los “n” puntos más cercanos. Se realiza el cálculo de la elevación de una celda con los puntos más próximos que tienen el mayor peso. El número de puntos cercanos que se extraen para hacer el cálculo se puede configurar, por defecto son 12.

Un aspecto a indicar es que el módulo IDW no trabaja adecuadamente con líneas de contorno, pero está bien como punto de partida.

```
(por defecto: 12 puntos)  
r.surf.idw input=Batimetria_P output=Prof_idw
```



Figura 146: Resultado de *Prof_idw* en escala de grises con Línea de Costa de referencia

(con 250 puntos)

```
r.surf.idw input=Batimetria_P output=Prof_idw.250 npoints=250
```



Figura 147: Resultado de Prof_idw.250 en escala de grises con Línea de Costa de referencia

4.6.1.2.2. Módulo *v.surf.bspline*

Para utilizar el método *v.surf.bspline* es necesario hacer la conversión de las líneas de contorno a puntos. Este método está preparado para realizar la interpolación de grandes conjuntos de puntos LIDAR.

Calcula la interpolación bilinear/bicubica con la regulación de Tykhonov [GRASS, 2012]. Destacar que sólo trabaja con valores positivos para la interpolación.

A continuación se describen los pasos para generar un MDE con valores positivos. En primer lugar pasamos los valores de la profundidad a valores positivos en la capa vectorial:

```
echo "UPDATE Batimetria_P SET PROFUNDIDA*(-1) WHERE PROFUNDIDA <  
0" | db.execute  
ó
```

```
v.db.update map= layer=1 Batimetria_P col=PROFUNDIDA qcol="(-1)*PROFUNDIDA"
```

Posteriormente generamos el mapa de puntos a partir de las líneas de contorno y hacemos la interpolación con el método bilinear y bicúbico:

```
v.to.points -v in=Batimetria_P out=batimetria_pts  
(con método bilinear)  
v.surf.bspline in=batimetria_pts raster=Prof_bspline_bil  
col=PROFUNDIDA method=bilinear layer=1  
(con método bicúbico)  
v.surf.bspline in=batimetria_pts raster=Prof_bspline_bic  
col=PROFUNDIDA method=bicubic layer=1
```

Resultado: Se ha obtenido un error del driver DBMI-DBF al ejecutar el primer paso.

4.6.1.2.3. Método v.surf.rst

Aproximación espacial y análisis topográfico para los puntos o isocotas en una capa vectorial de entrada a una capa raster en punto flotante utilizando la regularización spline con tensión.

El método RST no funciona bien en las zonas que tienen rápidos cambios topográficos.

```
# interpolation based on all points
```

```
v.surf.rst input=Batimetria_P zcol=PROFUNDIDA elev=Prof_rst
```

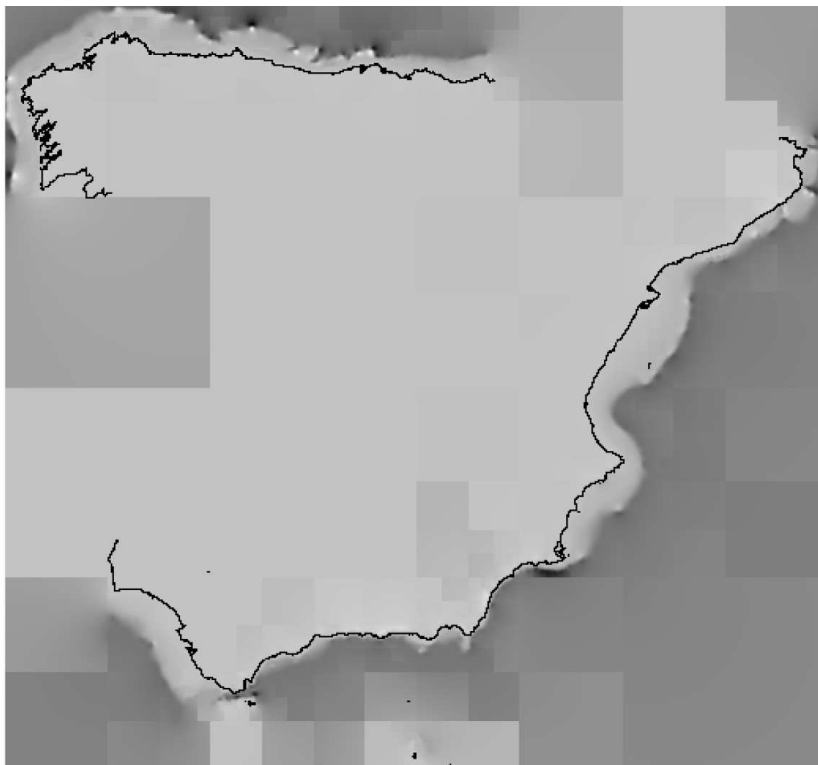


Figura 148: Resultado de Prof_rst en escala de grises con Línea de Costa de referencia

4.6.1.2.4. Módulo *r.surf.contour*

Este módulo está construido específicamente para la conversión de las líneas de contorno en el MDE. Genera un mapa de elevaciones raster a partir de unas líneas de contorno rasterizadas. Los valores de elevación se determinan utilizando procedimientos similares a los métodos manuales. Determina el valor de la elevación a partir de las dos líneas de contorno más cercanas (uphill y downhill).

Para las versiones anteriores a la 6.4.1 de GRASS es necesario hacer una conversión a número enteros de la capa raster [OSGEO, 2012].

Al tener instalado la versión 6.4.2 de GRASS, ejecutamos:

```
r.surf.contour in=Batimetria_P out=Prof_contour
```

Resultado: Se queda en 0% sin llegar a terminar. Tiempo en ejecución: 14 horas.

El funcionamiento de *r.surf.contour* es muy sensible a la resolución del mapa vectorial rasterizado, Si contiene múltiples líneas de contorno, pueden ocurrir diferentes anomalías. La velocidad de este módulo depende la distancia comprendida entre las líneas de contorno.

4.6.1.3. Conclusión

A la vista de los resultados obtenidos, utilizaremos como capa de entrada a la herramienta DSS la capa de profundidad obtenida a través del método “Rellenar celdas sin datos” de GvSIG. A través de este método se obtienen unos datos intermedios más lineales sin cambios demasiados bruscos entre las diferentes isolíneas que comprende el mapa de batimetría.

4.6.2. Estudio de la EMC (Evaluación Multicriterio) en un caso práctico

Se realiza un estudio de la aplicación de diferentes métodos de **Evaluación Multicriterio (EMC)** como son la **Media Lineal Ponderada (MLP)** y **Mediana Ponderada Ordenada (OWA)**. Se aplica con diferentes criterios físicos, operacionales y medioambientales propuestos para la obtención de resultados dentro del SIG GRASS.

Se obtienen diferentes resultados teniendo en cuenta los valores nulos de las capas de entrada o en contra partida, la asignación del valor 0.0 a dichos valores nulos, lo que les hace estar incluidos dentro de la evaluación.

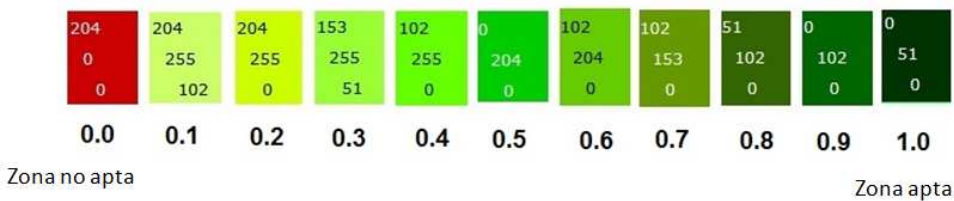
Dentro de cada ámbito existen criterios de exclusión total o parcial. Las restricciones o criterios totales corresponden con operaciones que devuelven cierto (valor 1.0) / falso (valor 0.0), y los factores o criterios parciales están asociadas a algún tipo de incertidumbre en el criterio de decisión, donde los valores entran en el intervalo [0.0, 1.0], donde 1.0 significa la certeza absoluta del cumplimiento del criterio y 0.0 que no se cumple.

La zona de estudio propuesta es la región de Andalucía por mayor cantidad de información en la base de datos.

4.6.2.1. Lenguaje de las siglas y paleta de colores de los resultados

CP := Criterio parcial
CT := Criterio total
RF := Regla física
RO := Regla operacional
REIA := Regla medioambiental

Paleta de colores verde para el intervalo [0-1]



Paleta de colores azul



Figura 149: Paleta de colores para el estudio de la EMC en un caso práctico

4.6.2.2. Listado de criterios y operaciones que intervienen

4.6.2.2.1. Reglas físicas

- **Criterio:** Zonas con distancia a la costa superior a 8 Km

Capa de entrada: Lin_Costa_P.shp

Operaciones:

1. Cálculo capa raster de distancias a la costa
2. Restringir la zona de estudio a la región de Andalucía o región donde se va realizar el estudio
3. Reclasificación: poner valor 1 si condición valor celda > 8 km, sino valor 0

Salida: Capa RF_DistCosta.tif (raster)

- **Criterio:** Zonas que cumplan una altura de ola significativa no superior a 14 m

Capa de entrada: Unión de todas las capas rasterHsMediaRaster_i.img

Operaciones:

1. Restringir a la región de Andalucía o región donde se va a realizar el estudio
2. Reclasificación: poner valor 1 si condición valor celda < 14 m, sino valor 0

Salida: Capa RF_Hs.tif (raster)

- **Operación:** AND de las capas RF_DistCosta.tif y Capa RF_Hs.tif

Salida: **RF_DCyHs.tif** (entrada de reglas operacionales y reglas de impacto medio ambiental)

Observaciones:

- La operación AND en raster
- Almacenar como resultado intermedio
- Capa resultante raster (posibilidad de pasar a vectorial para su almacenaje en PostGIS y visualización en MapServer).

4.6.2.2.2. Reglas Operacionales

- **Criterio:** Zonas excluyentes a un buffer de 2 millas náuticas (3704 metros) a las rutas de navegación.

Capa de entrada: Rutas_trans_maritimo.shp (sólo zona Andalucía)

Operaciones:

1. Aplicar buffer de 2 millas náuticas a las rutas marítimas
2. Asignar valores 1 fuera del buffer y 0 dentro del buffer.
3. Al ser una capa vectorial podemos operar tanto en vectorial como en raster. La Media Lineal Ponderada es una operación raster. Hay que considerar si en el paso anterior se ha vectorizado la capa, en el paso siguiente se podría aplicar un AND vectorial (jugar con los tiempos de respuesta).

Salida: RO_CT_RutasNav.tif (raster)

- **Operación:** AND de reglas físicas y reglas operacionales totales (RO_CT_RutasNav y RF_DCyHs)

Salida: **RF_RO_CT.tif** (raster)

A continuación se va aplicar la MLP de las reglas parciales operacionales.

- **Criterio:** Aplicación de operación difusa sobre la potencia del viento.

Capa de entrada: Viento_Anual.img (podría aplicarse sobre alguna estación del año: Viento_primavera.img, Viento_verano.img, Viento_otoño.img, Viento_invierno.img)

Operaciones:

1. Restringir a la región de Andalucía o región donde se va realizar el estudio
2. Aplicar fuzzy con operador lineal y valores 6, 8.5, 14 y 15.

Salida: RO_CP_VelViento.tif (raster)

- **Criterio:** Aplicación de operación difusa sobre la altura media de ola.

Operaciones:

1. Restringir a la región de Andalucía o región donde se va a realizar el estudio
2. Aplicar fuzzy con operador lineal y valores 0, 0, 0 y 14 (rampa descendente que asigna 1 a las celdas con valor 0 y 0 a las celdas con valor 14)

Salida: RO_CP_Hs.tif

- **Criterio:** Aplicación de operación difusa sobre las distancias a las rutas marítimas.

Capa entrada: Rutas_trans_maritimo.shp (sólo zona Andalucía)

Operación:

1. Cálculo capa raster de distancias a las rutas marítimas
2. Aplicar fuzzy sobre capa anterior con operador lineal y valores 2, 3, 5 y 20 (se asigna 0 a las celdas con valores menores a 2 o mayores a 20, 1 a las celdas con valores entre 3 y 5, y fuzzy a las celdas con valores entre 2 y 3 o entre 5 y 20)

Salida: RO_CP_DistRutasMarit.tif

- **Operación:** MLP con pesos

$$\begin{aligned}W1(\text{RO_CP_VelViento.tif}) &= 0.6 \\W2(\text{RO_CP_Hs.tif}) &= 0.3 \\W3(\text{RO_CP_DistRutasMarit.tif}) &= 0.1\end{aligned}$$

Salida: **RO_CP_MLP.tif**

- **Operación:** AND_fuzzy (mínimo) de RO_CP_MLP y RF_RO_CT

Salida: **RF_RO_salida.tif** (raster)

4.6.2.2.3. Restricciones de impacto medio ambiental

- **Criterio:** Zonas excluyentes de las zonas de Almadrabas

Entrada: Almadrabas_P.shp

Operaciones:

1. Restringir a la región de Andalucía o región donde se va realizar el estudio
2. Pasar a raster
3. Asignar valores 1 valores nulos y 0 dentro de la zona de Almadrabas.

Salida: REIA_Almadrabas.tif (raster)

- **Criterio:** Zonas excluyentes a las zonas MABs (zonas de reserva de la biosfera)

Entrada: MABs_P.shp

Operaciones:

1. Restringir a la región de Andalucía o región donde se va realizar el estudio
2. Pasar a raster
3. Asignar valores 1 valores nulos y 0 dentro de la zona de MABs.

Salida: REIA_MABs.tif

- **Operación:** MLP de las reglas medio ambientales, con pesos:

$$W1 (RIA_Almadrabas) = 0.7$$

$$W2 (RIA_MABs) = 0.3$$

Salida: **REIA_MLP.tif** (raster)

- **Operación:** AND_fuzzy (mínimo) de reglas de impacto medio ambiental y reglas físicas (RF_DCyHs y REIA_MLP)

Salida: **RF_REIA_salida.tif** (raster)

4.6.2.2.4. Regla de Decisión Final (OWA- Media Ponderada Ordenada)

- **Criterio:** Cálculo de la regla de decisión final mediante una Media Ponderada Ordenada (OWA) de las restricciones físicas y operacionales junto con las reglas físicas y reglas de impacto medio ambiental (**RF_RO_salida** y **RF_REIA_salida**)

$$W1 (RF_RO_salida) = 0.5$$

$$W2 (RF_REIA_salida) = 0.5$$

Se calcularán las capas según los pesos de orden:

- Z1=1 y Z2=0
Salida: **CapaResultado_1_0.tif**
- Z1=0 y Z2=1
Salida: **CapaResultado_0_1.tif**
- Z1=0.5 y Z2=0.5
Salida: **CapaResultado_0.5_0.5.tif**
- Z1=0.25 y Z2=0.75
Salida: **CapaResultado_0.25_0.75.tif**
- Z1=0.75 y Z2=0.25
Salida: **CapaResultado_0.75_0.25.tif**

4.6.2.3. Esquema del caso práctico

En el siguiente esquema se refleja de forma gráfica los criterios que intervienen, las salidas obtenidas y las operaciones aplicadas.

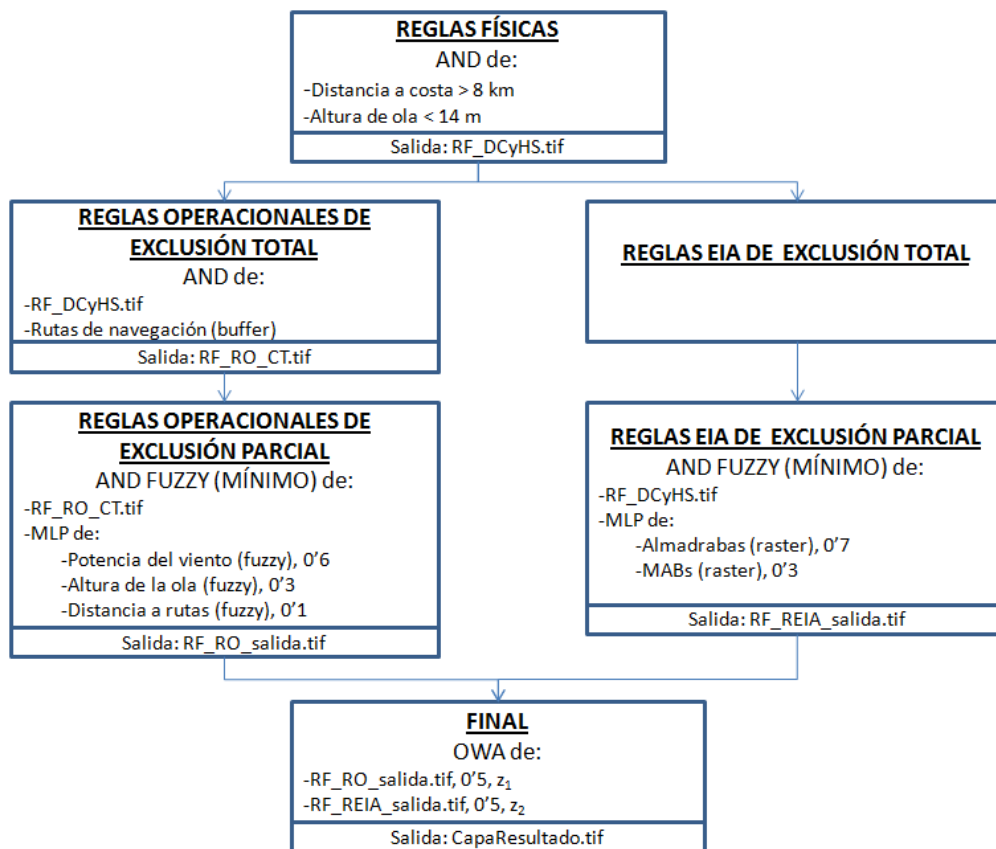


Figura 150: Esquema de operación del caso práctico

4.6.2.4. Consideraciones previas

Los comandos utilizados para las operaciones MLP, OWA, FUZZY y AND_FUZZY en GRASS están definidas para capas raster. Excluyen los valores nulos como entrada a la ecuación, lo que implica una exclusión de esas celdas de estudio.

Las operaciones OWA, FUZZY y AND_FUZZY que intervienen en este caso práctico son:

- **Operación OWA de dos entradas:**

```
r.mapcalc "capaOWA = if(isnull(capal) || isnull(capa2), null(),
    if(capal >= capa2, eval(((capal*w1*z1) + (capa2*w2*z2)) /
(w1*z1+w2*z2)), eval(((capal*w1*z2) + (capa2*w2*z1)) /
(w1*z2+w2*z1)))"
```

- **Operación OWA de tres entradas:**

```
r.mapcalc "capaOWA = if(isnull(capal) || isnull(capa2) ||
isnull(capas3), null(),
    if(capal >= capa2 && capa2 >= capas3, eval(((capal*w1*z1) +
(capas2*w2*z2) + (capas3*w3*z3)) / (w1*z1+w2*z2+w3*z3)),
    if(capal >= capas3 && capas3 > capa2, eval(((capal*w1*z1) +
(capas3*w3*z2) + (capa2*w2*z3)) / (w1*z1+w3*z2+w2*z3)),
    if(capas2 > capal && capal >= capas3, eval(((capa2*w2*z1) +
(capal*w1*z2) + (capas3*w3*z3)) / (w2*z1+w1*z2+w3*z3)),
    if(capas2 >= capas3 && capas3 > capal, eval(((capa2*w2*z1) +
(capas3*w3*z2) + (capal*w1*z3)) / (w2*z1+w3*z2+w1*z3)),
    if(capas3 > capal && capal >= capa2, eval(((capas3*w3*z1) +
(capal*w1*z2) + (capa2*w2*z3)) / (w3*z1+w1*z2+w2*z3)),
    if(capas3 > capa2 && capa2 > capal, eval(((capas3*w3*z1) +
(capas2*w2*z2) + (capal*w1*z3)) / (w3*z1+w2*z2+w1*z3)), null()
))))))"
```

- **Operación Fuzzy:**

```
r.mapcalc "capaFuzzy = if(isnull(capal), null(), if((capal <= A ||
capal >= D), 0.0, if((capal > A && capal < B), eval((capal - A)/(B -
A)), if((capal > C && capal < D), eval((D - capal)/(D - C)),
1.0))))"
```

- **Operación And fuzzy (mínimo):**

```
r.mapcalc "capaMin = if(isnull(capal) || isnull(capas2), null(),
    if(capal <= capas2, capal, capas2))"
```

Se observa a través de las ecuaciones que si alguna celda de las capas de entrada posee valor nulo, dicha celda no es contemplada en la operación.

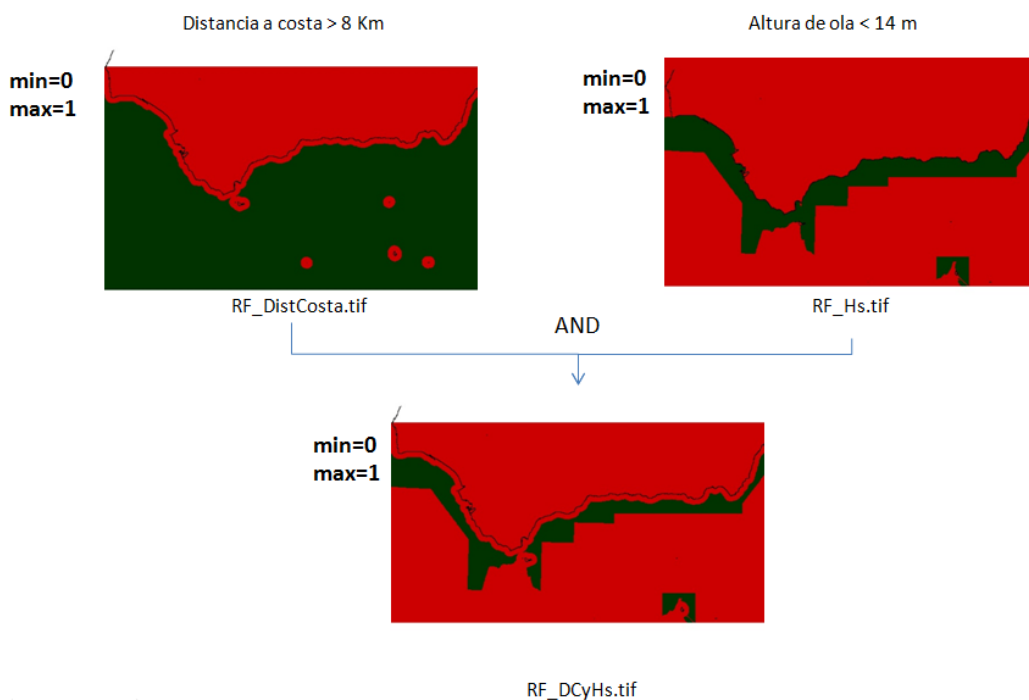
En este caso práctico, se ha aplicado la MLP a través de la operación OWA asignando pesos equitativos en los pesos de orden.

4.6.2.5. Estudio con valores nulos

El estudio preservando los valores nulos de las capas de entrada deriva un acotamiento de las zonas de resultado.

A continuación se van a mostrar los mapas intermedios y finales de la aplicación de los criterios teniendo en cuenta los valores nulos de las capas de entrada. Los mapas representados se visualizan junto con la línea de costa para tener un referente visual.

4.6.2.5.1. Aplicación de criterios físicos



Se aplican los dos criterios físicos de exclusión total y se combinan con una operación AND. En el caso del criterio de altura de ola significativa no superior a 14 m apreciamos que viene determinada por la capa de entrada. La capa de entrada a este criterio es la altura media de ola significativa, que alberga información en las zonas donde se han obtenido datos, el resto de celdas son valores nulos y el valor máximo de la capa es 2.7 metros.

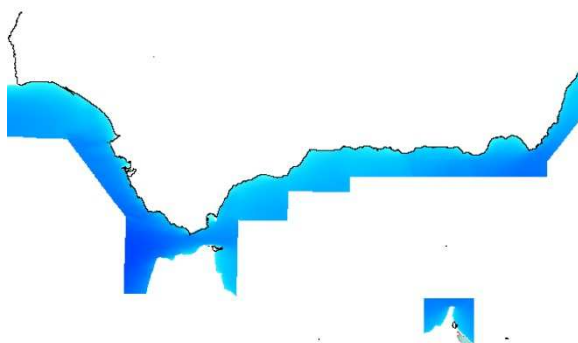
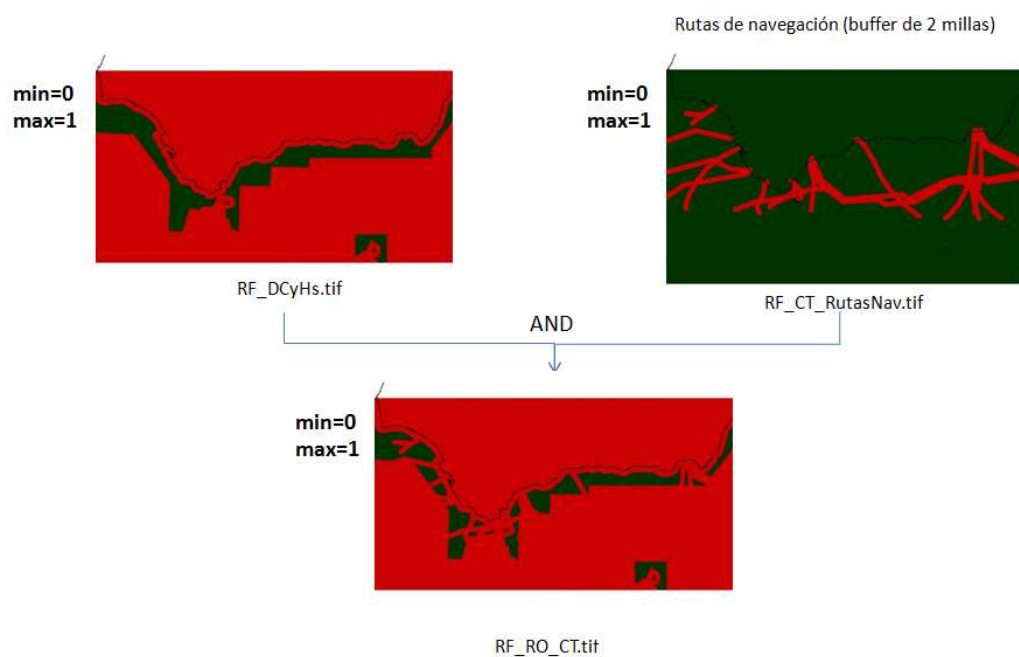


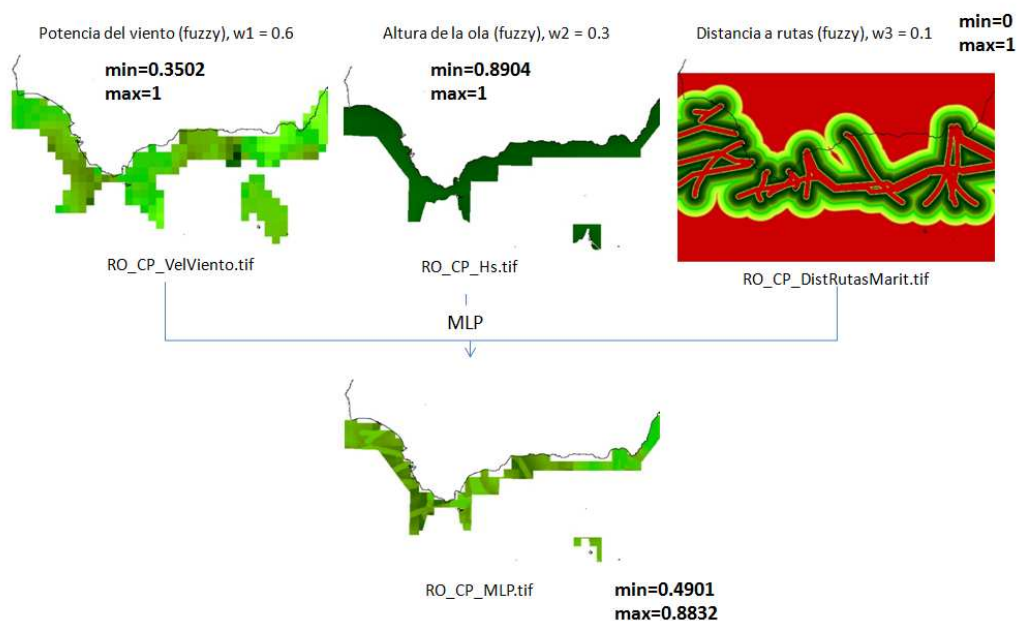
Figura 151: Capa raster de entrada Altura media de ola significativa

4.6.2.5.2. Aplicación de criterios operacionales de exclusión total



Se aplica un buffer a las rutas de navegación como criterio operacional de exclusión total y se combina con el mapa resultado de los criterios físicos mediante la operación AND.

4.6.2.5.3. Aplicación de criterios operacionales de exclusión parcial



En una primera parte se aplica la MLP sobre los criterios operacionales de exclusión parcial. En el criterio de la potencia del viento considerada de exclusión parcial, se aplica una operación fuzzy en la que está restringida por los datos de entrada. La capa de entrada es la potencia del viento que contiene información donde se han obtenido datos y el resto de valores son nulos, como se muestra en la siguiente figura. El valor mínimo de la capa raster es 6.42 KWh/m y el valor máximo es 8.58 KW/m.

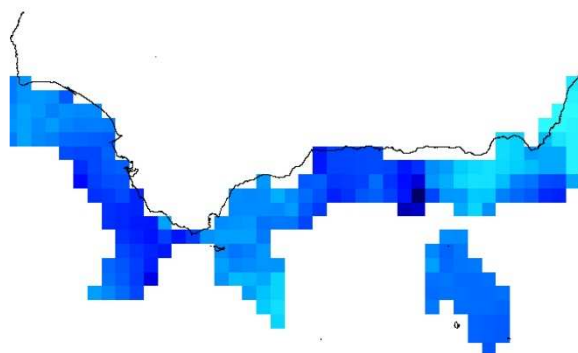
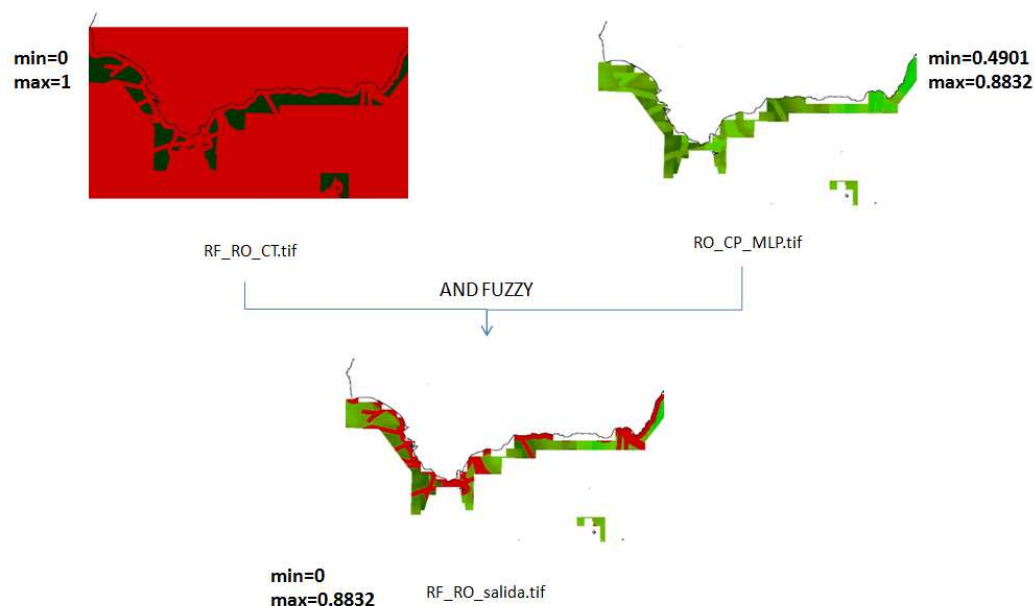


Figura 152: Capa raster de entrada de potencia del viento

En el criterio de altura de ola considerada de exclusión parcial, se aplica una operación fuzzy que de igual manera está restringida por los datos de entrada (ver *figura 151*).

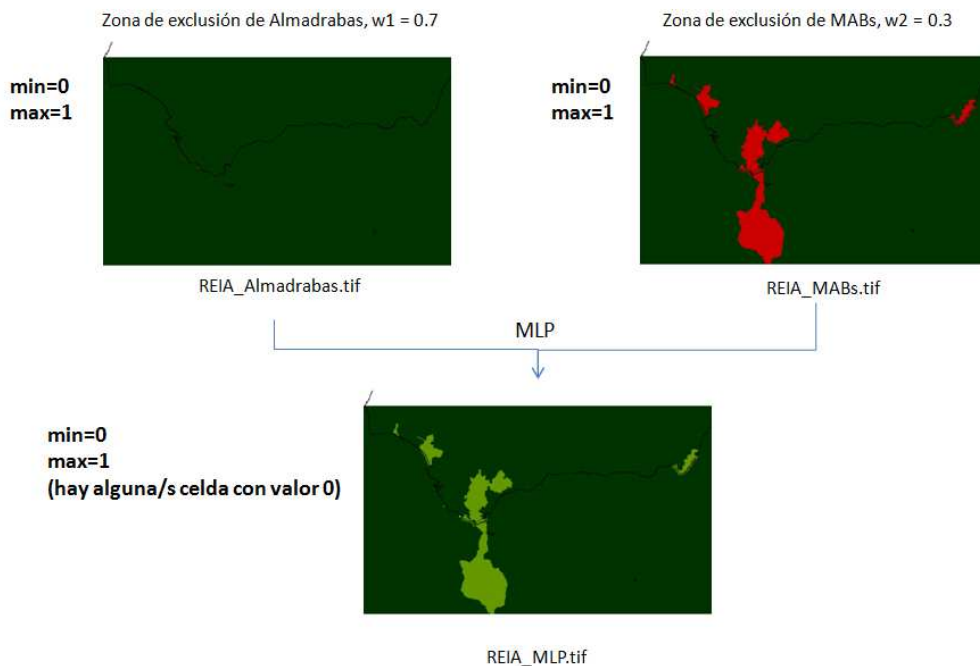
En el criterio de distancia a las rutas marítimas, se aplica una operación de distancias sobre las rutas marítimas y posteriormente se aplica una operación fuzzy.

El mapa resultante de la operación MLP vemos que está determinado por las celdas con datos de las capas de entrada.

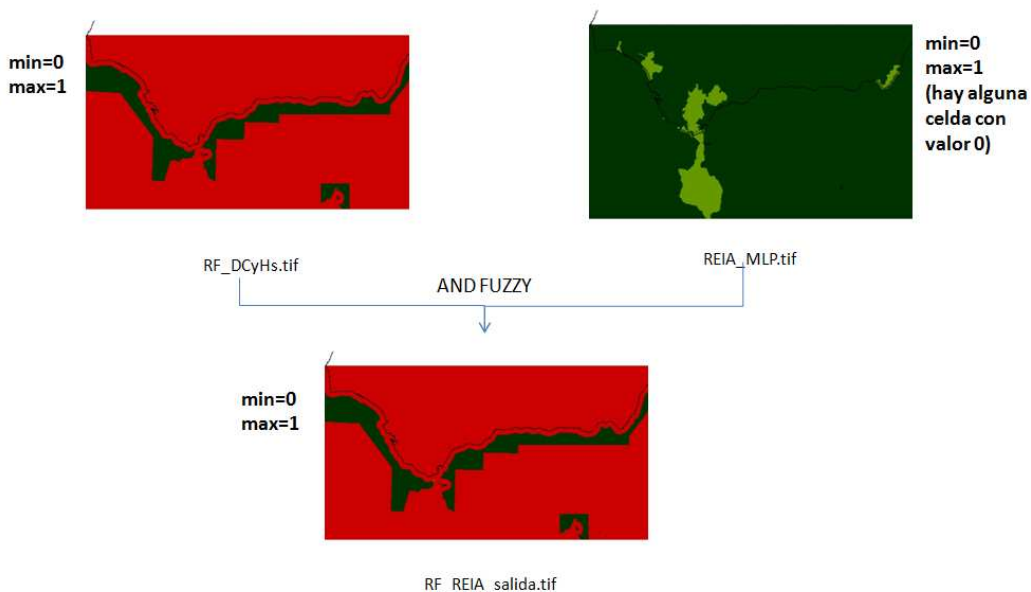


En esta segunda parte, se aplica la operación AND_FUZZY sobre el mapa resultado de la MLP y el mapa resultado del bloque de criterios operaciones de exclusión total. De igual manera, vemos que el mapa resultado de esta operación está acotado por aquellas celdas donde contienen datos.

4.6.2.5.4. Aplicación de criterios medioambientales de exclusión parcial



En los criterios medioambientales de exclusión parcial, se extraen las zonas de las almadrabas y las zonas de protección de la biosfera y se aplica la MLP. Como se ha asignado valor 1.0 fuera de las celdas que contienen datos, el mapa resultado de la MLP no va a estar acotado por los datos.



Posteriormente se aplica una operación AND_FUZZY sobre el mapa resultado de la MLP y el mapa resultado del bloque de criterios físicos, ya que en este caso no se han aplicado criterios medioambientales de exclusión total.

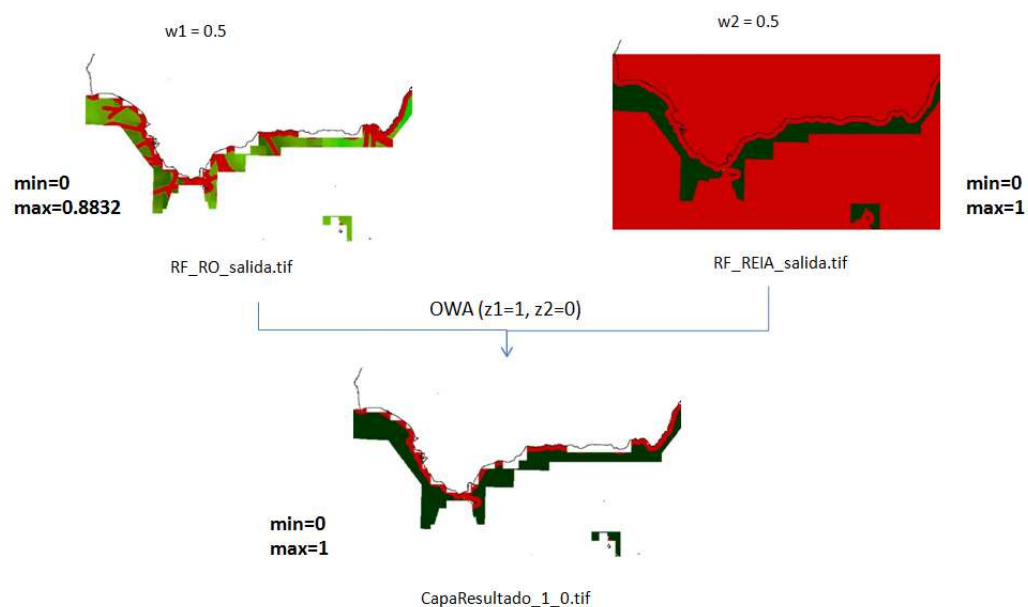
4.6.2.5.5. Toma de decisión final

La toma de decisión final se verá afectada por los valores nulos arrastrados en los mapas intermedios que se han generado.

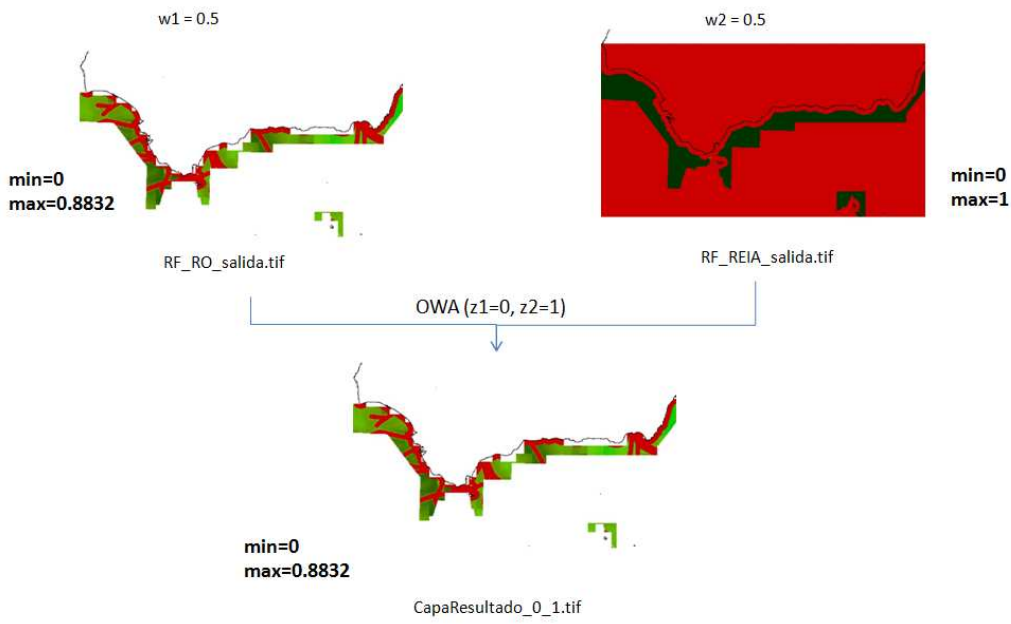
Se va a obtener 4 resultados teniendo en cuenta el riesgo según los pesos de orden aplicados en la operación OWA.

- **Riesgo máximo (OR):** con pesos de orden $z1=1.0$ y $z2=0.0$
- **Riesgo mínimo (AND):** con pesos de orden $z1=0.0$ y $z2=1.0$
- **Riesgo moderado mayor:** con pesos de orden $z1=0.75$ y $z2=0.25$
- **Riesgo moderado menor:** con pesos de orden $z1=0.25$ y $z2=0.75$
- **Sin riesgo (MLP):** con pesos de orden equitativos $z1=0.5$ y $z2=0.5$

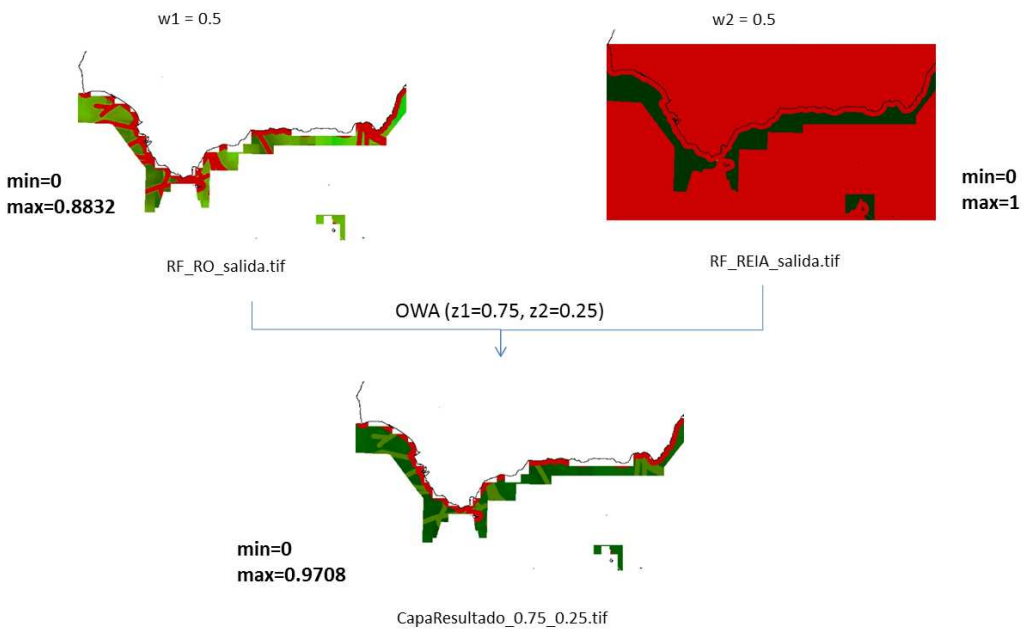
4.6.2.5.5.1. Riesgo máximo



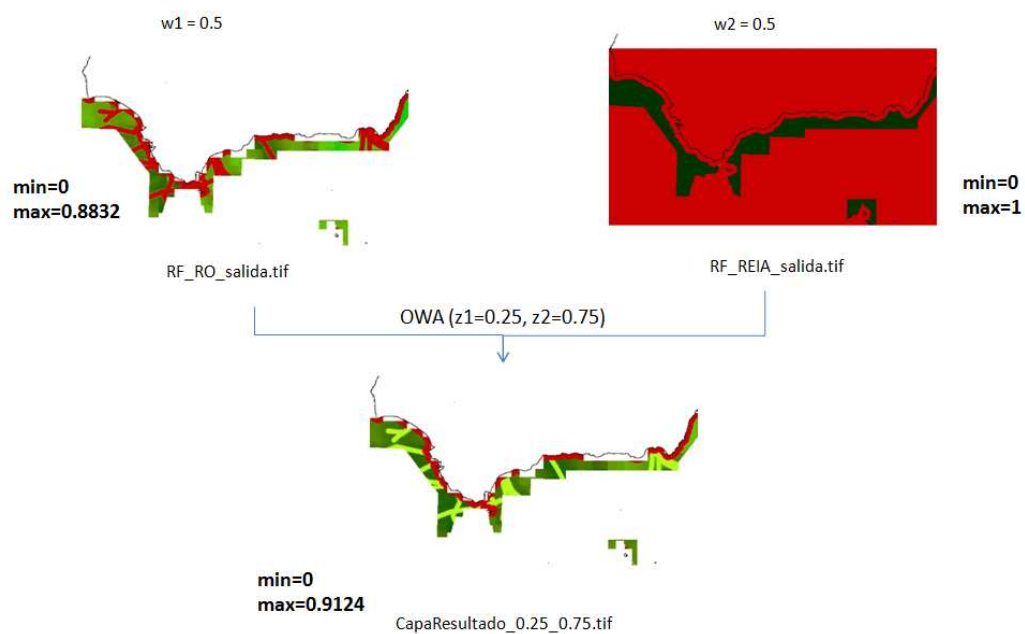
4.6.2.5.5.2. Riesgo mínimo



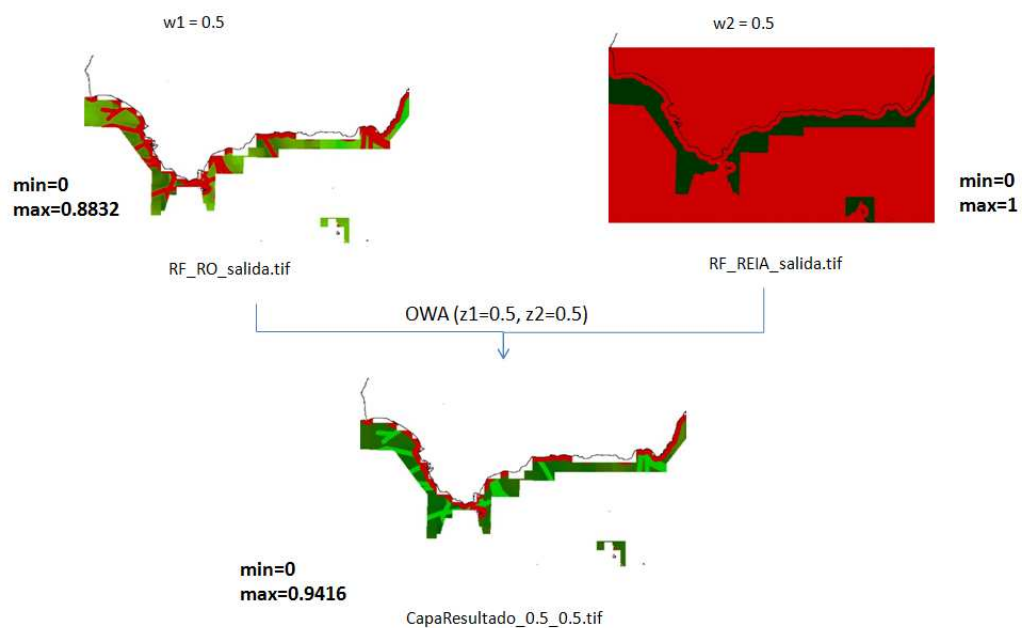
4.6.2.5.5.3. Riesgo moderado mayor



4.6.2.5.5.4. Riesgo moderado menor



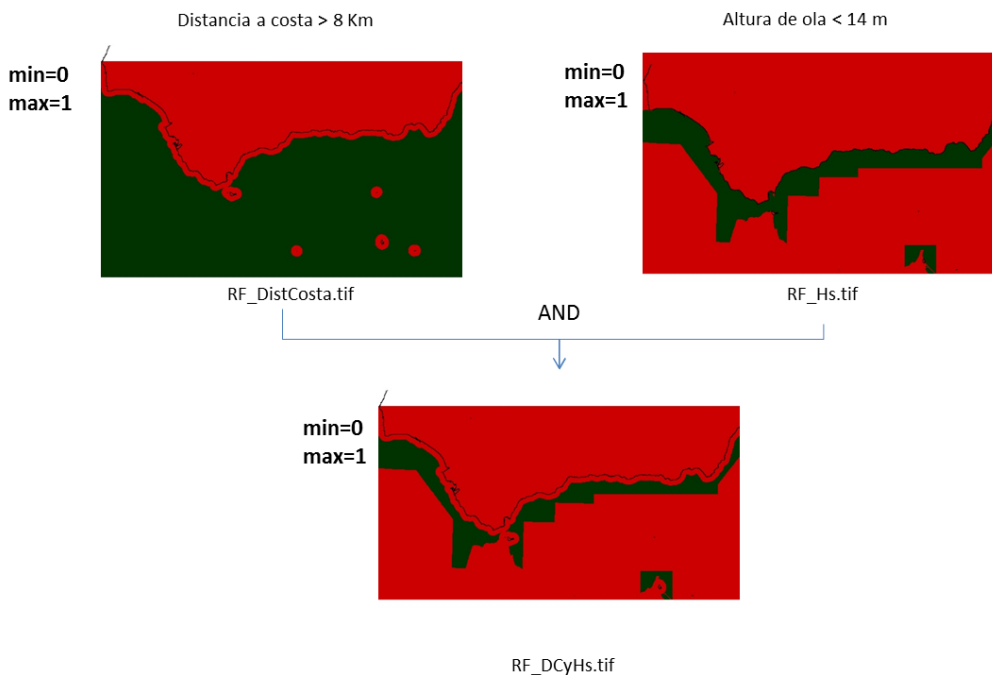
4.6.2.5.5.5. Sin riesgo



4.6.2.6. Estudio sin valores nulos

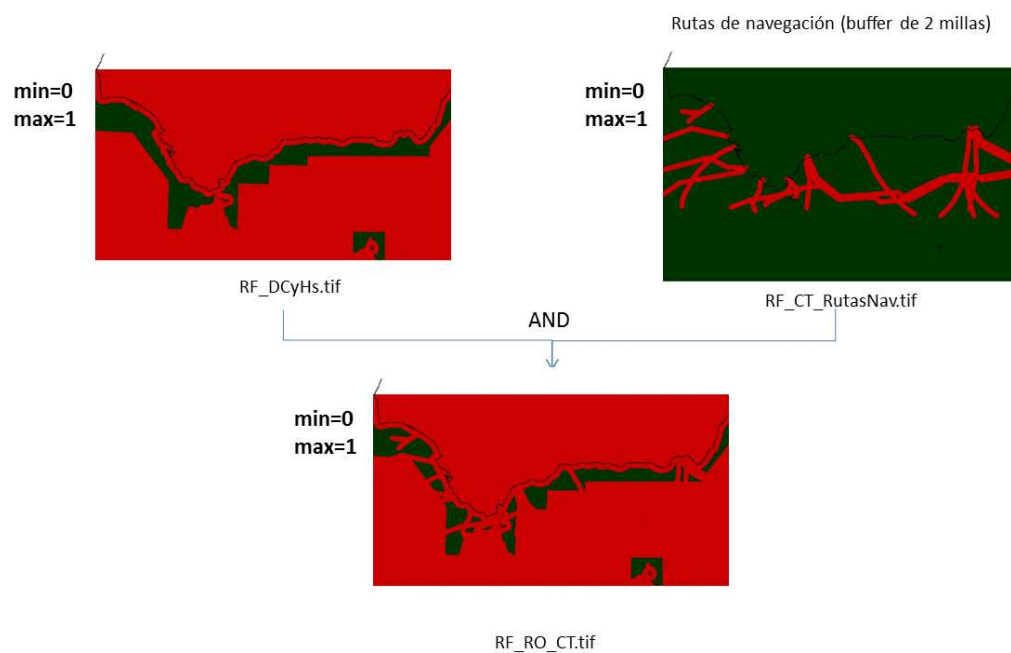
El estudio se ha realizado aplicando los criterios pero asignando valor 0.0 a aquellas celdas que contiene valores nulos en las capas raster de entrada a las operaciones MLP, OWA, FUZZY y AND_FUZZY. De esta manera, dichas celdas formarán parte de la valoración al realizar la EMC.

4.6.2.6.1. Aplicación de criterios físicos



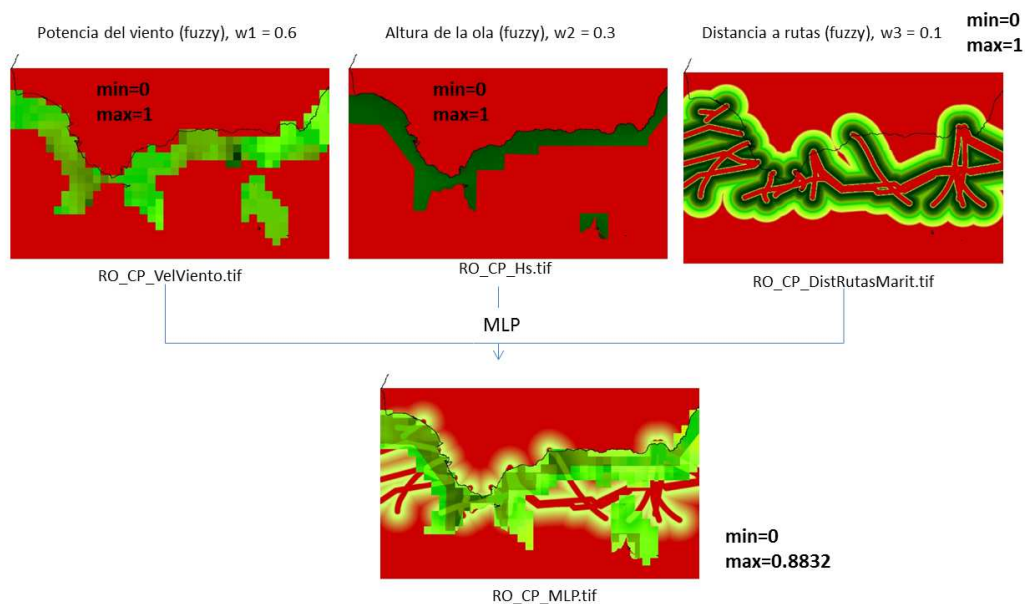
Se aplican los dos criterios físicos de exclusión total y se combinan con una operación AND. Este caso es el mismo que para el estudio con valores nulos.

4.6.2.6.2. Aplicación de criterios operacionales de exclusión total



Se aplica un buffer a las rutas de navegación como criterio operacional de exclusión total y se combina con el mapa resultado de los criterios físicos mediante la operación AND. Este caso es el mismo que para el estudio con valores nulos.

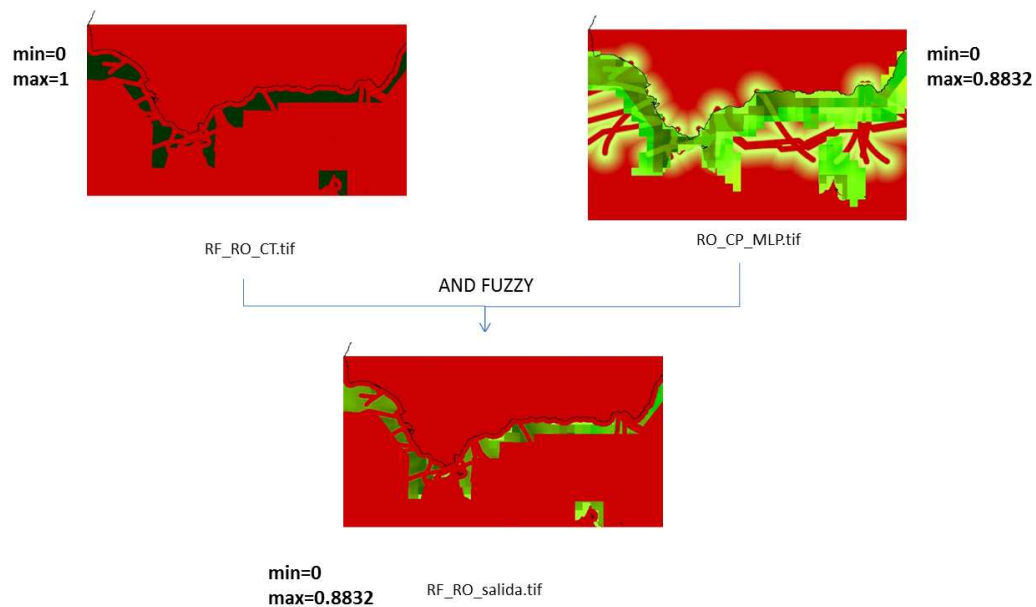
4.6.2.6.3. Aplicación de criterios operacionales de exclusión parcial



En una primera parte se aplica la MLP sobre los criterios operacionales de exclusión parcial. En el criterio de la potencia del viento considerada de exclusión parcial, se aplica una operación fuzzy en la que está restringida por los datos de entrada como vimos en el estudio con valores nulos. En este caso, antes emplear la operación fuzzy, se ha fijado el valor 0.0 a los valores nulos de la capa raster de entrada de la potencia del viento. Del mismo modo que para el criterio de altura de ola.

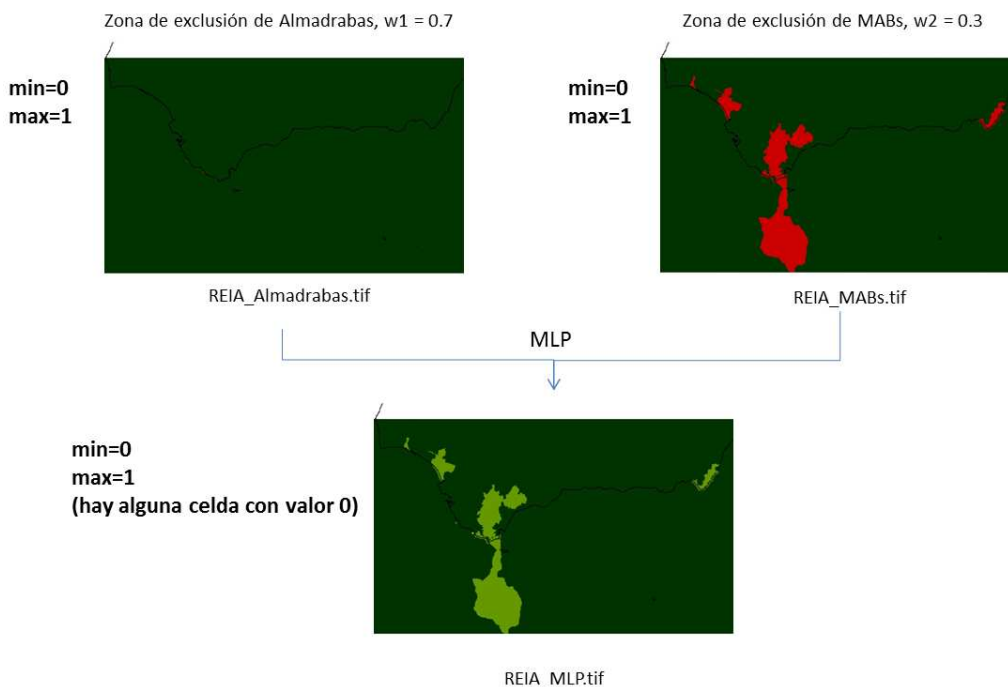
En el criterio de distancia a las rutas marítimas, se aplica una operación de distancias sobre las rutas marítimas y posteriormente se aplica una operación fuzzy.

En este caso contemplamos que una cantidad mayor de celdas han sido evaluadas en la operación MLP al no desestimar las celdas con valores nulos.

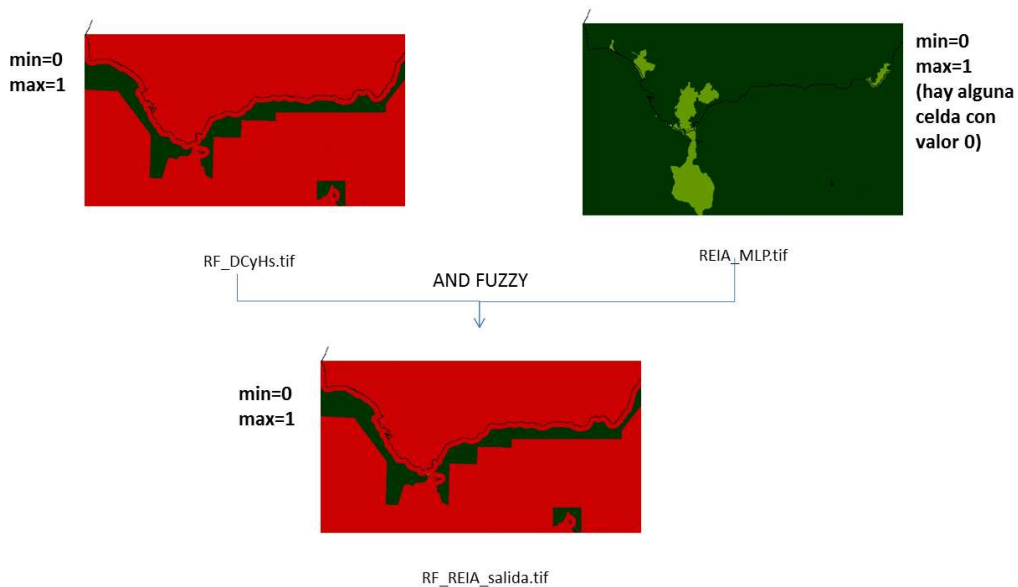


Posteriormente empleamos la operación AND_FUZZY sobre el mapa resultado de la MLP y el mapa resultado del bloque de criterios operaciones de exclusión total.

4.6.2.6.4. Aplicación de criterios medioambientales de exclusión parcial



En los criterios medioambientales de exclusión parcial, se extraen las zonas de las almadrabas y las zonas de protección de la biosfera y se aplica la MLP de igual modo que se hizo en el caso de estudio con valores nulos.



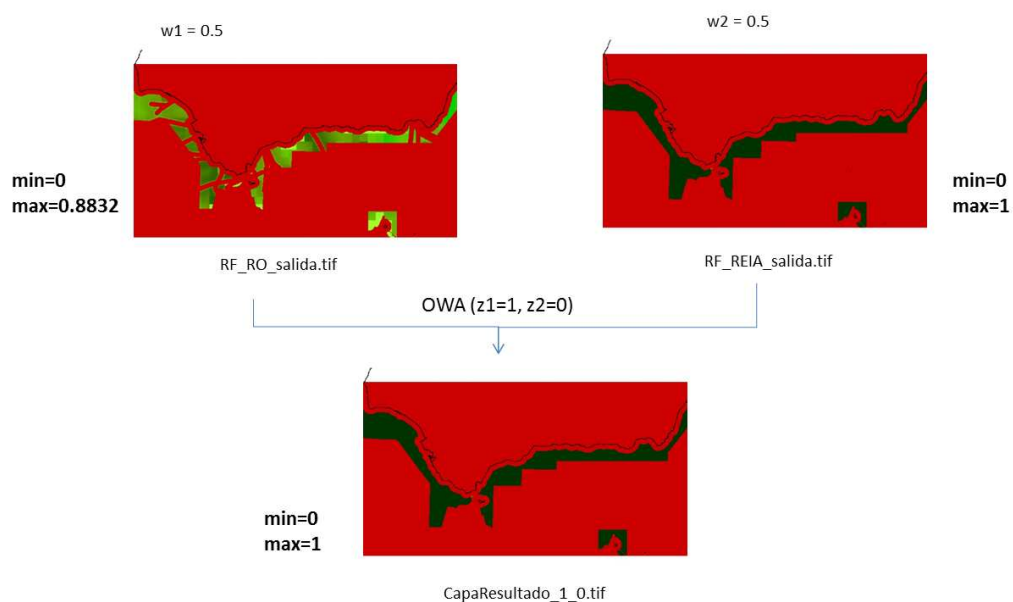
Posteriormente se aplica una operación AND_FUZZY sobre el mapa resultado de la MLP y el mapa resultado del bloque de criterios físicos, ya que en este caso no se han aplicado criterios medioambientales de exclusión total.

4.6.2.6.5. Toma de decisión final

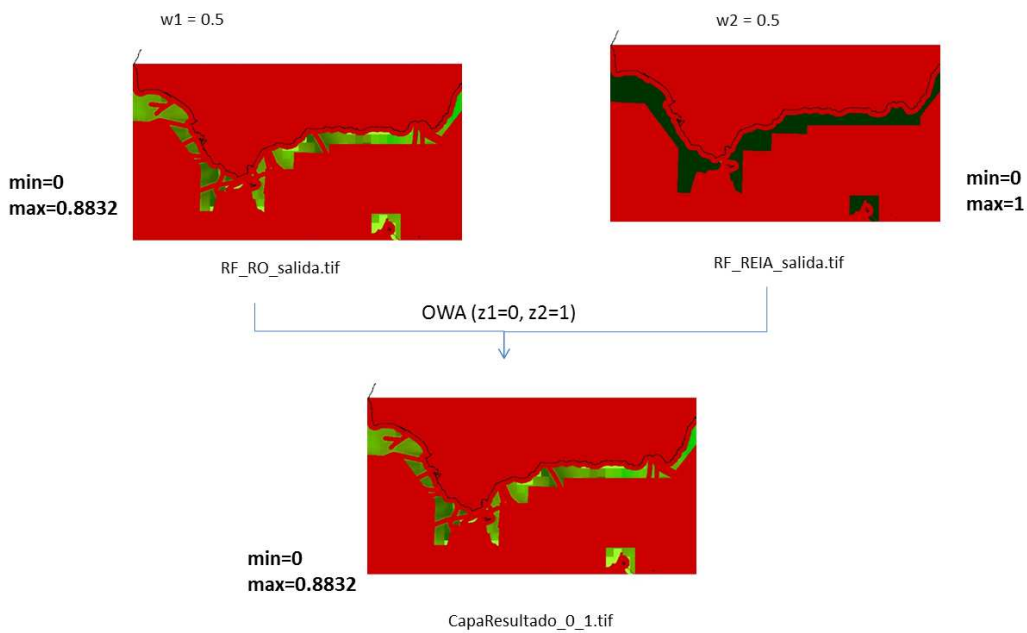
En este caso, el mapa resultado de la toma de decisión final no contendrá valores nulos y todas las celdas formarán parte de la EMC. Se va a obtener 4 resultados teniendo en cuenta el riesgo según los pesos de orden aplicados en la operación OWA.

- **Riesgo máximo (OR):** con pesos de orden $z1=1.0$ y $z2=0.0$
- **Riesgo mínimo (AND):** con pesos de orden $z1=0.0$ y $z2=1.0$
- **Riesgo moderado mayor:** con pesos de orden $z1=0.75$ y $z2=0.25$
- **Riesgo moderado menor:** con pesos de orden $z1=0.25$ y $z2=0.75$
- **Sin riesgo (MLP):** con pesos de orden equitativos $z1=0.5$ y $z2=0.5$

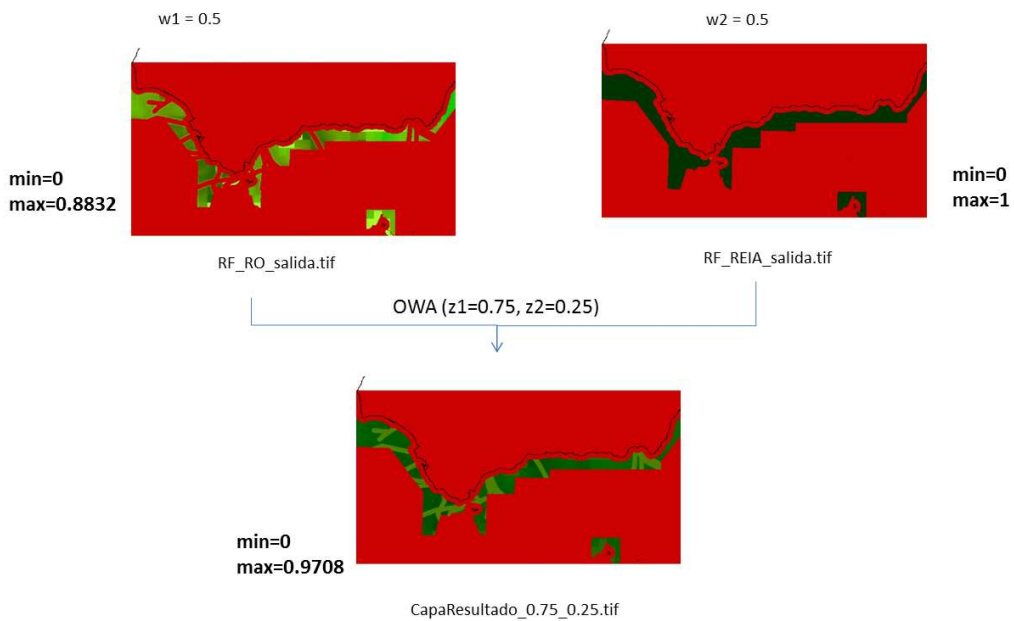
4.6.2.6.5.1. Riesgo máximo



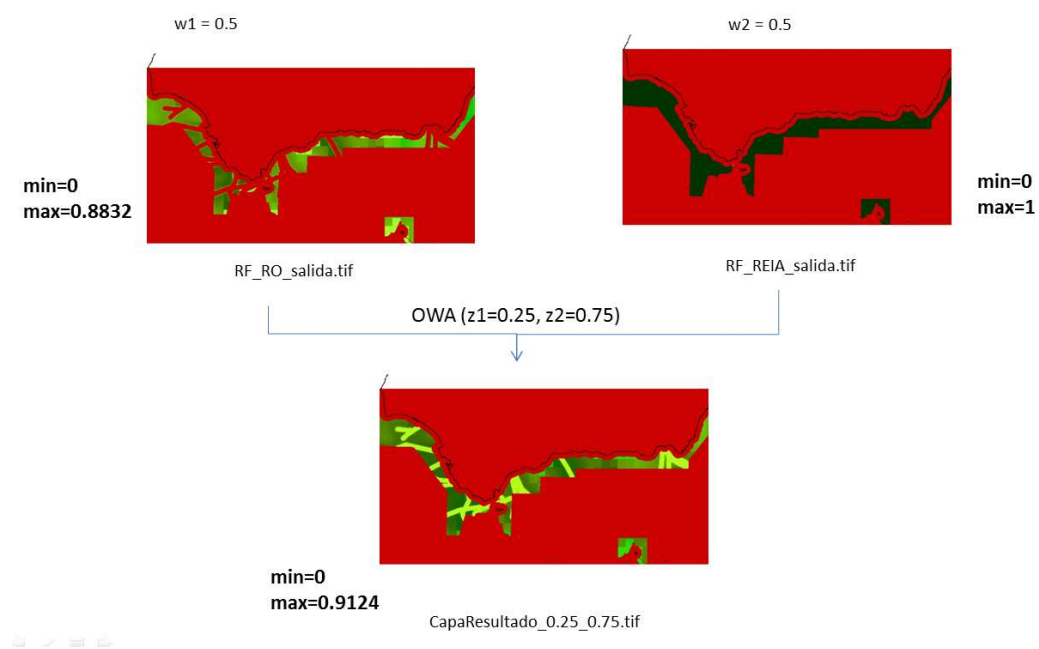
4.6.2.6.5.2. Riesgo mínimo



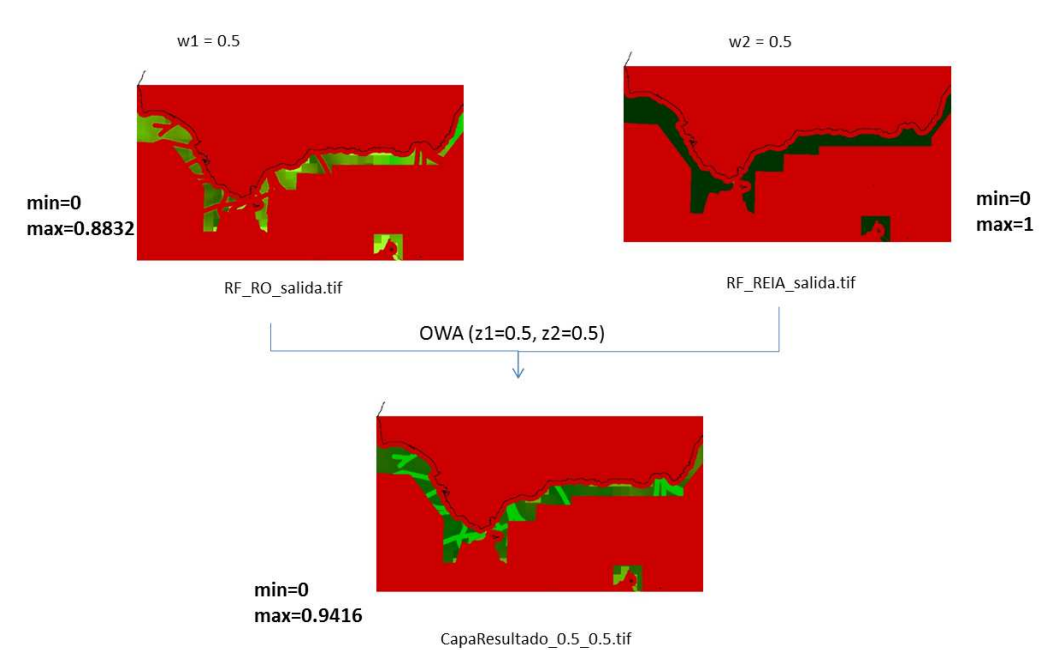
4.6.2.6.5.3. Riesgo moderado mayor



4.6.2.6.5.4. Riesgo moderado menor



4.6.2.6.5.5. Sin riesgo



4.6.2.7. Conclusiones

A la vista de los resultados, creemos conveniente aplicar el segundo caso de estudio a nuestros procedimientos de operación en la herramienta DSS. Elegimos el caso de estudio sin valores nulos porque de esta manera estamos evaluando todas las celdas que pertenecen a la región geográfica del escenario y no está restringido a aquellas zonas donde únicamente haya información en las capas de entrada.

4.6.3. Importación de las capas informativas de la BBDD OceanLider a PostgreSQL 9.1 + PostGIS 2.0

En este apartado se van a mencionar los pasos a seguir para la creación de la BBDD de OceanLider y su consiguiente población con las **capas informativas** de la BBDD proporcionada por PROES.

Se emplearán técnicas de importación múltiple de forma automática tanto para capas vectoriales (formato .SHP) como para capas raster (formato .TIF y .IMG), y técnicas de comprobación de errores en la importación múltiple.

4.6.3.1. Creación de la BBDD OceanLider en PostgreSQL

En primer lugar se va a crear la BBDD OceanLider dentro de PostgreSQL. Vamos a utilizar la herramienta *pgAdminIII* disponible en el paquete de instalación de PostgreSQL 9.1. Para la creación de la BBDD se puede realizar mediante dos vías: utilizando la interfaz gráfica de *pgAdminIII* o mediante sentencias SQL dentro de la consola de ejecución de sentencias SQL de *pgAdminIII*.

4.6.3.1.1. Creación de la base de datos utilizando la interfaz gráfica de *pgAdminIII*

Utilizando la interfaz gráfica, presionamos el botón derecho del ratón sobre “Databases” para crear una nueva base de datos.

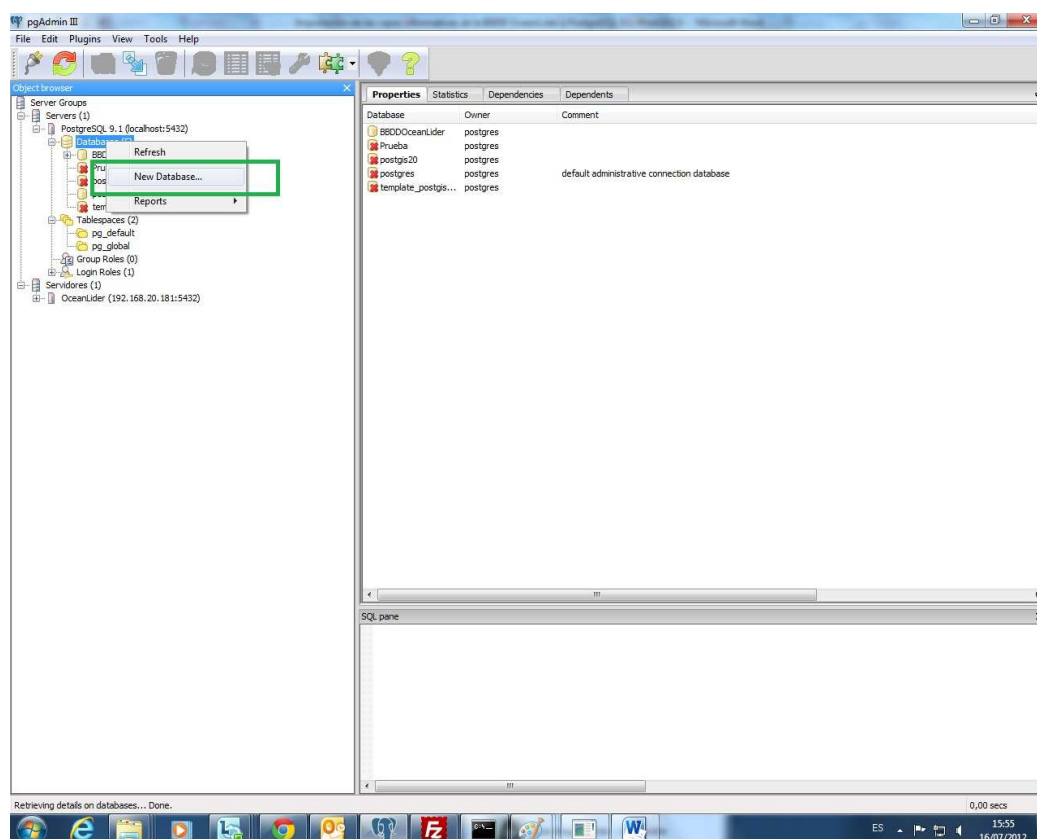


Figura 153: Creación de una nueva base de datos utilizando la interfaz gráfica de pgAdminIII

Nos mostrará un cuadro de dialogo en el que tendremos que indicar el nombre de la nueva base de datos, el propietario, la codificación y la plantilla a utilizar. Los datos de la nueva base de datos de OceanLider tendrá la siguiente información:

- **Nombre:** BBDDOceanLider
- **Propietario:** postgres
- **Codificación:** UTF-8
- **Plantilla:** postgis20

La plantilla *postgis20* es la que se utiliza para dar soporte tanto a formatos raster como vectorial. Es la plantilla que se crea en la instalación de PostGIS 2.0 y contiene todas las funciones geoespaciales de formato raster y vectorial.

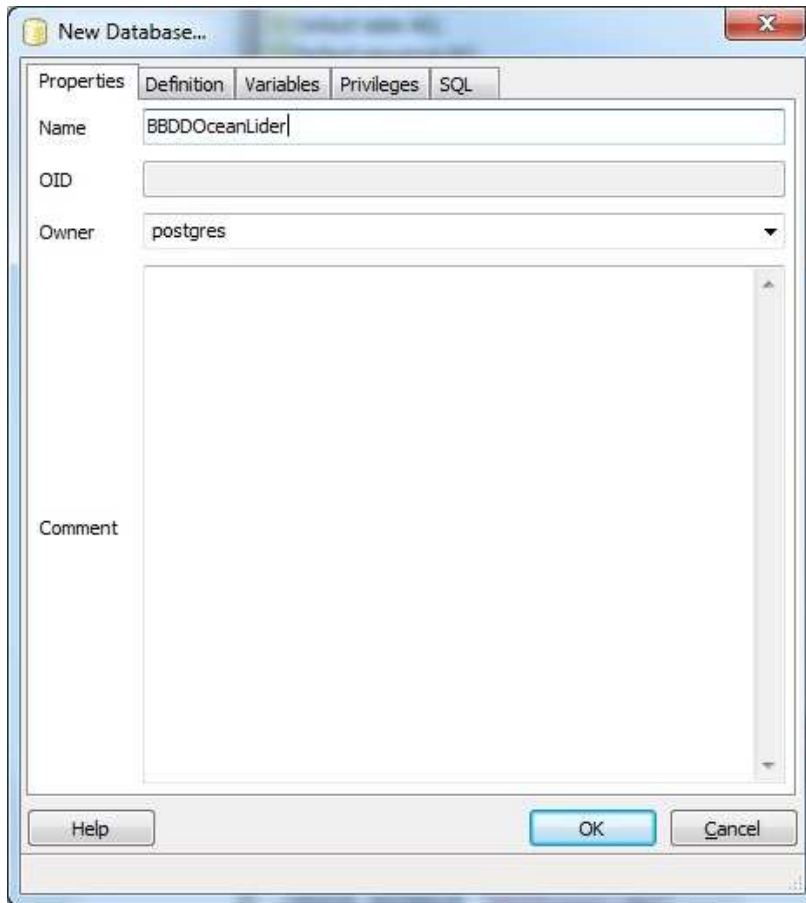


Figura 154: Pestaña de Propiedades en el cuadro de dialogo para la creación de una nueva base de datos utilizando la interfaz gráfica de pgAdminIII

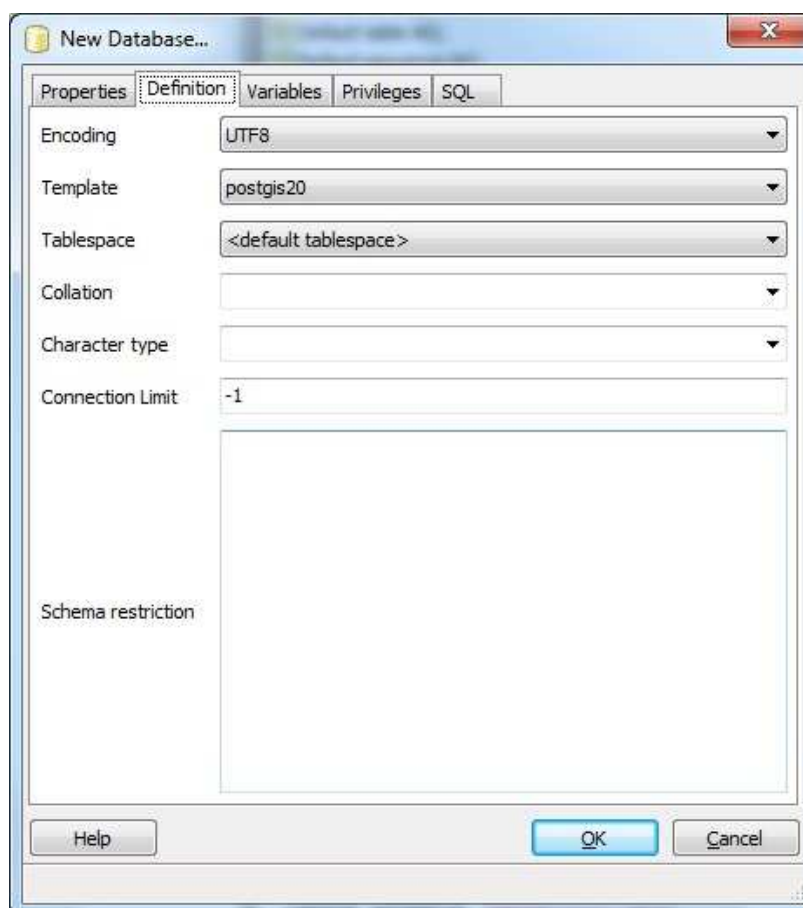


Figura 155: Pestaña de definición en el cuadro de dialogo para la creación de una nueva base de datos utilizando la interfaz gráfica de pgAdminIII

4.6.3.1.2. Creación de la base de datos utilizando la consola de ejecución de sentencias SQL de pgAdminIII

Utilizando la consola de ejecución de sentencias SQL de pgAdminIII podemos crear la base de datos directamente con código SQL. Para abrir la consola de ejecución de sentencias SQL seleccionamos el icono que se muestra a continuación ubicado en la barra de herramientas de pgAdminIII.



Figura 156: Icono de la consola de ejecución de sentencias SQL de pgAdminIII

Y ejecutamos el siguiente código:

```
CREATE DATABASE "BBDDOceanLider"  
  WITH OWNER = postgres  
       ENCODING = 'UTF8'  
       TABLESPACE = pg_default  
       LC_COLLATE = 'Spanish_Spain.1252'  
       LC_CTYPE = 'Spanish_Spain.1252'  
       CONNECTION LIMIT = -1;
```

4.6.3.2. Creación de la estructura de la BBDD de OceanLider con las capas informativas dentro de PostgreSQL

La BBDD de las capas informativas proporcionada por PROES está dividida en tres entidades: Península, Baleares y Canarias. Dentro de estos directorios se encuentra dividida la BBDD en diferentes subapartados dependiendo de la naturaleza de los datos. Cada subapartado contendrá diferentes capas raster o vectoriales. En su mayoría se utilizan capas vectoriales en formato Shapefile [Para más información ver apartado 4.6.3.3. *Poblar la BBDD de OceanLider en PostgreSQL*]

Por tanto, es necesaria hacer la división de la BBDD dentro de PostgreSQL en estas tres entidades. Para hacer esta división se crearán **tres esquemas** dentro de la base de datos nueva. Estos esquemas seguirán la nomenclatura de la división original: Península, Baleares y Canarias. Para crear un nuevo esquema pulsamos el botón derecho del ratón a 'Esquemas' y seleccionamos a "nuevo esquema".

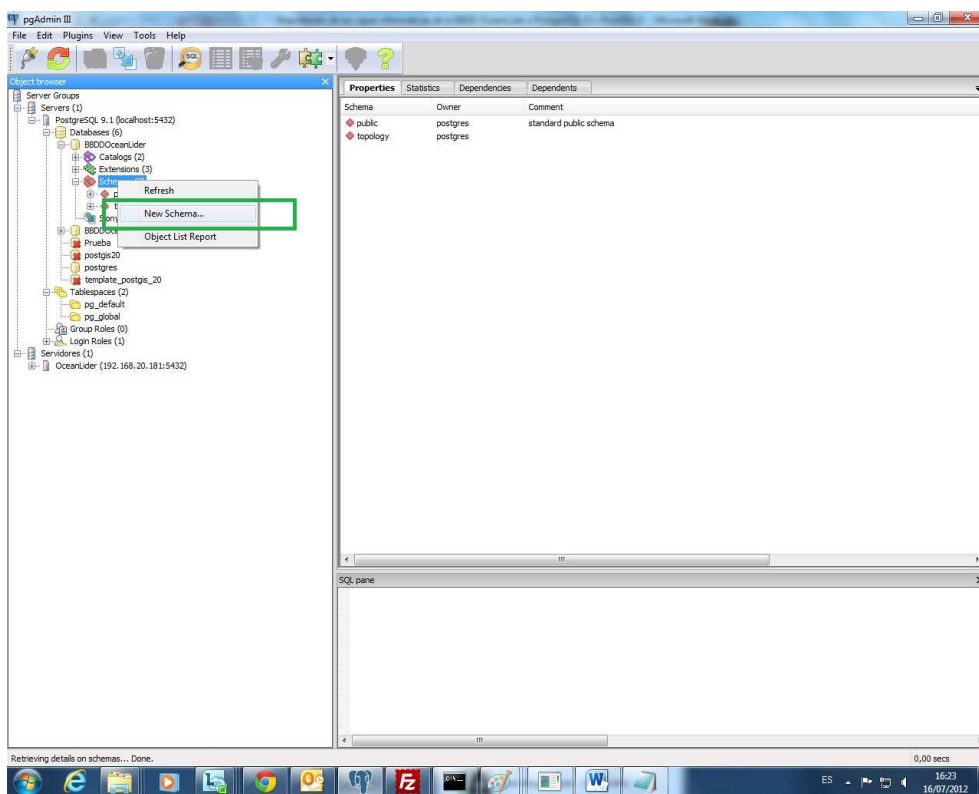


Figura 157: Creación de un nuevo esquema de la base de datos a través de la interfaz de pgAdminIII

Nos aparecerá un cuadro de diálogo en el que tendremos que completar el nombre del esquema y el propietario. Este paso se realiza para las tres entidades: Península, Baleares y Canarias.

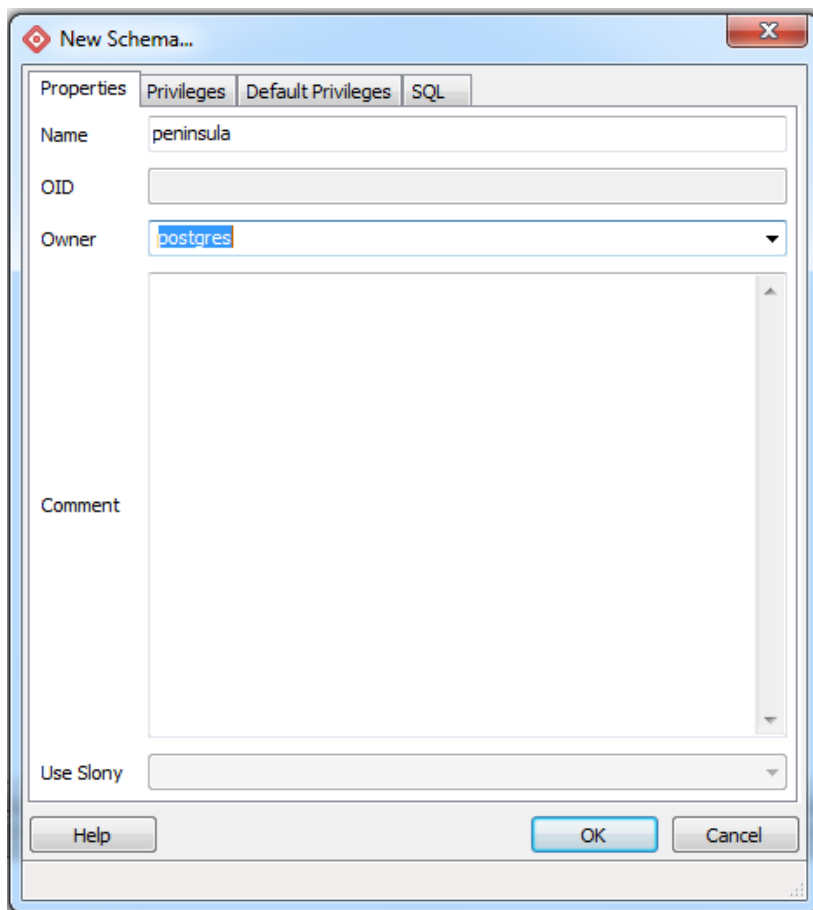


Figura 158: Cuadro de dialogo en la creación de un nuevo esquema de la base de datos utilizando la interfaz gráfica de pgAdminIII

Este paso se puede realizar también a través de la consola de ejecución de sentencias SQL de pgAdminIII ejecutando el siguiente código:

```
CREATE SCHEMA "peninsula"  
  AUTHORIZATION postgres;  
CREATE SCHEMA "balears"  
  AUTHORIZATION postgres;  
CREATE SCHEMA "canarias"  
  AUTHORIZATION postgres;
```

4.6.3.3. Poblar la BBDD de OceanLider en PostgreSQL

Cada esquema contendrá todas las capas informativas que cuelgan de los subdirectorios de cada entidad de la BBDD proporcionada por PROES. Al haber hecho la instalación de PostGIS 2.0 ya

tenemos a nuestra disposición el soporte para capas raster dentro de PostgreSQL (en la versión 1.5 poseía únicamente soporte vectorial y disponible el soporte raster como un paquete adicional). Esta característica brinda la oportunidad de que puedan alojar capas vectoriales y raster dentro de una misma base de datos.

4.6.3.3.1. Importación de las capas vectoriales de una entidad

Para hacer la importación automática de las capas vectoriales de una entidad dentro de PostgreSQL utilizaremos los comandos **shp2pgsql** y **psql**.

Para poder ejecutar estos comandos en cualquier directorio dentro de la consola de Windows, configuramos la variable PATH del sistema con la ruta a los archivos binarios de PostgreSQL:

```
set PATH=%PATH%;C:\Program Files\PostgreSQL\9.1\bin
```

Es importante saber que para utilizar estos comandos en la importación de capas vectoriales dentro de PostgreSQL 9.1+PostGIS 2.0 hay que indicar la codificación original de las capas y el sistema de coordenadas de referencia.

También es importante mencionar que los **esquemas** que se han creado dentro de la BBDD deben crearse con caracteres en **minúscula** para que funcione el siguiente código correctamente. Esquemas en la BBDDOceanLider:

- península
- baleares
- canarias

Los sistemas de coordenadas de referencia para las tres entidades son:

- Península: EPSG: 23030
- Baleares: EPSG: 23031
- Canarias: EPSG: 32628

La codificación utilizada en las capas de origen es LATIN1.

Antes de introducir el código de importación, abrimos una consola de Windows y nos ubicamos en la carpeta padre donde queremos que se despliegue la importación. Es decir, si queremos importar los ficheros Shape de la entidad Península debemos ejecutar la sentencia:

```
cd C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\P
```

Posteriormente ejecutamos el código:

```
for /D /R %x in (*) do (cd %x && for %f in (*.shp) do echo
shp2pgsql -W LATIN1 -s 23030 "%x\%f" peninsula.%~nf ^|psql -U
postgres -d BBDDOceanLider -h localhost -p 5432 >>
C:\Users\GSI\Desktop\sentencias_importacion_peninsula.bat)
```

Este código recorre las subcarpetas que cuelgan del directorio padre de la entidad Península e imprime los comandos de importación para los ficheros Shape que se encuentran ubicados dentro de dichos directorios en un fichero .BAT llamado “*sentencias_importacion_peninsula.bat*” que se creará en la ruta C:\Users\GSI\Desktop

Una vez se haya creado este fichero, ejecutamos la siguiente sentencia en la misma consola de Windows:

```
C:\Users\GSI\Desktop\sentencias_importacion_peninsula.bat
```

Es importante mencionar que hay que introducir la contraseña que se ha asignado al usuario “postgres” a medida que ejecuta las sentencias de importación. Los Shapes importados se localizan en el esquema *peninsula* dentro de PostgreSQL.

Nota: Una vez introducida la contraseña para el primer shape importado, podemos dar a la flecha de subida del teclado y luego a ENTER cada vez que se detenga a consultar de nuevo la contraseña para la importación del siguiente shape. O almacenamos la contraseña de postgres.

Realizamos estos mismos pasos para el resto de entidades: Baleares y Canarias.

Baleares:

```
cd C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\B
for /D /R %x in (*) do (cd %x && for %f in (*shp) do echo
shp2pgsql -W LATIN1 -s 23031 "%x\%f" baleares.%~nf ^|psql -U
postgres -d BBDDOceanLider -h localhost -p 5432 >>
C:\Users\GSI\Desktop\sentencias_importacion_baleares.bat)
C:\Users\GSI\Desktop\sentencias_importacion_baleares.bat
```

Canarias:

```
cd C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\C
for /D /R %x in (*) do (cd %x && for %f in (*shp) do echo
shp2pgsql -W LATIN1 -s 32628 "%x\%f" canarias.%~nf ^|psql -U
postgres -d BBDDOceanLider -h localhost -p 5432 >>
C:\Users\GSI\Desktop\sentencias_importacion_canarias.bat)
C:\Users\GSI\Desktop\sentencias_importacion_canarias.bat
```

ERROR DE CODIFICACIÓN

La codificación que usa Windows al crear los ficheros de importación en SQL es WIN1252 (en el lado del cliente) y es preciso hacer una transformación de WIN1252 a UTF8, que es la codificación de nuestra base de datos (codificación del servidor).

Nota: El comando **shp2pgsql** no permite una conversión de WIN1252 a LATIN1 (no se puede proveer la base de datos con la codificación LATIN1). Es posible que nos aparezca el siguiente error (*ERROR: secuencia de bytes no válida para codificación en UTF8*) con los pasos descritos anteriormente dependiendo de la codificación del sistema.

Vamos a realizar la importación por pasos y visualizamos el código generado por el comando **shp2pgsql** en los archivos SQL de importación por cada capa. Se va a realizar para península, pero hay que aplicarlo para el resto de entidades.

Creamos una carpeta llamada “*importacion_sql*” en el escritorio con la siguiente sentencia:

```
mkdir C:\Users\GSI\Desktop\importacion_sql_peninsula
```

Se va a realizar el ejemplo con la entidad Península. Ejecutamos las siguientes sentencias:

```
cd C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\P
for /D /R %x in (*) do (cd %x && for %f in (*.shp) do echo
shp2pgsql -W LATIN1 -s 23030 "%x\%f" peninsula.%~nf
^>C:\Users\GSI\Desktop\importacion_sql_peninsula\%~nf.sql >>
C:\Users\GSI\Desktop\sentencias_importacion_peninsula.bat)
```

Ejecutamos el .BAT:

```
C:\Users\GSI\Desktop\sentencias_importacion_peninsula.bat
```

En este punto podemos visualizar los archivos SQL que genera el comando **shp2pgsql** en la carpeta *importacion_sql*. Contiene las sentencias SQL de importación de los Shapefiles.

Si quisiéramos completar la importación introduciríamos las siguientes sentencias en la consola de Windows: (pero nos aparecerá el error de codificación)

```
cd C:\Users\GSI\Desktop\importacion_sql_peninsula
for %f in (*.sql) do psql -U postgres -d BBDDOceanLider -h
localhost -p 5432 -f "%f"
```

E introduciríamos la contraseña del usuario “postgres” a medida que se va ejecutando.

Nota a: Realizando este último paso no se solventa la importación de Shapes cuyo nombre contiene acentos o “ñ”.

Nota b: Los nombres introducidos de los Shapes en los esquemas de la BBDD aparecerán en minúsculas.

Si a la hora de importar nos muestra un error de codificación (*ERROR: secuencia de bytes no válida para codificación en UTF8*), ejecutamos los pasos que a continuación se describen. Observamos que el comando **shp2pgsql** genera en la primera línea de cada fichero de importación SQL la sentencia:

```
“SET CLIENT_ENCODING TO UTF8;”
```

Esto produce un error en la importación. Por tanto, mediante la consola de Windows introducimos los siguientes comandos:

```
mkdir C:\Users\GSI\Desktop\importacion_sql_codificacion_peninsula
```



```
cd C:\Users\GSI\Desktop\importacion_sql_peninsula
for %f in (*.sql) do (findstr /v "CLIENT_ENCODING" %~nf.sql >
C:\Users\GSI\Desktop\importacion_sql_codificacion_peninsula\%~nf.
sql)
cd C:\Users\GSI\Desktop\importacion_sql_codificacion_peninsula
for %f in (*.sql) do (psql -U postgres -d BBDDOceanLider -h
localhost -p 5432 -f %~nf.sql)
```

Este código crea una carpeta nueva llamada *importacion_sql_codificacion* donde se albergan los ficheros SQL de importación de los ShapeFiles eliminada la primera línea de codificación. Posteriormente recorre todos los ficheros de importación e importa dichos ShapeFiles a la base de datos llamada BBDDOceanLider en la máquina local.

Por tanto, en la carpeta *importacion_sql* tenemos los ficheros de importación originales generados por el comando **shp2pgs** y en la carpeta *importacion_sql_codificacion* tenemos los ficheros de importación sin la primera línea de codificación.

Los pasos para baleares y Canarias son:

Baleares

```
mkdir C:\Users\GSI\Desktop\importacion_sql_baleares
cd C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\B
for /D /R %x in (*) do (cd %x && for %f in (*.shp) do echo
shp2pgs -W LATIN1 -s 23031 "%x\%f" baleares.%~nf
^>C:\Users\GSI\Desktop\importacion_sql_baleares\%~nf.sql >>
C:\Users\GSI\Desktop\sentencias_importacion_baleares.bat)
C:\Users\GSI\Desktop\sentencias_importacion_baleares.bat
mkdir C:\Users\GSI\Desktop\importacion_sql_codificacion_baleares
cd C:\Users\GSI\Desktop\importacion_sql_baleares
for %f in (*.sql) do (findstr /v "CLIENT_ENCODING" %~nf.sql >
C:\Users\GSI\Desktop\importacion_sql_codificacion_baleares\%~nf.s
ql)
cd C:\Users\GSI\Desktop\importacion_sql_codificacion_baleares
for %f in (*.sql) do (psql -U postgres -d BBDDOceanLider -h
localhost -p 5432 -f %~nf.sql)
```

Canarias

```
mkdir C:\Users\GSI\Desktop\importacion_sql_canarias
cd C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\C
for /D /R %x in (*) do (cd %x && for %f in (*.shp) do echo
shp2pgs -W LATIN1 -s 32628 "%x\%f" canarias.%~nf
^>C:\Users\GSI\Desktop\importacion_sql_canarias\%~nf.sql >>
C:\Users\GSI\Desktop\sentencias_importacion_canarias.bat)
C:\Users\GSI\Desktop\sentencias_importacion_canarias.bat
mkdir C:\Users\GSI\Desktop\importacion_sql_codificacion_canarias
cd C:\Users\GSI\Desktop\importacion_sql_canarias
```

```
for %f in (*.sql) do (findstr /v "CLIENT_ENCODING" %~nf.sql >
C:\Users\GSI\Desktop\importacion_sql_codificacion_canarias\%~nf.s
ql)
cd C:\Users\GSI\Desktop\importacion_sql_codificacion_canarias
for %f in (*.sql) do (psql -U postgres -d BBDOceanLider -h
localhost -p 5432 -f %~nf.sql)
```

4.6.3.3.1.1. Problemática y resolución en la importación de Shapefiles con acentos y caracteres especiales en el nombre

Se ha detectado un error en la importación de los Shapefiles que contienen acentos o caracteres especiales en el nombre original.

En el caso que queramos mantener el nombre original de los ficheros y/o hacer la importación de los Shapes cuyo nombre contiene acentos, realizaremos la importación a través de **QuantumGIS** [como explica en el apartado 6.3.3.2. *Importar capas vectoriales*]. Realizando la importación a través de esta herramienta podemos conservar el nombre original de los esquemas: Península, Baleares y Canarias. Se realiza la importación capa a capa.

Los Shapes que contienen acentos en el nombre son:

- En la entidad **Península**:
 - Batimetría_P.shp
 - ZBaño_P.shp
- En la entidad **Baleares**:
 - Batimetría_B.shp
 - ZBaño_B.shp
- En la entidad **Canarias**:
 - Batimetría_Can_WGS84H28.shp
 - ZBaño_Can_WGS84H28.shp

Para realizar la importación completa de todos los shapes de la BBDD proporcionada por PROES, importamos estas capas con QuantumGIS:

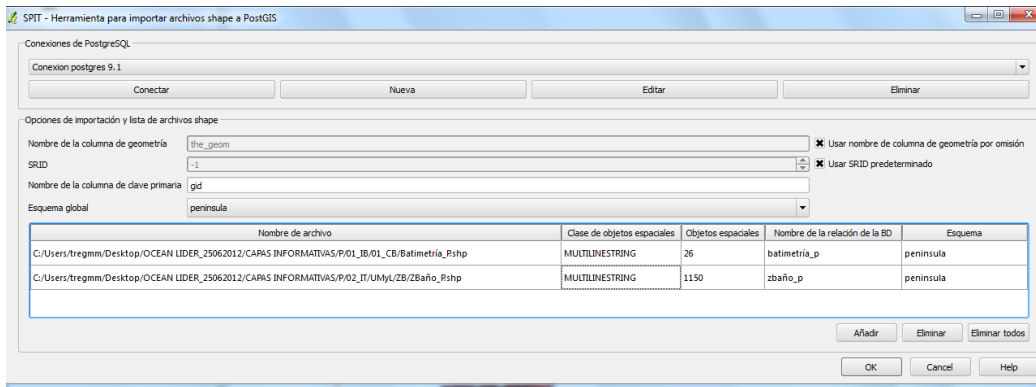


Figura 159: Cuadro de dialogo de QuantumGIS para la importación de Shapes a PostgreSQL

Se puede hacer la importación de forma múltiple indicando las rutas de los Shapes en el cuadro de dialogo una a una. Se ha especificado el esquema donde se importarán las capas (en este caso al esquema “*península*”) y se ha modificado el nombre que tendrá el Shape en PostgreSQL a minúsculas para conservar la misma nomenclatura que el resto de las capas importadas con el código anterior.

Realizamos este mismo paso para las entidades Baleares y Canarias.

Con estas capas conseguimos un error en la importación:

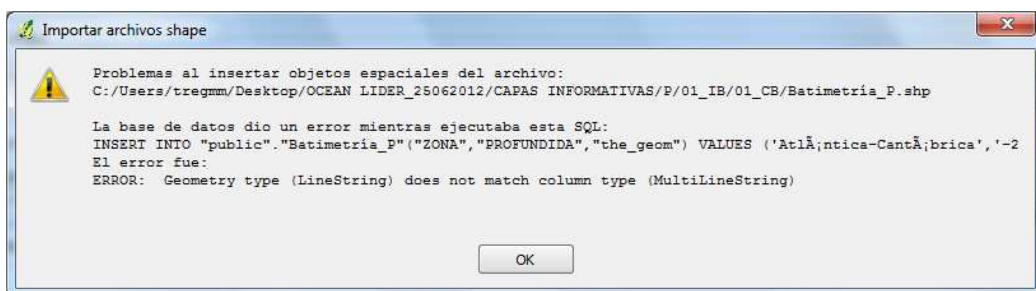


Figura 160: Error de QGIS en la importación de shape con objetos espaciales tipo **MULTILINESTRING**

Como no se ha solventado la problemática de importar los Shapes con acentos y “ñ” en el nombre, renombramos a mano estas capas y hacemos la importación con la sentencia que se indica más abajo ejecutando en la ventana de símbolos del sistema de Windows.

Por ejemplo, para renombrar el Shapefile de **Batimetría_P** (ubicada en C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\P\01_IB\01_CB),

recordamos que hay que renombrar cualquier fichero asociado al mismo, es decir, renombramos sin acento a:

```
Batimetria_P.dbf
Batimetria_P.prj
Batimetria_P.sbn
Batimetria_P.sbx
Batimetria_P.shp
Batimetria_P.shp.xml
Batimetria_P.shx
```

Y a continuación en la consola de Windows introducimos la siguiente sentencia:

```
shp2pgsql -W LATIN1 -s 23030 "C:\Users\GSI\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\P\01_IB\01_CB\Batimetria_P.shp"
peninsula.Batimetria_P |psql -U postgres -d BBDDOceanLider -h
localhost -p 5432
```

De forma similar se hará para el resto de los shaps.

Para **ZBaño_P** (ubicada en C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\P\02_IT\UMyL\ZB), renombramos:

```
ZBano_P.dbf
ZBano_P.prj
ZBano_P.sbn
ZBano_P.sbx
ZBano_P.shp
ZBano_P.shp.xml
ZBano_P.shx
```

Ejecutamos en CMD:

```
shp2pgsql -W LATIN1 -s 23030 "C:\Users\GSI\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\P\02_IT\UMyL\ZB\ZBano_P.shp"
peninsula.ZBano_P |psql -U postgres -d BBDDOceanLider -h
localhost -p 5432
```

Para **Batimetría_B** (ubicada en C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\B\01_IB\01_CB), renombramos:

```
Batimetria_B.dbf
Batimetria_B.prj
Batimetria_B.sbn
Batimetria_B.sbx
Batimetria_B.shp
Batimetria_B.shp.xml
Batimetria_B.shx
```

Ejecutamos en CMD:

```
shp2pgsql -W LATIN1 -s 23031 "C:\Users\GSI\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\B\01_IB\01_CB\Batimetria_B.shp"
baleares.Batimetria_B |psql -U postgres -d BBDDOceanLider -h
localhost -p 5432
```

Para **ZBaño_B** (ubicada en C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\B\02_IT\UMyL\ZB), renombramos:

```
ZBano_B.dbf
ZBano_B.prj
ZBano_B.sbn
ZBano_B.sbx
ZBano_B.shp
ZBano_B.shp.xml
ZBano_B.shx
```

Ejecutamos en CMD:

```
shp2pgsql -W LATIN1 -s 23031 "C:\Users\GSI\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\B\02_IT\UMyL\ZB\ZBano_B.shp"
baleares.ZBano_B |psql -U postgres -d BBDDOceanLider -h localhost
-p 5432
```

Para **Batimetría_Can_WGS84H28** (ubicada en C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\C\01_IB\01_CB), renombramos:

```
Batimetria_Can_WGS84H28.dbf
Batimetria_Can_WGS84H28.prj
Batimetria_Can_WGS84H28.sbn
Batimetria_Can_WGS84H28.sbx
Batimetria_Can_WGS84H28.shp
Batimetria_Can_WGS84H28.shp.xml
Batimetria_Can_WGS84H28.shx
```

Ejecutamos en CMD:

```
shp2pgsql -W LATIN1 -s 32628 "C:\Users\GSI\Desktop\OCEAN
LIDER_25062012\CAPAS
INFORMATIVAS\C\01_IB\01_CB\Batimetria_Can_WGS84H28.shp"
canarias.Batimetria_Can_WGS84H28 |psql -U postgres -d
BBDDOceanLider -h localhost -p 5432
```

Para **ZBaño_Can_WGS84H28** (ubicada en C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\C\02_IT\UMyL\ZB), renombramos:

```
ZBano_Can_WGS84H28.dbf
ZBano_Can_WGS84H28.prj
```

```
ZBano_Can_WGS84H28.sbn
ZBano_Can_WGS84H28.sbx
ZBano_Can_WGS84H28.shp
ZBano_Can_WGS84H28.shp.xml
ZBano_Can_WGS84H28.shx
```

Ejecutamos en CMD:

```
shp2pgsql -W LATIN1 -s 32628 "C:\Users\GSI\Desktop\OCEAN
LIDER_25062012\CAPAS
INFORMATIVAS\C\02_IT\UMyL\ZB\ZBano_Can_WGS84H28.shp"
canarias.ZBano_Can_WGS84H28 |psql -U postgres -d BBDDOceanLider -
h localhost -p 5432
```

4.6.3.3.2. Importación de las capas raster de una entidad

De forma similar que en apartado para la importación de capas vectoriales, ejecutamos el siguiente código en la consola de símbolos del sistema de Windows para las capas raster:

Para la entidad **Península**:

```
cd C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\P
for /D /R %x in (*) do (cd %x && for %f in (*.img *.tif) do echo
raster2pgsql -s 23030 -I -C -M "%x\%f" -F peninsula.%~nf ^| psql
-U postgres -d BBDDOceanLider -h localhost -p 5432 >>
C:\Users\GSI\Desktop\sentencias_importacion_peninsula_raster.bat)
C:\Users\GSI\Desktop\sentencias_importacion_peninsula_raster.bat
```

Para la entidad **Baleares**:

```
cd C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\B
for /D /R %x in (*) do (cd %x && for %f in (*.img *.tif) do echo
raster2pgsql -s 23031 -I -C -M "%x\%f" -F baleares.%~nf ^| psql -
U postgres -d BBDDOceanLider -h localhost -p 5432 >>
C:\Users\GSI\Desktop\sentencias_importacion_baleares_raster.bat)
C:\Users\GSI\Desktop\sentencias_importacion_baleares_raster.bat
```

Para la entidad **Canarias**:

```
cd C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\C
for /D /R %x in (*) do (cd %x && for %f in (*.img *.tif) do echo
raster2pgsql -s 32628 -I -C -M "%x\%f" -F canarias.%~nf ^| psql -
U postgres -d BBDDOceanLider -h localhost -p 5432 >>
C:\Users\GSI\Desktop\sentencias_importacion_canarias_raster.bat)
C:\Users\GSI\Desktop\sentencias_importacion_canarias_raster.bat
```

4.6.3.4. Comprobación de resultados

4.6.3.4.1. Resumen de las capas vectoriales y raster de la BBDD OceanLider proporcionada por PROES

Es importante hacer un resumen de los Shapes y capas raster que contiene la BBDD proporcionada por PROES para contrarrestar con las capas importadas en PostgreSQL. A continuación se muestran los resultados de este resumen.

4.6.3.4.1.1. Península

```
cd C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\C
for /D /R %x in (*) do (cd %x && for %f in (*.shp) do echo %f)>
C:\Users\GSI\Desktop\shapes_importar_peninsula.txt)
```

Shapes importados en el esquema <i>peninsula</i>	
Batimetria_P.shp	ZEPIMs_P.shp
Lin_Costa_P.shp	IEZH_P.shp
Aguas_Costeras_P.shp	PPNNs_P.shp
CCAA_P.shp	Bentos_punt_P.shp
DDHH_P.shp	Biocenosis_CVa_P.shp
DDMM_P.shp	Biocenosis_Eco_P.shp
Deslinde_P.shp	bionomia_Mur_P.shp
Deslinde_point_P.shp	Habitat_marinos_P.shp
Linea_Base_P.shp	Habitats_N-NW_P.shp
Mar_territorial_P.shp	Habitats_SW_Med_P.shp
500Lin_Costa_P.shp	Habitat_mar_N-NW.shp
Fondos_P.shp	Habitat_mar_P.shp
Formas_line_P.shp	Habitat_mar_SW_Med.shp
Formas_P.shp	Cetaceos_P.shp
Geofis_P.shp	AToN_P.shp
Texturas_fondo_P.shp	DPP_PE_P.shp
SAL_P.shp	Ext_aridos_P.shp
Patrim_P.shp	Ext_aridos_point_P.shp
Patrim_point_P.shp	OZE_P.shp
AEIV_P.shp	ProtecConducciones_P.shp
aicp_P.shp	Vertido_dragado_P.shp
aopn_P.shp	presiones_P.shp
IBAs_P.shp	RMO_P.shp
iba_recom_P.shp	DST_P.shp
ibm_P.shp	Areas_eolicas_P.shp
ZMES_P.shp	Conducciones_P.shp
AIN_DOT_P.shp	conexiones_red_P.shp
BBPPro_P.shp	EEAL_P.shp
EENN_P.shp	Vertidos_P.shp
Micror_P.shp	Vertido_material_militar_P.shp
MMNat_P.shp	ZONA_CABLES_P.shp

MMNat_point_P.shp PaiPPro_P.shp PEIN_P.shp PEP_P.shp PParNNat_P.shp PPNat_P.shp PPPPer_P.shp PPRR_P.shp Refugio_caza_P.shp RRF_P.shp RRNN_P.shp RR_ley_prop_P.shp ZEPVN_P.shp ZIAE_P.shp ZZHH_P.shp DipEuro_P.shp LICs_P.shp ZEPAs.shp GeoPP_P.shp MABs_P.shp OSPAR_AMPs_P.shp P_Humanidad_P.shp RAMSARs_P.shp	Militar_P.shp Acuicultura_P.shp Acuicultura_point_P.shp ALMADRABAS_p.shp Almadrabas_point_P.shp Arrecifes_P.shp Arrecifes_point_P.shp Caladeros_P.shp Cria_moluscos_lin_P.shp Cria_moluscos_P.shp Piscifactoria_P.shp Piscifactoria_point_P.shp RRMM_P.shp Viveros_P.shp Viveros_point_P.shp ZPIPs_P.shp ZBano_P.shp
--	--

Un TOTAL de: 102 shapes

```
cd C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\C
for /D /R %x in (*) do (cd %x && for %f in (*.img *.tif) do echo
%f)>> C:\Users\GSI\Desktop\raster_importar_peninsula.txt)
```

Capas raster (formato IMG o TIF) importados en el esquema <i>peninsula</i>	
FLUORS_P.tif OD_P.tif SAL_P.tif TEMP_P.tif TURB_P.tif maxmuecant11.img maxmuemed1.img maxvivcant.img maxvivmed.img medmuecant1.img medmuemed1.img medvivcant.img medvivmed.img HsMediaRaster.img HsMediaRaster_1.img HsMediaRaster_10.img	PwDEF_Can.img PwDEF_Med.img PwJJARaster.img PwJJARaster_1.img PwJJARaster_10.img PwJJARaster_11.img PwJJARaster_12.img PwJJARaster_13.img PwJJARaster_14.img PwJJARaster_15.img PwJJARaster_16.img PwJJARaster_17.img PwJJARaster_18.img PwJJARaster_19.img PwJJARaster_20.img

HsMediaRaster_11.img	PwJJARaster_21.img
HsMediaRaster_12.img	PwJJARaster_22.img
HsMediaRaster_13.img	PwJJARaster_23.img
HsMediaRaster_14.img	PwJJARaster_24.img
HsMediaRaster_15.img	PwJJARaster_25.img
HsMediaRaster_16.img	PwJJARaster_26.img
HsMediaRaster_17.img	PwJJARaster_27.img
HsMediaRaster_18.img	PwJJARaster_28.img
HsMediaRaster_19.img	PwJJARaster_29.img
HsMediaRaster_2.img	PwJJARaster_30.img
HsMediaRaster_20.img	PwJJARaster_31.img
HsMediaRaster_21.img	PwJJARaster_32.img
HsMediaRaster_22.img	PwJJARaster_33.img
HsMediaRaster_23.img	PwJJARaster_4.img
HsMediaRaster_24.img	PwJJARaster_5.img
HsMediaRaster_25.img	PwJJARaster_6.img
HsMediaRaster_26.img	PwJJARaster_7.img
HsMediaRaster_27.img	PwJJARaster_8.img
HsMediaRaster_28.img	PwJJARaster_9.img
HsMediaRaster_29.img	PwMAMRaster.img
HsMediaRaster_30.img	PwMAMRaster_0.img
HsMediaRaster_31.img	PwMAMRaster_1.img
HsMediaRaster_32.img	PwMAMRaster_10.img
HsMediaRaster_33.img	PwMAMRaster_11.img
HsMediaRaster_34.img	PwMAMRaster_12.img
HsMediaRaster_35.img	PwMAMRaster_13.img
HsMediaRaster_4.img	PwMAMRaster_14.img
HsMediaRaster_5.img	PwMAMRaster_15.img
HsMediaRaster_6.img	PwMAMRaster_16.img
HsMediaRaster_7.img	PwMAMRaster_17.img
HsMediaRaster_8.img	PwMAMRaster_18.img
HsMediaRaster_9.img	PwMAMRaster_19.img
PwDEFRaster.img	PwMAMRaster_20.img
PwDEFRaster_0.img	PwMAMRaster_21.img
PwDEFRaster_1.img	PwMAMRaster_22.img
PwDEFRaster_10.img	PwMAMRaster_23.img
PwDEFRaster_11.img	PwMAMRaster_24.img
PwDEFRaster_12.img	PwMAMRaster_25.img
PwDEFRaster_13.img	PwMAMRaster_26.img
PwDEFRaster_14.img	PwMAMRaster_27.img
PwDEFRaster_15.img	PwMAMRaster_28.img
PwDEFRaster_16.img	PwMAMRaster_29.img
PwDEFRaster_17.img	PwMAMRaster_30.img
PwDEFRaster_18.img	PwMAMRaster_31.img
PwDEFRaster_19.img	PwMAMRaster_32.img
PwDEFRaster_20.img	

PwDEFRaster_21.img	PwMAMRaster_33.img
PwDEFRaster_22.img	PwMAMRaster_4.img
PwDEFRaster_23.img	PwMAMRaster_5.img
PwDEFRaster_24.img	PwMAMRaster_6.img
PwDEFRaster_25.img	PwMAMRaster_7.img
PwDEFRaster_26.img	PwMAMRaster_8.img
PwDEFRaster_27.img	PwMAMRaster_9.img
PwDEFRaster_28.img	PwMediaRaster.img
PwDEFRaster_29.img	PwMediaRaster_0.img
PwDEFRaster_30.img	PwMediaRaster_1.img
PwDEFRaster_31.img	PwMediaRaster_10.img
PwDEFRaster_32.img	PwMediaRaster_11.img
PwDEFRaster_33.img	PwMediaRaster_12.img
PwDEFRaster_34.img	PwMediaRaster_13.img
PwDEFRaster_4.img	PwMediaRaster_14.img
PwDEFRaster_5.img	PwMediaRaster_15.img
PwDEFRaster_6.img	PwMediaRaster_16.img
PwDEFRaster_7.img	PwSONRaster.img
PwDEFRaster_8.img	PwSONRaster_0.img
PwDEFRaster_9.img	PwSONRaster_1.img
PwMediaRaster_17.img	PwSONRaster_10.img
PwMediaRaster_18.img	PwSONRaster_11.img
PwMediaRaster_19.img	PwSONRaster_12.img
PwMediaRaster_20.img	PwSONRaster_13.img
PwMediaRaster_21.img	PwSONRaster_14.img
PwMediaRaster_22.img	PwSONRaster_15.img
PwMediaRaster_23.img	PwSONRaster_16.img
PwMediaRaster_24.img	PwSONRaster_17.img
PwMediaRaster_25.img	PwSONRaster_18.img
PwMediaRaster_26.img	PwSONRaster_19.img
PwMediaRaster_27.img	PwSONRaster_20.img
PwMediaRaster_28.img	PwSONRaster_21.img
PwMediaRaster_29.img	PwSONRaster_22.img
PwMediaRaster_30.img	PwSONRaster_23.img
PwMediaRaster_31.img	PwSONRaster_24.img
PwMediaRaster_32.img	PwSONRaster_25.img
PwMediaRaster_33.img	PwSONRaster_26.img
PwMediaRaster_4.img	PwSONRaster_27.img
PwMediaRaster_5.img	PwSONRaster_28.img
PwMediaRaster_6.img	PwSONRaster_29.img
PwMediaRaster_7.img	PwSONRaster_30.img
PwMediaRaster_8.img	PwSONRaster_31.img
PwMediaRaster_9.img	PwSONRaster_32.img
PwMEDIA_Can.img	PwSONRaster_33.img
PwMEDIA_Med.img	PwSONRaster_4.img
	PwSONRaster_5.img

	PwSONRaster_6.img PwSONRaster_7.img PwSONRaster_8.img PwSONRaster_9.img
--	--

Un TOTAL de: 228 capas raster importadas

4.6.3.4.1.2. Baleares

```
cd C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\B
for /D /R %x in (*) do (cd %x && for %f in (*.shp) do echo %f>>
C:\Users\GSI\Desktop\shapes_importar_baleares.txt)
```

Shapes importados en el esquema <i>baleares</i>	
Batimetria_B.shp	Patrim_Mundial_B.shp
Lin_Costa_B.shp	RAMSARs_B.shp
Aguas_costeras_B.shp	ZEPIMs_B.shp
CCAA_B.shp	PPNNs_B.shp
DDHH_B.shp	Fanerog_B.shp
DDMM_B.shp	Habitats_B.shp
Deslinde_B.shp	Habitat_fanerog_B.shp
Linea_base_B.shp	Cetaceo_B.shp
Mar_territorial_B.shp	AToN_B.shp
500Lin_Costa_B.shp	DPP_PE_B.shp
Geofis_B.shp	Ext_aridos_B.shp
Patrim_B.shp	OZE_B.shp
Patrim_point_B.shp	Zonas_fondeo_B.shp
AEIV_B.shp	Zonas_Prohibidas_B.shp
aicp_B.shp	presiones_B.shp
aopn_B.shp	RMO_B.shp
Atun_B.shp	Areas_eolicas_B.shp
IBAs_B.shp	Conducciones_lineales_B.shp
iba_recom_B.shp	EEAL_B.shp
ibm_B.shp	Vertidos_B.shp
AP_PTI_B.shp	ZONA_CABLES_B.shp
LICcientifico_B.shp	Acuicultura_B.shp
MMNat_B.shp	Arrecifes_B.shp
ParaPro_B.shp	Cria_moluscos_B.shp
PPNat_B.shp	RRMM_B.shp
RRNN_B.shp	Viveros_refug_B.shp
LICs_B.shp	ZBano_B.shp
ZEPAs_B.shp	
MABs_B.shp	

Un TOTAL de: 56 shapes

```
cd C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\B
```

```
for /D /R %x in (*) do (cd %x && for %f in (*.img *.tif) do echo %f)>> C:\Users\GSI\Desktop\raster_importar_baleares.txt)
```

Capas raster (formato IMG o TIF) importados en el esquema <i>baleares</i>	
maxmuemed2.img	PwMAMRaster.img
maxvivmed.img	PwMAMRaster_1.img
medmuemed1.img	PwMAMRaster_2.img
medvivmed.img	PwMAMRaster_3.img
HsMediaRaster.img	PwMAM_B.img
HsMediaRaster_1.img	PwMediaRaster.img
HsMediaRaster_2.img	PwMediaRaster_1.img
HsMediaRaster_3.img	PwMediaRaster_2.img
pwdefraster.img	PwMediaRaster_3.img
pwdefraster_0.img	PwMEDIA_B.img
pwdefraster_1.img	PwSONRaster.img
pwdefraster_2.img	PwSONRaster_1.img
pwdefraster_3.img	PwSONRaster_2.img
PwDEF_B.img	PwSONRaster_3.img
PwJJARaster.img	PwSON_B.img
PwJJARaster_1.img	
PwJJARaster_2.img	
PwJJARaster_3.img	
PwJJA_B.img	

Un TOTAL de: 34 capas raster

4.6.3.4.1.3. Canarias

```
cd C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\C
for /D /R %x in (*) do (cd %x && for %f in (*.shp) do echo %f)>>
C:\Users\GSI\Desktop\shapes_importar_canarias.txt)
```

Shapes importados en el esquema <i>canarias</i>	
Batimetria_Can_WGS84H28.shp	RRNN_Can_WGS84H28.shp
Lin_Costa_Can_WGS84H28.shp	LICs_Can_WGS84H28.shp
Toponimia_Can_WGS84H28.shp	ZEPAs_Can_WGS84H28.shp
Toponimia_costa_Can_WGS84H28.shp	MABs_Can_WGS84H28.shp
Aguas_costeras_Can_WGS84H28.shp	RAMSARs_Can_WGS84H28.shp
CCAA_Can_WGS84H28.shp	PPNNs_Can_WGS84H28.shp
DDMM_Can_WGS84H28.shp	Bentos_Can_WGS84H28.shp
Linea_Base_Can_WGS84H28.shp	Habitats_Can_WGS84H281.shp
Mar_territorial_Can_WGS84H28.shp	AToN_Can_WGS84H28.shp
ZMT_Can_WGS84H28.shp	DPP_Canarias_WGS84H28.shp
500Lin_Costa_Can_WGS84.shp	Ext_aridos_Can_WGS84H28.shp
Fondos_Can_WGS84H28.shp	OZE_Can_WGS84H28.shp
Formas_Can_WGS84H28.shp	presiones_Can_WGS84H28.shp
Formas_line_Can_WGS84H28.shp	RMO_Can_WGS84H28.shp
Geofis_Can_WGS84H28.shp	DST_ZE_Can_WGS84H28.shp

FLUORS_Can_WGS84H28.shp OD_Can_WGS84H28.shp SAL_Can_WGS84H28.shp TEMP_Can_WGS84H28.shp TURB_Can_WGS84H28.shp Patrim_Can_point_WGS84H28.shp Patrim_Can_WGS84H28.shp aopn_Can_WGS84H28.shp IBAs_Can_WGS84H28.shp iba_recom_Can_WGS84H28.shp ibm_Can_WGS84H28.shp ZMES_Can_WGS84H28.shp MMNat_Can_WGS84H28.shp Otros_Can_WGS84H28.shp PaiPro_Can_WGS84H28.shp PPNat_Can_WGS84H28.shp PPRur_Can_WGS84H28.shp	Areas_eolicas_Can_WGS84H28.shp Conducciones_Line_Can_WGS84H28.shp EEAL_Can_WGS84H28.shp ProtecConducciones_Can_WGS84H28.shp Vertidos_Can_WGS84H28.shp Militar_Can_WGS84H28.shp Acuicultura_Can_WGS84.shp Arrecifes_Can_WGS84H28.shp Piscifactoria_Can_WGS84H28.shp RRMM_Can_WGS84H28.shp Tipo_pesca_Can_WGS84H28.shp ZBano_Can_WGS84H28.shp
--	--

Un **TOTAL** de: **59** shapes

```
cd C:\Users\GSI\Desktop\OCEAN LIDER_25062012\CAPAS INFORMATIVAS\C
for /D /R %x in (*) do (cd %x && for %f in (*.img *.tif) do echo
%f>> C:\Users\GSI\Desktop\raster_importar_canarias.txt)
```

Capas raster (formato IMG o TIF) importados en el esquema canarias	
maxmuecan1.img	PwMEDIA_C.img
maxvivcan.img	PwMedia_M091.img
medmuecan1.img	PwMedia_M092.img
medvivcan.img	PwMedia_M093.img
HsMedia_M091.img	PwMedia_M094.img
HsMedia_M092.img	PwMedia_M095.img
HsMedia_M093.img	PwMedia_M101.img
HsMedia_M094.img	PwMedia_M102.img
HsMedia_M095.img	PwMedia_M103.img
HsMedia_M101.img	PwMedia_M104.img
HsMedia_M102.img	PwMedia_M105.img
HsMedia_M103.img	PwMedia_M106.img
HsMedia_M104.img	PwMedia_M111.img
HsMedia_M105.img	PwMedia_M112.img
HsMedia_M106.img	PwMedia_M113.img
HsMedia_M111.img	PwMedia_M114.img
HsMedia_M112.img	PwMedia_M121.img
HsMedia_M113.img	PwMedia_M122.img
HsMedia_M114.img	PwMedia_M123.img
HsMedia_M121.img	PwMedia_M124.img
HsMedia_M122.img	PwMedia_M125.img
HsMedia_M123.img	PwMedia_M126.img
HsMedia_M124.img	PwMedia_M127.img

HsMedia_M125.img	PwMedia_M128.img
HsMedia_M126.img	PwMedia_M131.img
HsMedia_M127.img	PwMedia_M132.img
HsMedia_M128.img	PwMedia_M141.img
HsMedia_M131.img	PwMAM_C.img
HsMedia_M132.img	PwMAM_M091.img
HsMedia_M141.img	PwMAM_M092.img
PwDEF_C.img	PwMAM_M093.img
PwDEF_M091.img	PwMAM_M094.img
PwDEF_M092.img	PwMAM_M095.img
PwDEF_M093.img	PwMAM_M101.img
PwDEF_M094.img	PwMAM_M102.img
PwDEF_M095.img	PwMAM_M103.img
PwDEF_M101.img	PwMAM_M104.img
PwDEF_M102.img	PwMAM_M105.img
PwDEF_M103.img	PwMAM_M106.img
PwDEF_M104.img	PwMAM_M111.img
PwDEF_M105.img	PwMAM_M112.img
PwDEF_M106.img	PwMAM_M113.img
PwDEF_M111.img	PwMAM_M114.img
PwDEF_M112.img	PwMAM_M121.img
PwDEF_M113.img	PwMAM_M122.img
PwDEF_M114.img	PwMAM_M123.img
PwDEF_M121.img	PwSON_C.img
PwDEF_M122.img	PwSON_M091.img
PwDEF_M123.img	PwSON_M092.img
PwDEF_M124.img	PwSON_M093.img
PwDEF_M125.img	PwSON_M094.img
PwDEF_M126.img	PwSON_M095.img
PwDEF_M127.img	PwSON_M101.img
PwDEF_M128.img	PwSON_M102.img
PwDEF_M131.img	PwSON_M103.img
PwDEF_M132.img	PwSON_M104.img
PwDEF_M141.img	PwSON_M105.img
PwJJA_C.img	PwSON_M106.img
PwJJA_M091.img	PwSON_M111.img
PwJJA_M092.img	PwSON_M112.img
PwJJA_M093.img	PwSON_M113.img
PwJJA_M094.img	PwSON_M114.img
PwJJA_M095.img	PwSON_M121.img
PwJJA_M101.img	PwSON_M122.img
PwJJA_M102.img	PwSON_M123.img
PwJJA_M103.img	PwSON_M124.img
PwJJA_M104.img	PwSON_M125.img
PwJJA_M105.img	PwSON_M126.img
PwJJA_M106.img	PwSON_M127.img
PwJJA_M111.img	PwSON_M128.img
PwJJA_M112.img	PwSON_M131.img

PwJJA_M113.img PwJJA_M114.img PwJJA_M121.img PwJJA_M122.img PwJJA_M123.img PwJJA_M124.img PwJJA_M125.img PwJJA_M126.img PwJJA_M127.img PwJJA_M128.img PwJJA_M131.img PwJJA_M132.img PwJJA_M141.img PwMAM_M124.img PwMAM_M125.img PwMAM_M126.img PwMAM_M127.img PwMAM_M128.img PwMAM_M131.img PwMAM_M132.img PwMAM_M141.img	PwSON_M132.img PwSON_M141.img
--	----------------------------------

Un **TOTAL** de: 165 capas raster

4.6.3.4.2. Capas vectoriales y raster en la BBDDOceanLider ubicadas en PostgreSQL tras la importación

Puede ocurrir que a la hora de hacer la importación de las capas a PostgreSQL no se haya realizado correctamente debido a un error humano (mala introducción de la contraseña) o un error computacional (por ejemplo, para las capas raster puede que nos quedemos sin memoria en caso que el raster tenga un gran tamaño). Para comprobar que todas las capas han sido importadas correctamente, podemos realizar las siguientes consultas SQL en el ejecutor de consultas SQL dentro de pgAdminIII.

4.6.3.4.2.1. Comprobación de Península

Para ver las tablas contenidas en el esquema 'peninsula' en PostgreSQL:

```
SELECT tablename FROM pg_tables WHERE schemaname = 'peninsula';
```

Capas que se tenían que importar	Capas importadas en PostgreSQL
500Lin_Costa_P.shp	500lin_costa_p
Acuicultura_P.shp	acuicultura_p
Acuicultura_point_P.shp	acuicultura_point_p
AEIV_P.shp	aeiv_p
Aguas_Costeras_P.shp	aguas_costeras_p
aicp_P.shp	aicp_p
AIN_DOT_P.shp	ain_dot_p
ALMADRABAS_p.shp	almadrabas_p
Almadrabas_point_P.shp	almadrabas_point_p
aopn_P.shp	aopn_p
Areas_eolicas_P.shp	areas_eolicas_p
Arrecifes_P.shp	arrecifes_p
Arrecifes_point_P.shp	arrecifes_point_p
AToN_P.shp	aton_p
Batimetria_P.shp	batimetria_p
BBPPro_P.shp	bbppro_p
Bentos_punt_P.shp	bentos_punt_p
Biocenosis_CVa_P.shp	biocenosis_cva_p
Biocenosis_Eco_P.shp	biocenosis_eco_p
bionomia_Mur_P.shp	bionomia_mur_p
Caladeros_P.shp	caladeros_p
CCAA_P.shp	ccaa_p
Cetaceos_P.shp	cetaceos_p
Conducciones_P.shp	conducciones_p
conexiones_red_P.shp	conexiones_red_p
Cria_moluscos_lin_P.shp	cria_moluscos_lin_p
Cria_moluscos_P.shp	cria_moluscos_p
DDHH_P.shp	ddhh_p
DDMM_P.shp	ddmm_p
Deslinde_P.shp	deslinde_p
Deslinde_point_P.shp	deslinde_point_p
DipEuro_P.shp	dipeuro_p
DPP_PE_P.shp	dpp_pe_p
DST_P.shp	dst_p
EEAL_P.shp	eeal_p

EENN_P.shp	eenn_p
Ext_aridos_P.shp	ext_aridos_p
Ext_aridos_point_P.shp	ext_aridos_point_p
FLUORS_P.tif	
Fondos_P.shp	fondos_p
Formas_line_P.shp	formas_line_p
Formas_P.shp	formas_p
Geofis_P.shp	geofis_p
GeoPP_P.shp	geopp_p
Habitat_mar_N-NW.shp	habitat_mar_n-nw
Habitat_mar_P.shp	habitat_mar_p
Habitat_mar_SW_Med.shp	habitat_mar_sw_med
Habitat_marinos_P.shp	habitat_marinos_p
Habitats_N-NW_P.shp	habitats_n-nw_p
Habitats_SW_Med_P.shp	habitats_sw_med_p
HsMediaRaster.img	hsmediaraster
HsMediaRaster_1.img	hsmediaraster_1
HsMediaRaster_10.img	hsmediaraster_10
HsMediaRaster_11.img	hsmediaraster_11
HsMediaRaster_12.img	hsmediaraster_12
HsMediaRaster_13.img	hsmediaraster_13
HsMediaRaster_14.img	hsmediaraster_14
HsMediaRaster_15.img	hsmediaraster_15
HsMediaRaster_16.img	hsmediaraster_16
HsMediaRaster_17.img	hsmediaraster_17
HsMediaRaster_18.img	hsmediaraster_18
HsMediaRaster_19.img	hsmediaraster_19
HsMediaRaster_2.img	hsmediaraster_2
HsMediaRaster_20.img	hsmediaraster_20
HsMediaRaster_21.img	hsmediaraster_21
HsMediaRaster_22.img	hsmediaraster_22
HsMediaRaster_23.img	hsmediaraster_23
HsMediaRaster_24.img	hsmediaraster_24
HsMediaRaster_25.img	hsmediaraster_25
HsMediaRaster_26.img	hsmediaraster_26
HsMediaRaster_27.img	hsmediaraster_27
HsMediaRaster_28.img	hsmediaraster_28

HsMediaRaster_29.img	hsmediaraster_29
HsMediaRaster_3.img	hsmediaraster_3
HsMediaRaster_30.img	hsmediaraster_30
HsMediaRaster_31.img	hsmediaraster_31
HsMediaRaster_32.img	hsmediaraster_32
HsMediaRaster_33.img	hsmediaraster_33
HsMediaRaster_34.img	hsmediaraster_34
HsMediaRaster_35.img	hsmediaraster_35
HsMediaRaster_4.img	hsmediaraster_4
HsMediaRaster_5.img	hsmediaraster_5
HsMediaRaster_6.img	hsmediaraster_6
HsMediaRaster_7.img	hsmediaraster_7
HsMediaRaster_8.img	hsmediaraster_8
HsMediaRaster_9.img	hsmediaraster_9
iba_recom_P.shp	iba_recom_p
IBAs_P.shp	ibas_p
ibm_P.shp	ibm_p
IEZH_P.shp	iezh_p
LICs_P.shp	lics_p
Lin_Costa_P.shp	lin_costa_p
Linea_Base_P.shp	linea_base_p
MABs_P.shp	mabs_p
Mar_territorial_P.shp	mar_territorial_p
maxmuecant11.img	maxmuecant11
maxmuemed1.img	maxmuemed1
maxvivcant.img	maxvivcant
maxvivmed.img	maxvivmed
medmuecant1.img	medmuecant1
medmuemed1.img	medmuemed1
medvivcant.img	medvivcant
medvivmed.img	medvivmed
Micror_P.shp	micror_p
Militar_P.shp	militar_p
MMNat_P.shp	mmnat_p
MMNat_point_P.shp	mmnat_point_p
OD_P.tif	od_p
OSPAR_AMPs_P.shp	ospar_amps_p

OZE_P.shp	oze_p
P_Humanidad_P.shp	p_humanidad_p
PaiPPro_P.shp	paipro_p
Patrim_P.shp	patrim_p
Patrim_point_P.shp	patrim_point_p
PEIN_P.shp	pein_p
PEP_P.shp	pep_p
Piscifactoria_P.shp	piscifactoria_p
Piscifactoria_point_P.shp	piscifactoria_point_p
PParNNat_P.shp	pparnnat_p
PPNat_P.shp	ppnat_p
PPNNs_P.shp	ppnns_p
PPPer_P.shp	ppper_p
PPRR_P.shp	
presiones_P.shp	presiones_p
ProtecConducciones_P.shp	protecconducciones_p
PwDEF_Can.img	pwdef_can
PwDEF_Med.img	pwdef_med
PwDEFRaster.img	pwdefraster
PwDEFRaster_0.img	pwdefraster_0
PwDEFRaster_1.img	pwdefraster_1
PwDEFRaster_10.img	pwdefraster_10
PwDEFRaster_11.img	pwdefraster_11
PwDEFRaster_12.img	pwdefraster_12
PwDEFRaster_13.img	pwdefraster_13
PwDEFRaster_14.img	pwdefraster_14
PwDEFRaster_15.img	pwdefraster_15
PwDEFRaster_16.img	pwdefraster_16
PwDEFRaster_17.img	pwdefraster_17
PwDEFRaster_18.img	pwdefraster_18
PwDEFRaster_19.img	pwdefraster_19
PwDEFRaster_2.img	pwdefraster_2
PwDEFRaster_20.img	pwdefraster_20
PwDEFRaster_21.img	pwdefraster_21
PwDEFRaster_22.img	pwdefraster_22
PwDEFRaster_23.img	pwdefraster_23
PwDEFRaster_24.img	pwdefraster_24

PwDEFRaster_25.img	pwdefraster_25
PwDEFRaster_26.img	pwdefraster_26
PwDEFRaster_27.img	pwdefraster_27
PwDEFRaster_28.img	pwdefraster_28
PwDEFRaster_29.img	pwdefraster_29
PwDEFRaster_3.img	pwdefraster_3
PwDEFRaster_30.img	pwdefraster_30
PwDEFRaster_31.img	pwdefraster_31
PwDEFRaster_32.img	pwdefraster_32
PwDEFRaster_33.img	pwdefraster_33
PwDEFRaster_34.img	pwdefraster_34
PwDEFRaster_4.img	pwdefraster_4
PwDEFRaster_5.img	pwdefraster_5
PwDEFRaster_6.img	pwdefraster_6
PwDEFRaster_7.img	pwdefraster_7
PwDEFRaster_8.img	pwdefraster_8
PwDEFRaster_9.img	pwdefraster_9
PwJJARaster.img	pwjjaraster
PwJJARaster_1.img	pwjjaraster_1
PwJJARaster_10.img	pwjjaraster_10
PwJJARaster_11.img	pwjjaraster_11
PwJJARaster_12.img	pwjjaraster_12
PwJJARaster_13.img	pwjjaraster_13
PwJJARaster_14.img	pwjjaraster_14
PwJJARaster_15.img	pwjjaraster_15
PwJJARaster_16.img	pwjjaraster_16
PwJJARaster_17.img	pwjjaraster_17
PwJJARaster_18.img	pwjjaraster_18
PwJJARaster_19.img	pwjjaraster_19
PwJJARaster_2.img	pwjjaraster_2
PwJJARaster_20.img	pwjjaraster_20
PwJJARaster_21.img	pwjjaraster_21
PwJJARaster_22.img	pwjjaraster_22
PwJJARaster_23.img	pwjjaraster_23
PwJJARaster_24.img	pwjjaraster_24
PwJJARaster_25.img	pwjjaraster_25
PwJJARaster_26.img	pwjjaraster_26

PwJJARaster_27.img	pwjjaraster_27
PwJJARaster_28.img	pwjjaraster_28
PwJJARaster_29.img	pwjjaraster_29
PwJJARaster_3.img	pwjjaraster_3
PwJJARaster_30.img	pwjjaraster_30
PwJJARaster_31.img	pwjjaraster_31
PwJJARaster_32.img	pwjjaraster_32
PwJJARaster_33.img	pwjjaraster_33
PwJJARaster_4.img	pwjjaraster_4
PwJJARaster_5.img	pwjjaraster_5
PwJJARaster_6.img	pwjjaraster_6
PwJJARaster_7.img	pwjjaraster_7
PwJJARaster_8.img	pwjjaraster_8
PwJJARaster_9.img	pwjjaraster_9
PwMAMRaster.img	pwmamraster
PwMAMRaster_0.img	pwmamraster_0
PwMAMRaster_1.img	pwmamraster_1
PwMAMRaster_10.img	pwmamraster_10
PwMAMRaster_11.img	pwmamraster_11
PwMAMRaster_12.img	pwmamraster_12
PwMAMRaster_13.img	pwmamraster_13
PwMAMRaster_14.img	pwmamraster_14
PwMAMRaster_15.img	pwmamraster_15
PwMAMRaster_16.img	pwmamraster_16
PwMAMRaster_17.img	pwmamraster_17
PwMAMRaster_18.img	pwmamraster_18
PwMAMRaster_19.img	pwmamraster_19
PwMAMRaster_2.img	pwmamraster_2
PwMAMRaster_20.img	pwmamraster_20
PwMAMRaster_21.img	pwmamraster_21
PwMAMRaster_22.img	pwmamraster_22
PwMAMRaster_23.img	pwmamraster_23
PwMAMRaster_24.img	pwmamraster_24
PwMAMRaster_25.img	pwmamraster_25
PwMAMRaster_26.img	pwmamraster_26
PwMAMRaster_27.img	pwmamraster_27
PwMAMRaster_28.img	pwmamraster_28

PwMAMRaster_29.img	pwmamraster_29
PwMAMRaster_3.img	pwmamraster_3
PwMAMRaster_30.img	pwmamraster_30
PwMAMRaster_31.img	pwmamraster_31
PwMAMRaster_32.img	pwmamraster_32
PwMAMRaster_33.img	pwmamraster_33
PwMAMRaster_4.img	pwmamraster_4
PwMAMRaster_5.img	pwmamraster_5
PwMAMRaster_6.img	pwmamraster_6
PwMAMRaster_7.img	pwmamraster_7
PwMAMRaster_8.img	pwmamraster_8
PwMAMRaster_9.img	pwmamraster_9
PwMEDIA_Can.img	pwmedia_can
PwMEDIA_Med.img	pwmedia_med
PwMediaRaster.img	pwmediaraster
PwMediaRaster_0.img	pwmediaraster_0
PwMediaRaster_1.img	pwmediaraster_1
PwMediaRaster_10.img	pwmediaraster_10
PwMediaRaster_11.img	pwmediaraster_11
PwMediaRaster_12.img	pwmediaraster_12
PwMediaRaster_13.img	pwmediaraster_13
PwMediaRaster_14.img	pwmediaraster_14
PwMediaRaster_15.img	pwmediaraster_15
PwMediaRaster_16.img	pwmediaraster_16
PwMediaRaster_17.img	pwmediaraster_17
PwMediaRaster_18.img	pwmediaraster_18
PwMediaRaster_19.img	pwmediaraster_19
PwMediaRaster_2.img	pwmediaraster_2
PwMediaRaster_20.img	pwmediaraster_20
PwMediaRaster_21.img	pwmediaraster_21
PwMediaRaster_22.img	pwmediaraster_22
PwMediaRaster_23.img	pwmediaraster_23
PwMediaRaster_24.img	pwmediaraster_24
PwMediaRaster_25.img	pwmediaraster_25
PwMediaRaster_26.img	pwmediaraster_26
PwMediaRaster_27.img	pwmediaraster_27
PwMediaRaster_28.img	pwmediaraster_28

PwMediaRaster_29.img	pwmediaraster_29
PwMediaRaster_3.img	pwmediaraster_3
PwMediaRaster_30.img	pwmediaraster_30
PwMediaRaster_31.img	pwmediaraster_31
PwMediaRaster_32.img	pwmediaraster_32
PwMediaRaster_33.img	pwmediaraster_33
PwMediaRaster_4.img	pwmediaraster_4
PwMediaRaster_5.img	pwmediaraster_5
PwMediaRaster_6.img	pwmediaraster_6
PwMediaRaster_7.img	pwmediaraster_7
PwMediaRaster_8.img	pwmediaraster_8
PwMediaRaster_9.img	pwmediaraster_9
PwSONRaster.img	pwsorraster
PwSONRaster_0.img	pwsorraster_0
PwSONRaster_1.img	pwsorraster_1
PwSONRaster_10.img	pwsorraster_10
PwSONRaster_11.img	pwsorraster_11
PwSONRaster_12.img	pwsorraster_12
PwSONRaster_13.img	pwsorraster_13
PwSONRaster_14.img	pwsorraster_14
PwSONRaster_15.img	pwsorraster_15
PwSONRaster_16.img	pwsorraster_16
PwSONRaster_17.img	pwsorraster_17
PwSONRaster_18.img	pwsorraster_18
PwSONRaster_19.img	pwsorraster_19
PwSONRaster_2.img	pwsorraster_2
PwSONRaster_20.img	pwsorraster_20
PwSONRaster_21.img	pwsorraster_21
PwSONRaster_22.img	pwsorraster_22
PwSONRaster_23.img	pwsorraster_23
PwSONRaster_24.img	pwsorraster_24
PwSONRaster_25.img	pwsorraster_25
PwSONRaster_26.img	pwsorraster_26
PwSONRaster_27.img	pwsorraster_27
PwSONRaster_28.img	pwsorraster_28
PwSONRaster_29.img	pwsorraster_29
PwSONRaster_3.img	pwsorraster_3

PwSONRaster_30.img	pwsorraster_30
PwSONRaster_31.img	pwsorraster_31
PwSONRaster_32.img	pwsorraster_32
PwSONRaster_33.img	pwsorraster_33
PwSONRaster_4.img	pwsorraster_4
PwSONRaster_5.img	pwsorraster_5
PwSONRaster_6.img	pwsorraster_6
PwSONRaster_7.img	pwsorraster_7
PwSONRaster_8.img	pwsorraster_8
PwSONRaster_9.img	pwsorraster_9
RAMSARs_P.shp	ramsars_p
Refugio_caza_P.shp	refugio_caza_p
RMO_P.shp	rmo_p
RR_ley_prop_P.shp	rr_ley_prop_p
RRF_P.shp	rrf_p
RRMM_P.shp	rrmm_p
RRNN_P.shp	rrnn_p
SAL_P.shp	sal_p
SAL_P.tif	
TEMP_P.tif	temp_p
Texturas_fondo_P.shp	texturas_fondo_p
TURB_P.tif	turb_p
Vertido_dragado_P.shp	vertido_dragado_p
Vertido_material_militar_P.shp	vertido_material_militar_p
Vertidos_P.shp	vertidos_p
Viveros_P.shp	viveros_p
Viveros_point_P.shp	viveros_point_p
ZBano_P.shp	zbanop
ZEPAs.shp	zepas
ZEPIMs_P.shp	zepims_p
ZEPVN_P.shp	zepvn_p
ZIAE_P.shp	ziae_p
ZMES_P.shp	zmes_p
ZONA_CABLES_P.shp	zona_cables_p
ZPIPs_P.shp	zpips_p
ZZHH_P.shp	zzhh_p

Comprobamos que **no** se han importado correctamente en el esquema ‘peninsula’ las siguientes capas (se han indicado con color rojo):

- **FLUORS_P.tif**
- **PPRR_P.shp**
- **SAL_P.tif**

4.6.3.4.2.2. Comprobación de Baleares

Para ver las tablas contenidas en el esquema ‘baleares’ en PostgreSQL:

```
SELECT tablename FROM pg_tables WHERE schemaname = 'baleares';
```

Capas que se tenían que importar	Capas importadas en PostgreSQL
500Lin_Costa_B.shp	500lin_costa_b
Acuicultura_B.shp	acuicultura_b
AEIV_B.shp	aeiv_b
Aguas_costeras_B.shp	aguas_costeras_b
aicp_B.shp	aicp_b
aopn_B.shp	aopn_b
AP_PTI_B.shp	ap_pti_b
Areas_eolicas_B.shp	areas_eolicas_b
Arrecifes_B.shp	arrecifes_b
AToN_B.shp	aton_b
Atun_B.shp	atun_b
Batimetria_B.shp	batimetria_b
CCAA_B.shp	ccaa_b
Cetaceo_B.shp	cetaceo_b
Conducciones_lineales_B.shp	conducciones_lineales_b
Cria_moluscos_B.shp	cria_moluscos_b
DDHH_B.shp	ddhh_b
DDMM_B.shp	ddmm_b
Deslinde_B.shp	deslinde_b
DPP_PE_B.shp	eeal_b
EEAL_B.shp	ext_aridos_b
Ext_aridos_B.shp	fanerog_b
Fanerog_B.shp	geofis_b
Geofis_B.shp	habitat_fanerog_b
Habitat_fanerog_B.shp	habitats_b

Habitats_B.shp	hsmediaraster
HsMediaRaster.img	hsmediaraster_1
HsMediaRaster_1.img	hsmediaraster_2
HsMediaRaster_2.img	hsmediaraster_3
HsMediaRaster_3.img	iba_recom_b
iba_recom_B.shp	ibas_b
IBAs_B.shp	ibm_b
ibm_B.shp	licientifico_b
LICientifico_B.shp	lics_b
LICs_B.shp	lin_costa_b
Lin_Costa_B.shp	linea_base_b
Linea_base_B.shp	mabs_b
MABs_B.shp	mar_territorial_b
Mar_territorial_B.shp	maxmuemed2
maxmuemed2.img	maxvivmed
maxvivmed.img	medmuemed1
medmuemed1.img	medvivmed
medvivmed.img	mmnat_b
MMNat_B.shp	oze_b
OZE_B.shp	parapro_b
ParaPro_B.shp	patrim_b
Patrim_B.shp	patrim_mundial_b
Patrim_Mundial_B.shp	patrim_point_b
Patrim_point_B.shp	ppnat_b
PPNat_B.shp	ppnns_b
PPNNs_B.shp	presiones_b
presiones_B.shp	pwdef_b
PwDEF_B.img	pwdefraster
pwdefraster.img	pwdefraster_0
pwdefraster_0.img	pwdefraster_1
pwdefraster_1.img	
pwdefraster_2.img	pwdefraster_2
pwdefraster_3.img	pwdefraster_3
PwJJA_B.img	pwjja_b
PwJJARaster.img	pwjjaraster
PwJJARaster_1.img	pwjjaraster_1
PwJJARaster_2.img	pwjjaraster_2

PwJJARaster_3.img	pwjjaraster_3
PwMAM_B.img	pwmam_b
PwMAMRaster.img	pwmamraster
PwMAMRaster_1.img	pwmamraster_1
PwMAMRaster_2.img	pwmamraster_2
PwMAMRaster_3.img	pwmamraster_3
PwMEDIA_B.img	pwmedia_b
PwMediaRaster.img	pwmediaraster
PwMediaRaster_1.img	pwmediaraster_1
PwMediaRaster_2.img	pwmediaraster_2
PwMediaRaster_3.img	pwmediaraster_3
PwSON_B.img	pwson_b
PwSONRaster.img	pwsonraster
PwSONRaster_1.img	pwsonraster_1
PwSONRaster_2.img	pwsonraster_2
PwSONRaster_3.img	pwsonraster_3
RAMSARs_B.shp	ramsars_b
RMO_B.shp	rmo_b
RRMM_B.shp	rrmm_b
RRNN_B.shp	rrnn_b
Vertidos_B.shp	vertidos_b
Viveros_refug_B.shp	viveros_refug_b
ZBano_B.shp	zbanob_b
ZEPAs_B.shp	zepas_b
ZEPIMs_B.shp	zepims_b
ZONA_CABLES_B.shp	zona_cables_b
Zonas_fondeo_B.shp	zonas_fondeo_b
Zonas_Prohibidas_B.shp	zonas_prohibidas_b

Comprobamos que no se han importado correctamente en el esquema 'balears' las siguientes capas (se han indicado con color rojo):

- DPP_PE_B.shp

4.6.3.4.2.3. Comprobación de Canarias

Para ver las tablas contenidas en el esquema 'canarias':

```
SELECT tablename FROM pg_tables WHERE schemaname = 'canarias';
```

Capas que se tenían que importar	Capas importadas en PostgreSQL
500Lin_Costa_Can_WGS84.shp	500lin_costa_can_wgs84
Acuicultura_Can_WGS84.shp	acuicultura_can_wgs84
Aguas_costeras_Can_WGS84H28.shp	aguas_costeras_can_wgs84h28
aopn_Can_WGS84H28.shp	aopn_can_wgs84h28
Areas_eolicas_Can_WGS84H28.shp	areas_eolicas_can_wgs84h28
Arrecifes_Can_WGS84H28.shp	arrecifes_can_wgs84h28
AToN_Can_WGS84H28.shp	aton_can_wgs84h28
Batimetria_Can_WGS84H28.shp	batimetria_can_wgs84h28
Bentos_Can_WGS84H28.shp	bentos_can_wgs84h28
CCAA_Can_WGS84H28.shp	ccaa_can_wgs84h28
Conducciones_Line_Can_WGS84H28.shp	conducciones_line_can_wgs84h28
DDMM_Can_WGS84H28.shp	ddmm_can_wgs84h28
DPP_Canarias_WGS84H28.shp	dpp_canarias_wgs84h28
DST_ZE_Can_WGS84H28.shp	dst_ze_can_wgs84h28
EEAL_Can_WGS84H28.shp	eeal_can_wgs84h28
Ext_aridos_Can_WGS84H28.shp	ext_aridos_can_wgs84h28
FLUORS_Can_WGS84H28.shp	fluors_can_wgs84h28
Fondos_Can_WGS84H28.shp	fondos_can_wgs84h28
Formas_Can_WGS84H28.shp	formas_can_wgs84h28
Formas_line_Can_WGS84H28.shp	formas_line_can_wgs84h28
Geofis_Can_WGS84H28.shp	geofis_can_wgs84h28
Habitats_Can_WGS84H281.shp	habitats_can_wgs84h281
HsMedia_M091.img	hsmedia_m091
HsMedia_M092.img	hsmedia_m092
HsMedia_M093.img	hsmedia_m093
HsMedia_M094.img	hsmedia_m094
HsMedia_M095.img	hsmedia_m095
HsMedia_M101.img	hsmedia_m101
HsMedia_M102.img	hsmedia_m102
HsMedia_M103.img	hsmedia_m103
HsMedia_M104.img	hsmedia_m104
HsMedia_M105.img	hsmedia_m105
HsMedia_M106.img	hsmedia_m106
HsMedia_M111.img	hsmedia_m111
HsMedia_M112.img	hsmedia_m112

HsMedia_M113.img	hsmedia_m112
HsMedia_M114.img	hsmedia_m113
HsMedia_M121.img	hsmedia_m114
HsMedia_M122.img	hsmedia_m121
HsMedia_M123.img	hsmedia_m122
HsMedia_M124.img	hsmedia_m123
HsMedia_M125.img	hsmedia_m124
HsMedia_M126.img	hsmedia_m125
HsMedia_M127.img	hsmedia_m126
HsMedia_M128.img	hsmedia_m127
HsMedia_M131.img	hsmedia_m128
HsMedia_M132.img	hsmedia_m131
HsMedia_M141.img	hsmedia_m132
iba_recom_Can_WGS84H28.shp	hsmedia_m141
IBAs_Can_WGS84H28.shp	iba_recom_can_wgs84h28
ibm_Can_WGS84H28.shp	ibas_can_wgs84h28
LICs_Can_WGS84H28.shp	ibm_can_wgs84h28
Lin_Costa_Can_WGS84H28.shp	lics_can_wgs84h28
Linea_Base_Can_WGS84H28.shp	lin_costa_can_wgs84h28
MABs_Can_WGS84H28.shp	linea_base_can_wgs84h28
Mar_territorial_Can_WGS84H28.shp	mabs_can_wgs84h28
maxmuecan1.img	mar_territorial_can_wgs84h28
maxvivcan.img	maxmuecan1
medmuecan1.img	maxvivcan
medvivcan.img	medmuecan1
Militar_Can_WGS84H28.shp	medvivcan
MMNat_Can_WGS84H28.shp	militar_can_wgs84h28
OD_Can_WGS84H28.shp	mmnat_can_wgs84h28
Otros_Can_WGS84H28.shp	od_can_wgs84h28
OZE_Can_WGS84H28.shp	otros_can_wgs84h28
PaiPro_Can_WGS84H28.shp	oze_can_wgs84h28
Patrim_Can_point_WGS84H28.shp	paipro_can_wgs84h28
Patrim_Can_WGS8H284.shp	patrim_can_point_wgs84h28
Piscifactoria_Can_WGS84H28.shp	patrim_can_wgs8h284
PPNat_Can_WGS84H28.shp	piscifactoria_can_wgs84h28
PPNNs_Can_WGS84H28.shp	ppnat_can_wgs84h28
PPRur_Can_WGS84H28.shp	ppnns_can_wgs84h28

presiones_Can_WGS84H28.shp	pprur_can_wgs84h28
ProtecConducciones_Can_WGS84H28.shp	presiones_can_wgs84h28
PwDEF_C.img	protecconducciones_can_wgs84h28
PwDEF_M091.img	pwdef_c
PwDEF_M092.img	pwdef_m091
PwDEF_M093.img	pwdef_m092
PwDEF_M094.img	pwdef_m093
PwDEF_M095.img	pwdef_m094
PwDEF_M101.img	pwdef_m095
PwDEF_M102.img	pwdef_m101
PwDEF_M103.img	pwdef_m102
PwDEF_M104.img	pwdef_m103
PwDEF_M105.img	pwdef_m104
PwDEF_M106.img	pwdef_m105
PwDEF_M111.img	pwdef_m106
PwDEF_M112.img	pwdef_m111
PwDEF_M113.img	pwdef_m112
PwDEF_M114.img	pwdef_m113
PwDEF_M121.img	pwdef_m114
PwDEF_M122.img	pwdef_m121
PwDEF_M123.img	pwdef_m122
PwDEF_M124.img	pwdef_m123
PwDEF_M125.img	pwdef_m124
PwDEF_M126.img	pwdef_m125
PwDEF_M127.img	pwdef_m126
PwDEF_M128.img	pwdef_m127
PwDEF_M131.img	pwdef_m128
PwDEF_M132.img	pwdef_m131
PwDEF_M141.img	pwdef_m132
PwJJA_C.img	pwdef_m141
PwJJA_M091.img	pwjja_c
PwJJA_M092.img	pwjja_m091
PwJJA_M093.img	pwjja_m092
PwJJA_M094.img	pwjja_m093
PwJJA_M095.img	pwjja_m094
PwJJA_M101.img	pwjja_m095
PwJJA_M102.img	pwjja_m101

PwJJA_M103.img	pwjja_m102
PwJJA_M104.img	pwjja_m103
PwJJA_M105.img	pwjja_m104
PwJJA_M106.img	pwjja_m105
PwJJA_M111.img	pwjja_m106
PwJJA_M112.img	pwjja_m111
PwJJA_M113.img	pwjja_m112
PwJJA_M114.img	pwjja_m113
PwJJA_M121.img	pwjja_m114
PwJJA_M122.img	pwjja_m121
PwJJA_M123.img	pwjja_m122
PwJJA_M124.img	pwjja_m123
PwJJA_M125.img	pwjja_m124
PwJJA_M126.img	pwjja_m125
PwJJA_M127.img	pwjja_m126
PwJJA_M128.img	pwjja_m127
PwJJA_M131.img	pwjja_m128
PwJJA_M132.img	pwjja_m131
PwJJA_M141.img	pwjja_m132
PwMAM_C.img	pwjja_m141
PwMAM_M091.img	pwmam_c
PwMAM_M092.img	pwmam_m091
PwMAM_M093.img	pwmam_m092
PwMAM_M094.img	pwmam_m093
PwMAM_M095.img	pwmam_m094
PwMAM_M101.img	pwmam_m095
PwMAM_M102.img	pwmam_m101
PwMAM_M103.img	pwmam_m102
PwMAM_M104.img	pwmam_m103
PwMAM_M105.img	pwmam_m104
PwMAM_M106.img	pwmam_m105
PwMAM_M111.img	pwmam_m106
PwMAM_M112.img	pwmam_m111
PwMAM_M113.img	pwmam_m112
PwMAM_M114.img	pwmam_m113
PwMAM_M121.img	pwmam_m114
PwMAM_M122.img	pwmam_m121

PwMAM_M123.img	pwmam_m122
PwMAM_M124.img	pwmam_m123
PwMAM_M125.img	pwmam_m124
PwMAM_M126.img	pwmam_m125
PwMAM_M127.img	pwmam_m126
PwMAM_M128.img	pwmam_m127
PwMAM_M131.img	pwmam_m128
PwMAM_M132.img	pwmam_m131
PwMAM_M141.img	pwmam_m132
PwMEDIA_C.img	pwmam_m141
PwMedia_M091.img	pwmedia_c
PwMedia_M092.img	pwmedia_m091
PwMedia_M093.img	pwmedia_m092
PwMedia_M094.img	pwmedia_m093
PwMedia_M095.img	pwmedia_m094
PwMedia_M101.img	pwmedia_m095
PwMedia_M102.img	pwmedia_m101
PwMedia_M103.img	pwmedia_m102
PwMedia_M104.img	pwmedia_m103
PwMedia_M105.img	pwmedia_m104
PwMedia_M106.img	pwmedia_m105
PwMedia_M111.img	pwmedia_m106
PwMedia_M112.img	pwmedia_m111
PwMedia_M113.img	pwmedia_m112
PwMedia_M114.img	pwmedia_m113
PwMedia_M121.img	pwmedia_m114
PwMedia_M122.img	pwmedia_m121
PwMedia_M123.img	pwmedia_m122
PwMedia_M124.img	pwmedia_m123
PwMedia_M125.img	pwmedia_m124
PwMedia_M126.img	pwmedia_m125
PwMedia_M127.img	pwmedia_m126
PwMedia_M128.img	pwmedia_m127
PwMedia_M131.img	pwmedia_m128
PwMedia_M132.img	pwmedia_m131
PwMedia_M141.img	pwmedia_m132
PwSON_C.img	pwmedia_m141

PwSON_M091.img	pwson_c
PwSON_M092.img	pwson_m091
PwSON_M093.img	pwson_m092
PwSON_M094.img	pwson_m093
PwSON_M095.img	pwson_m094
PwSON_M101.img	pwson_m095
PwSON_M102.img	pwson_m101
PwSON_M103.img	pwson_m102
PwSON_M104.img	pwson_m103
PwSON_M105.img	pwson_m104
PwSON_M106.img	pwson_m105
PwSON_M111.img	pwson_m106
PwSON_M112.img	pwson_m111
PwSON_M113.img	pwson_m112
PwSON_M114.img	pwson_m113
PwSON_M121.img	pwson_m114
PwSON_M122.img	pwson_m121
PwSON_M123.img	pwson_m122
PwSON_M124.img	pwson_m123
PwSON_M125.img	pwson_m124
PwSON_M126.img	pwson_m125
PwSON_M127.img	pwson_m126
PwSON_M128.img	pwson_m127
PwSON_M131.img	pwson_m128
PwSON_M132.img	pwson_m131
PwSON_M141.img	pwson_m132
RAMSARs_Can_WGS84H28.shp	pwson_m141
RMO_Can_WGS84H28.shp	ramsars_can_wgs84h28
RRMM_Can_WGS84H28.shp	rmo_can_wgs84h28
RRNN_Can_WGS84H28.shp	rrmm_can_wgs84h28
SAL_Can_WGS84H28.shp	rrnn_can_wgs84h28
TEMP_Can_WGS84H28.shp	sal_can_wgs84h28
Tipo_pesca_Can_WGS84H28.shp	temp_can_wgs84h28
Toponimia_Can_WGS84H28.shp	tipo_pesca_can_wgs84h28
Toponimia_costa_Can_WGS84H28.shp	toponimia_can_wgs84h28
TURB_Can_WGS84H28.shp	toponimia_costa_can_wgs84h28
Vertidos_Can_WGS84H28.shp	turb_can_wgs84h28

ZBano_Can_WGS84H28.shp	vertidos_can_wgs84h28
ZEPAs_Can_WGS84H28.shp	zbanos_can_wgs84h28
ZMES_Can_WGS84H28.shp	zepas_can_wgs84h28
ZMT_Can_WGS84H28.shp	zmes_can_wgs84h28
	zmt_can_wgs84h28

Se han importado todas las capas raster y vectoriales correctamente en el esquema 'canarias'.

4.6.3.4.3. Importación de las capas fallidas

Tras comprobar las capas que no han sido importadas, se procede a realizar de nuevo la importación:

- Para **FLUORS_P.tif**:

```
raster2pgsql -s 23030 -I -C -M "C:\Users\GSI\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\P\02_IT\PO\FLUORS_P.tif" -F
peninsula.FLUORS_P | psql -U postgres -d BBDDOceanLider -h
localhost -p 5432
```

Resultado: "out of memory". No se puede importar esta capa dentro de PostgreSQL porque no se dispone de suficiente memoria.

- Para **PPRR_P.shp**:

```
shp2pgsql -W LATIN1 -s 23030 "C:\Users\GSI\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\P\02_IT\CyMA\E\AU\PPRR_P.shp"
peninsula.PPRR_P | psql -U postgres -d BBDDOceanLider -h localhost
-p 5432
```

Resultado: DBF file can not be opened. No se puede importar esta capa dentro de PostgreSQL porque no se encuentra el fichero DBF para esta Shape.

- Para **SAL_P.tif**:

Tenemos que renombrar esta capa raster ya que se encuentra una capa vectorial con el mismo nombre dentro de PostgreSQL.

```
raster2pgsql -s 23030 -I -C -M "C:\Users\GSI\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\P\02_IT\PO\SAL_P.tif" -F
peninsula.SAL_P_raster | psql -U postgres -d BBDDOceanLider -h
localhost -p 5432
```

Resultado: La capa ha sido importada con éxito con el nombre "sal_p_raster" en el esquema 'peninsula'

- Para **DPP_PE_B.shp**:

```
shp2pgsql -W LATIN1 -s 23031 "C:\Users\GSI\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\B\02_IT\UMyL\DPP\DPP_PE_B.shp"
baleares.DPP_PE_B |psql -U postgres -d BBDDOceanLider -h
localhost -p 5432
```

Resultado: DBF file can not be opened. No se puede importar esta capa dentro de PostgreSQL porque nos e encuentra el fichero DBF para este Shape.

4.6.3.5. Conclusión

Se han utilizado técnicas de importación múltiple de forma automática con código CMD de Windows y comandos de PostgreSQL. Al ser un proceso automático es necesario realizar una comprobación final para examinar que todas las capas (raster y vectoriales) han sido importadas.

Tras contrarrestar los resultados finales de la importación, se ha visto que ha habido casos fallidos de algunas capas. Se ha vuelto a realizar la importación de estas capas y se ha detectado la causa del fallo.

Para las capas que ha sido posible resolver el conflicto del fallo, se han importado con éxito a PostgreSQL, pero ha habido tres capas en las que no ha sido posible. Los conflictos que intervienen es la falta del archivo DBF (tabla de atributos asociados a la capa vectorial) en dos Shapefiles y el tamaño excesivo de una capa raster.

La BBDD dentro de PostgreSQL ha sido dividida en tres esquemas según la clasificación en entidades de la BBDD original (proporcionada por PROES). Al utilizar el comando *shp2pgsql* en la importación fuerza a que los nombres de las capas importadas permanezcan en minúscula y también ha sido necesario resolver el conflicto de la importación de las capas cuyo nombre contiene acentos o caracteres especiales.

4.6.4. Unión de capas raster y vectoriales de la BBDDOceanLider en la proyección WGS84

4.6.4.1. Contexto del contenido de la base de datos geográfica

La base de datos geográfica perteneciente al ámbito del proyecto está dividida en dos grupos, el grupo de *capas informativas* y de *capas de restricciones*. En el grupo de capas informativas están aquellas capas que contienen información de diferente índole recolectada de diversas fuentes gubernamentales y autonómicas por organismos del proyecto. El grupo de capas de restricciones contiene capas que han sido evaluadas por organismos del proyecto a partir de las capas informativas como objetivo de fijar valores de idoneidad a las capas de índole medioambiental. Cada grupo está a su vez subdividida en tres entidades que son Península, Baleares y Canarias. Cada entidad contiene capas raster y vectoriales en una proyección y sistema de referencia dada por su ubicación en la geografía española:

- **Península:** código EPSG 23030 que corresponde con la proyección UTM ED50 Huso 30 N

- **Baleares:** código EPSG 23031 que corresponde con la proyección UTM ED50 Huso 31 N
- **Canarias:** código EPSG 32628 que corresponde con la proyección UTM WGS84 Huso 28 N

El **Sistema de Coordenadas Universal Transversal de Mercator** (en inglés Universal Transverse Mercator, **UTM**) es un sistema de coordenadas basado en la proyección cartográfica transversa de Mercator, que se construye como la proyección de Mercator normal, pero en vez de hacerla tangente al Ecuador, se la hace tangente a un meridiano.

A diferencia del sistema de coordenadas geográficas, expresadas en longitud y latitud, las magnitudes en el sistema UTM se expresan en metros únicamente al nivel del mar, que es la base de la proyección del elipsoide de referencia.

El **WGS84** es un sistema de coordenadas geográficas mundial que permite localizar cualquier punto de la Tierra (sin necesitar otro de referencia) por medio de tres unidades dadas. **WGS84** son las siglas en inglés de **World Geodetic System 84** (que significa Sistema Geodésico Mundial 1984).

Se trata de un estándar en geodesia, cartografía, y navegación, que data de 1984. Se estima un error de cálculo menor a 2 cm, por lo que es en la que se basa el Sistema de Posicionamiento Global (GPS).

Por una cuestión de practicidad, se suele proyectar este sistema de coordenadas geodésicas (expresados en grados, minutos, segundos) a algún otro sistema de coordenadas cartesiano (pasar de un modelo 3D a uno 2D) llamados sistema de proyección típicamente UTM que se expresan en metros (en orden a su relación a un punto de origen arbitrario) que facilita cálculos de distancia y superficie. [Calderón, Álvarez, Axpuc, 2011]

Uno de los cometidos a la hora de trabajar con capas de distintas regiones del territorio español en diferentes sistemas de referencia es la necesidad de trabajar de forma conjunta en un único sistema de referencia. El sistema de referencia elegido es el **WGS84 con código EPSG 4326**.

4.6.4.2. Unión de capas raster y vectoriales de Península, Baleares y Canarias

En un principio, se importaron todas las capas de los dos grupos en dos bases de datos dentro de PostGIS:

- **BBDOceanLider:** base de datos geográfica que contiene las capas vectoriales del grupo de capas informativas.
- **BBDOceanLider_restricciones:** base de datos geográfica que contiene las capas vectoriales del grupo de capas de restricciones del ámbito medioambiental.

Posteriormente se decidió utilizar únicamente las capas informativas para que la herramienta DSS fuera más genérica y flexible, sin tener prefijados los valores en algunos puntos de la toma de decisión. Aunque el software que se entrega utiliza la base de datos de restricciones porque el

estudio indica que hay zonas omitidas en las capas medioambientales por análisis de las entidades del proyecto.

Por tanto, en este apartado se va a comentar el trabajo realizado para ambas bases de datos y se van a describir los pasos para realizar la unión de las diferentes capas de estas dos bases de datos..

Como se ha comentado anteriormente, para que el usuario tenga la libertad de escoger cualquier región geográfica de estudio de la zona marítima española en la herramienta DSS, es necesario hacer la unión de las diferentes capas de mismo contenido procedentes de las tres entidades (Península, Baleares y Canarias) y reproyectarlas a WGS84.

En la unión de las diferentes **capas vectoriales** que están dentro de estas entidades (Península, Baleares y Canarias) se ha hecho uso de operaciones dentro de PostGIS con capas albergadas en la base de datos *BBDDOceanLider* y *BBDDOceanLider_restricciones*, de forma independiente.

En la unión de las diferentes **capas raster** que están dentro de estas entidades (Península, Baleares y Canarias) se han utilizado operaciones dentro del SIG GRASS de las capas albergadas en el directorio de la base de datos del ámbito de proyecto OceanLider bajo el sistema de ficheros de Windows.

4.6.4.2.1. Unión de capas vectoriales

La base de datos *BBDDOceanLider* y *BBDDOceanLider_restricciones* albergan cuatro esquemas:

- **Esquema ‘península’**: Contiene las capas vectoriales de los datos asociados al proyecto que están ubicados en la Península y zona marítima Peninsular. Las capas que están bajo este esquema tienen el sistema de referencia EPSG 23030.
- **Esquema ‘baleares’**: Contiene las capas vectoriales informativas de los datos asociados al proyecto que están ubicados en las Islas Baleares y zona marítima de Baleares. Las capas que están bajo este esquema tienen el sistema de referencia EPSG 23031.
- **Esquema ‘canarias’**: Contiene las capas vectoriales de los datos asociados al proyecto que están ubicados en las Islas Canarias y zona marítima de Canarias. Las capas que están bajo este esquema tienen el sistema de referencia EPSG 32628.
- **Esquema ‘public’**: Esquema por defecto. Contiene las capas vectoriales de los datos asociados al proyecto que son unión de las tres entidades y reproyectadas al sistema de referencia con código EPSG 4326. Son las capas de entrada que utiliza la herramienta DSS para operar.

Por tanto, las capas de unión estarán ubicadas en el esquema ‘public’ y son las que utilizaremos como capas de entrada en la herramienta DSS. Haremos uso de las capas de los diferentes esquemas para hacer la unión y reproyección de las capas como veremos a continuación.

Vamos a utilizar la consola de sentencias SQL dentro de pgAdmin III que dispone PostgreSQL. Los ejemplos expuestos son ejecutados sobre la *BBDDOceanLider*.

Los pasos de ejecución para realizar la unión y reproyección:

- 1- Buscar las capas vectoriales que tienen la misma información para las tres entidades geográficas. Por ejemplo la línea de costa para cada entidad, estas capas son: 'lin_costa_p' (en el esquema península), 'lin_costa_b' (en el esquema baleares) y 'lin_costa_can_wgs84h28' (en el esquema canarias).

- 2- Consultar los nombres de las columnas y tipo de datos de las columnas de cada capa vectorial y verificar que concuerda con las columnas para las diferentes entidades. Por ejemplo para la línea de costa:

```
SELECT column_name FROM information_schema.columns WHERE
table_name = 'lin_costa_p';
SELECT column_name FROM information_schema.columns WHERE
table_name = 'lin_costa_b';
SELECT column_name FROM information_schema.columns WHERE
table_name = 'lin_costa_can_wgs84h28';
```

- 3- Consultar el tipo de geometría de cada capa vectorial y verificar que concuerda con las geometrías para las diferentes entidades. Por ejemplo para la línea de costa:

```
SELECT DISTINCT(st_geometrytype(geom)) FROM
peninsula.lin_costa_p;
SELECT DISTINCT(st_geometrytype(geom)) FROM
baleares.lin_costa_b;
SELECT DISTINCT(st_geometrytype(geom)) FROM
canarias.lin_costa_can_wgs84h28;
```

- 4- Crear la nueva tabla de unión en el esquema 'public' con las columnas de las tres capas vectoriales pertenecientes a las diferentes entidades y mismo tipo de datos. Por ejemplo para la línea de costa:

```
CREATE TABLE public.lin_costa (
    gid serial NOT NULL,
    linea character varying (5),
    zona character varying(50),
    CONSTRAINT lin_costa_pkey PRIMARY KEY (gid)
);
```

- 5- Añadir el campo de geometría a la nueva tabla con el EPSG 4326. Hay que tener en cuenta el tipo de geometría que consultamos en el paso 2 y el sistema de referencia espacial que es el 4326 para las capas de unión. Por ejemplo para la línea de costa:

```
SELECT
AddGeometryColumn('public','lin_costa','geom',4326,'MultiLi
neString',2);
```

- 6- Y por último insertamos la información de las capas vectoriales de las tres entidades a la nueva tabla y re proyectamos en el campo de geometría con la función *ST_Transform* de PostGIS. Por ejemplo para la línea de costa:

```
INSERT INTO public.lin_costa ("linea", "zona", "geom") (  
SELECT  
"linea", "zona", ST_Transform(ST_Transform(geom, 23030), 4326)  
FROM peninsula.lin_costa_p  
UNION  
SELECT  
"linea", "zona", ST_Transform(ST_Transform(geom, 23031), 4326)  
FROM baleares.lin_costa_b  
UNION  
SELECT  
"linea", "zona", ST_Transform(ST_Transform(geom, 32628), 4326)  
FROM canarias.lin_costa_can_wgs84h28);
```

Nos pueden surgir algunos imprevistos en cuanto a los tipos de datos que contienen las columnas de las capas vectoriales. Puede que no coincida el tipo de datos de una columna con el resto de capas a las que queremos hacer unión. En este caso creamos la nueva tabla con un tipo de datos más general que los de las entidades, por ejemplo, añadiendo más tamaño de la cadena de caracteres. O en caso que una de las capas contenga una columna que no esté presente en las capas de las demás entidades, lo que se hace es crear la nueva tabla con dicha columna y a la hora de la unión ignorar dicho campo poniendo 'null' en las capas que no lo contengan.

También hay que tener en cuenta a la hora de hacer la importación en GRASS desde PostGIS de las capas vectoriales de unión, el comando *v.in.ogr* hace una truncación sobre el dato numérico de más de 10 decimales. Si hay datos numéricos con más de 10 decimales en el rango (0-1), va a ser truncado siempre a 0. Por tanto, es imprescindible utilizar el dato numérico de PostgreSQL con precisión y escala al crear la nueva capa vectorial de unión [PostgreSQL c, 2013].

En el *Anexo E* y *F* se dispone el fichero SQL de la unión de las diferentes capas vectoriales de la base de datos de restricciones e informativas.

4.6.4.2.2. Unión de capas raster

Se va a utilizar el sistema SIG GRASS para la unión de capas raster ubicadas en el sistema de ficheros de Windows. Se va a unir capas raster con diferentes códigos de proyección y van a ser re proyectadas al sistema de coordenadas WGS84.

Al utilizar GRASS con capas raster con diferentes códigos de proyección es necesario crear localidades que se definan con dichos códigos. Dentro de estas localidades se importan las capas raster que sigan dicha proyección y sistema de referencia, y posteriormente se re proyectan a la localización con el código EPSG 4326.

Por tanto, antes de realizar los pasos para hacer la unión y re proyección, es necesario crear las cuatro localidades dentro del SIG GRASS y los ficheros de configuración del entorno de trabajo de GRASS.

Para crear las localidades dentro del entorno del SIG GRASS podemos seguir los pasos descritos en el apartado 6.1.2.1. *Creación de MAPSET y LOCATION* del manual de instalación.

Las localidades que debemos crear son:

- **Península:** Se importan las capas raster correspondientes a Península y zona marítima Peninsular. Se crea con el código EPSG 23030.
- **Baleares:** Se importan las capas raster correspondientes a Baleares y zona marítima de Baleares. Se crea con el código EPSG 23031.
- **Canarias:** Se importan las capas raster correspondientes a Canarias y zona marítima de Canarias. Se crea con el código EPSG 32628.
- **WGS84:** Es la localización que se crea para la herramienta DSS. Se crea con el código EPSG 4326.

Dentro de cada localización, debemos crear un MAPSET llamado 'DSS', que es sobre el que lanzaremos las operaciones.

Los ficheros de configuración que debemos crear son cuatro correspondientes a las cuatro localizaciones creadas. Les ubicaremos sobre la carpeta raíz del usuario de Windows (en nuestro caso C:\Users\tregmm).

- Fichero de configuración **.grassrc6_peninsula** contiene:

```
GISDBASE: C:\Users\tregmm\Documents\GISDataBase
LOCATION_NAME: Peninsula
MAPSET: DSS
GRASS_GUI: text
```

- Fichero de configuración **.grassrc6_baleres** contiene:

```
GISDBASE: C:\Users\tregmm\Documents\GISDataBase
LOCATION_NAME: Baleares
MAPSET: DSS
GRASS_GUI: text
```

- Fichero de configuración **.grassrc6_canarias** contiene:

```
GISDBASE: C:\Users\tregmm\Documents\GISDataBase
LOCATION_NAME: Canarias
MAPSET: DSS
GRASS_GUI: text
```

- Fichero de configuración **.grass6_wgs84** contiene:

```
GISDBASE: C:\Users\tregmm\Documents\GISDataBase
LOCATION_NAME: WGS84
MAPSET: DSS
GRASS_GUI: text
```


Vamos a construir un .BAT con diferentes sentencias y operaciones de GRASS para posteriormente ejecutarlo en Windows (Ver apartado 6.3.1. *Configuración para la ejecución de comandos y scripts de GRASS en CMD de Windows*). Los pasos que se describen a continuación son el ejemplo de la unión y reproyección de la capa raster del viento anual. Los pasos son los siguientes:

1. Configurar el fichero del entorno de trabajo de GRASS con la localización de Península

```
SET GISRC=C:\Users\tregmm\.grassrc6_peninsula
```
2. Configurar la región de trabajo dentro de la localización de Península

```
g.region vect=lin_costa_p ewres=300 nsres=300
```
3. Importar la capa raster del viento anual de la Península

```
r.in.gdal -o  
input="C:\Users\tregmm\Desktop\BBDD_Ocean_Lider\P\02_Capas\  
02_IT\RE\VI\Viento_Anual.img" output=Viento_Anual_p
```
4. Configurar el fichero del entorno de trabajo de GRASS con la localización Baleares

```
SET GISRC=C:\Users\tregmm\.grassrc6_baleares
```
5. Configurar la región de trabajo dentro de la localización Baleares

```
g.region vect=lin_costa_b ewres=300 nsres=300
```
6. Importar la capa raster del viento anual de Baleares

```
r.in.gdal -o  
input="C:\Users\tregmm\Desktop\BBDD_Ocean_Lider\B\02_Capas\  
02_IT\RE\VI\Viento_Anual.img" output=Viento_Anual_b
```
7. Configurar el fichero del entorno de trabajo de GRASS con la localización Canarias

```
SET GISRC=C:\Users\tregmm\.grassrc6_canarias
```
8. Configurar la región de trabajo dentro de la localización Canarias

```
g.region vect=lin_costa_c ewres=300 nsres=300
```
9. Importar la capa raster del viento anual de Canarias

```
r.in.gdal -o  
input="C:\Users\tregmm\Desktop\BBDD_Ocean_Lider\C\02_Capas\  
02_IT\RE\VI\viento_anual.img" output=Viento_Anual_c
```
10. Configurar el fichero del entorno de trabajo de GRASS con la localización WGS84

```
SET GISRC=C:\Users\tregmm\.grassrc6_wgs84
```

11. Configurar la región de trabajo dentro de la localización WGS84

```
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300
ewres=300 rows=3224 cols=3453
```

12. Hacer la reproyección de las capas raster del viento de las localización Península, Baleares y Canarias a la proyección WGS84

```
r.proj input=Viento_Anual_p location=Peninsula mapset=DSS
r.proj input=Viento_Anual_b location=Baleares mapset=DSS
r.proj input=Viento_Anual_c location=Canarias mapset=DSS
```

13. Realizar la unión de las capas raster del viento

```
r.patch input=Viento_Anual_p,Viento_Anual_b,Viento_Anual_c
output=Viento_Anual
```

14. Exportar el mapa de salida al sistema de ficheros de Windows

```
r.out.gdal input=Viento_Anual
output="C:\Users\tregmm\Desktop\Union\vientoanual.tif"
```

Hay algunos casos en los que se ha dividido a la vez por subzonas la recopilación de la información en una entidad. Por ejemplo para la altura media de ola significativa, tenemos una colección de capas raster en cada entidad (Península, Baleares y Canarias). En necesario hacer la unión previa de la colección de capas raster de cada entidad, pero puede ser una tarea ardua si se llega a un número elevado de capas raster. Por este motivo, nos hemos apoyado en el lenguaje de comandos batch de la consola de Windows.

Vamos a utilizar las sentencias FOR y el empleo de variables para hacer la unión de forma automática, combinándolo con comandos de GRASS.

Para la asignación y empleo de variables dentro de un bucle *for* es preciso la extensión de variables dentro del script. Por defecto el intérprete de comandos tiene desactivada la expansión diferida (a menos que se haya arrancado con /V:ON o bien se hayan modificado ciertas claves de registro para cambiar el comportamiento general del intérprete de comandos), y por ello hay que prestar especial atención a la expansión de las variables de entorno en la línea de comando antes de que esta pase a ejecución, para eso usamos la opción *enabledelayedexpansion* de **setlocal**. Luego se utilizará el delimitador de variable "!", en vez del delimitador normal que es "%".

Por ejemplo, para la unión de la colección de capas raster sobre la altura media de ola significativa en la Península los pasos son los siguientes:

1. Configurar el fichero del entorno de trabajo de GRASS con la localización de Península

```
SET GISRC=C:\Users\tregmm\.grassrc6_peninsula
```

2. Configurar la región de trabajo dentro de la localización de Península

```
g.region vect=lin_costa_p ewres=300 nsres=300
```

3. Importación de la colección raster de altura media de ola significativa:

```
for %%f in ("C:\Users\tregmm\Desktop\OCEAN
LIDER_25062012\CAPAS
INFORMATIVAS\P\02_IT\RE\OL\HsMedia*img") do (r.in.gdal -o
input="%%f" output=%%~nf)
```

4. Configurar extensión de variables

```
setlocal enabledelayedexpansion
```

5. Resetar la variable 'var'

```
set var=
```

6. Recopilar en una variable los nombres de las capas raster de la altura media de ola significativa de Península

```
for %%f in ("C:\Users\tregmm\Desktop\OCEAN
LIDER_25062012\CAPAS
INFORMATIVAS\P\02_IT\RE\OL\HsMedia*img") do (set
var=%%~nf,!var!)
```

7. Unión de la colección raster de la altura media de ola significativa de Península

```
r.patch input=%var% output=HsMediaUnion_p
```

El script completo para la altura media de ola significativa con colección de capas raster en las tres entidades (Península, Baleares y Canarias) se detalla a continuación.

```
REM Unión de colección raster Hs de Península
SET GISRC=C:\Users\tregmm\.grassrc6_peninsula
g.region vect=lin_costa_p ewres=300 nsres=300
for %%f in ("C:\Users\tregmm\Desktop\OCEAN LIDER_25062012\CAPAS
INFORMATIVAS\P\02_IT\RE\OL\HsMedia*img") do (r.in.gdal -o
input="%%f" output=%%~nf)
setlocal enabledelayedexpansion
set var=
for %%f in ("C:\Users\tregmm\Desktop\OCEAN LIDER_25062012\CAPAS
INFORMATIVAS\P\02_IT\RE\OL\HsMedia*img") do (set var=%%~nf,!var!)
r.patch input=%var% output=HsMediaUnion_p
REM Unión de colección raster Hs de Baleares
SET GISRC=C:\Users\tregmm\.grassrc6_baleares
g.region vect=lin_costa_b ewres=300 nsres=300
```

```

for %%f in ("C:\Users\tregmm\Desktop\OCEAN LIDER_25062012\CAPAS
INFORMATIVAS\B\02_IT\RE\OL\HsMedia*img") do (r.in.gdal -o
input="%%f" output=%%~nf)
setlocal enabledelayedexpansion
set var=
for %%f in ("C:\Users\tregmm\Desktop\OCEAN LIDER_25062012\CAPAS
INFORMATIVAS\B\02_IT\RE\OL\HsMedia*img") do (set var=%%~nf,!var!)
r.patch input=%var% output=HsMediaUnion_b
REM Unión de colección raster Hs de Canarias
SET GISRC=C:\Users\tregmm\grassrc6_canarias
g.region vect=lin_costa_c ewres=300 nsres=300
for %%f in ("C:\Users\tregmm\Desktop\OCEAN LIDER_25062012\CAPAS
INFORMATIVAS\C\02_IT\RE\OL\HsMedia*img") do (r.in.gdal -o
input="%%f" output=%%~nf)
setlocal enabledelayedexpansion
set var=
for %%f in ("C:\Users\tregmm\Desktop\OCEAN LIDER_25062012\CAPAS
INFORMATIVAS\C\02_IT\RE\OL\HsMedia*img") do (set var=%%~nf,!var!)
r.patch input=%var% output=HsMediaUnion_c
REM Unión final y reproyección a WGS84
SET GISRC=C:\Users\tregmm\grassrc6_wgs84
REM important rows and cols
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300
rows=3224 cols=3453
r.proj input=HsMediaUnion_p location=Peninsula mapset=DSS
r.proj input=HsMediaUnion_b location=Baleares mapset=DSS
r.proj input=HsMediaUnion_c location=Canarias mapset=DSS
r.patch input=HsMediaUnion_p,HsMediaUnion_b,HsMediaUnion_c
output=HsMediaUnion
r.out.gdal input=HsMediaUnion
output="C:\Users\tregmm\Desktop\Union\unionhsmedia.tif"

```

En el *Anexo G* se dispone el fichero .BAT de la unión de diferentes capas raster informativas.

Capítulo

5

VALIDACIÓN Y PRUEBAS

En este apartado se va hacer un resumen de las validaciones, donde se mostrará un caso de prueba por cada criterio y por combinación. Se puede encontrar el desarrollo del plan de validación y pruebas completo en el *Anexo I*.

5.1. Resultado de las operaciones de los criterios y subcriterios propuestos para la herramienta DSS

A continuación se muestran los resultados de las pruebas realizadas sobre los criterios/subcriterios propuestos para la herramienta OceanLider. Las pruebas se han realizado de forma unitaria para cada criterio/subcriterio y se han aplicado sobre la costa y zona marítima de Andalucía.

Se van a mostrar los resultados con la paleta de colores de la *figura 161*, en el *anexo I* podemos encontrar los resultados con las dos paletas de colores propuestas para el proyecto.


COLOR	RANGO	RGB	TEXTO
	0	255, 0, 0	No Apta
	{0, 0.1}	254, 235, 226	Alta Restricción
	{0.1, 0.2}	252, 219, 192	
	{0.2, 0.3}	249, 178, 119	
	{0.3, 0.4}	247, 150, 70	
	{0.4, 0.5}	255, 255, 159	Media Restricción
	{0.5, 0.6}	248, 242, 0	
	{0.6, 0.7}	220, 255, 209	Baja Restricción
	{0.7, 0.8}	162, 255, 133	
	{0.8, 0.9}	97, 255, 47	
	{0.9, 1}	0, 208, 134	
	1	0, 176, 80	Apta
	-1	204, 102, 0	Tierra

Figura 161: Paleta de colores 1

5.1.1.1. Criterios físicos

- **Distancia a la costa.** De exclusión parcial. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa (valor A, valor B, valor C y valor D)

Los valores de los parámetros introducidos en la interfaz:

<input checked="" type="checkbox"/> Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km. 	
Valor A	8
Valor B	9
Valor C	20
Valor D	25
Peso del subcriterio:	0.0

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el criterio.

```

/* ***** OPERACIONES DE CONFIGURACIÓN ***** */

//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Balears
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_272
//Rasterización por valor de una capa vectorial
v.to.rast input=R_272 layer=1 type=point,line,area output=R_273 use=val
value=1 rows=4096
//Creación de un mapa de distancias
r.grow.distance input=R_273 distance=R_274 metric=Euclidean
//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_275
//Rasterización por valor de una capa vectorial
v.to.rast input=R_275 layer=1 type=point,line,area output=R_276 use=val
value=0 rows=4096
//Establecimiento de un valor para los valores nulos
r.null map=R_276 null=1
//Multiplicación de dos capas raster
r.mapcalc "R_277 = R_274 * R_276 "
//Multiplicación de una capa raster por un escalar
r.mapcalc "R_278 = R_277 * 100"

```

```
//Operación difusa
r.mapcalc "R_279 = if(isnull( R_278 ) , null(), if((R_278 <= 8 || R_278
>= 25), 0.0, if((R_278 > 8 && R_278 < 9), eval((R_278 - 8)/(9 - 8 )),
if((R_278 > 20 && R_278 < 25 ), eval((25 - R_278)/(25 - 20)), 1.0)))"
//Establecimiento de un valor para los valores nulos
r.null map=R_279 value=0

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_280 = if(isnull(ccaa),R_279, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.736927215493 s=34.847278777993 e=-3.830121796875 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_279
output="C:\Users\GSI\OceanLider\Resultados\2\RI_R_279.tif"
r.out.gdal input=R_280
output="C:\Users\GSI\OceanLider\Resultados\2\RF_R_280.tif"
```

Visualización del resultado:



- **Profundidad.** De exclusión parcial. Se aplica una operación difusa sobre la profundidad extrayendo los parámetros del dispositivo (parámetros: Profundidad mínima, ProfMin, y

profundidad máxima, ProfMax). Las medidas se aplican en metros. Se consideran los siguientes valores que procesa internamente:

valor A:= ProfMin-5m, valor B:= ProfMin, valor C:= ProfMax, valor D:= ProfMax+5m

Nota: Se puede proponer otra constante distinta a ± 5 metros para calcular los valores A y D.

Características del dispositivo

Nombre:	<input type="text" value="Dispositivo 2"/>
Tipo de dispositivo:	<input type="text" value="De corrientes - Fuera de costa: flotante"/>
Tipo de captador:	<input type="text" value="Eje horizontal"/>
Altura de buje:	<input type="text" value="5"/> metros
Diámetro rotor:	<input type="text" value="2"/> metros
Profundidad máxima:	<input type="text" value="-50"/> metros
Profundidad mínima:	<input type="text" value="-15"/> metros
Altura de ola máxima admitida:	<input type="text" value="0.2"/> metros
Velocidad de la corriente máxima admitida:	<input type="text" value="0.6"/> m/s
Potencia unitaria:	<input type="text" value="10"/> vatios
Separación entre dispositivos de una misma línea:	<input type="text" value="2"/> kilómetros
Separación entre líneas de dispositivos:	<input type="text" value="2"/> kilómetros

Selección en interfaz:

Profundidad. Se aplica una operación difusa sobre la profundidad extrayendo los parámetros del dispositivo. Las medidas se aplican en metros. 

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el criterio.


```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */

//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Baleares
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa raster ubicada en el sistema de
ficheros
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_281
//Operación difusa
r.mapcalc "R_282 = if(isnull( R_281 ) , null(), if((R_281 <= -55.0 ||
R_281 >= -10.0), 0.0, if((R_281 > -55.0 && R_281 < -50.0), eval((R_281 -
-55.0)/(-50.0 --55.0 )), if((R_281 > -15.0 && R_281 < -10.0 ), eval((-
10.0 - R_281 )/(-10.0 - -15.0)), 1.0)))"

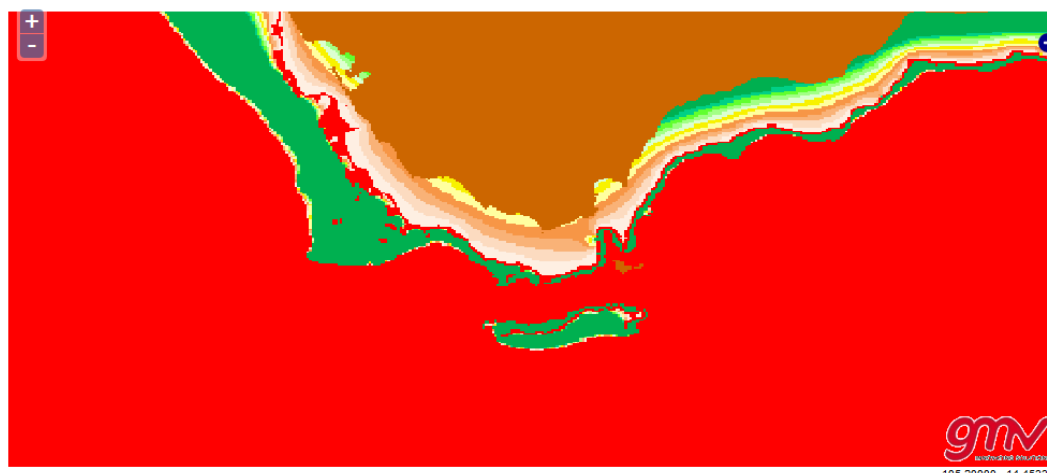
/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BEDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_283 = if(isnull(ccaa),R_282, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.716053259111 s=35.243885290361 e=-3.939985078125 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_282
output="C:\Users\GSI\OceanLider\Resultados\3\RI_R_282.tif"
r.out.gdal input=R_283
output="C:\Users\GSI\OceanLider\Resultados\3\RF_R_283.tif"
```

Visualización del resultado:



- **Supervivencia del dispositivo.** De exclusión total. El usuario puede seleccionar varios subcriterios. En caso que se seleccione más de un subcriterio se le da la opción al usuario de escoger el método de combinación de los subcriterios: método de razonamiento más que pesimista (multiplicación), pesimista (mínimo), optimista (máximo) y neutro (MLP).
 - ✓ **Altura de ola significativa.** Se extrae el valor mayor de ola significativa admitido configurado en el dispositivo, HsMax. Se hace una reclasificación sobre el raster de altura de ola media.

Si $Altura_ola_media(Hs) \geq 0$ & $Hs \leq HsMax$ entonces

Altura_ola_significante := 1

Sino

Altura_ola_significante := 0

Selección en la interfaz:

- Altura de ola significativa.** Se extrae el valor máximo de ola significativa admitido configurado en el dispositivo. 

Se muestran los resultados con una altura máxima admitida de 0.2 metros configurado en los parámetros del dispositivo:

Características del dispositivo

Nombre:

Tipo de dispositivo:

Tipo de captador:

Altura de buje: metros

Diámetro rotor: metros

Profundidad máxima: metros

Profundidad mínima: metros

Altura de ola máxima admitida: metros

Velocidad de la corriente máxima admitida: m/s

Potencia unitaria: vatios

Separación entre dispositivos de una misma línea: kilómetros

Separación entre líneas de dispositivos: kilómetros

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */  
  
//Configuración de la localización y Mapset del proyecto en el entorno  
de trabajo de GRASS  
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84  
//Configuración de la región que engloba a Península, Canarias y  
Balears  
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224  
cols=3453  
  
/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */  
  
//Importación en GRASS de una capa raster ubicada en el sistema de  
ficheros  
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\unionhsmmedia.tif"  
output=R_284
```

```

//Reclasificación de la capa raster
r.reclass input=R_284 output=R_285
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_285.txt"
//Copia de una capa raster
r.mapcalc "R_286 = R_285"
//Establecimiento de un valor para los valores nulos
r.null map=R_286 null=0

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_287 = if(isnull(ccaa),R_286, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.758899871743 s=35.418567840493 e=-3.896039765625 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_286
output="C:\Users\GSI\OceanLider\Resultados\4\RI_R_286.tif"
r.out.gdal input=R_287
output="C:\Users\GSI\OceanLider\Resultados\4\RF_R_287.tif"

```

Fichero de reclasificación "C:\Users\tregmm\OceanLider\Resultados\reclass_R_285.txt":

```

0 thru 0.2 = 1
* = 0

```

Visualización del resultado:



Los resultados de las pruebas del subcriterio *Velocidad de corriente* perteneciente a este criterio se encuentra en el *Anexo I*.

5.1.1.2. Criterios operacionales

- **Mantenimiento de la posición.** De exclusión parcial. El usuario puede seleccionar varios subcriterios. En caso que se seleccione más de un subcriterio, el método de razonamiento para la combinación de los subcriterios es el más que pesimista (multiplicación).
- ✓ *Facilidad de anclaje al fondo (composición del sustrato marino).* Se extraen los valores de idoneidad de las zonas configuradas para la capa Geofis dependiendo del tipo de dispositivo elegido y se le deja la libertad al usuario para poder cambiar estos valores a través de la interfaz.

Los valores de idoneidad configurados para cada agrupación por defecto para cada tipo de dispositivo se reflejan en la siguiente tabla.

NOMBRE	NOMBRE CAPA	METADATOS	Tipo de Capa	CORRIENTES		OLEAJE		
				FUERA DE COSTA (Apoyados sobre base de gravedad o pilotados)	FUERA DE COSTA (Fondeados)	FUERA DE COSTA (Apoyados sobre base de gravedad o pilotados)	FUERA DE COSTA (Flotantes Fondeados)	
MORFOLOGÍA FONDOS MARINOS								
Sustrato de los fondos	Geofis_P.shp	Cartografía vectorial poligonal de los tipos de sustrato de los fondos marinos del litoral de la Península Ibérica, Ceuta y Melilla. Se encuentra en ETRS89 huso 30.	Roca	Excluyente parcial	0,5	0,75	0,5	0,75
			Vegetado		1	1	1	1
			Blando Cohesivo		0	0,5	0	0,5
			Blando no Cohesivo		1	1	1	1

Características del dispositivo

Nombre:

Tipo de dispositivo:

Largo: metros

Ancho: metros

Alto: metros

Profundidad máxima: metros

Profundidad mínima: metros

Altura de ola máxima admitida: metros

Velocidad de la corriente máxima admitida: m/s

Potencia unitaria: vatios

Separación entre dispositivos de una misma línea: kilómetros

Separación entre líneas de dispositivos: kilómetros

Selección en la interfaz:

Facilidad de anclaje al fondo (composición del sustrato marino) ⓘ

Roca	<input type="text" value="0.5"/>
Vegetado	<input type="text" value="1"/>
Blando cohesivo	<input type="text" value="0"/>
Blando no cohesivo	<input type="text" value="1"/>

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```

/* ***** OPERACIONES DE CONFIGURACIÓN ***** */

//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
    
```

Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Balears
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributo
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_296 layer=geofis
where="Agrup='Roca'"
//Rasterización por valor de una capa vectorial
v.to.rast input=R_296 layer=1 type=point,line,area output=R_297 use=val
value=0.5 rows=4096
//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributo
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_298 layer=geofis
where="Agrup='Vegetado'"
//Rasterización por valor de una capa vectorial
v.to.rast input=R_298 layer=1 type=point,line,area output=R_299 use=val
value=1 rows=4096
//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributo
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_300 layer=geofis
where="Agrup='Blando cohesivo'"
//Rasterización por valor de una capa vectorial
v.to.rast input=R_300 layer=1 type=point,line,area output=R_301 use=val
value=0 rows=4096
//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributo
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_302 layer=geofis
where="Agrup='Blando no cohesivo'"
//Rasterización por valor de una capa vectorial
v.to.rast input=R_302 layer=1 type=point,line,area output=R_303 use=val
value=1 rows=4096
//Operación de unión de dos capas raster
r.patch input=R_297,R_299 output=R_304
r.patch input=R_301,R_303 output=R_305
r.patch input=R_304,R_305 output=R_306
//Establecimiento de un valor para los valores nulos
r.null map=R_306 null=1

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

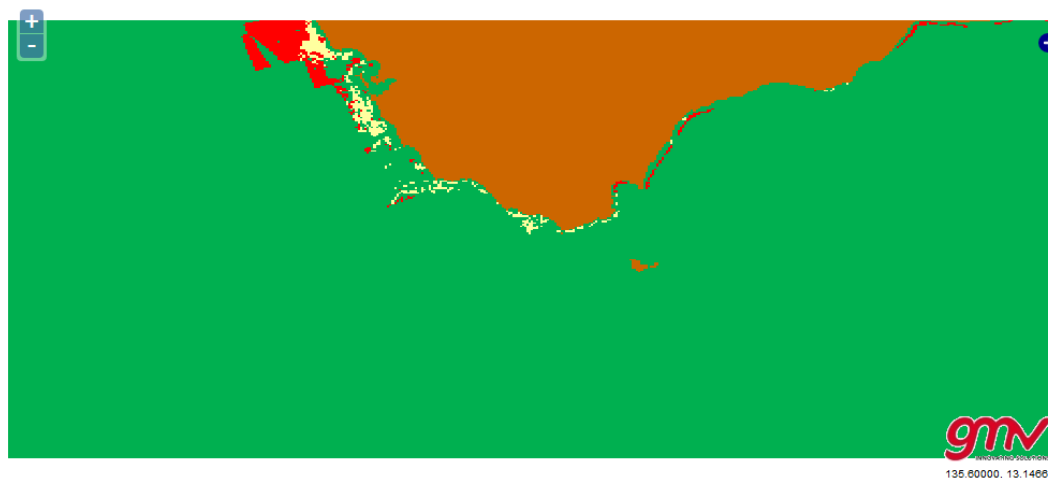
//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
```

```
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_307 = if(isnull(ccaa),R_306, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.716053259111 s=35.243885290361 e=-3.939985078125 w=-
7.477582734375 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_306
output="C:\Users\GSI\OceanLider\Resultados\7\RI_R_306.tif"
r.out.gdal input=R_307
output="C:\Users\GSI\OceanLider\Resultados\7\RF_R_307.tif"
```


Visualización del resultado:



Los resultados de las pruebas del *subcriterio Facilidad de anclaje al fondo (morfología del fondo marino)* perteneciente a este criterio se encuentra en el *Anexo I*.

- **Viabilidad de instalación en zonas ocupadas.** De exclusión parcial. El usuario puede seleccionar varios subcriterios. Si se selecciona más de un subcriterio el método de razonamiento para la combinación de subcriterios es el pesimista (mínimo de los valores).
 - ✓ **Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español.** Se aplica una operación difusa sobre la distancia a las zonas ocupadas por las ayudas a la navegación del sistema portuario español. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Los valores de los parámetros introducidos en la interfaz:

Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español. Se aplica una operación difusa sobre la distancia a las zonas ocupadas por las ayudas a la navegación del sistema portuario español. Las medidas se aplican en Km. 

De subida ▾	
Valor A	0.5
Valor B	1
Valor C	99999
Valor D	99999

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */

//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Baleares
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributos
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=aton output=R_332
//Rasterización por valor de una capa vectorial
v.to.rast input=R_332 layer=1 type=point,line,area output=R_333 use=val
value=0 rows=4096
//Cálculo de un mapa de distancias
r.grow.distance input=R_333 distance=R_334 metric=Euclidean
//Multiplicación de una capa raster por un escalar
r.mapcalc "R_335 = R_334 * 100"
//Operación difusa
r.mapcalc "R_336 = if(isnull( R_335 ), null(), if((R_335 <= 0.5 ||
R_335 >= 99999), 0.0, if((R_335 > 0.5 && R_335 < 1), eval((R_335 -
0.5)/(1 -0.5 )), if((R_335 > 99999 && R_335 < 99999 ), eval((99999 -
R_335)/(99999 - 99999)), 1.0)))"

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_337 = if(isnull(ccaa),R_336, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */
```

```
//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.868762817717 s=35.264758911467 e=-3.830121796875 w=-
7.477582734375 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_336
output="C:\Users\GSI\OceanLider\Resultados\10\RI_R_336.tif"
r.out.gdal input=R_337
output="C:\Users\GSI\OceanLider\Resultados\10\RF_R_337.tif"
```

Visualización del resultado:



Los resultados de las pruebas de los subcriterios: *Viabilidad de instalación en zonas designadas como Dominio Público Portuario, Viabilidad de instalación en zonas preseleccionadas para la extracción de áridos, Viabilidad de instalación en zonas ocupadas por conducciones submarinas, Viabilidad de instalación en zonas preseleccionadas para el vertido de material de dragado, Viabilidad de instalación en zonas ocupadas por sensores instrumentales de medición de parámetros oceanográficos y Viabilidad de instalación en zonas ocupadas por dispositivos de separación del tráfico marítimo*, que pertenecen a este criterio, se encuentra en el Anexo I.

- **Viabilidad de instalación en zonas restringidas.** De exclusión parcial. El usuario puede escoger varios subcriterios. Si se selecciona más de un subcriterio, el método de razonamiento para la combinación de subcriterios es el pesimista (mínimo de los valores).
 - ✓ *Viabilidad de instalación en zonas restringidas para uso militar.* Se aplica una operación difusa sobre la distancia a las zonas restringidas para uso militar. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Los valores de los parámetros introducidos en la interfaz:

Viabilidad de instalación en zonas restringidas para uso militar. Se aplica una operación difusa sobre la distancia a las zonas restringidas para uso militar. Las medidas se aplican en Km. 

De subida ▾	
Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="2"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */
//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Baleares
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributos
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=militar output=R_380
//Rasterización por valor de una capa vectorial
v.to.rast input=R_380 layer=1 type=point,line,area output=R_381 use=val
value=0 rows=4096
//Cálculo de un mapa de distancias
r.grow.distance input=R_381 distance=R_382 metric=Euclidean
//Multiplicación de una capa raster por un escalar
r.mapcalc "R_383 = R_382 * 100"
//Operación difusa
r.mapcalc "R_384 = if(isnull( R_383 ), null(), if((R_383 <= 1 || R_383
>= 99999), 0.0, if((R_383 > 1 && R_383 < 2), eval((R_383 - 1)/(2 -1 )),
if((R_383 > 99999 && R_383 < 99999 ), eval((99999 - R_383 )/(99999 -
99999)), 1.0)))"

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_385 = if(isnull(ccaa),R_384, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */
```

```
//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.846790161467 s=35.286731567717 e=-3.786176484375 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_384
output="C:\Users\GSI\OceanLider\Resultados\17\RI_R_384.tif"
r.out.gdal input=R_385
output="C:\Users\GSI\OceanLider\Resultados\17\RF_R_385.tif"
```

Visualización del resultado:



Los resultados de los subcriterios: *Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías*, *Viabilidad de instalación en zonas restringidas por la presencia de vertidos* y *Viabilidad de instalación en zonas restringidas por la existencia de vertidos de material militar*, pertenecientes a este criterio, se encuentran en el *Anexo I*.

5.1.1.3. Criterios recurso energético

La influencia del recurso energético en la toma de decisión se calcula mediante un único criterio. De exclusión parcial. Es dependiente del tipo de dispositivo: se aplica la densidad de potencia para tipos de dispositivo de oleaje, y la marea media para tipos de dispositivo de corrientes. Se aplica las siguientes fórmulas según los valores del raster de entrada y el parámetro *valor_máximo* configurado por el usuario.

- **Oleaje** (capa raster de *potencia*):
 - *De 0 a VALOR MÁXIMO KWh/m* aplicamos la ecuación: Valor índice= Valor del mapa raster/VALOR MÁXIMO
 - *+de VALOR MÁXIMO KWh/m*: Valor constante de 1

Características del dispositivo

Nombre:	<input type="text" value="Dispositivo 1"/>
Tipo de dispositivo:	<input type="text" value="Undimotriz - Fuera de costa: flotante"/>
Largo:	<input type="text" value="5"/> metros
Ancho:	<input type="text" value="2"/> metros
Alto:	<input type="text" value="10"/> metros
Profundidad máxima:	<input type="text" value="-100"/> metros
Profundidad mínima:	<input type="text" value="-20"/> metros
Altura de ola máxima admitida:	<input type="text" value="1.6"/> metros
Velocidad de la corriente máxima admitida:	<input type="text" value="0.6"/> m/s
Potencia unitaria:	<input type="text" value="10"/> vatios
Separación entre dispositivos de una misma línea:	<input type="text" value="2"/> kilómetros
Separación entre líneas de dispositivos:	<input type="text" value="2"/> kilómetros

Los valores de los parámetros introducidos en la interfaz:

Recurso energético

Peso criterio:

Densidad. Las unidades del valor máximo es en KWh/m. ⓘ

Valor Máximo

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el criterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */
```

```
//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Balears
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa raster ubicada en el sistema de
ficheros
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\pw.tif" output=R_417
//Aplicación de una fórmula en un intervalo sobre una capa raster
r.mapcalc "R_418= if(R_417 >= 0 && R_417 < 0.3, eval(R_417/0.3), 1)"
//Establecimiento de un valor para los valores nulos
r.null map=R_418 null=0

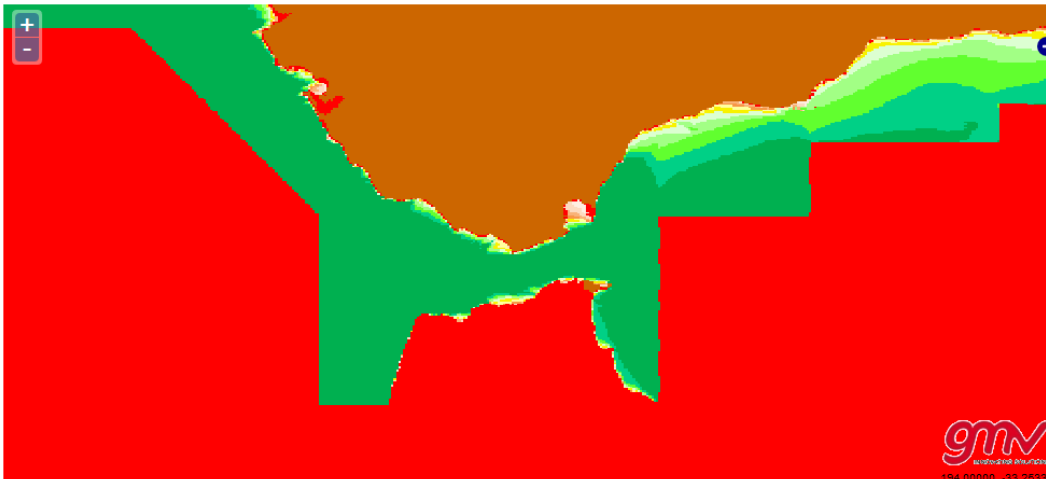
/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_419 = if(isnull(ccaa),R_418, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.825916540361 s=35.243885290361 e=-3.808149140625 w=-
7.301801484375 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_418
output="C:\Users\GSI\OceanLider\Resultados\22\RI_R_418.tif"
r.out.gdal input=R_419
output="C:\Users\GSI\OceanLider\Resultados\22\RF_R_419.tif"
```

Visualización del resultado:




El resultado del criterio de recurso energético para tipos de dispositivo **de corrientes** se encuentra en el *Anexo I*.

5.1.1.4. Criterios medioambientales

- **Espacios protegidos.** De exclusión parcial. El usuario puede seleccionar varios subcriterios. Si se seleccionan más de uno se le da la opción al usuario de escoger el método de razonamiento para la combinación de los subcriterios: más que pesimista (multiplicación de valores), pesimista (mínimo), optimista (máximo) y neutro (MLP).
- ✓ **Reservas de la biosfera.** Se aplican las áreas correspondientes a las reservas de la biosfera configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

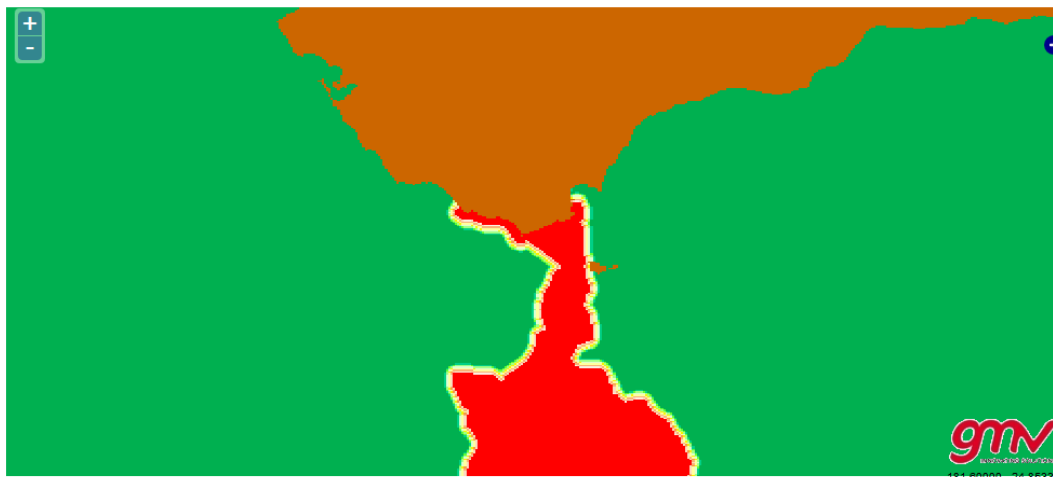
Reservas de la biosfera 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mabs output=R_453
v.to.rast input=R_453 layer=1 type=point,line,area output=R_454 use=val
value=0.0 rows=4096
r.grow.distance input=R_454 distance=R_455 metric=euclidean
r.mapcalc "R_456 = R_455 * 100"
r.mapcalc "R_457 = if(isnull( R_456 ), null(), if((R_456 <= 1 || R_456
>= 99999), 0.0, if((R_456 > 1 && R_456 < 4), eval((R_456 - 1)/(4 -1 )),
if((R_456 > 99999 && R_456 < 99999 ), eval((99999 - R_456 )/(99999 -
99999)), 1.0))))"
r.patch input=R_454,R_457 output=R_458
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_459 = if(isnull(ccaa),R_458, -1.0)"
g.region n=36.781971227861 s=35.177967321611 e=-3.764203828125 w=-
7.367719453125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_458
output="C:\Users\GSI\OceanLider\Resultados\26\RI_R_458.tif"
r.out.gdal input=R_459
output="C:\Users\GSI\OceanLider\Resultados\26\RF_R_459.tif"
```

Visualización del resultado:



Los resultados de las pruebas de los subcriterios: *Geoparques, Patrimonio de la humanidad, Red de Áreas Marinas Protegidas del Convenio OSPAR, Zonas Especialmente Protegidas de Importancia para el Mediterráneo, Sitios reconocidos con el Diploma Europeo de Calidad de Espacios Protegidos, Lugares de Importancia Comunitaria, Zonas de Especial Protección para las Aves, Parques Nacionales, Biotopos protegidos, Microrreservas, Monumentos Naturales, Parajes Naturales, Parques Naturales, Reservas con ley de protección propia, Reservas Naturales y Zonas de Especial Protección de los Valores Naturales*, pertenecientes a este criterio, se encuentran en el *Anexo I*.

- **Biodiversidad.** De exclusión parcial. El usuario puede seleccionar varios subcriterios. Si se seleccionan más de uno se le da la opción al usuario de escoger el método de razonamiento para la combinación de los subcriterios: más que pesimista (multiplicación de valores), pesimista (mínimo), optimista (máximo) y neutro (MLP).
 - ✓ **Áreas de interés para la conservación de los cetáceos.** Se aplican las áreas correspondientes a las áreas de interés para la conservación de los cetáceos configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Áreas de interés para la conservación de los cetáceos

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=aicp output=R_867
v.to.rast input=R_867 layer=1 type=point,line,area output=R_868 use=val
value=0.0 rows=4096
r.grow.distance input=R_868 distance=R_869 metric=euclidean
r.mapcalc "R_870 = R_869 * 100"
r.mapcalc "R_871 = if(isnull( R_870 ), null(), if((R_870 <= 1 || R_870
>= 99999), 0.0, if((R_870 > 1 && R_870 < 4), eval((R_870 - 1)/(4 -1 )),
if((R_870 > 99999 && R_870 < 99999 ), eval((99999 - R_870 )/(99999 -
99999)), 1.0))))"
r.patch input=R_868,R_871 output=R_872
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_873 = if(isnull(ccaa),R_872, -1.0)"
g.region n=36.803943884111 s=35.068104040361 e=-3.808149140625 w=-
7.433637421875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_872
output="C:\Users\GSI\OceanLider\Resultados\49\RI_R_872.tif"
r.out.gdal input=R_873
output="C:\Users\GSI\OceanLider\Resultados\49\RF_R_873.tif"
```


Visualización del resultado:



Los resultados de las pruebas de los subcriterios: *Áreas compatibles con la figura de Parque Nacional, Áreas importantes para las aves, Zonas propuestas para realizar el Inventario de la biodiversidad marina, Áreas importantes para las aves marinas, Zona marina especialmente sensible y Rutas migratorias de cetáceos*, pertenecientes a este criterio, se encuentran en el *Anexo I*.

- **Usos pesqueros.** De exclusión parcial. El usuario puede seleccionar varios subcriterios. Si se seleccionan más de uno se le da la opción al usuario de escoger el método de razonamiento para la combinación de los subcriterios: más que pesimista (multiplicación de valores), pesimista (mínimo), optimista (máximo) y neutro (MLP).
 - ✓ **Instalaciones de acuicultura.** Se aplican las áreas correspondientes a las instalaciones de acuicultura configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Instalaciones de acuicultura 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=acuicultura
output=R_916
v.to.rast input=R_916 layer=1 type=point,line,area output=R_917 use=val
value=0.0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=acuicultura_buf
output=R_918
v.to.rast input=R_918 layer=1 type=point,line,area output=R_919 use=val
value=0.0 rows=4096
r.patch input=R_917,R_919 output=R_920
r.grow.distance input=R_920 distance=R_921 metric=euclidean
r.mapcalc "R_922 = R_921 * 100"
r.mapcalc "R_923 = if(isnull( R_922 ), null(), if((R_922 <= 1 || R_922
>= 99999), 0.0, if((R_922 > 1 && R_922 < 4), eval((R_922 - 1)/(4 -1 )),
if((R_922 > 99999 && R_922 < 99999 ), eval((99999 - R_922 )/(99999 -
99999)), 1.0))))"
r.patch input=R_920,R_923 output=R_924
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_925 = if(isnull(ccaa),R_924, -1.0)"
g.region n=36.890735473967 s=35.045032348967 e=-3.742231171875 w=-
7.455610078125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_924
output="C:\Users\GSI\OceanLider\Resultados\56\RI_R_924.tif"
r.out.gdal input=R_925
output="C:\Users\GSI\OceanLider\Resultados\56\RF_R_925.tif"
```

Visualización del resultado:



Los resultados de las pruebas de los subcriterios: *Almadrabas, Arrecifes, Caladeros, Viveros, Zonas protegidas de interés pesquero, Piscifactorías, Reservas marinas y Zonas de producción de moluscos y de otros invertebrados marinos*, pertenecientes a este criterio, se encuentran en el *Anexo I*.

5.2. Resultados de la combinación de criterios

A continuación se muestra un extracto de los resultados de las pruebas de combinación realizadas sobre los criterios propuestos para la herramienta OceanLider. Las pruebas se han realizado en la costa y zona marítima de Andalucía. En el *Anexo I* se puede ver los resultados de las pruebas completas en la combinación de criterios.

Se van a mostrar los resultados con la paleta de colores de la *figura 161*, en el *anexo I* podemos encontrar los resultados con las dos paletas de colores propuestas para el proyecto.

5.2.1. Combinación de criterios físicos

5.2.1.1. Combinación 4

Combinación de dos criterios: **Distancia a la costa** (exclusión parcial) y **profundidad** (exclusión parcial). Se pueden combinar estos dos criterios parciales mediante los diferentes métodos de razonamiento: *más que pesimista, pesimista, optimista y MLP*.

Los parámetros que se han configurado para estos criterios son:

- ✓ Para el criterio distancia a la costa se han configurado los valores: valor A->8 kilómetros, valor B-> 9 kilómetros, valor C-> 20 kilómetros y valor D-> 25 kilómetros.

Distancia a la costa

Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km.

Valor A	8
Valor B	9
Valor C	20
Valor D	25
Peso del subcriterio:	0.0

- ✓ Para el criterio profundidad se extraen los parámetros profundidad máxima y mínima del dispositivo al que está asociado.

Características del dispositivo

Nombre:

Tipo de dispositivo:

Largo: metros

Ancho: metros

Alto: metros

Profundidad máxima: metros

Profundidad mínima: metros

Altura de ola máxima admitida: metros

Velocidad de la corriente máxima admitida: m/s

Potencia unitaria: vatios

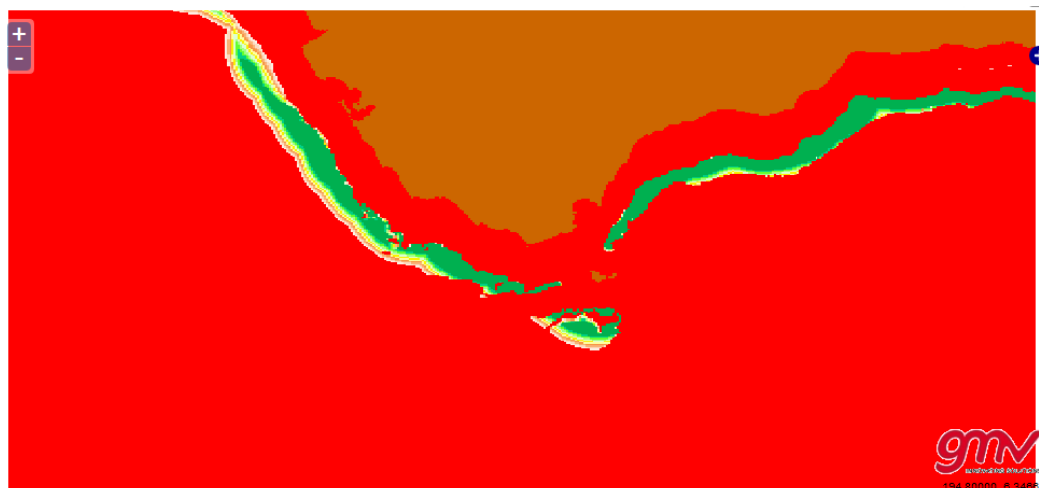
Separación entre dispositivos de una misma línea: kilómetros

Separación entre líneas de dispositivos: kilómetros

Para la combinación de los criterios distancia a la costa y profundidad utilizando el método de razonamiento: **pesimista** (mínimo de los valores)

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_1186
```

```
v.to.rast input=R_1186 layer=1 type=point,line,area output=R_1187
use=val value=1 rows=4096
r.grow.distance input=R_1187 distance=R_1188 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_1189
v.to.rast input=R_1189 layer=1 type=point,line,area output=R_1190
use=val value=0 rows=4096
r.null map=R_1190 null=1
r.mapcalc "R_1191 = R_1188 * R_1190 "
r.mapcalc "R_1192 = R_1191 * 100"
r.mapcalc "R_1193 = if(isnull( R_1192 ) , null(), if((R_1192 <= 8 ||
R_1192 >= 25), 0.0, if((R_1192 > 8 && R_1192 < 9), eval((R_1192 - 8)/(9
-8 )), if((R_1192 > 20 && R_1192 < 25 ), eval((25 - R_1192 )/(25 - 20)),
1.0))))"
r.null map=R_1193 value=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_1194
r.mapcalc "R_1195 = if(isnull( R_1194 ) , null(), if((R_1194 <= -105.0
|| R_1194 >= -15.0), 0.0, if((R_1194 > -105.0 && R_1194 < -100.0),
eval((R_1194 - -105.0)/(-100.0 --105.0 )), if((R_1194 > -20.0 && R_1194
< -15.0 ), eval((-15.0 - R_1194 )/(-15.0 - -20.0)), 1.0))))"
r.mapcalc "R_1196 = if(isnull( R_1193 ) || isnull( R_1195 ), null(),
if(R_1193 <= R_1195 , R_1193 , R_1195 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1197 = if(isnull(ccaa),R_1196, -1.0)"
g.region n=36.868762817717 s=35.088977661467 e=-3.742231171875 w=-
7.521528046875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1193
output="C:\Users\GSI\OceanLider\Resultados\75\RI_R_1193.tif"
r.out.gdal input=R_1195
output="C:\Users\GSI\OceanLider\Resultados\75\RI_R_1195.tif"
r.out.gdal input=R_1197
output="C:\Users\GSI\OceanLider\Resultados\75\RF_R_1197.tif"
```



5.2.2. Combinación de criterios operacionales

5.2.2.1. Combinación 8


Combinación de tres criterios operacionales: **mantenimiento de la posición** (de exclusión parcial), **viabilidad de instalación en zonas ocupadas** (de exclusión parcial) y **viabilidad de instalación en zonas restringidas** (de exclusión parcial).

- En el criterio **mantenimiento de la posición** han intervenido dos subcriterios que se combinan por defecto con el método de razonamiento: **más que pesimista** (multiplicación de valores): *facilidad de anclaje al fondo (composición del sustrato marino)* y *facilidad de anclaje al fondo (morfología del fondo marino)*.
- En el criterio **viabilidad de instalación en zonas ocupadas** han intervenido dos subcriterios que se combinan por defecto con el método **pesimista** (mínimo de los valores): *viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español* y *viabilidad de instalación en zonas designadas como Dominio Público Portuario*.
- En el criterio **viabilidad de instalación en zonas restringidas** han intervenido dos subcriterios que se combinan por defecto con el método de razonamiento **pesimista** (mínimo de los valores): *viabilidad de instalación en zonas restringidas para uso militar* y *viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías*.

Los parámetros establecidos en cada subcriterio:

- ✓ *Para el subcriterio de facilidad de anclaje al fondo (composición del sustrato marino):*
Se extraen los valores de idoneidad de las zonas configuradas para la capa Geofis dependiendo del tipo de dispositivo elegido (en este escenario se ha configurado un tipo

de dispositivo undimotriz- fuera costa (flotante)) y se le deja la libertad al usuario para poder cambiar estos valores a través de la interfaz. Los valores de idoneidad configurados en la interfaz:

Facilidad de anclaje al fondo (composición del sustrato marino) 


Roca	<input type="text" value="0.5"/>
Vegetado	<input type="text" value="1"/>
Blando cohesivo	<input type="text" value="0"/>
Blando no cohesivo	<input type="text" value="1"/>

- ✓ *Para el subcriterio de facilidad de anclaje al fondo (morfología al fondo marino):* Se extraen los valores de idoneidad de las zonas configuradas en la capa Formas dependiendo del tipo de dispositivo elegido (en este escenario se ha configurado un tipo de dispositivo undimotriz- fuera costa (flotante)) y se le deja la libertad al usuario de cambiar los valores según la agrupación para los diferentes tipos. Los valores de idoneidad configurados en la interfaz:

Facilidad de anclaje al fondo (morfología del fondo marino) 

Grupo0	<input type="text" value="0"/>
Grupo1	<input type="text" value="1"/>

- ✓ *Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español.*

Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español. Se aplica una operación difusa sobre la distancia a las zonas ocupadas por las ayudas a la navegación del sistema portuario español. Las medidas se aplican en Km. 

De subida ▾

Valor A	<input type="text" value="0.5"/>
Valor B	<input type="text" value="1"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>

- ✓ *Viabilidad de instalación en zonas desinadas como Dominio Público Portuario.*

Viabilidad de instalación en zonas designadas como Dominio Público Portuario. Se aplica una operación difusa sobre la distancia a las zonas designadas como Dominio Público Portuario. Las medidas se aplican en Km. 

De subida ▾

Valor A	<input type="text" value="0"/>
Valor B	<input type="text" value="1"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>

✓ *Viabilidad de instalación en zonas restringidas para uso militar.*

Viabilidad de instalación en zonas restringidas para uso militar. Se aplica una operación difusa sobre la distancia a las zonas restringidas para uso militar. Las medidas se aplican en Km. 

De subida ▼	
Valor A	1
Valor B	2
Valor C	99999
Valor D	99999

✓ *Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías.*

Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Las medidas se aplican en Km. 

De subida ▼	
Valor A	1.5
Valor B	3
Valor C	99999
Valor D	99999

El método de razonamiento que se ha escogido para la combinación de los tres criterios: **más que pesimista (multiplicación de valores)**

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1330 layer=geofis
where="Agrup='Roca'"
v.to.rast input=R_1330 layer=1 type=point,line,area output=R_1331
use=val value=0.5 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1332 layer=geofis
where="Agrup='Vegetado'"
v.to.rast input=R_1332 layer=1 type=point,line,area output=R_1333
use=val value=1 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1334 layer=geofis
where="Agrup='Blando cohesivo'"
v.to.rast input=R_1334 layer=1 type=point,line,area output=R_1335
use=val value=0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1336 layer=geofis
where="Agrup='Blando no cohesivo'"
v.to.rast input=R_1336 layer=1 type=point,line,area output=R_1337
use=val value=1 rows=4096
r.patch input=R_1331,R_1333 output=R_1338
r.patch input=R_1335,R_1337 output=R_1339
```

```
r.patch input=R_1338,R_1339 output=R_1340
r.null map=R_1340 null=1
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1341 layer=formas
where="(TIPO='Megaripples') OR (TIPO='Abanico deltaico') OR
(TIPO='Abanico deltaico profundo') OR (TIPO='Caynon submarino') OR
(TIPO='Caynon submarino') OR (TIPO='Llanuras abisales') OR
(TIPO='Morfologia de crestas y valles') OR (TIPO='Megaripples') OR
(TIPO='Ondas de arena') OR (TIPO='Relieve volcanico')"
v.to.rast input=R_1341 layer=1 type=point,line,area output=R_1342
use=val value=0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1343 layer=formas
where="(TIPO='Ripples') OR (TIPO='Marcas de arrastre') OR (TIPO='Talud
continental') OR (TIPO='Afloramiento rocoso') OR (TIPO='Playas
sumergidas') OR (TIPO='Plataforma continental')"
v.to.rast input=R_1343 layer=1 type=point,line,area output=R_1344
use=val value=1 rows=4096
r.patch input=R_1342,R_1344 output=R_1345
r.null map=R_1345 null=1
r.mapcalc "R_1366 = R_1340 * R_1345 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=zona_cables output=R_1346
v.to.rast input=R_1346 layer=1 type=point,line,area output=R_1347
use=val value=0 rows=4096
r.grow.distance input=R_1347 distance=R_1348 metric=euclidean
r.mapcalc "R_1349 = R_1348 * 100"
r.mapcalc "R_1350 = if(isnull( R_1349 ), null(), if((R_1349 <= 1.5 ||
R_1349 >= 15), 0.0, if((R_1349 > 1.5 && R_1349 < 3), eval((R_1349 -
1.5)/(3 - 1.5)), if((R_1349 > 6 && R_1349 < 15), eval((15 - R_1349
)/(15 - 6)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=militar output=R_1361
v.to.rast input=R_1361 layer=1 type=point,line,area output=R_1362
use=val value=0 rows=4096
r.grow.distance input=R_1362 distance=R_1363 metric=euclidean
r.mapcalc "R_1364 = R_1363 * 100"
r.mapcalc "R_1365 = if(isnull( R_1364 ), null(), if((R_1364 <= 1 ||
R_1364 >= 99999), 0.0, if((R_1364 > 1 && R_1364 < 2), eval((R_1364 -
1)/(2 - 1)), if((R_1364 > 99999 && R_1364 < 99999), eval((99999 -
R_1364)/(99999 - 99999)), 1.0))))"
r.mapcalc "R_1367 = if(isnull( R_1340 ) || isnull( R_1345 ), null(),
if(R_1340 <= R_1345 , R_1340 , R_1345 ))"
r.mapcalc "R_1368 = if(isnull( R_1367 ) || isnull( R_1345 ), null(),
if(R_1367 <= R_1345 , R_1367 , R_1345 ))"
r.mapcalc "R_1369 = if(isnull( R_1368 ) || isnull( R_1350 ), null(),
if(R_1368 <= R_1350 , R_1368 , R_1350 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=aton output=R_1351
v.to.rast input=R_1351 layer=1 type=point,line,area output=R_1352
use=val value=0 rows=4096
r.grow.distance input=R_1352 distance=R_1353 metric=euclidean
r.mapcalc "R_1354 = R_1353 * 100"
r.mapcalc "R_1355 = if(isnull( R_1354 ), null(), if((R_1354 <= 0.5 ||
R_1354 >= 99999), 0.0, if((R_1354 > 0.5 && R_1354 < 1), eval((R_1354 -
```

```

0.5)/(1 -0.5 )), if((R_1354 > 99999 && R_1354 < 99999 ), eval((99999 -
R_1354)/(99999 - 99999)), 1.0)))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=dpp output=R_1356
v.to.rast input=R_1356 layer=1 type=point,line,area output=R_1357
use=val value=0 rows=4096
r.grow.distance input=R_1357 distance=R_1358 metric=euclidean
r.mapcalc "R_1359 = R_1358 * 100"
r.mapcalc "R_1360 = if(isnull( R_1359 ), null(), if((R_1359 <= 0 ||
R_1359 >= 99999), 0.0, if((R_1359 > 0 && R_1359 < 1), eval((R_1359 -
0)/(1 -0 )), if((R_1359 > 99999 && R_1359 < 99999 ), eval((99999 -
R_1359)/(99999 - 99999)), 1.0))))"
r.mapcalc "R_1370 = if(isnull( R_1340 ) || isnull( R_1345 ), null(),
if(R_1340 <= R_1345 , R_1340 , R_1345 ))"
r.mapcalc "R_1371 = if(isnull( R_1370 ) || isnull( R_1345 ), null(),
if(R_1370 <= R_1345 , R_1370 , R_1345 ))"
r.mapcalc "R_1372 = if(isnull( R_1371 ) || isnull( R_1350 ), null(),
if(R_1371 <= R_1350 , R_1371 , R_1350 ))"
r.mapcalc "R_1373 = if(isnull( R_1372 ) || isnull( R_1365 ), null(),
if(R_1372 <= R_1365 , R_1372 , R_1365 ))"
r.mapcalc "R_1374 = if(isnull( R_1373 ) || isnull( R_1355 ), null(),
if(R_1373 <= R_1355 , R_1373 , R_1355 ))"
r.mapcalc "R_1375 = R_1366 * R_1369 * R_1374 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1376 = if(isnull(ccaa),R_1375, -1.0)"
g.region n=36.7819712227861 s=35.024158727861 e=-3.830121796875 w=-
7.433637421875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1340
output="C:\Users\GSI\OceanLider\Resultados\83\RI_R_1340.tif"
r.out.gdal input=R_1345
output="C:\Users\GSI\OceanLider\Resultados\83\RI_R_1345.tif"
r.out.gdal input=R_1350
output="C:\Users\GSI\OceanLider\Resultados\83\RI_R_1350.tif"
r.out.gdal input=R_1365
output="C:\Users\GSI\OceanLider\Resultados\83\RI_R_1365.tif"
r.out.gdal input=R_1355
output="C:\Users\GSI\OceanLider\Resultados\83\RI_R_1355.tif"
r.out.gdal input=R_1360
output="C:\Users\GSI\OceanLider\Resultados\83\RI_R_1360.tif"
r.out.gdal input=R_1376
output="C:\Users\GSI\OceanLider\Resultados\83\RF_R_1376.tif"

```



5.2.3. Combinación de criterios medioambientales

5.2.3.1. Combinación 10


Combinación de tres subcriterios utilizando el método de razonamiento MLP para su combinación **del criterio conservación de la biodiversidad** (de exclusión parcial): *reservas de la biosfera, lugares de importancia comunitaria y zonas de especial protección para las aves.*

Se le deja la libertad al usuario para configurar el valor de idoneidad para las áreas que intervienen en cada subcriterio y se le muestra uno por defecto dependiendo del tipo de dispositivo. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.


El tipo de dispositivo al que esté ligado el escenario: **undimotriz- fuera costa (flotante).**

Los parámetros establecidos en cada subcriterio:


- ✓ *Subcriterio Reservas de la biosfera*

<input checked="" type="checkbox"/> Reservas de la biosfera 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.7

✓ *Subcriterio Lugares de Importancia Comunitaria*

<input checked="" type="checkbox"/> Lugares de Importancia Comunitaria 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.2

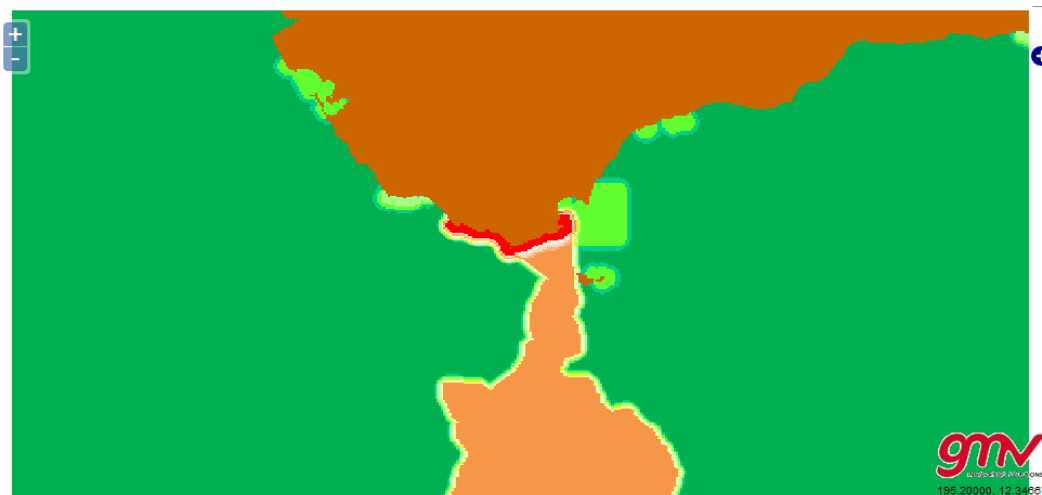
✓ *Subcriterio Zonas de Especial Protección para las Aves*

<input checked="" type="checkbox"/> Zonas de Especial Protección para las Aves 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.1

Los pesos de cada subcriterio que interviene: **0.7** a reservas de la biosfera, **0.2** a lugares de importancia comunitaria y **0.1** a zonas de especial protección para las aves.

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
```

```
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepas output=R_4443
v.to.rast input=R_4443 layer=1 type=point,line,area output=R_4444
use=val value=0.0 rows=4096
r.grow.distance input=R_4444 distance=R_4445 metric=euclidean
r.mapcalc "R_4446 = R_4445 * 100"
r.mapcalc "R_4447 = if(isnull( R_4446 ), null(), if((R_4446 <= 1 ||
R_4446 >= 99999), 0.0, if((R_4446 > 1 && R_4446 < 4), eval((R_4446 -
1)/(4 -1 )), if((R_4446 > 99999 && R_4446 < 99999 ), eval((99999 -
R_4446)/(99999 - 99999)), 1.0))))"
r.patch input=R_4444,R_4447 output=R_4448
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mabs output=R_4449
v.to.rast input=R_4449 layer=1 type=point,line,area output=R_4450
use=val value=0.0 rows=4096
r.grow.distance input=R_4450 distance=R_4451 metric=euclidean
r.mapcalc "R_4452 = R_4451 * 100"
r.mapcalc "R_4453 = if(isnull( R_4452 ), null(), if((R_4452 <= 1 ||
R_4452 >= 99999), 0.0, if((R_4452 > 1 && R_4452 < 4), eval((R_4452 -
1)/(4 -1 )), if((R_4452 > 99999 && R_4452 < 99999 ), eval((99999 -
R_4452)/(99999 - 99999)), 1.0))))"
r.patch input=R_4450,R_4453 output=R_4454
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=lics output=R_4455
v.to.rast input=R_4455 layer=1 type=point,line,area output=R_4456
use=val value=0.0 rows=4096
r.grow.distance input=R_4456 distance=R_4457 metric=euclidean
r.mapcalc "R_4458 = R_4457 * 100"
r.mapcalc "R_4459 = if(isnull( R_4458 ), null(), if((R_4458 <= 1 ||
R_4458 >= 99999), 0.0, if((R_4458 > 1 && R_4458 < 4), eval((R_4458 -
1)/(4 -1 )), if((R_4458 > 99999 && R_4458 < 99999 ), eval((99999 -
R_4458)/(99999 - 99999)), 1.0))))"
r.patch input=R_4456,R_4459 output=R_4460
r.mapcalc "R_4461 = 0.1 * R_4448 + 0.7 * R_4454 + 0.2 * R_4460 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_4462 = if(isnull(ccaa),R_4461, -1.0)"
g.region n=36.824817505217 s=35.132922973967 e=-3.808149140625 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_4448
output="C:\Users\GSI\OceanLider\Resultados\90\RI_R_4448.tif"
r.out.gdal input=R_4454
output="C:\Users\GSI\OceanLider\Resultados\90\RI_R_4454.tif"
r.out.gdal input=R_4460
output="C:\Users\GSI\OceanLider\Resultados\90\RI_R_4460.tif"
r.out.gdal input=R_4462
output="C:\Users\GSI\OceanLider\Resultados\90\RF_R_4462.tif"
```



5.2.4. Combinación de criterios físicos y criterios operacionales

5.2.4.1. Combinación 17

Combinación de **criterio distancia a la costa** (criterio físico de exclusión parcial), **criterio profundidad** (criterio físico de exclusión parcial), **criterio supervivencia del dispositivo** (criterio físico de exclusión total) y **criterio viabilidad de instalación en zonas restringidas** (criterio operacional de exclusión parcial).

En el **criterio supervivencia del dispositivo** intervienen los dos subcriterios: *altura de ola significativa* y *velocidad de corriente*. Se ha elegido el método de razonamiento **optimista** para su combinación.

En el **criterio viabilidad de instalación en zonas restringidas** intervienen los subcriterios: *viabilidad de instalación en zonas restringidas para uso militar*, *viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías*, *viabilidad de instalación en zonas restringidas por la presencia de vertidos* y *viabilidad de instalación en zonas restringidas por la existencia de vertidos de material militar*. Por defecto se combinan con el método de razonamiento **pesimista**.

Y el método de razonamiento elegido para la combinación de criterios físicos parciales ha sido el **pesimista**.

Los parámetros que intervienen:

- ✓ *En el subcriterio altura de ola significativa*. Se extrae el valor mayor de ola significativa admitido configurado en el dispositivo con un valor de 2.7 metros

Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

- ✓ En el subcriterio *velocidad de corriente*. Se extrae el valor de velocidad máxima de corriente configurado en el dispositivo con un valor de 1.7 m/s.
- ✓ En el criterio *profundidad*. Se extrae el valor de profundidad máxima y mínima configurado en el dispositivo con un valor de -100 y -20 metros respectivamente.
- ✓ En el criterio *distancia a la costa*.

Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km.

Valor A	8
Valor B	9
Valor C	20
Valor D	25
Peso del subcriterio:	0.0

- ✓ En el subcriterio *viabilidad de instalación en zonas restringidas para uso militar*.

Viabilidad de instalación en zonas restringidas para uso militar. Se aplica una operación difusa sobre la distancia a las zonas restringidas para uso militar. Las medidas se aplican en Km.

De subida ▾

Valor A	1
Valor B	2
Valor C	99999
Valor D	99999

- ✓ En el subcriterio *viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías*.

Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Las medidas se aplican en Km.

Normal ▾

Valor A	1.5
Valor B	3
Valor C	6
Valor D	15

- ✓ En el subcriterio *viabilidad de instalación en zonas restringidas por la presencia de vertidos*.

Viabilidad de instalación en zonas restringidas por la presencia de vertidos. Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de vertidos. Las medidas se aplican en Km.

De subida ▾	
Valor A	1
Valor B	2
Valor C	99999
Valor D	99999

✓ *En el subcriterio viabilidad de instalación en zonas restringidas por la existencia de vertidos de material militar.*

Viabilidad de instalación en zonas restringidas por la existencia de vertidos de material militar. Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de vertidos de material militar. Las medidas se aplican en Km.

De subida ▾	
Valor A	1
Valor B	2
Valor C	99999
Valor D	99999

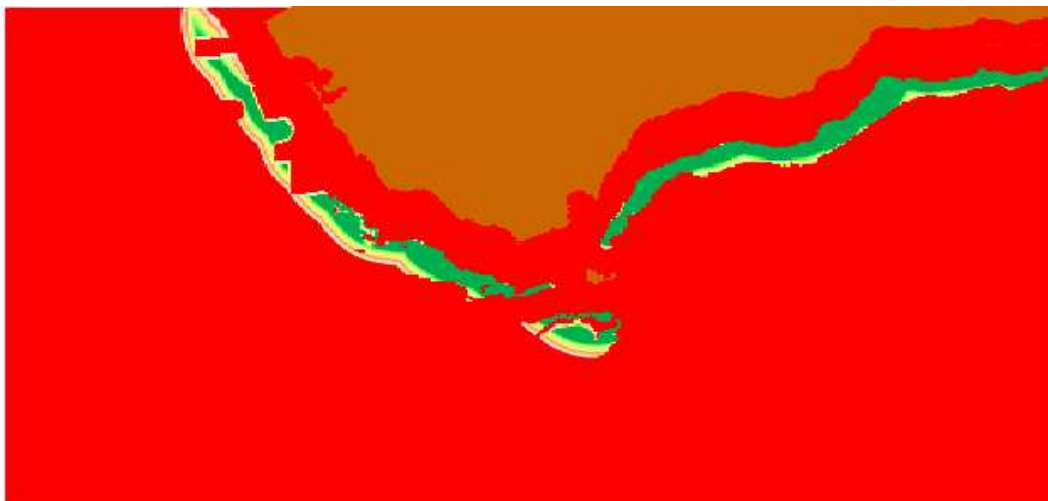
No hace falta configurar los parámetros de la toma de decisión final. Se han escogido criterios físicos y criterios operacionales, sólo interviene una rama en la toma de decisión.

Internamente el motor hace la combinación mediante el mínimo de los valores del resultado de los criterios físicos y el resultado de los criterios operacionales.

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\unionhsmmedia.tif"
output=R_3809
r.reclass input=R_3809 output=R_3810
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_3810.txt"
r.mapcalc "R_3811 = R_3810"
r.null map=R_3811 null=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\medviv.tif"
output=R_3817
r.reclass input=R_3817 output=R_3818
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_3818.txt"
r.mapcalc "R_3819 = R_3818"
r.null map=R_3819 null=0
r.mapcalc "R_3845 = if(isnull( R_3811 ) || isnull( R_3819 ), null(),
if(R_3811 >= R_3819 , R_3811 , R_3819 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_3825
v.to.rast input=R_3825 layer=1 type=point,line,area output=R_3826
use=val value=1 rows=4096
r.grow.distance input=R_3826 distance=R_3827 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_3828
v.to.rast input=R_3828 layer=1 type=point,line,area output=R_3829
use=val value=0 rows=4096
```

```
r.null map=R_3829 null=1
r.mapcalc "R_3830 = R_3827 * R_3829 "
r.mapcalc "R_3831 = R_3830 * 100"
r.mapcalc "R_3832 = if(isnull( R_3831 ) , null(), if((R_3831 <= 8 ||
R_3831 >= 25), 0.0, if((R_3831 > 8 && R_3831 < 9), eval((R_3831 - 8)/(9
-8 )), if((R_3831 > 20 && R_3831 < 25 ), eval((25 - R_3831)/(25 - 20)),
1.0))))"
r.null map=R_3832 value=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_3838
r.mapcalc "R_3839 = if(isnull( R_3838 ) , null(), if((R_3838 <= -105.0
|| R_3838 >= -15.0), 0.0, if((R_3838 > -105.0 && R_3838 < -100.0),
eval((R_3838 - -105.0)/(-100.0 --105.0 )), if((R_3838 > -20.0 && R_3838
< -15.0 ), eval((-15.0 - R_3838 )/(-15.0 - -20.0)), 1.0))))"
r.mapcalc "R_3846 = if(isnull( R_3832 ) || isnull( R_3839 ), null(),
if(R_3832 <= R_3839 , R_3832 , R_3839 ))"
r.mapcalc "R_3847 = if(isnull( R_3846 ) || isnull( R_3839 ), null(),
if(R_3846 <= R_3839 , R_3846 , R_3839 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=vertido_material_militar
output=R_3812
v.to.rast input=R_3812 layer=1 type=point,line,area output=R_3813
use=val value=0 rows=4096
r.grow.distance input=R_3813 distance=R_3814 metric=euclidean
r.mapcalc "R_3815 = R_3814 * 100"
r.mapcalc "R_3816 = if(isnull( R_3815 ) , null(), if((R_3815 <= 1 ||
R_3815 >= 99999), 0.0, if((R_3815 > 1 && R_3815 < 2), eval((R_3815 -
1)/(2 -1 )), if((R_3815 > 99999 && R_3815 < 99999 ), eval((99999 -
R_3815)/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=vertidos output=R_3820
v.to.rast input=R_3820 layer=1 type=point,line,area output=R_3821
use=val value=0 rows=4096
r.grow.distance input=R_3821 distance=R_3822 metric=euclidean
r.mapcalc "R_3823 = R_3822 * 100"
r.mapcalc "R_3824 = if(isnull( R_3823 ) , null(), if((R_3823 <= 1 ||
R_3823 >= 99999), 0.0, if((R_3823 > 1 && R_3823 < 2), eval((R_3823 -
1)/(2 -1 )), if((R_3823 > 99999 && R_3823 < 99999 ), eval((99999 -
R_3823)/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=militar output=R_3833
v.to.rast input=R_3833 layer=1 type=point,line,area output=R_3834
use=val value=0 rows=4096
r.grow.distance input=R_3834 distance=R_3835 metric=euclidean
r.mapcalc "R_3836 = R_3835 * 100"
r.mapcalc "R_3837 = if(isnull( R_3836 ) , null(), if((R_3836 <= 1 ||
R_3836 >= 99999), 0.0, if((R_3836 > 1 && R_3836 < 2), eval((R_3836 -
1)/(2 -1 )), if((R_3836 > 99999 && R_3836 < 99999 ), eval((99999 -
R_3836)/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=zona_cables output=R_3840
v.to.rast input=R_3840 layer=1 type=point,line,area output=R_3841
use=val value=0 rows=4096
r.grow.distance input=R_3841 distance=R_3842 metric=euclidean
r.mapcalc "R_3843 = R_3842 * 100"
```

```
r.mapcalc "R_3844 = if(isnull( R_3843 ) , null(), if((R_3843 <= 1.5 ||
R_3843 >= 15), 0.0, if((R_3843 > 1.5 && R_3843 < 3), eval((R_3843 -
1.5)/(3 -1.5 )), if((R_3843 > 6 && R_3843 < 15 ), eval((15 - R_3843
)/(15 - 6)), 1.0))))"
r.mapcalc "R_3848 = if(isnull( R_3816 ) || isnull( R_3824 ), null(),
if(R_3816 <= R_3824 , R_3816 , R_3824 ))"
r.mapcalc "R_3849 = if(isnull( R_3848 ) || isnull( R_3824 ), null(),
if(R_3848 <= R_3824 , R_3848 , R_3824 ))"
r.mapcalc "R_3850 = if(isnull( R_3849 ) || isnull( R_3837 ), null(),
if(R_3849 <= R_3837 , R_3849 , R_3837 ))"
r.mapcalc "R_3851 = if(isnull( R_3850 ) || isnull( R_3847 ), null(),
if(R_3850 <= R_3847 , R_3850 , R_3847 ))"
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_3852 = if(isnull(ccaa),R_3851, -1.0)"
g.region n=36.803943884111 s=34.980213415361 e=-3.874067109375 w=-
7.257856171875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_3811
output="C:\Users\GSI\OceanLider\Resultados\103\RI_R_3811.tif"
r.out.gdal input=R_3819
output="C:\Users\GSI\OceanLider\Resultados\103\RI_R_3819.tif"
r.out.gdal input=R_3832
output="C:\Users\GSI\OceanLider\Resultados\103\RI_R_3832.tif"
r.out.gdal input=R_3839
output="C:\Users\GSI\OceanLider\Resultados\103\RI_R_3839.tif"
r.out.gdal input=R_3816
output="C:\Users\GSI\OceanLider\Resultados\103\RI_R_3816.tif"
r.out.gdal input=R_3824
output="C:\Users\GSI\OceanLider\Resultados\103\RI_R_3824.tif"
r.out.gdal input=R_3837
output="C:\Users\GSI\OceanLider\Resultados\103\RI_R_3837.tif"
r.out.gdal input=R_3844
output="C:\Users\GSI\OceanLider\Resultados\103\RI_R_3844.tif"
r.out.gdal input=R_3852
output="C:\Users\GSI\OceanLider\Resultados\103\RF_R_3852.tif"
```




5.2.5. Combinación de criterios físicos y criterios de recurso energético

5.2.5.1. Combinación 18

Combinación del **criterio distancia a la costa** (criterio físico de exclusión parical) y **criterio de recurso energético- densidad** (criterio de recurso energético de exclusión parcial).

Los parámetros que intervienen:

✓ *En el criterio distancia a la costa.*

Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km. 

Valor A	<input type="text" value="8"/>
Valor B	<input type="text" value="9"/>
Valor C	<input type="text" value="20"/>
Valor D	<input type="text" value="25"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

✓ *En el criterio de recurso energético- densidad.* Se aplica por tener el tipo de dispositivo undimotriz ligado al escenario. Hay que tener en cuenta que al introducir valores máximos mayores que 0 el resultado tenderá a 0.

Recurso energético

Peso criterio:

Densidad. Las unidades del valor máximo es en KWh/m.

Valor Máximo

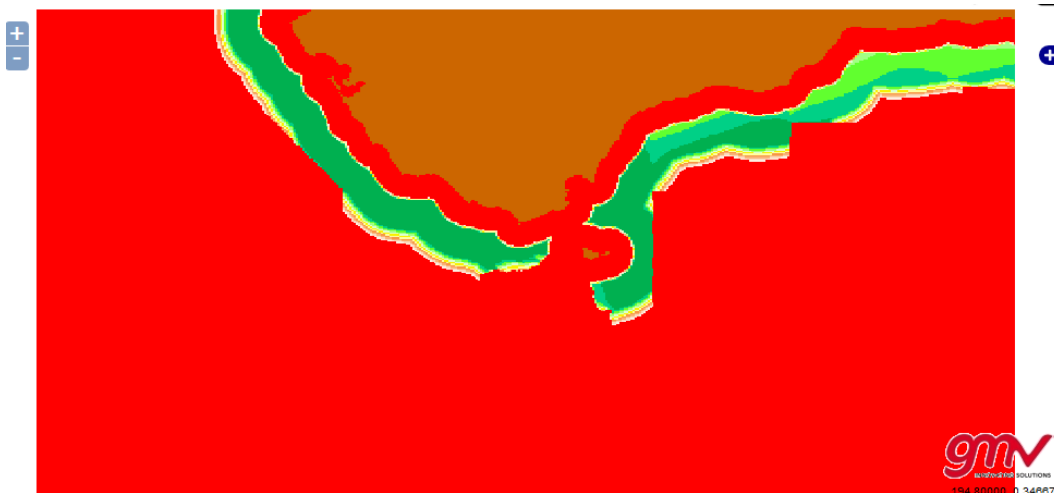
No hace falta configurar los parámetros de la toma de decisión final. Se han escogido criterios físicos y criterios de recurso energético, sólo interviene una rama en la toma de decisión.

Internamente el motor hace la combinación mediante el mínimo de los valores del resultado de los criterios físicos y el resultado de los criterios de recurso energético.

```

SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_3853
v.to.rast input=R_3853 layer=1 type=point,line,area output=R_3854
use=val value=1 rows=4096
r.grow.distance input=R_3854 distance=R_3855 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_3856
v.to.rast input=R_3856 layer=1 type=point,line,area output=R_3857
use=val value=0 rows=4096
r.null map=R_3857 null=1
r.mapcalc "R_3858 = R_3855 * R_3857 "
r.mapcalc "R_3859 = R_3858 * 100"
r.mapcalc "R_3860 = if(isnull( R_3859 ), null(), if((R_3859 <= 8 ||
R_3859 >= 25), 0.0, if((R_3859 > 8 && R_3859 < 9), eval((R_3859 - 8)/(9
-8)), if((R_3859 > 20 && R_3859 < 25 ), eval((25 - R_3859)/(25 - 20)),
1.0)))"
r.null map=R_3860 value=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\pw.tif" output=R_3861
r.mapcalc "R_3862= if(R_3861 >= 0 && R_3861 < 0.3, eval(R_3861/0.3), 1)"
r.null map=R_3862 null=0
r.mapcalc "R_3863 = if(isnull( R_3862 ) || isnull( R_3860 ), null(),
if(R_3862 <= R_3860 , R_3862 , R_3860 )"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_3864 = if(isnull(ccaa),R_3863, -1.0)"
g.region n=36.781971227861 s=35.002186071611 e=-3.808149140625 w=-
7.367719453125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_3860
output="C:\Users\GSI\OceanLider\Resultados\104\RI_R_3860.tif"
r.out.gdal input=R_3862
output="C:\Users\GSI\OceanLider\Resultados\104\RI_R_3862.tif"
r.out.gdal input=R_3864
output="C:\Users\GSI\OceanLider\Resultados\104\RF_R_3864.tif"

```



5.2.6. Combinación de criterios físicos y criterios medioambientales


5.2.6.1. Combinación 20

Combinación del criterio **distancia a la costa** (criterio físico de exclusión parcial) y **espacios protegidos** (criterio medio ambiental de exclusión parcial).

En el criterio de **espacios protegidos** intervienen los subcriterios: *reservas de la biosfera*, *lugares de importancia comunitaria* y *zonas de especial protección para las aves*. El método de razonamiento elegido para su combinación es el **pesimista**.


Los parámetros que intervienen:

- ✓ En el criterio *distancia a la costa*.

Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km. 


Valor A	<input type="text" value="8"/>
Valor B	<input type="text" value="9"/>
Valor C	<input type="text" value="20"/>
Valor D	<input type="text" value="25"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

- ✓ En el subcriterio *reservas de la biosfera*

Reservas de la biosfera 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ En el subcriterio de lugares de importancia comunitaria

Lugares de Importancia Comunitaria 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ En el subcriterio de zonas de especial protección para las aves

Zonas de Especial Protección para las Aves 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

Se le deja la libertad al usuario para configurar el valor de idoneidad para las áreas que intervienen en cada subcriterio medioambiental y se le muestra uno por defecto dependiendo del tipo de dispositivo. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

El tipo de dispositivo al que esté ligado el escenario: **undimotriz- fuera costa (flotante)**.

No hace falta configurar los parámetros de la toma de decisión final. Se han escogido criterios físicos y criterios medioambientales, sólo interviene una rama en la toma de decisión.

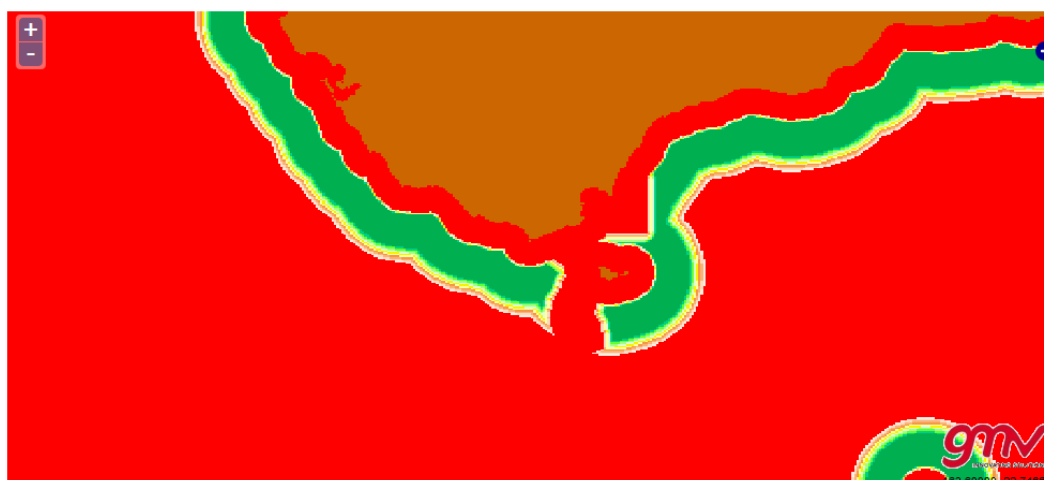
Internamente el motor hace la combinación mediante el mínimo de los valores del resultado de los criterios físicos y el resultado de los criterios medioambientales.

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_4058
v.to.rast input=R_4058 layer=1 type=point,line,area output=R_4059
use=val value=1 rows=4096
r.grow.distance input=R_4059 distance=R_4060 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_4061
v.to.rast input=R_4061 layer=1 type=point,line,area output=R_4062
use=val value=0 rows=4096
r.null map=R_4062 null=1
r.mapcalc "R_4063 = R_4060 * R_4062 "
r.mapcalc "R_4064 = R_4063 * 100"
r.mapcalc "R_4065 = if(isnull( R_4064 ) , null(), if((R_4064 <= 8 ||
R_4064 >= 25), 0.0, if((R_4064 > 8 && R_4064 < 9), eval((R_4064 - 8)/(9
-8 )), if((R_4064 > 20 && R_4064 < 25 ), eval((25 - R_4064 )/(25 - 20)),
1.0))))"
r.null map=R_4065 value=0
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mabs output=R_4052
v.to.rast input=R_4052 layer=1 type=point,line,area output=R_4053
use=val value=0.0 rows=4096
r.grow.distance input=R_4053 distance=R_4054 metric=euclidean
r.mapcalc "R_4055 = R_4054 * 100"
r.mapcalc "R_4056 = if(isnull( R_4055 ) , null(), if((R_4055 <= 1 ||
R_4055 >= 99999), 0.0, if((R_4055 > 1 && R_4055 < 4), eval((R_4055 -
1)/(4 -1 )), if((R_4055 > 99999 && R_4055 < 99999 ), eval((99999 -
R_4055 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4053,R_4056 output=R_4057
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=lics output=R_4066
v.to.rast input=R_4066 layer=1 type=point,line,area output=R_4067
use=val value=0.0 rows=4096
r.grow.distance input=R_4067 distance=R_4068 metric=euclidean
r.mapcalc "R_4069 = R_4068 * 100"
r.mapcalc "R_4070 = if(isnull( R_4069 ) , null(), if((R_4069 <= 1 ||
R_4069 >= 99999), 0.0, if((R_4069 > 1 && R_4069 < 4), eval((R_4069 -
1)/(4 -1 )), if((R_4069 > 99999 && R_4069 < 99999 ), eval((99999 -
R_4069 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4067,R_4070 output=R_4071
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepas output=R_4072
```

```

v.to.rast input=R_4072 layer=1 type=point,line,area output=R_4073
use=val value=0.0 rows=4096
r.grow.distance input=R_4073 distance=R_4074 metric=euclidean
r.mapcalc "R_4075 = R_4074 * 100"
r.mapcalc "R_4076 = if(isnull( R_4075 ), null(), if((R_4075 <= 1 ||
R_4075 >= 99999), 0.0, if((R_4075 > 1 && R_4075 < 4), eval((R_4075 -
1)/(4 - 1)), if((R_4075 > 99999 && R_4075 < 99999), eval((99999 -
R_4075)/(99999 - 99999)), 1.0))))"
r.patch input=R_4073,R_4076 output=R_4077
r.mapcalc "R_4078 = if(isnull( R_4057 ) || isnull( R_4071 ), null(),
if(R_4057 <= R_4071 , R_4057 , R_4071 )"
r.mapcalc "R_4079 = if(isnull( R_4078 ) || isnull( R_4071 ), null(),
if(R_4078 <= R_4071 , R_4078 , R_4071 )"
r.mapcalc "R_4080 = if(isnull( R_4079 ) || isnull( R_4065 ), null(),
if(R_4079 <= R_4065 , R_4079 , R_4065 )"
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_4081 = if(isnull(ccaa),R_4080, -1.0)"
g.region n=36.758899536467 s=35.198840942717 e=-3.874067109375 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_4065
output="C:\Users\GSI\OceanLider\Resultados\106\RI_R_4065.tif"
r.out.gdal input=R_4057
output="C:\Users\GSI\OceanLider\Resultados\106\RI_R_4057.tif"
r.out.gdal input=R_4071
output="C:\Users\GSI\OceanLider\Resultados\106\RI_R_4071.tif"
r.out.gdal input=R_4077
output="C:\Users\GSI\OceanLider\Resultados\106\RI_R_4077.tif"
r.out.gdal input=R_4081
output="C:\Users\GSI\OceanLider\Resultados\106\RF_R_4081.tif"

```



5.2.7. Combinación de criterios físicos, criterios operacionales y criterios medioambientales

5.2.7.1. Combinación 21

Combinación del criterio **profundidad** (criterio físico de exclusión parcial), **distancia a la costa** (criterio físico de exclusión parcial), **mantenimiento de la posición** (criterio operacional de exclusión parcial), **viabilidad de instalación en zonas ocupadas** (criterio operacional de exclusión parcial), **viabilidad de instalación en zonas restringidas** (criterio operacional de exclusión parcial), y **usos pesqueros** (criterio medio ambiental de exclusión parcial).

En el criterio **mantenimiento de la posición** intervienen los subcriterios: *facilidad de anclaje al fondo (composición del sustrato marino)* y *facilidad de anclaje al fondo (morfología del fondo marino)*. El método de razonamiento por defecto para su combinación es el **más que pesimista** (multiplicación de los valores).

En el criterio **viabilidad de instalación en zonas ocupadas** intervienen los subcriterios: *viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español* y *viabilidad de instalación en zonas designadas como dominio público portuario*. El método de razonamiento por defecto para su combinación es el **pesimista** (mínimo de los valores).

En el criterio **viabilidad de instalación en zonas restringidas** intervienen los subcriterios: *viabilidad de instalación en zonas restringidas para uso militar* y *viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías*. El método de razonamiento por defecto para su combinación es el **pesimista** (mínimo de los valores).

En el criterio de **usos pesqueros** intervienen los subcriterios: *arrecifes, caladeros y zonas de producción de moluscos y otros invertebrados marinos*. El método de razonamiento elegido es el **pesimista** (mínimo de los valores).

El método de razonamiento elegido para la combinación de los criterios físicos parciales es el **pesimista** (mínimo de los valores).

El método de razonamiento elegido para la combinación de los criterios operacionales parciales es el **pesimista** (mínimo de los valores).

Para la evaluación multicriterio, en la toma de decisión final se configuran los pesos:

- Rama de criterios operacionales: 0.25
- Rama de criterios medioambientales: 0.75

Y los pesos de orden:

- Peso de orden primero: 0.5
- Peso de orden segundo: 0.5

Los parámetros que intervienen:

- ✓ Criterio *profundidad*: los parámetros profundidad máxima y mínima: -100 metros y -20 metros configurados en los parámetros del dispositivo ligado al escenario.
- ✓ *En el criterio distancia a la costa.*

Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km.

Valor A	8
Valor B	9
Valor C	20
Valor D	25
Peso del subcriterio:	0.0

- ✓ *En el subcriterio de facilidad de anclaje al fondo (composición del sustrato marino)*: Se extraen los valores de idoneidad de las zonas configuradas para la capa Geofis dependiendo del tipo de dispositivo elegido (en este escenario se ha configurado un tipo de dispositivo undimotriz- fuera costa (flotante)) y se le deja la libertad al usuario para poder cambiar estos valores a través de la interfaz. Los valores de idoneidad configurados en la interfaz:

Facilidad de anclaje al fondo (composición del sustrato marino)


Roca	0.5
Vegetado	1
Blando cohesivo	0
Blando no cohesivo	1

- ✓ *En el subcriterio de facilidad de anclaje al fondo (morfología al fondo marino)*: Se extraen los valores de idoneidad de las zonas configuradas en la capa Formas dependiendo del tipo de dispositivo elegido (en este escenario se ha configurado un tipo de dispositivo undimotriz- fuera costa (flotante)) y se le deja la libertad al usuario de cambiar los valores según la agrupación para los diferentes tipos. Los valores de idoneidad configurados en la interfaz:

Facilidad de anclaje al fondo (morfología del fondo marino)

Grupo0	0
Grupo1	1

- ✓ *En el subcriterio viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español.*

Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español. Se aplica una operación difusa sobre la distancia a las zonas ocupadas por las ayudas a la navegación del sistema portuario español. Las medidas se aplican en Km. 

De subida ▾

Valor A	0.5
Valor B	1
Valor C	99999
Valor D	99999

✓ En el subcriterio viabilidad de instalación en zonas designadas como Dominio Público Portuario.

Viabilidad de instalación en zonas designadas como Dominio Público Portuario. Se aplica una operación difusa sobre la distancia a las zonas designadas como Dominio Público Portuario. Las medidas se aplican en Km. 

De subida ▾

Valor A	0
Valor B	1
Valor C	99999
Valor D	99999

✓ En el subcriterio viabilidad de instalación en zonas restringidas para uso militar.

Viabilidad de instalación en zonas restringidas para uso militar. Se aplica una operación difusa sobre la distancia a las zonas restringidas para uso militar. Las medidas se aplican en Km. 

De subida ▾

Valor A	1
Valor B	2
Valor C	99999
Valor D	99999


✓ En el subcriterio viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías.

Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Las medidas se aplican en Km. 

De subida ▾


Valor A	1.5
Valor B	3
Valor C	99999
Valor D	99999

✓ En el subcriterio arrecifes

Arrecifes 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *En el subcriterio caladeros*

Caladeros 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *En el subcriterio zonas de producción de moluscos y otros invertebrados marinos*

Zonas de producción de moluscos y de otros invertebrados marinos 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_4110
v.to.rast input=R_4110 layer=1 type=point,line,area output=R_4111
use=val value=1 rows=4096
```

```
r.grow.distance input=R_4111 distance=R_4112 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_4113
v.to.rast input=R_4113 layer=1 type=point,line,area output=R_4114
use=val value=0 rows=4096
r.null map=R_4114 null=1
r.mapcalc "R_4115 = R_4112 * R_4114 "
r.mapcalc "R_4116 = R_4115 * 100"
r.mapcalc "R_4117 = if(isnull( R_4116 ), null(), if((R_4116 <= 8 ||
R_4116 >= 25), 0.0, if((R_4116 > 8 && R_4116 < 9), eval((R_4116 - 8)/(9
-8 )), if((R_4116 > 20 && R_4116 < 25 ), eval((25 - R_4116 )/(25 - 20)),
1.0))))"
r.null map=R_4117 value=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_4128
r.mapcalc "R_4129 = if(isnull( R_4128 ), null(), if((R_4128 <= -105.0
|| R_4128 >= -15.0), 0.0, if((R_4128 > -105.0 && R_4128 < -100.0),
eval((R_4128 - -105.0)/(-100.0 --105.0 )), if((R_4128 > -20.0 && R_4128
< -15.0 ), eval((-15.0 - R_4128 )/(-15.0 - -20.0)), 1.0))))"
r.mapcalc "R_4146 = if(isnull( R_4117 ) || isnull( R_4129 ), null(),
if(R_4117 <= R_4129 , R_4117 , R_4129 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_4082 layer=geofis
where="Agrup='Roca'"
v.to.rast input=R_4082 layer=1 type=point,line,area output=R_4083
use=val value=0.5 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_4084 layer=geofis
where="Agrup='Vegetado'"
v.to.rast input=R_4084 layer=1 type=point,line,area output=R_4085
use=val value=1 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_4086 layer=geofis
where="Agrup='Blando cohesivo'"
v.to.rast input=R_4086 layer=1 type=point,line,area output=R_4087
use=val value=0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_4088 layer=geofis
where="Agrup='Blando no cohesivo'"
v.to.rast input=R_4088 layer=1 type=point,line,area output=R_4089
use=val value=1 rows=4096
r.patch input=R_4083,R_4085 output=R_4090
r.patch input=R_4087,R_4089 output=R_4091
r.patch input=R_4090,R_4091 output=R_4092
r.null map=R_4092 null=1
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_4130 layer=formas
where="(TIPO='Megaripples') OR (TIPO='Abanico deltaico') OR
(TIPO='Abanico deltaico profundo') OR (TIPO='Caynon submarino') OR
(TIPO='Caynon submarino') OR (TIPO='Llanuras abisales') OR
(TIPO='Morfologia de crestas y valles') OR (TIPO='Megaripples') OR
(TIPO='Ondas de arena') OR (TIPO='Relieve volcanico')"
v.to.rast input=R_4130 layer=1 type=point,line,area output=R_4131
use=val value=0 rows=4096
```

```

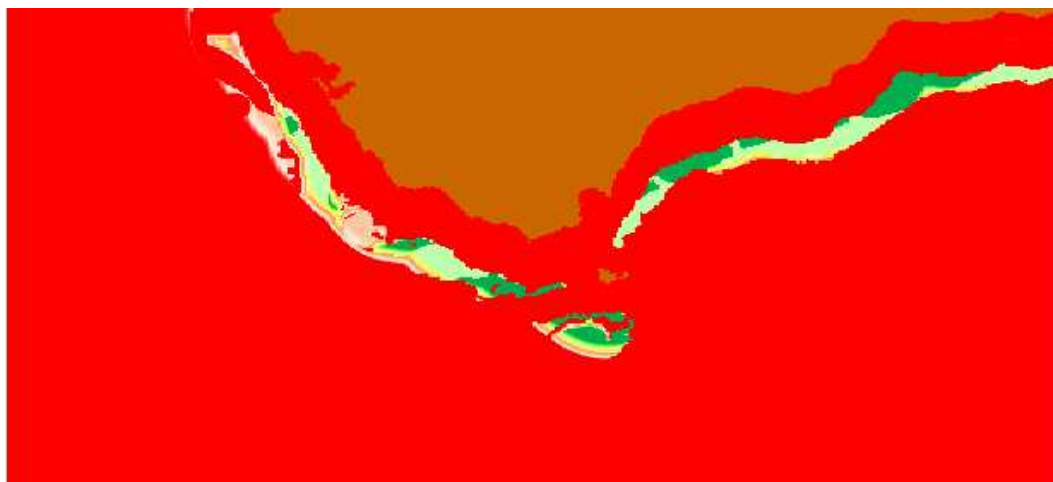
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_4132 layer=formas
where="(TIPO='Ripples') OR (TIPO='Marcas de arrastre') OR (TIPO='Talud
continental') OR (TIPO='Afloramiento rocoso') OR (TIPO='Playas
sumergidas') OR (TIPO='Plataforma continental')"
v.to.rast input=R_4132 layer=1 type=point,line,area output=R_4133
use=val value=1 rows=4096
r.patch input=R_4131,R_4133 output=R_4134
r.null map=R_4134 null=1
r.mapcalc "R_4147 = R_4092 * R_4134 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=aton output=R_4105
v.to.rast input=R_4105 layer=1 type=point,line,area output=R_4106
use=val value=0 rows=4096
r.grow.distance input=R_4106 distance=R_4107 metric=euclidean
r.mapcalc "R_4108 = R_4107 * 100"
r.mapcalc "R_4109 = if(isnull( R_4108 ) , null(), if((R_4108 <= 0.5 ||
R_4108 >= 99999), 0.0, if((R_4108 > 0.5 && R_4108 < 1), eval((R_4108 -
0.5)/(1 -0.5 )), if((R_4108 > 99999 && R_4108 < 99999 ), eval((99999 -
R_4108 )/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=dpp output=R_4118
v.to.rast input=R_4118 layer=1 type=point,line,area output=R_4119
use=val value=0 rows=4096
r.grow.distance input=R_4119 distance=R_4120 metric=euclidean
r.mapcalc "R_4121 = R_4120 * 100"
r.mapcalc "R_4122 = if(isnull( R_4121 ) , null(), if((R_4121 <= 0 ||
R_4121 >= 99999), 0.0, if((R_4121 > 0 && R_4121 < 1), eval((R_4121 -
0)/(1 -0 )), if((R_4121 > 99999 && R_4121 < 99999 ), eval((99999 -
R_4121 )/(99999 - 99999)), 1.0))))"
r.mapcalc "R_4148 = if(isnull( R_4092 ) || isnull( R_4134 ) , null(),
if(R_4092 <= R_4134 , R_4092 , R_4134 )"
r.mapcalc "R_4149 = if(isnull( R_4148 ) || isnull( R_4134 ) , null(),
if(R_4148 <= R_4134 , R_4148 , R_4134 )"
r.mapcalc "R_4150 = if(isnull( R_4149 ) || isnull( R_4109 ) , null(),
if(R_4149 <= R_4109 , R_4149 , R_4109 )"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=militar output=R_4123
v.to.rast input=R_4123 layer=1 type=point,line,area output=R_4124
use=val value=0 rows=4096
r.grow.distance input=R_4124 distance=R_4125 metric=euclidean
r.mapcalc "R_4126 = R_4125 * 100"
r.mapcalc "R_4127 = if(isnull( R_4126 ) , null(), if((R_4126 <= 1 ||
R_4126 >= 99999), 0.0, if((R_4126 > 1 && R_4126 < 2), eval((R_4126 -
1)/(2 -1 )), if((R_4126 > 99999 && R_4126 < 99999 ), eval((99999 -
R_4126 )/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=zona_cables output=R_4135
v.to.rast input=R_4135 layer=1 type=point,line,area output=R_4136
use=val value=0 rows=4096
r.grow.distance input=R_4136 distance=R_4137 metric=euclidean
r.mapcalc "R_4138 = R_4137 * 100"
r.mapcalc "R_4139 = if(isnull( R_4138 ) , null(), if((R_4138 <= 1.5 ||
R_4138 >= 15), 0.0, if((R_4138 > 1.5 && R_4138 < 3), eval((R_4138 -

```



```
1.5)/(3 -1.5 )), if((R_4138 > 6 && R_4138 < 15 ), eval((15 - R_4138
)/(15 - 6)), 1.0))))"
r.mapcalc "R_4151 = if(isnull( R_4092 ) || isnull( R_4134 ), null(),
if(R_4092 <= R_4134 , R_4092 , R_4134 ))"
r.mapcalc "R_4152 = if(isnull( R_4151 ) || isnull( R_4134 ), null(),
if(R_4151 <= R_4134 , R_4151 , R_4134 ))"
r.mapcalc "R_4153 = if(isnull( R_4152 ) || isnull( R_4109 ), null(),
if(R_4152 <= R_4109 , R_4152 , R_4109 ))"
r.mapcalc "R_4154 = if(isnull( R_4153 ) || isnull( R_4122 ), null(),
if(R_4153 <= R_4122 , R_4153 , R_4122 ))"
r.mapcalc "R_4155 = if(isnull( R_4154 ) || isnull( R_4127 ), null(),
if(R_4154 <= R_4127 , R_4154 , R_4127 ))"
r.mapcalc "R_4156 = R_4147 * R_4150 * R_4155 "
r.mapcalc "R_4157 = if(isnull( R_4156 ) || isnull( R_4146 ), null(),
if(R_4156 <= R_4146 , R_4156 , R_4146 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=cria_moluscos
output=R_4093
v.to.rast input=R_4093 layer=1 type=point,line,area output=R_4094
use=val value=0.0 rows=4096
r.grow.distance input=R_4094 distance=R_4095 metric=euclidean
r.mapcalc "R_4096 = R_4095 * 100"
r.mapcalc "R_4097 = if(isnull( R_4096 ) , null(), if((R_4096 <= 1 ||
R_4096 >= 99999), 0.0, if((R_4096 > 1 && R_4096 < 4), eval((R_4096 -
1)/(4 -1 )), if((R_4096 > 99999 && R_4096 < 99999 ), eval((99999 -
R_4096)/(99999 - 99999)), 1.0))))"
r.patch input=R_4094,R_4097 output=R_4098
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=caladeros output=R_4099
v.to.rast input=R_4099 layer=1 type=point,line,area output=R_4100
use=val value=0.0 rows=4096
r.grow.distance input=R_4100 distance=R_4101 metric=euclidean
r.mapcalc "R_4102 = R_4101 * 100"
r.mapcalc "R_4103 = if(isnull( R_4102 ) , null(), if((R_4102 <= 1 ||
R_4102 >= 99999), 0.0, if((R_4102 > 1 && R_4102 < 4), eval((R_4102 -
1)/(4 -1 )), if((R_4102 > 99999 && R_4102 < 99999 ), eval((99999 -
R_4102)/(99999 - 99999)), 1.0))))"
r.patch input=R_4100,R_4103 output=R_4104
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=arrecifes output=R_4140
v.to.rast input=R_4140 layer=1 type=point,line,area output=R_4141
use=val value=0.0 rows=4096
r.grow.distance input=R_4141 distance=R_4142 metric=euclidean
r.mapcalc "R_4143 = R_4142 * 100"
r.mapcalc "R_4144 = if(isnull( R_4143 ) , null(), if((R_4143 <= 1 ||
R_4143 >= 99999), 0.0, if((R_4143 > 1 && R_4143 < 4), eval((R_4143 -
1)/(4 -1 )), if((R_4143 > 99999 && R_4143 < 99999 ), eval((99999 -
R_4143)/(99999 - 99999)), 1.0))))"
r.patch input=R_4141,R_4144 output=R_4145
r.mapcalc "R_4158 = if(isnull( R_4098 ) || isnull( R_4104 ), null(),
if(R_4098 <= R_4104 , R_4098 , R_4104 ))"
r.mapcalc "R_4159 = if(isnull( R_4158 ) || isnull( R_4104 ), null(),
if(R_4158 <= R_4104 , R_4158 , R_4104 ))"
r.mapcalc "R_4160 = if(isnull( R_4159 ) || isnull( R_4146 ), null(),
if(R_4159 <= R_4146 , R_4159 , R_4146 ))"
```

```
r.mapcalc "R_4161 = if(isnull( R_4157 ) || isnull( R_4160 ),
null(),if(R_4157 >= R_4160 , eval(((R_4157 * 0.25 * 0.5) + (R_4160 *
0.75 * 0.5 )) / (0.25 * 0.5 + 0.75 * 0.5 )), eval(((R_4157 * R_4157 *
0.5 ) + (R_4160 * 0.75 * 0.5 )) / (0.25 * 0.5 + 0.75 * 0.5))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_4162 = if(isnull(ccaa),R_4161, -1.0)"
g.region n=36.714954223967 s=35.198840942717 e=-3.852094453125 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_4117
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4117.tif"
r.out.gdal input=R_4129
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4129.tif"
r.out.gdal input=R_4092
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4092.tif"
r.out.gdal input=R_4134
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4134.tif"
r.out.gdal input=R_4109
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4109.tif"
r.out.gdal input=R_4122
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4122.tif"
r.out.gdal input=R_4127
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4127.tif"
r.out.gdal input=R_4139
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4139.tif"
r.out.gdal input=R_4098
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4098.tif"
r.out.gdal input=R_4104
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4104.tif"
r.out.gdal input=R_4145
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4145.tif"
r.out.gdal input=R_4162
output="C:\Users\GSI\OceanLider\Resultados\107\RF_R_4162.tif"
```



5.2.8. Combinación de criterios físicos, criterios operacionales y criterios de recurso energético

5.2.8.1. Combinación 22

Combinación de criterio **distancia a la costa** (criterio físico de exclusión parcial), **viabilidad de instalación en zonas ocupadas** (criterio operacional de exclusión parcial), **viabilidad de instalación en zonas restringidas** (criterio operacional de exclusión parcial), y **densidad** (criterio de recurso energético de exclusión parcial).

En el criterio **viabilidad de instalación en zonas ocupadas** intervienen los subcriterios: *viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español* y *viabilidad de instalación en zonas designadas como dominio público portuario*. El método de razonamiento por defecto para su combinación es el **pesimista** (mínimo de los valores).

En el criterio **viabilidad de instalación en zonas restringidas** intervienen los subcriterios: *viabilidad de instalación en zonas restringidas para uso militar* y *viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías*. El método de razonamiento por defecto para su combinación es el **pesimista** (mínimo de los valores).

Para combinar los criterios operacionales parciales se ha elegido el método de razonamiento **MLP** con un peso de **0.6** para el *criterio viabilidad de instalación en zonas ocupadas* y **0.4** para el *criterio viabilidad de instalación en zonas restringidas*.

Para la evaluación multicriterio, en la toma de decisión final se configuran los pesos:


- Rama de criterios operacionales: 0.4
- Rama de criterios de recurso energético: 0.6

Y los pesos de orden:

- Peso de orden primero: 0.5
- Peso de orden segundo: 0.5

Los parámetros que intervienen:

✓ *En el criterio distancia a la costa.*

Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km. 

Valor A	<input type="text" value="8"/>
Valor B	<input type="text" value="9"/>
Valor C	<input type="text" value="20"/>
Valor D	<input type="text" value="25"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

- ✓ Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español.

Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español. Se aplica una operación difusa sobre la distancia a las zonas ocupadas por las ayudas a la navegación del sistema portuario español. Las medidas se aplican en Km.

De subida ▼

Valor A	0.5
Valor B	1
Valor C	99999
Valor D	99999

- ✓ Viabilidad de instalación en zonas desinadas como Dominio Público Portuario.

Viabilidad de instalación en zonas designadas como Dominio Público Portuario. Se aplica una operación difusa sobre la distancia a las zonas designadas como Dominio Público Portuario. Las medidas se aplican en Km.

De subida ▼

Valor A	0
Valor B	1
Valor C	99999
Valor D	99999

- ✓ Viabilidad de instalación en zonas restringidas para uso militar.

Viabilidad de instalación en zonas restringidas para uso militar. Se aplica una operación difusa sobre la distancia a las zonas restringidas para uso militar. Las medidas se aplican en Km.

De subida ▼

Valor A	1
Valor B	2
Valor C	99999
Valor D	99999

- ✓ Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías.

Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Las medidas se aplican en Km.

De subida ▼

Valor A	1.5
Valor B	3
Valor C	99999
Valor D	99999

- ✓ En el criterio recurso energético-densidad. Se aplica por tener el tipo de dispositivo undimotriz ligado al escenario. Hay que tener en cuenta que al introducir valores máximos mayores que 0 el resultado tenderá a 0.

Recurso energético

Peso criterio:

Densidad. Las unidades del valor máximo es en KWh/m. 

Valor Máximo

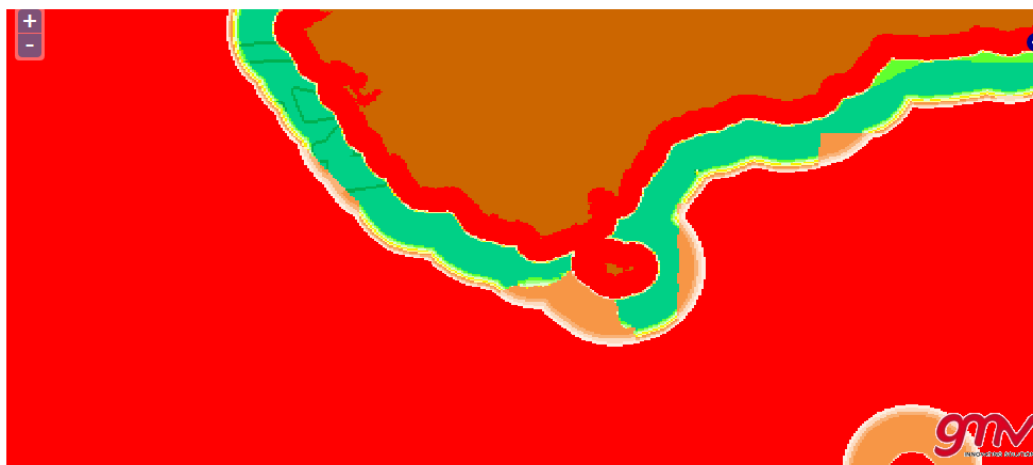
```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_5533
v.to.rast input=R_5533 layer=1 type=point,line,area output=R_5534
use=val value=1 rows=4096
r.grow.distance input=R_5534 distance=R_5535 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_5536
v.to.rast input=R_5536 layer=1 type=point,line,area output=R_5537
use=val value=0 rows=4096
r.null map=R_5537 null=1
r.mapcalc "R_5538 = R_5535 * R_5537 "
r.mapcalc "R_5539 = R_5538 * 100"
r.mapcalc "R_5540 = if(isnull( R_5539 ), null(), if((R_5539 <= 8 ||
R_5539 >= 25), 0.0, if((R_5539 > 8 && R_5539 < 9), eval((R_5539 - 8)/(9
-8 )), if((R_5539 > 20 && R_5539 < 25 ), eval((25 - R_5539 )/(25 - 20)),
1.0)))"
r.null map=R_5540 value=0
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=aton output=R_5528
v.to.rast input=R_5528 layer=1 type=point,line,area output=R_5529
use=val value=0 rows=4096
r.grow.distance input=R_5529 distance=R_5530 metric=euclidean
r.mapcalc "R_5531 = R_5530 * 100"
r.mapcalc "R_5532 = if(isnull( R_5531 ), null(), if((R_5531 <= 0.5 ||
R_5531 >= 99999), 0.0, if((R_5531 > 0.5 && R_5531 < 1), eval((R_5531 -
0.5)/(1 -0.5 )), if((R_5531 > 99999 && R_5531 < 99999 ), eval((99999 -
R_5531 )/(99999 - 99999)), 1.0)))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=dpp output=R_5541
v.to.rast input=R_5541 layer=1 type=point,line,area output=R_5542
use=val value=0 rows=4096
r.grow.distance input=R_5542 distance=R_5543 metric=euclidean
r.mapcalc "R_5544 = R_5543 * 100"
r.mapcalc "R_5545 = if(isnull( R_5544 ), null(), if((R_5544 <= 0 ||
R_5544 >= 99999), 0.0, if((R_5544 > 0 && R_5544 < 1), eval((R_5544 -
0)/(1 -0 )), if((R_5544 > 99999 && R_5544 < 99999 ), eval((99999 -
R_5544 )/(99999 - 99999)), 1.0)))"
r.mapcalc "R_5556 = if(isnull( R_5532 ) || isnull( R_5545 ), null(),
if(R_5532 <= R_5545 , R_5532 , R_5545 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=militar output=R_5546
```

```

v.to.rast input=R_5546 layer=1 type=point,line,area output=R_5547
use=val value=0 rows=4096
r.grow.distance input=R_5547 distance=R_5548 metric=euclidean
r.mapcalc "R_5549 = R_5548 * 100"
r.mapcalc "R_5550 = if(isnull( R_5549 ) , null(), if((R_5549 <= 1 ||
R_5549 >= 99999), 0.0, if((R_5549 > 1 && R_5549 < 2), eval((R_5549 -
1)/(2 -1 )), if((R_5549 > 99999 && R_5549 < 99999 ), eval((99999 -
R_5549 )/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=zona_cables output=R_5551
v.to.rast input=R_5551 layer=1 type=point,line,area output=R_5552
use=val value=0 rows=4096
r.grow.distance input=R_5552 distance=R_5553 metric=euclidean
r.mapcalc "R_5554 = R_5553 * 100"
r.mapcalc "R_5555 = if(isnull( R_5554 ) , null(), if((R_5554 <= 1.5 ||
R_5554 >= 15), 0.0, if((R_5554 > 1.5 && R_5554 < 3), eval((R_5554 -
1.5)/(3 -1.5 )), if((R_5554 > 6 && R_5554 < 15 ), eval((15 - R_5554
)/(15 - 6)), 1.0))))"
r.mapcalc "R_5557 = if(isnull( R_5532 ) || isnull( R_5545 ), null(),
if(R_5532 <= R_5545 , R_5532 , R_5545 )"
r.mapcalc "R_5558 = if(isnull( R_5557 ) || isnull( R_5545 ), null(),
if(R_5557 <= R_5545 , R_5557 , R_5545 )"
r.mapcalc "R_5559 = if(isnull( R_5558 ) || isnull( R_5550 ), null(),
if(R_5558 <= R_5550 , R_5558 , R_5550 )"
r.mapcalc "R_5560 = 0.6 * R_5556 + 0.4 * R_5559"
r.mapcalc "R_5561 = if(isnull( R_5560 ) || isnull( R_5540 ), null(),
if(R_5560 <= R_5540 , R_5560 , R_5540 )"
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\pw.tif" output=R_5526
r.mapcalc "R_5527= if(R_5526 >= 0 && R_5526 < 0.3, eval(R_5526/0.3), 1)"
r.null map=R_5527 null=0
r.mapcalc "R_5562 = if(isnull( R_5527 ) || isnull( R_5540 ), null(),
if(R_5527 <= R_5540 , R_5527 , R_5540 )"
r.mapcalc "R_5563 = if(isnull( R_5561 ) || isnull( R_5562 ),
null(),if(R_5561 >= R_5562 , eval(((R_5561 * 0.4 * 0.5) + (R_5562 * 0.6
* 0.5 )) / (0.4 * 0.5 + 0.6 * 0.5 )), eval(((R_5561 * R_5561 * 0.5 ) +
(R_5562 * 0.6 * 0.5 )) / (0.4 * 0.5 + 0.6 * 0.5))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_5564 = if(isnull(ccaa),R_5563, -1.0)"
g.region n=36.803943884111 s=35.199939977861 e=-3.830121796875 w=-
7.499555390625 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_5540
output="C:\Users\GSI\OceanLider\Resultados\108\RI_R_5540.tif"
r.out.gdal input=R_5532
output="C:\Users\GSI\OceanLider\Resultados\108\RI_R_5532.tif"
r.out.gdal input=R_5545
output="C:\Users\GSI\OceanLider\Resultados\108\RI_R_5545.tif"
r.out.gdal input=R_5550
output="C:\Users\GSI\OceanLider\Resultados\108\RI_R_5550.tif"
r.out.gdal input=R_5555
output="C:\Users\GSI\OceanLider\Resultados\108\RI_R_5555.tif"
r.out.gdal input=R_5527
output="C:\Users\GSI\OceanLider\Resultados\108\RI_R_5527.tif"

```

```
r.out.gdal input=R_5564  
output="C:\Users\GSI\OceanLider\Resultados\108\RF_R_5564.tif"
```



5.2.9. Combinación de criterios físicos, criterios de recurso energético y criterios medioambientales

5.2.9.1. Combinación 23

Combinación de los criterios: **distancia a la costa** (criterio físico de exclusión parcial), **densidad** (criterio de recurso energético de exclusión parcial), **espacios protegidos** (criterio medio ambiental de exclusión parcial) y **usos pesqueros** (criterio medio ambiental de exclusión parcial).

En el **criterio de espacios protegidos** intervienen tres subcriterios combinados con el método **optimista**: *subcriterio reservas de la biosfera*, *subcriterio lugares de importancia comunitaria y zonas de especial protección para las aves*. El método de razonamiento elegido para su combinación es el **pesimista** (mínimo de los valores).

En el **criterio de usos pesqueros** intervienen tres subcriterios combinados con el método **optimista**: *subcriterio arrecifes, caladeros y subcriterio zonas de producción de moluscos y de otros invertebrados marinos*. El método de razonamiento elegido para su combinación es el **pesimista** (mínimo de los valores).

Se le deja la libertad al usuario para configurar el valor de idoneidad para las áreas que intervienen en cada subcriterio medioambiental y se le muestra uno por defecto dependiendo del tipo de dispositivo. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

El tipo de dispositivo al que esté ligado el escenario: **undimotriz- fuera costa (flotante)**.

Para la combinación de criterios medioambientales de exclusión parcial se ha elegido el método de razonamiento: **optimista** (máximo de los valores).

Para la evaluación multicriterio, en la toma de decisión final se configuran los pesos:

- Rama de criterios de recurso energético: 0.4
- Rama de criterios medioambientales: 0.6

Y los pesos de orden:

- Peso de orden primero: 0.5
- Peso de orden segundo: 0.5

Los parámetros que intervienen:

- ✓ *En el criterio distancia a la costa.*

Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km. 🌐

Valor A	<input type="text" value="8"/>
Valor B	<input type="text" value="9"/>
Valor C	<input type="text" value="20"/>
Valor D	<input type="text" value="25"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

- *En el criterio recurso energético-densidad.* Se aplica por tener el tipo de dispositivo undimotriz ligado al escenario. Hay que tener en cuenta que al introducir valores máximos mayores que 0 el resultado tenderá a 0.


Recurso energético

Peso criterio:

Densidad. Las unidades del valor máximo es en KWh/m. 🌐


Valor Máximo

- ✓ *En el subcriterio reservas de la biosfera*

Reservas de la biosfera 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ En el subcriterio lugares de importancia comunitaria

Lugares de Importancia Comunitaria 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.2

✓ En el subcriterio zonas de especial protección para las aves


Zonas de Especial Protección para las Aves 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.1


✓ En el subcriterio arrecifes

<input checked="" type="checkbox"/> Arrecifes 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.6

✓ *En el subcriterio caladeros*

<input checked="" type="checkbox"/> Caladeros 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.3

✓ *En el subcriterio zonas de producción de moluscos y de otros invertebrados marinos*

<input checked="" type="checkbox"/> Zonas de producción de moluscos y de otros invertebrados marinos 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.1

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_4234
v.to.rast input=R_4234 layer=1 type=point,line,area output=R_4235
use=val value=1 rows=4096
r.grow.distance input=R_4235 distance=R_4236 metric=euclidean
```

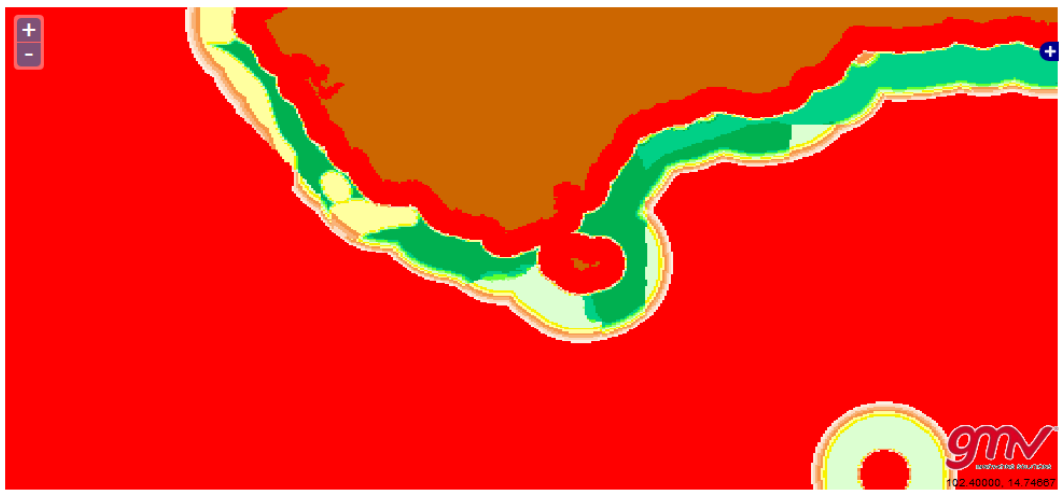
```
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_4237
v.to.rast input=R_4237 layer=1 type=point,line,area output=R_4238
use=val value=0 rows=4096
r.null map=R_4238 null=1
r.mapcalc "R_4239 = R_4236 * R_4238 "
r.mapcalc "R_4240 = R_4239 * 100"
r.mapcalc "R_4241 = if(isnull( R_4240 ) , null(), if((R_4240 <= 8 ||
R_4240 >= 25), 0.0, if((R_4240 > 8 && R_4240 < 9), eval((R_4240 - 8)/(9
- 8 ))), if((R_4240 > 20 && R_4240 < 25 ), eval((25 - R_4240 )/(25 - 20)),
1.0))))"
r.null map=R_4241 value=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\pw.tif" output=R_4208
r.mapcalc "R_4209= if(R_4208 >= 0 && R_4208 < 0.3, eval(R_4208/0.3), 1)"
r.null map=R_4209 null=0
r.mapcalc "R_4248 = if(isnull( R_4209 ) || isnull( R_4241 ), null(),
if(R_4209 <= R_4241 , R_4209 , R_4241 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=cria_moluscos
output=R_4202
v.to.rast input=R_4202 layer=1 type=point,line,area output=R_4203
use=val value=0.0 rows=4096
r.grow.distance input=R_4203 distance=R_4204 metric=euclidean
r.mapcalc "R_4205 = R_4204 * 100"
r.mapcalc "R_4206 = if(isnull( R_4205 ) , null(), if((R_4205 <= 1 ||
R_4205 >= 99999), 0.0, if((R_4205 > 1 && R_4205 < 4), eval((R_4205 -
1)/(4 - 1 ))), if((R_4205 > 99999 && R_4205 < 99999 ), eval((99999 -
R_4205 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4203,R_4206 output=R_4207
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=caladeros output=R_4216
v.to.rast input=R_4216 layer=1 type=point,line,area output=R_4217
use=val value=0.0 rows=4096
r.grow.distance input=R_4217 distance=R_4218 metric=euclidean
r.mapcalc "R_4219 = R_4218 * 100"
r.mapcalc "R_4220 = if(isnull( R_4219 ) , null(), if((R_4219 <= 1 ||
R_4219 >= 99999), 0.0, if((R_4219 > 1 && R_4219 < 4), eval((R_4219 -
1)/(4 - 1 ))), if((R_4219 > 99999 && R_4219 < 99999 ), eval((99999 -
R_4219 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4217,R_4220 output=R_4221
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=arrecifes output=R_4242
v.to.rast input=R_4242 layer=1 type=point,line,area output=R_4243
use=val value=0.0 rows=4096
r.grow.distance input=R_4243 distance=R_4244 metric=euclidean
r.mapcalc "R_4245 = R_4244 * 100"
r.mapcalc "R_4246 = if(isnull( R_4245 ) , null(), if((R_4245 <= 1 ||
R_4245 >= 99999), 0.0, if((R_4245 > 1 && R_4245 < 4), eval((R_4245 -
1)/(4 - 1 ))), if((R_4245 > 99999 && R_4245 < 99999 ), eval((99999 -
R_4245 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4243,R_4246 output=R_4247
r.mapcalc "R_4249 = if(isnull( R_4207 ) || isnull( R_4221 ), null(),
if(R_4207 <= R_4221 , R_4207 , R_4221 ))"
r.mapcalc "R_4250 = if(isnull( R_4249 ) || isnull( R_4221 ), null(),
if(R_4249 <= R_4221 , R_4249 , R_4221 ))"
```

```

v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=lics output=R_4210
v.to.rast input=R_4210 layer=1 type=point,line,area output=R_4211
use=val value=0.0 rows=4096
r.grow.distance input=R_4211 distance=R_4212 metric=euclidean
r.mapcalc "R_4213 = R_4212 * 100"
r.mapcalc "R_4214 = if(isnull( R_4213 ) , null(), if((R_4213 <= 1 ||
R_4213 >= 99999), 0.0, if((R_4213 > 1 && R_4213 < 4), eval((R_4213 -
1)/(4 -1 )), if((R_4213 > 99999 && R_4213 < 99999 ), eval((99999 -
R_4213)/(99999 - 99999)), 1.0))))"
r.patch input=R_4211,R_4214 output=R_4215
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepas output=R_4222
v.to.rast input=R_4222 layer=1 type=point,line,area output=R_4223
use=val value=0.0 rows=4096
r.grow.distance input=R_4223 distance=R_4224 metric=euclidean
r.mapcalc "R_4225 = R_4224 * 100"
r.mapcalc "R_4226 = if(isnull( R_4225 ) , null(), if((R_4225 <= 1 ||
R_4225 >= 99999), 0.0, if((R_4225 > 1 && R_4225 < 4), eval((R_4225 -
1)/(4 -1 )), if((R_4225 > 99999 && R_4225 < 99999 ), eval((99999 -
R_4225)/(99999 - 99999)), 1.0))))"
r.patch input=R_4223,R_4226 output=R_4227
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mabs output=R_4228
v.to.rast input=R_4228 layer=1 type=point,line,area output=R_4229
use=val value=0.0 rows=4096
r.grow.distance input=R_4229 distance=R_4230 metric=euclidean
r.mapcalc "R_4231 = R_4230 * 100"
r.mapcalc "R_4232 = if(isnull( R_4231 ) , null(), if((R_4231 <= 1 ||
R_4231 >= 99999), 0.0, if((R_4231 > 1 && R_4231 < 4), eval((R_4231 -
1)/(4 -1 )), if((R_4231 > 99999 && R_4231 < 99999 ), eval((99999 -
R_4231)/(99999 - 99999)), 1.0))))"
r.patch input=R_4229,R_4232 output=R_4233
r.mapcalc "R_4251 = if(isnull( R_4207 ) || isnull( R_4221 ) , null(),
if(R_4207 <= R_4221 , R_4207 , R_4221 )"
r.mapcalc "R_4252 = if(isnull( R_4251 ) || isnull( R_4221 ) , null(),
if(R_4251 <= R_4221 , R_4251 , R_4221 )"
r.mapcalc "R_4253 = if(isnull( R_4252 ) || isnull( R_4247 ) , null(),
if(R_4252 <= R_4247 , R_4252 , R_4247 )"
r.mapcalc "R_4254 = if(isnull( R_4253 ) || isnull( R_4215 ) , null(),
if(R_4253 <= R_4215 , R_4253 , R_4215 )"
r.mapcalc "R_4255 = if(isnull( R_4254 ) || isnull( R_4227 ) , null(),
if(R_4254 <= R_4227 , R_4254 , R_4227 )"
r.mapcalc "R_4256 = if(isnull( R_4250 ) || isnull( R_4255 ) , null(),
if(R_4250 >= R_4255 , R_4250 , R_4255 )"
r.mapcalc "R_4257 = if(isnull( R_4256 ) || isnull( R_4241 ) , null(),
if(R_4256 <= R_4241 , R_4256 , R_4241 )"
r.mapcalc "R_4258 = if(isnull( R_4257 ) || isnull( R_4248 ) ,
null(),if(R_4257 >= R_4248 , eval(((R_4257 * 0.6 * 0.5) + (R_4248 * 0.4
* 0.5 )) / (0.6 * 0.5 + 0.4 * 0.5 )), eval(((R_4257 * R_4257 * 0.5 ) +
(R_4248 * 0.4 * 0.5 )) / (0.6 * 0.5 + 0.4 * 0.5))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096

```

```
r.mapcalc "R_4259 = if(isnull(ccaa),R_4258, -1.0)"
g.region n=36.781971227861 s=35.112049352861 e=-3.698285859375 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_4241
output="C:\Users\GSI\OceanLider\Resultados\109\RI_R_4241.tif"
r.out.gdal input=R_4207
output="C:\Users\GSI\OceanLider\Resultados\109\RI_R_4207.tif"
r.out.gdal input=R_4221
output="C:\Users\GSI\OceanLider\Resultados\109\RI_R_4221.tif"
r.out.gdal input=R_4247
output="C:\Users\GSI\OceanLider\Resultados\109\RI_R_4247.tif"
r.out.gdal input=R_4215
output="C:\Users\GSI\OceanLider\Resultados\109\RI_R_4215.tif"
r.out.gdal input=R_4227
output="C:\Users\GSI\OceanLider\Resultados\109\RI_R_4227.tif"
r.out.gdal input=R_4233
output="C:\Users\GSI\OceanLider\Resultados\109\RI_R_4233.tif"
r.out.gdal input=R_4209
output="C:\Users\GSI\OceanLider\Resultados\109\RI_R_4209.tif"
r.out.gdal input=R_4259
output="C:\Users\GSI\OceanLider\Resultados\109\RF_R_4259.tif"
```



5.2.10. Combinación de criterios físicos, criterios operacionales, criterios medioambientales y criterios de recurso energético

5.2.10.1. Combinación 24

Combinación de los criterios: **profundidad** (criterio físico de exclusión parcial), **viabilidad de instalación en zonas ocupadas** (criterio operacional de exclusión parcial), **espacios protegidos** (criterio medio ambiental de exclusión parcial), y **densidad** (criterio de recurso energético de exclusión parcial).

En el criterio **viabilidad de instalación en zonas ocupadas** intervienen los subcriterios: *viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español* y *viabilidad de instalación en zonas designadas como dominio público portuario*. El método de razonamiento por defecto para su combinación es el **pesimista** (mínimo de los valores).

En el **criterio de espacios protegidos** intervienen tres subcriterios combinados con el método **pesimista**: *subcriterio reservas de la biosfera*, *subcriterio lugares de importancia comunitaria* y *zonas de especial protección para las aves*. El método de razonamiento elegido para su combinación es el **pesimista** (mínimo de los valores).

Se le deja la libertad al usuario para configurar el valor de idoneidad para las áreas que intervienen en cada subcriterio medioambiental y se le muestra uno por defecto dependiendo del tipo de dispositivo. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

El tipo de dispositivo al que esté ligado el escenario: **undimotriz- fuera costa (flotante)**.

Para la evaluación multicriterio, en la toma de decisión final se configuran los pesos:


- Rama de criterios operacionales: 0.2
- Rama de criterios medioambientales: 0.5
- Rama de criterios de recurso energético: 0.3

Y los pesos de orden:

- Peso de orden primero: 0.33
- Peso de orden segundo: 0.33
- Peso de orden tercero: 0.33

Los parámetros que intervienen:

- ✓ El criterio *profundidad*: los parámetros profundidad máxima y mínima del dispositivo al que está ligado y tiene los valores -100 y -20, respectivamente.
- ✓ Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español.

Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español. Se aplica una operación difusa sobre la distancia a las zonas ocupadas por las ayudas a la navegación del sistema portuario español. Las medidas se aplican en Km. 

De subida ▾

Valor A	0.5
Valor B	1
Valor C	99999
Valor D	99999

✓ *Viabilidad de instalación en zonas desinadas como Dominio Público Portuario.*

Viabilidad de instalación en zonas designadas como Dominio Público Portuario. Se aplica una operación difusa sobre la distancia a las zonas designadas como Dominio Público Portuario. Las medidas se aplican en Km. 

De subida ▾

Valor A	0
Valor B	1
Valor C	99999
Valor D	99999

✓ *En el criterio recurso energético-densidad.* Se aplica por tener el tipo de dispositivo undimotriz ligado al escenario. Hay que tener en cuenta que al introducir valores máximos mayores que 0 el resultado tenderá a 0.

Recurso energético

Peso criterio:

Densidad. Las unidades del valor máximo es en KWh/m. 

Valor Máximo

✓ *En el subcriterio reservas de la biosfera*

Reservas de la biosfera 

Valor de idoneidad	<input type="text" value="0.0"/>
Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="4"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

✓ *En el subcriterio lugares de importancia comunitaria*

Lugares de Importancia Comunitaria

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.2

✓ En el subcriterio zonas de especial protección para las aves

 Zonas de Especial Protección para las Aves

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.1

```

SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_4290
r.mapcalc "R_4291 = if(isnull( R_4290 ), null(), if((R_4290 <= -105.0
|| R_4290 >= -15.0), 0.0, if((R_4290 > -105.0 && R_4290 < -100.0),
eval((R_4290 - -105.0)/(-100.0 --105.0 )), if((R_4290 > -20.0 && R_4290
< -15.0 ), eval((-15.0 - R_4290 )/(-15.0 - -20.0)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=aton output=R_4268
v.to.rast input=R_4268 layer=1 type=point,line,area output=R_4269
use=val value=0 rows=4096
r.grow.distance input=R_4269 distance=R_4270 metric=euclidean
r.mapcalc "R_4271 = R_4270 * 100"
r.mapcalc "R_4272 = if(isnull( R_4271 ), null(), if((R_4271 <= 0.5 ||
R_4271 >= 99999), 0.0, if((R_4271 > 0.5 && R_4271 < 1), eval((R_4271 -
0.5)/(1 -0.5 )), if((R_4271 > 99999 && R_4271 < 99999 ), eval((99999 -
R_4271 )/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=dpp output=R_4285
v.to.rast input=R_4285 layer=1 type=point,line,area output=R_4286
use=val value=0 rows=4096

```

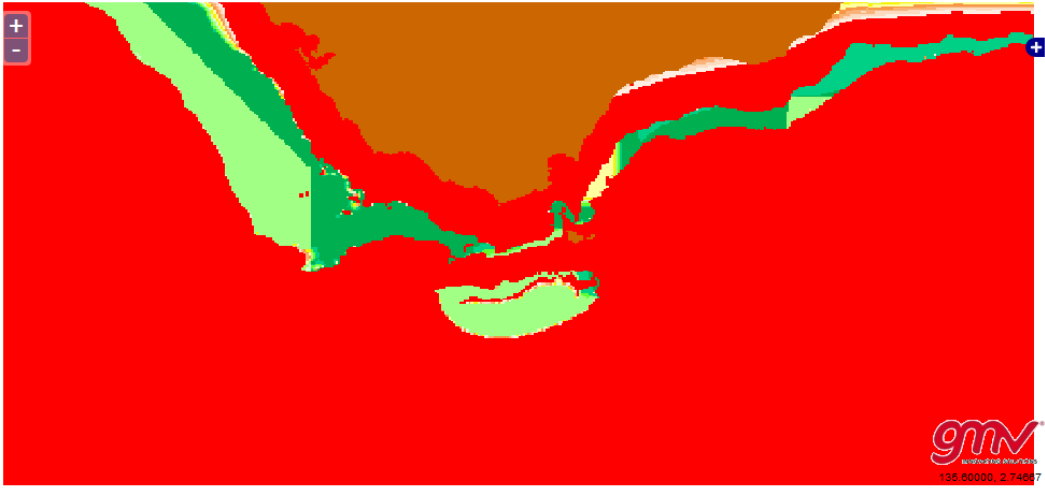


```
r.grow.distance input=R_4286 distance=R_4287 metric=euclidean
r.mapcalc "R_4288 = R_4287 * 100"
r.mapcalc "R_4289 = if(isnull( R_4288 ) , null(), if((R_4288 <= 0 ||
R_4288 >= 99999), 0.0, if((R_4288 > 0 && R_4288 < 1), eval((R_4288 -
0)/(1 -0 )), if((R_4288 > 99999 && R_4288 < 99999 ), eval((99999 -
R_4288 )/(99999 - 99999)), 1.0))))"
r.mapcalc "R_4292 = if(isnull( R_4272 ) || isnull( R_4289 ), null(),
if(R_4272 <= R_4289 , R_4272 , R_4289 ))"
r.mapcalc "R_4293 = if(isnull( R_4292 ) || isnull( R_4291 ), null(),
if(R_4292 <= R_4291 , R_4292 , R_4291 ))"
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\pw.tif" output=R_4266
r.mapcalc "R_4267= if(R_4266 >= 0 && R_4266 < 0.3, eval(R_4266/0.3), 1)"
r.null map=R_4267 null=0
r.mapcalc "R_4294 = if(isnull( R_4267 ) || isnull( R_4291 ), null(),
if(R_4267 <= R_4291 , R_4267 , R_4291 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=lics output=R_4260
v.to.rast input=R_4260 layer=1 type=point,line,area output=R_4261
use=val value=0.0 rows=4096
r.grow.distance input=R_4261 distance=R_4262 metric=euclidean
r.mapcalc "R_4263 = R_4262 * 100"
r.mapcalc "R_4264 = if(isnull( R_4263 ) , null(), if((R_4263 <= 1 ||
R_4263 >= 99999), 0.0, if((R_4263 > 1 && R_4263 < 4), eval((R_4263 -
1)/(4 -1 )), if((R_4263 > 99999 && R_4263 < 99999 ), eval((99999 -
R_4263 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4261,R_4264 output=R_4265
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepas output=R_4273
v.to.rast input=R_4273 layer=1 type=point,line,area output=R_4274
use=val value=0.0 rows=4096
r.grow.distance input=R_4274 distance=R_4275 metric=euclidean
r.mapcalc "R_4276 = R_4275 * 100"
r.mapcalc "R_4277 = if(isnull( R_4276 ) , null(), if((R_4276 <= 1 ||
R_4276 >= 99999), 0.0, if((R_4276 > 1 && R_4276 < 4), eval((R_4276 -
1)/(4 -1 )), if((R_4276 > 99999 && R_4276 < 99999 ), eval((99999 -
R_4276 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4274,R_4277 output=R_4278
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mabs output=R_4279
v.to.rast input=R_4279 layer=1 type=point,line,area output=R_4280
use=val value=0.0 rows=4096
r.grow.distance input=R_4280 distance=R_4281 metric=euclidean
r.mapcalc "R_4282 = R_4281 * 100"
r.mapcalc "R_4283 = if(isnull( R_4282 ) , null(), if((R_4282 <= 1 ||
R_4282 >= 99999), 0.0, if((R_4282 > 1 && R_4282 < 4), eval((R_4282 -
1)/(4 -1 )), if((R_4282 > 99999 && R_4282 < 99999 ), eval((99999 -
R_4282 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4280,R_4283 output=R_4284
r.mapcalc "R_4295 = if(isnull( R_4265 ) || isnull( R_4278 ), null(),
if(R_4265 <= R_4278 , R_4265 , R_4278 ))"
r.mapcalc "R_4296 = if(isnull( R_4295 ) || isnull( R_4278 ), null(),
if(R_4295 <= R_4278 , R_4295 , R_4278 ))"
r.mapcalc "R_4297 = if(isnull( R_4296 ) || isnull( R_4291 ), null(),
if(R_4296 <= R_4291 , R_4296 , R_4291 ))"
```

```

r.mapcalc "R_4298 = if(isnull(R_4293) || isnull(R_4297) ||
isnull(R_4294), null(),if(R_4293 >= R_4297 && R_4297 >= R_4294,
eval(((R_4293*0.2*0.33) + (R_4297*0.5*0.33) + (R_4294*0.3*0.33)) /
(0.2*0.33+0.5*0.33+0.3*0.33)),if(R_4293 >= R_4294 && R_4294 > R_4297,
eval(((R_4293*0.2*0.33) + (R_4294*0.3*0.33) + (R_4297*0.5*0.33)) /
(0.2*0.33+0.3*0.33+0.5*0.33)),if(R_4297 > R_4293 && R_4293 >= R_4294,
eval(((R_4297*0.5*0.33) + (R_4293*0.2*0.33) + (R_4294*0.3*0.33))
/(0.5*0.33+0.2*0.33+0.3*0.33)),if(R_4297 >= R_4294 && R_4294 > R_4293,
eval(((R_4297*0.5*0.33) + (R_4294*0.3*0.33) + (R_4293*0.2*0.33)) /
(0.5*0.33+0.3*0.33+ 0.2*0.33)),if(R_4294 > R_4293 && R_4293 >= R_4297,
eval(((R_4294*0.3*0.33) + (R_4293*0.2*0.33) + (R_4297*0.5*0.33)) /
(0.3*0.33+0.2*0.33+0.5*0.33)),if(R_4294 > R_4297 && R_4297 > R_4293,
eval(((R_4294*0.3*0.33) + (R_4297*0.5*0.33) + (R_4293*0.2*0.33)) /
(0.3*0.33+0.5*0.33+0.2*0.33)), null())))))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_4299 = if(isnull(ccaa),R_4298, -1.0)"
g.region n=36.692981567717 s=35.045032348967 e=-3.786176484375 w=-
7.301801484375 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_4291
output="C:\Users\GSI\OceanLider\Resultados\111\RI_R_4291.tif"
r.out.gdal input=R_4272
output="C:\Users\GSI\OceanLider\Resultados\111\RI_R_4272.tif"
r.out.gdal input=R_4289
output="C:\Users\GSI\OceanLider\Resultados\111\RI_R_4289.tif"
r.out.gdal input=R_4265
output="C:\Users\GSI\OceanLider\Resultados\111\RI_R_4265.tif"
r.out.gdal input=R_4278
output="C:\Users\GSI\OceanLider\Resultados\111\RI_R_4278.tif"
r.out.gdal input=R_4284
output="C:\Users\GSI\OceanLider\Resultados\111\RI_R_4284.tif"
r.out.gdal input=R_4267
output="C:\Users\GSI\OceanLider\Resultados\111\RI_R_4267.tif"
r.out.gdal input=R_4299
output="C:\Users\GSI\OceanLider\Resultados\111\RF_R_4299.tif"

```



6.1. Manual de instalación

En este apartado se explica los pasos a seguir para realizar la instalación de la herramienta DSS. Se detalla la instalación en un ordenador haciendo la función de un servidor local.

Se ha diseñado la herramienta DSS para que pueda ser distribuida, donde la base de datos puede estar albergada en un servidor y la aplicación en otro diferente. La instalación que se describe a continuación estará integrada en un único servidor.

Se va a realizar la instalación sobre un ordenador con Sistema Operativo **Windows 7**. Los programas que hacen falta para su instalación:

- NetBeans 7.0/7.1
- GRASS 6.4
- PostgreSQL 9.1 + PostGIS 2.0

La instalación consta de varias partes, en primer lugar se va a realizar la instalación de NetBeans que es un entorno de desarrollo que provee un servidor local integrado llamado Glassfish para poder arrancar la herramienta web. En NetBeans importaremos los proyectos que la herramienta DSS precisa para funcionar. Luego, se instalará GRASS que es el SIG donde se ejecutarán las operaciones de los scripts operacionales. Por último concluimos la instalación del sistema gestor de base de datos donde estarán alojados los datos, que es PostgreSQL como sistema gestor de base de datos relacional y su complemento PostGIS para datos geoespaciales.

6.1.1. Instalación de NetBeans (Glassfish)

Realizamos la instalación de NetBeans 7.1 con el ejecutable dispuesto para Windows siguiendo los pasos que nos ofrece la guía. Si se prefiere, se puede hacer la descarga de la última actualización de NetBeans en la página <http://netbeans.org>, pero hay que asegurar que el paquete de instalación contiene el servidor GlassFish.

6.1.1.1. Importación del código fuente y configuración de parámetros en código fuente

Una vez instalado NetBeans, abrimos el entorno de desarrollo e importamos los proyectos involucrados en la herramienta. Los proyectos son:

- DAO
- Entities
- Motor
- ORPAWeb

Se deben cambiar los parámetros que vinculan a la base de datos donde están las capas informativas para que trabaje de forma local, estas variables están bajo la clase **bd.DatosBBDD** en el paquete **Entities**:

```
public static String hostBBDDOceanLider="localhost";  
public static String passBBDDOceanLider="postgres";  
public static String userBBDDOceanLider="postgres";
```

6.1.2. Instalación de GRASS

Instalamos GRASS 6.4 con el ejecutable dispuesto para Windows siguiendo los pasos que nos ofrece la guía. La última versión estable es la versión 6.4.2, si se prefiere se puede descargar la última versión en la página <http://grass.osgeo.org/download/software/ms-windows/>

6.1.2.1. Creación de MAPSET y LOCATION

GRASS cuenta con una base de datos interna que se debe crear para establecer el entorno de trabajo. DATABASE es el directorio principal que contiene la estructura en directorios de trabajo. De este directorio colgarán varios LOCATION, cada uno de ellos es un directorio que corresponde a un proyecto de trabajo.

Cada LOCATION utiliza un único sistema de referencia que se debe especificar para trabajar en él. Para especificar el sistema de referencia de una localización se puede utilizar el código EPSG de la zona.

Dentro de cada localización, se ubican varios MAPSET correspondientes a los directorios de cada usuario donde se almacenan los mapas que han creado y que solamente dichos usuarios podrán modificar.

De este modo, tendremos que crear el espacio LOCATION y MAPSET que utilizará la herramienta DSS donde se lanzarán las operaciones SIG de forma automática.

Para no tener problemas con el directorio base de la base de datos interna de GRASS (DATABASE), modificamos el nombre del directorio accediendo a la carpeta *Documentos* de

nuestro usuario que es donde se habrá instalado por defecto. Modificamos el nombre de la DATABASE sin espacios “GISDataBase”.

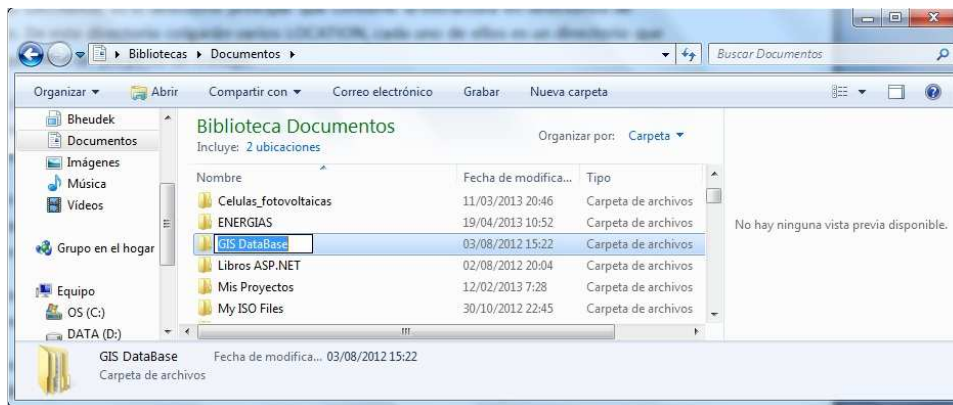


Figura 162: Modificación del nombre del directorio DATABASE de GRASS

Abrimos GRASS 6.4.2 TlTk GUI, nos aparece la siguiente ventana, donde nos indica el directorio base para la base interna de GRASS (nuestro DATABASE).



Figura 163: Ventana principal de GRASS TlTk GUI

Para crear una localización nueva (LOCATION), en concreto la de la herramienta DSS, presionamos sobre el botón de definir una localización con código EPSG. Los datos que indicamos para crear la nueva localización son:

- Nombre de la nueva localización: **WGS84**
- Número de código EPSG de la proyección: **4326**

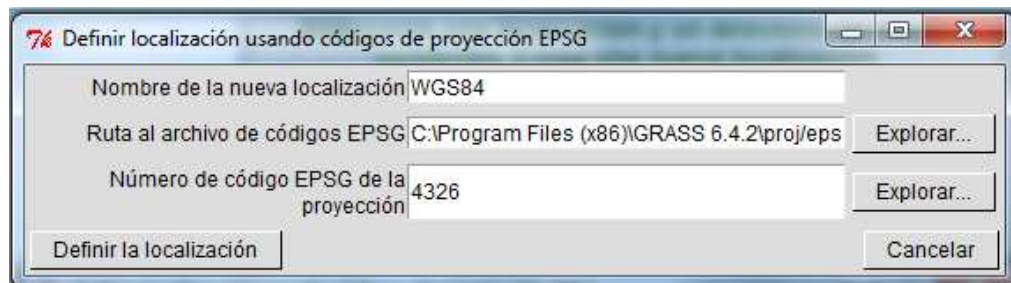


Figura 164: Ventana de creación de una localización de GRASS con código de proyección EPSG

Una vez introducidos los datos, presionamos sobre el botón de “*Definir la localización*”. Creamos la localización, creamos el MAPSET de la herramienta DSS dentro de la misma. Teniendo seleccionada la localización que hemos creado, introducimos el nombre del nuevo MAPSET sobre el cuadro de texto que indica “*Crear nuevo directorio de mapas en la localización seleccionada*” con el nombre “DSS”, como se indica en la siguiente figura.

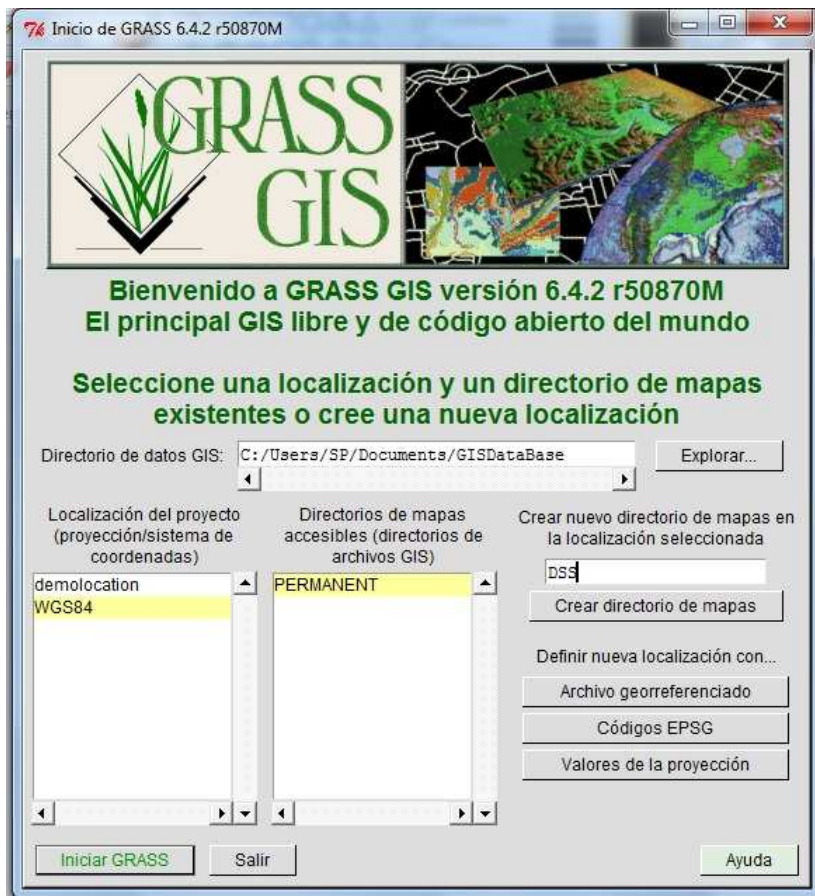


Figura 165: Ventana de GRASS TclTk GUI para la creación de un MAPSET

6.1.2.2. Configuración del archivo .grassrc6

Para poder ejecutar de forma automática operaciones de GRASS sobre el sistema es necesario configurar el archivo “.grassrc6_wgs84”. Este fichero contiene la información del directorio de la base de datos interna de GRASS (DATABASE), la localización (LOCATION), directorio de mapas (MAPSET) y modo (GRASS_GUI), donde se ejecutarán las operaciones.

Actualmente el contenido de este fichero es:

```
GISDBASE: C:\Users\SP\Documents\GISDataBase
LOCATION_NAME: WGS84
MAPSET: DSS
GRASS_GUI: text
```

Es necesario configurar estos parámetros que utilizará la herramienta DSS. Tenemos que cambiar el directorio de la variable GISDBASE al que se ha configurado en el paso anterior como DATABASE.

6.1.2.3. Configuración de variables de entorno de Windows

Es necesaria la configuración de variables de entorno de Windows para lanzar la ejecución de operaciones fuera del entorno de GRASS. Para configurar variables de entorno de Windows nos vamos a “Panel de Control->Sistema y Seguridad->Sistema->Cambiar configuración->Pestaña de opciones avanzadas->Variables de entorno”. Nos aparece la siguiente ventana.

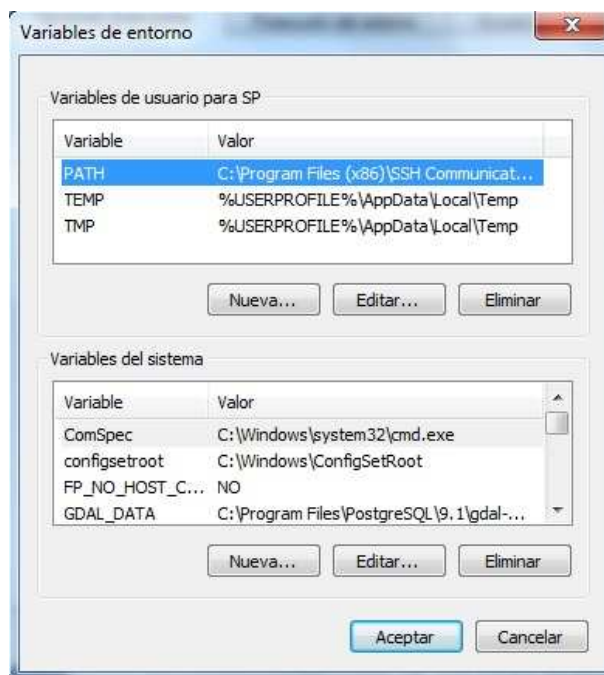


Figura 166: Ventana de configuración de variables de entorno de Windows

Creamos las siguientes variables nuevas **del sistema**. GISBASE es el directorio base donde se ha instalado GRASS y GISRC es la ruta al fichero de configuración de .grassrc6. Las rutas van acorde con nuestro nombre de usuario de Windows. En caso que se haya instalado GRASS bajo otro directorio hay que indicar la ruta donde se haya realizado la instalación.

```
GISBASE := C:\Program Files\GRASS 6.4.2
GISRC := C:\Users\SP\OceanLider\.grassrc6_wgs84
```

En la variable Path (no PATH) de variable del sistema añadimos las siguientes direcciones:

```
Path := C:\Program Files\GRASS 6.4.2\lib; C:\Program Files\GRASS  
6.4.2\extralib; C:\Program Files\GRASS 6.4.2\bin; C:\Program  
Files\GRASS 6.4.2\scripts
```

Una vez configuramos las variables de entorno, **reiniciamos el ordenador**.

6.1.2.4. Posibles conflictos en el lanzamiento de operaciones fuera del entorno de GRASS

Es posible que aparezcan algunos conflictos cuando ejecutemos operaciones de GRASS sobre la consola del sistema de Windows (CMD). Estos conflictos tienen que ver con que en diferentes versiones de Windows necesita de bibliotecas que no se han instalado en GRASS.

Para comprobar que funciona correctamente abrimos una consola de símbolos del sistema de Windows (CMD) y ejecutamos una sentencia de comandos de GRASS, por ejemplo:

```
g.list type=rast
```

Si al introducir una sentencia de comandos de GRASS nos aparece alguna alerta de error, es posible que nos falte alguna biblioteca instalada. Los posibles conflictos que se han encontrado son con las bibliotecas **libiconv2** y **libintl3**. Para solucionarlo se deben colocar los archivos *libiconv2.dll* y *libintl3.dll* en la carpeta *extralib* del directorio principal de GRASS. Estos archivos se proporcionan en la documentación del proyecto.

6.1.3. Instalar y repoblar BBDD sobre PostgreSQL

La herramienta DSS maneja dos bases de datos según la naturaleza de los datos:

- **DSS_nueva**: es una base de datos relacional que se instala sobre PostgreSQL 9.1. Contiene las entidades y relaciones de la herramienta DSS.
- **BBDDOceanLider**: es una base de datos espacial que se instala sobre PostgreSQL 9.1+PostGIS 2.0. Contiene las capas vectoriales de entrada que precisa la herramienta DSS.

6.1.3.1. Instalar PostgreSQL 9.1 + PostGIS 2.0

Realizamos la instalación de PostgreSQL 9.1 con el ejecutable dispuesto para Windows siguiendo los pasos que nos ofrece la guía. Según los datos que utiliza la herramienta DSS, se ha configurado el nombre y contraseña de Postgres:

- Nombre: postgres
- Contraseña: postgres

Se puede configurar como se desee. Si se configura con otro nombre o contraseña hay que tener en cuenta la modificación de los parámetros en el código fuente sobre el acceso a la base de datos.

Una vez se instale PostgreSQL, el ejecutable ofrece la posibilidad de completar la instalación con otros complementos. Se puede instalar PostGIS 2.0 a través del mismo ejecutable de PostgreSQL 9.1 o a través del ejecutable para Windows que en este último caso se tendrá que ejecutar después de haber instalado PostgreSQL sin complementos.

6.1.3.2. Crear y repoblar la base de datos DSS_nueva

Para crear la base de datos **DSS_nueva**, lo vamos a realizar sobre pgAdmin III que es una aplicación gráfica que dispone PostgreSQL para gestionar bases de datos.

Presionamos el botón derecho del ratón sobre “Databases” para crear una nueva base de datos.

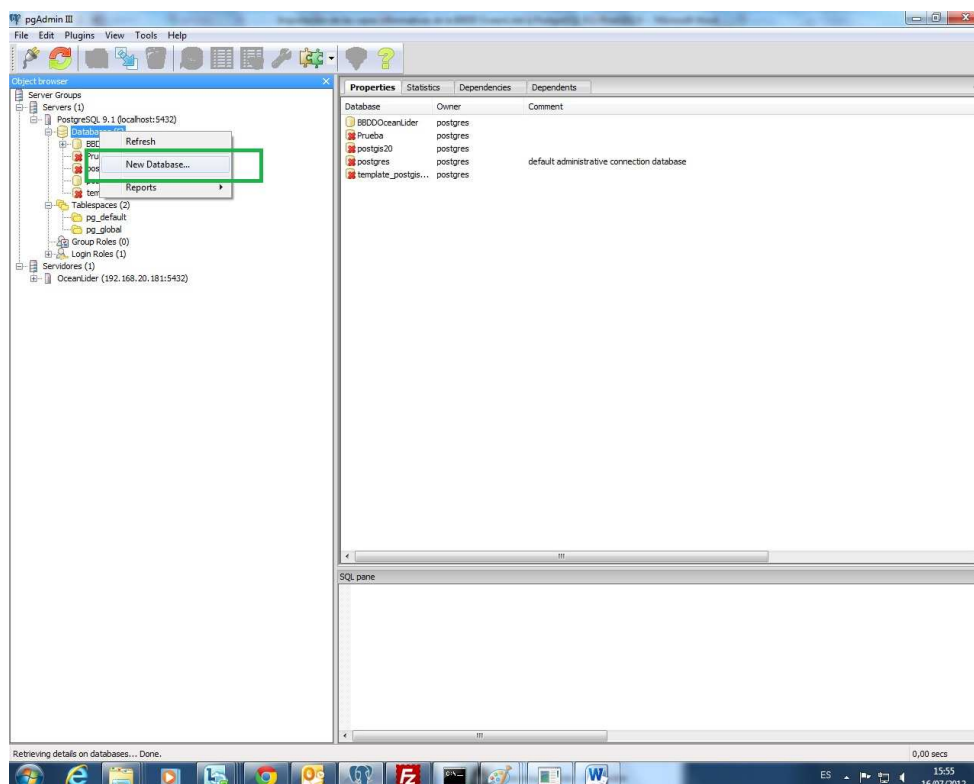


Figura 167: Creación de una nueva base de datos utilizando la interfaz gráfica de pgAdminIII

Nos mostrará un cuadro de dialogo en el que tendremos que indicar el nombre de la nueva base de datos, el propietario, la codificación y la plantilla a utilizar. La base de datos **DSS_nueva** tiene asociada la siguiente información:

- **Nombre:** DSS_nueva
- **Propietario:** postgres
- **Codificación:** UTF-8
- **Plantilla:** postgis20

La plantilla *postgis20* se crea en la instalación de PostGIS 2.0 y contiene todas las funciones geoespaciales de formato raster y vectorial. Vamos a utilizar esta plantilla para albergar información espacial en algunos campos, como es el campo de la región geográfica para un escenario.

Para repoblar la base de datos, utilizamos la consola de sentencias SQL de pgAdmin III. Importamos el fichero SQL de creación de tablas y relaciones de la base de datos *DSS_nueva* y ejecutamos. Posteriormente abrimos el de inserción y de nuevo damos a ejecutar. Ambos ficheros SQL están disponibles en la documentación del proyecto.

6.1.3.3. Crear y repoblar la base de datos BBDDOceanLider

Para crear y repoblar la base de datos *BBDDOceanLider* que contiene las capas informativas se puede seguir los pasos descritos en el apartado 4.6.3. *Importación de las capas informativas de la BBDDOceanLider a PostgreSQL 9.1 + PostGIS 2.0.*

Para que no sea una tarea tan tediosa, se ha previsto en la documentación del proyecto un backup de la base de datos *BBDDOceanLider* llamado "*BBDDOceanLider.backup*". Para realizar la restauración, en primer lugar debemos crear la base de datos según se describe a través de la interfaz pgAdminIII en el apartado anterior con los datos:

- **Nombre:** BBDDOceanLider
- **Propietario:** postgres
- **Codificación:** UTF-8
- **Plantilla:** postgis20

Posteriormente escribimos en la consola de símbolos del sistema de Windows (CMD):

```
set PATH=%PATH%;C:\Program Files\PostgreSQL\9.1\bin
psql -U postgres -d BBDDOceanLider -h localhost -p 5432 -f
ruta\BBDDOceanLider.backup
```

Donde *ruta* será la ruta donde se alberga la copia de seguridad de la base de datos *BBDDOceanLider*.

La base de datos *BBDDOceanLider* contiene cuatro esquemas según la procedencia de los datos:

- **Esquema 'península':** Contiene las capas vectoriales informativas de los datos asociados al proyecto que están ubicados en la Península y zona marítima Peninsular. Las capas que están bajo este esquema tienen el sistema de referencia EPSG 23030.

- **Esquema ‘balears’**: Contiene las capas vectoriales informativas de los datos asociados al proyecto que están ubicados en las Islas Baleares y zona marítima de Baleares. Las capas que están bajo este esquema tienen el sistema de referencia EPSG 23031.
- **Esquema ‘canarias’**: Contiene las capas vectoriales informativas de los datos asociados al proyecto que están ubicados en las Islas Canarias y zona marítima de Canarias. Las capas que están bajo este esquema tienen el sistema de referencia EPSG 32628.
- **Esquema ‘public’**: Contiene las capas vectoriales informativas de los datos asociados al proyecto que son unión de las tres entidades y reproyectadas al sistema de referencia con código EPSG 4326. Son las capas de entrada que utiliza la herramienta DSS para operar.

También se ha previsto en la documentación del proyecto un backup de la base de datos *BBDDOceanLider_restricciones* en caso que se requiera llamado “*BBDDOceanLider_restricciones.backup*”.

6.1.4. Instalar directorios de la herramienta OceanLider

Finalmente, tenemos que instalar el **sistema de ficheros de la herramienta DSS** en el directorio principal del usuario que estemos utilizando en Windows. La carpeta que alberga el sistema de ficheros que utiliza la herramienta DSS se llama *OceanLider* y de ella cuelgan una serie de directorios y ficheros:

- **OceanLider**: carpeta raíz del sistema de ficheros de la herramienta DSS
 - **Resultados**: directorio que almacena los resultados de los escenarios que se hayan ejecutado. En él se crearán carpetas correspondientes a cada escenario ejecutado que contendrán el script operacional, las capas raster generadas del resultado final y resultados intermedios, ficheros de seguimiento de errores, y los ficheros .MAP para la visualización de las capas raster.
 - **Union**: directorio que almacena las capas raster perteneciente a capas de entrada en la herramienta DSS. También contiene los ficheros .MAP de algunos mapas de visualización en la herramienta DSS.
 - **.grassrc6_wgs84**: fichero de configuración del entorno de trabajo de GRASS para establecer la localización (LOCATION) y directorio de mapas (MAPSET).

6.2. Manual de usuario



En este apartado se van a describir los pasos para ejecutar las acciones básicas en la herramienta DSS. Las acciones son:

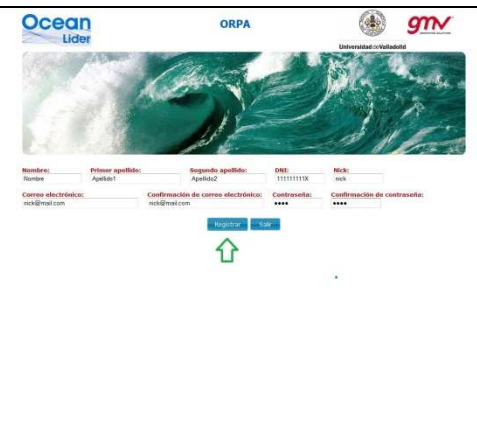
- Registrarse
- Entrar en el sistema
- Crear un nuevo escenario
- Crear un nuevo dispositivo

- Visualizar las características de un dispositivo
- Seleccionar el área de estudio de un escenario
- Cargar un escenario creado
- Cambiar las características de un escenario
- Configuración de criterios
- Configuración de la toma de decisión final
- Ejecutar un escenario
- Visualización de resultados
- Salir del sistema

6.2.1. Registrarse

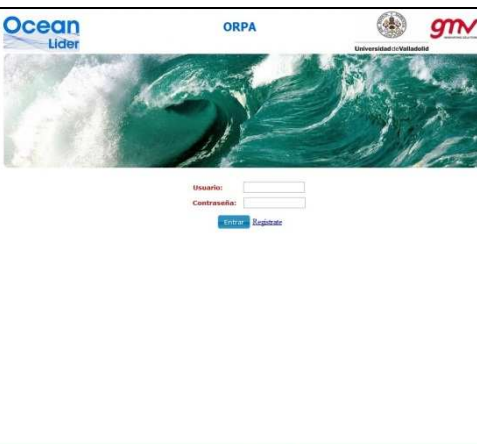
Es necesario registrarse para entrar en el sistema.


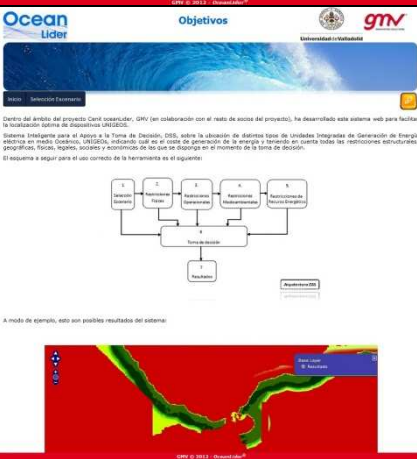
<p>Paso 1: Página de inicio</p> <p>Nos situamos en la página principal</p>	
<p>Paso 2: Entrar en página de registro</p> <p>Presionamos el link de 'Registrarse' de la página principal y nos lleva a la página de registro de usuarios</p>	

<p>Paso 3: Registrarse</p> <p>Rellenamos los campos para el registro y clicamos en el botón de Registrar. El sistema dejará registrarse si los campos son completados, la repetición de campos es correcta para el correo electrónico y contraseña, y la dirección de correo no esté registrada con anterioridad en el sistema.</p>	
--	--

6.2.2. Entrar en el sistema

Se explican los pasos para entrar en el sistema una vez el usuario esté registrado.

<p>Paso 1: Página de inicio</p> <p>Nos situamos en la página principal</p>	
---	---

<p>Paso 2: Entrar en el sistema</p> <p>Rellenamos los campos de 'Usuario' y 'Contraseña'. En el campo 'Usuario' introducimos nuestro correo electrónico (el que dimos de alta en el registro de usuario) y la contraseña (la que dimos de alta en el registro de usuario)</p>	
<p>Paso 3: Página objetivos del sistema</p> <p>Una vez el sistema nos identifique, accedemos al sistema en la página de objetivos donde se nos explica brevemente la herramienta</p>	

6.2.3. Crear un nuevo escenario

Una vez hayamos accedido al sistema, podemos crear un nuevo escenario.

<p>Paso 1: Página de objetivos</p> <p>Una vez hayamos entrado en el sistema, nos ubicamos en la página de objetivos</p>	
<p>Paso 2: Selección escenario</p> <p>Presionamos en el botón de 'Selección Escenario' y nos dirige a la página de cargar/crear escenarios</p>	
<p>Paso 3: Página selección escenario</p> <p>La página de selección de escenarios nos muestra dos pestañas según la acción que deseamos realizar: cargar un escenario creado o crear un nuevo escenario</p>	

Paso 4: Selección pestaña de crear nuevo escenario

Seleccionamos la pestaña de crear un nuevo escenario






Paso 5: Pestaña de crear un nuevo escenario

Nos muestra el contenido de la pestaña de crear un nuevo escenario. Debemos rellenar los campos: nombre, dispositivo y selección de área. En la elección de un dispositivo, podemos elegir un dispositivo del desplegable o crear un nuevo dispositivo para posterior seleccionarle. [Ver acción de crear un nuevo dispositivo]

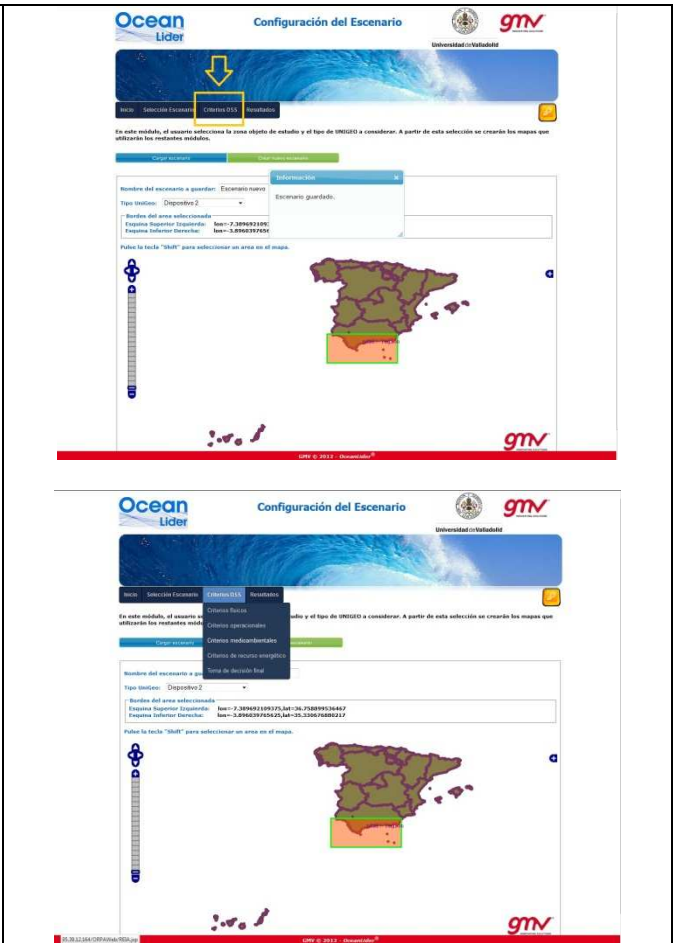
En la elección del área de estudio tenemos que seleccionar una región del mapa mostrado. [Ver acción de seleccionar área de estudio de un escenario]



<p>Paso 6: Guardar escenario</p> <p>Rellenamos los campos de nombre, seleccionamos un dispositivo y seleccionamos un área del mapa mostrado. Una vez configuremos el escenario nuevo, clicamos en el botón de ‘<i>Guardar Escenario</i>’ para crear el nuevo escenario.</p>	
<p>Paso 7: Mensaje confirmación</p> <p>El sistema nos muestra un mensaje de confirmación para crear el escenario</p>	
<p>Paso 8: Aceptar mensaje escenario almacenado</p> <p>Una vez aceptemos el mensaje de confirmación de crear un escenario nuevo, el sistema nos avisará con un mensaje indicando que el escenario ha sido guardado correctamente</p>	

Paso 9: Configurar criterios del escenario almacenado

A partir de este momento nos aparecerán las pestañas para configurar los criterios del escenario en la barra superior [Ver acción de configuración de criterios]



6.2.4. Crear un nuevo dispositivo

Podemos crear un nuevo dispositivo cuando estemos configurando un nuevo escenario o cuando queramos cambiar las características de un escenario ya creado aún si procesar.

Paso 1.a: Pestaña de crear un nuevo escenario

Una vez ubicados en la pestaña de crear un nuevo escenario, presionamos el botón de ‘*Crear nuevo dispositivo*’



Paso 1.b: Pestaña cargar escenario con escenario cargado

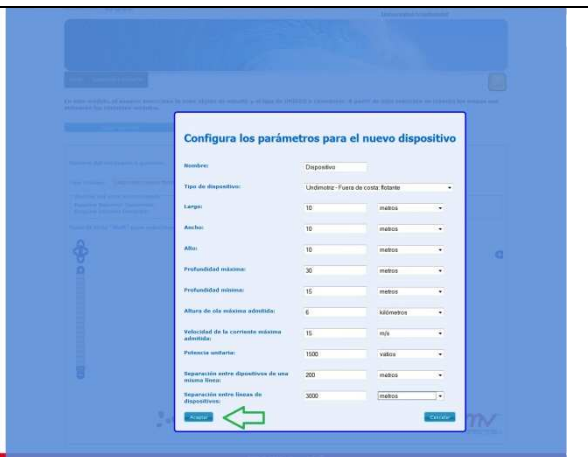
Una vez ubicados en la carga de un escenario con un escenario cargado, presionamos el botón de ‘*Crear nuevo dispositivo*’



	
<p>Paso 2: Ventana de configurar un nuevo dispositivo</p> <p>Nos aparece en la interfaz la ventana de configuración de un nuevo dispositivo. Introducimos los campos necesarios para su creación.</p>	

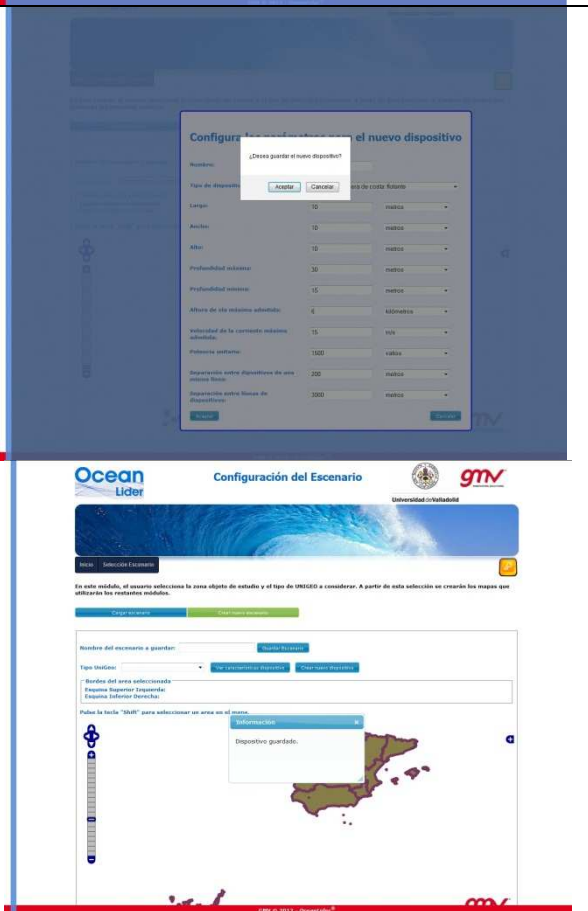
Paso 3: Guardar nuevo dispositivo

Una vez rellenados los parámetros del dispositivo, presionamos en el botón de *Aceptar* para almacenar el nuevo dispositivo.



Paso 4: Mensaje confirmación guardar dispositivo

El sistema muestra un mensaje de confirmación para almacenar el nuevo dispositivo. Una vez aceptado, el sistema almacenará el dispositivo y si no ha habido ningún error en la base de datos, nos responderá con un mensaje de que el dispositivo ha sido almacenado



Paso 5: Disposición del dispositivo en el desplegable

Una vez el escenario ha sido almacenado correctamente estará disponible en el desplegable de selección de dispositivo



6.2.5. Visualizar características dispositivo

Podemos visualizar las características de un dispositivo cuando estemos creando un nuevo escenario o cuando cargamos un escenario ya creado.

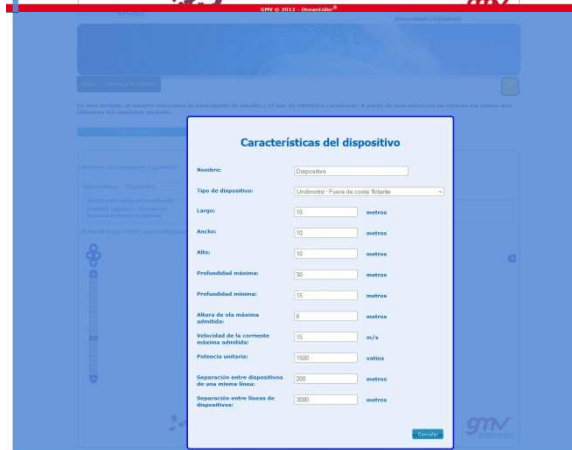
Paso 1: Seleccionar dispositivo en desplegable

Ubicados en la pestaña de crear un nuevo escenario o en la pestaña de cargar un escenario con un escenario cargado, aparece un desplegable con los dispositivos disponibles.



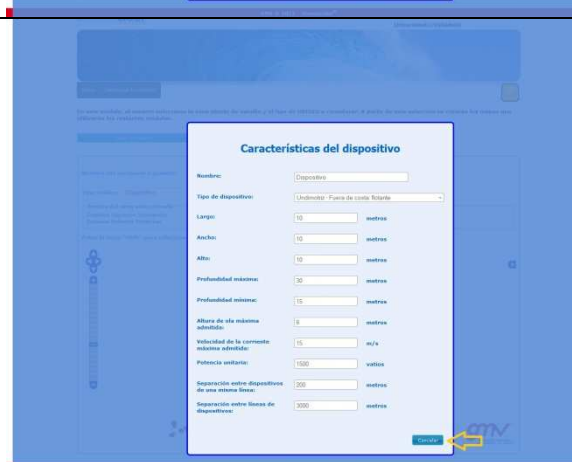
Paso 2: Selección de un dispositivo

Una vez seleccionado el dispositivo, presionamos el botón de ‘Ver características del dispositivo’ y nos aparecerá en la interfaz una ventana con los parámetros del dispositivo seleccionado





Paso 3: Salir de la ventana de características del dispositivo

Para salir de la ventana de las características del dispositivo, presionamos en el botón de *Cancelar*



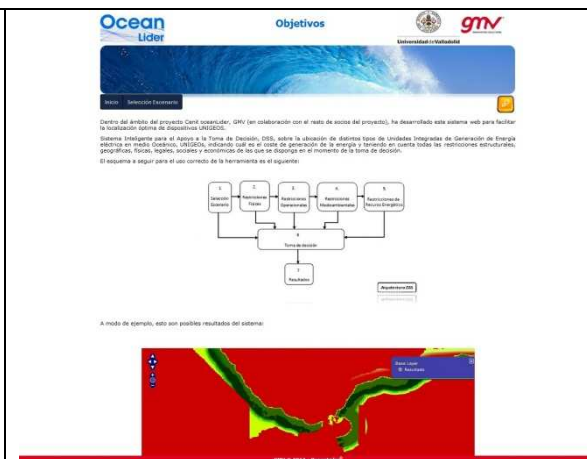
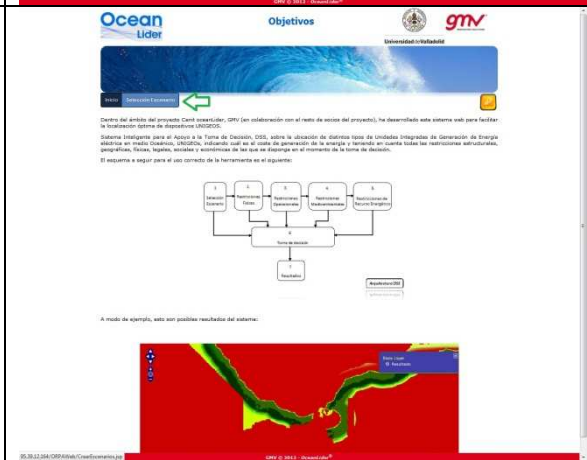

6.2.6. Seleccionar área de estudio de un escenario




Debemos seleccionar el área de estudio de un escenario cuando estemos creando un nuevo escenario o cuando queramos cambiar las características de un escenario ya creado pero no procesado.

<p>Paso 1: Pestaña de crear nuevo escenario o cargar un escenario con un escenario cargado</p> <p>Sobre el mapa que se muestra presionamos la tecla <i>Shift</i> de nuestro teclado y con el ratón presionamos con el botón izquierdo arrastrando, obteniendo un área dibujada sobre el mapa.</p>	
<p>Paso 2: Obtención del área de estudio del escenario</p>	

6.2.7. Cargar un escenario creado

Una vez hayamos accedido al sistema, podemos cargar un escenario creado.

<p>Paso 1: Página de objetivos</p> <p>Una vez hayamos entrado en el sistema, nos ubicamos en la página de objetivos</p>	
<p>Paso 2: Selección escenario</p> <p>Clicamos en el botón de ‘<i>Selección Escenario</i>’ y nos dirige a la página de cargar/crear escenarios</p>	
<p>Paso 3: Página selección escenario</p> <p>La página de selección de escenarios nos muestra dos pestañas según la acción que deseamos realizar: cargar un escenario creado o crear un nuevo escenario. Por defecto el sistema muestra la pestaña de cargar escenarios. En el contenido de la pestaña de cargar escenarios aparece un desplegable con todos los escenarios creados hasta el momento.</p>	

<p>Paso 4: Selección de escenario creado en el desplegable</p> <p>Seleccionamos un escenario disponible en el desplegable.</p>	
<p>Paso 5: Botón cargar escenario</p> <p>Presionamos en el botón de 'Cargar escenario'.</p> <p>Tiene que haber un escenario seleccionado en el desplegable para poder ejecutar este paso.</p>	
<p>Paso 6: Visualización de las características del escenario seleccionado</p> <p>El sistema muestra las características del escenario que hemos seleccionado.</p>	

Paso 7: Configurar criterios del escenario almacenado

A partir de este momento nos aparecerán las pestañas para configurar los criterios del escenario en la barra superior [Ver acción de configuración de criterios]



6.2.8. Cambiar características de un escenario

Se puede cambiar las características de un escenario si todavía no ha sido procesado.

Paso 1: Cargamos un escenario creado
 [Ver acción de cargar un escenario creado]



Paso 2: Cambiamos características de un escenario sin procesar
 Podemos cambiar el nombre, dispositivo o zona de estudio de un escenario

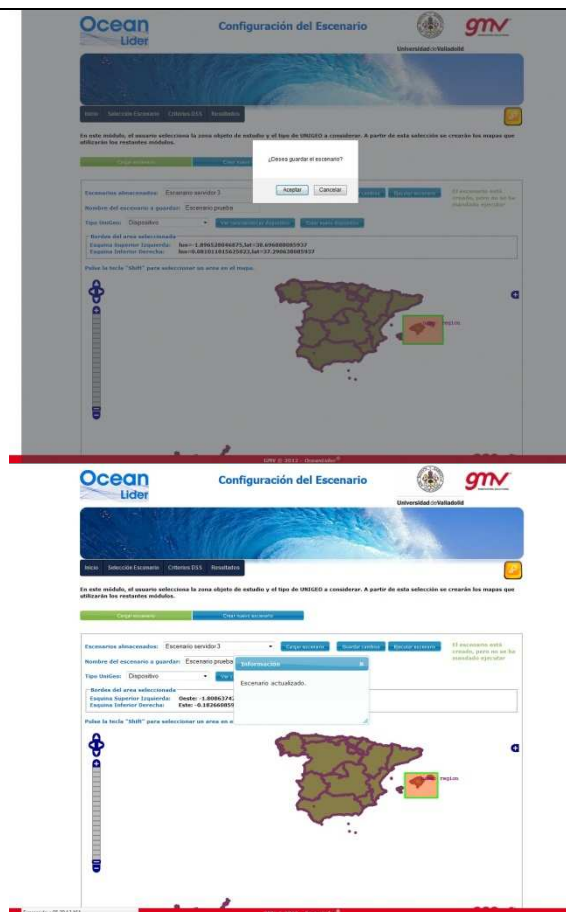


Paso 3: Guardar cambios
 Presionamos sobre el botón de 'Guardar Cambios'



Paso 4: Mensaje confirmación guardar cambios de un escenario

El sistema muestra un mensaje de confirmación para guardar los cambios introducidos en el escenario seleccionado. Aceptamos confirmación y si se ha tramitado correctamente nos aparece un mensaje de aviso de que los cambios han sido guardados.

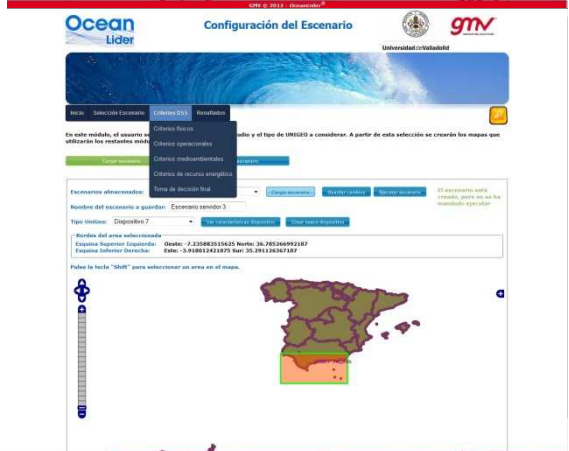


6.2.9. Configuración de criterios

Se puede configurar los criterios de un escenario creado o de un escenario cargado aún sin ejecutar.

Paso 1.a: Cargamos un escenario sin ejecutar

[Ver acción cargar un escenario creado]



Paso 1.b: Creamos un nuevo escenario

[Ver acción de crear un nuevo escenario]



<p>Paso 2: Accedemos al apartado de criterios y seleccionamos alguna de las opciones disponibles (criterios físicos, criterios operacionales, criterios medio ambientales o criterios de recurso energético).</p>	
<p>Paso 3: Seleccionamos/Deseleccionamos criterios</p> <p>Seleccionamos o deseccionamos los criterios que están disponibles para ese escenario e introducimos los parámetros necesarios que necesite para la configuración del criterio</p>	

Paso 4: Presionamos botón de ‘Guardar Parámetros’

Para almacenar los criterios seleccionados o actualizar los criterios que deseccionemos, presionamos sobre el botón de ‘Guardar Parámetros’



6.2.10. Configuración de la toma de decisión final

Esta acción se tiene que realizar si se ha configurado más de dos ramas: rama de criterios operacionales, rama de criterios medioambientales y rama de criterios de recurso energético. La interfaz no dejará configurar estos parámetros si no se ha elegido más de dos ramas.

Paso 1.a: Cargamos un escenario sin ejecutar

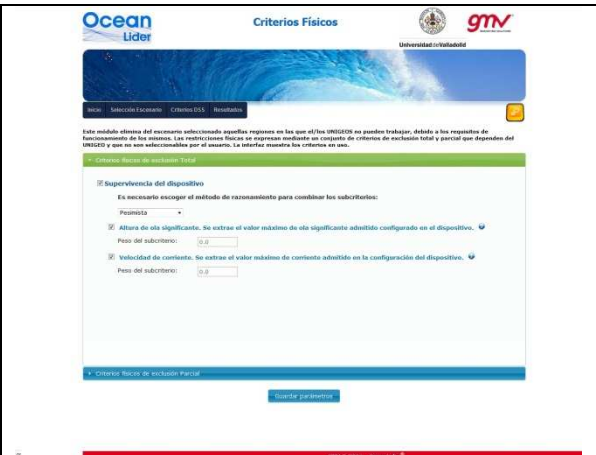
[Ver acción cargar un escenario creado]



<p>Paso 1.b: Creamos un nuevo escenario</p> <p>[Ver acción de crear un nuevo escenario]</p>	

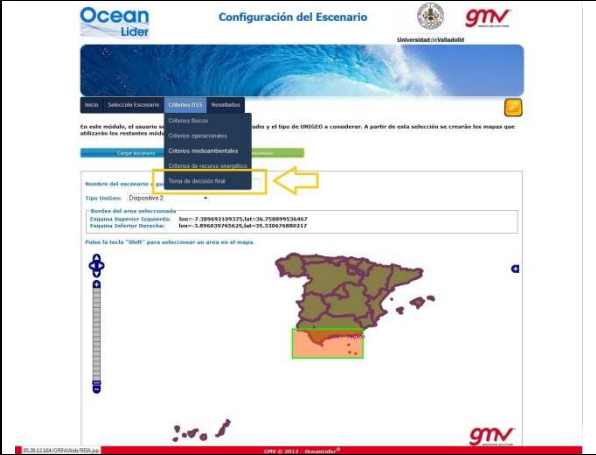
Paso 2: Configuramos al menos dos ramas de criterios operacionales, criterios medio ambientales o criterios de recurso energético.

[Ver acción de configuración de criterios]

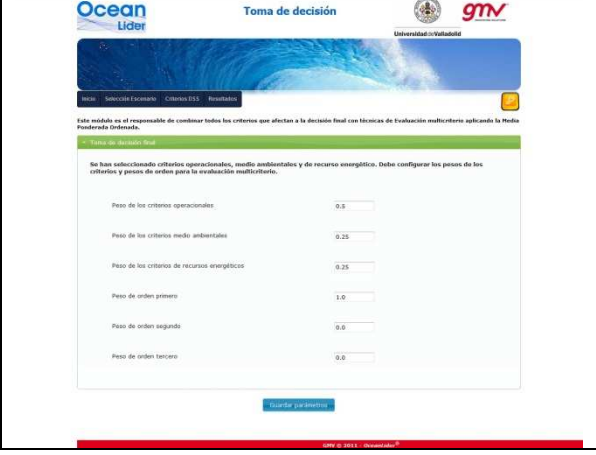


Paso 3: Entramos en el apartado de Toma de decisión final

Una vez configurados más de dos ramas de criterios, se puede configurar la toma de decisión final.

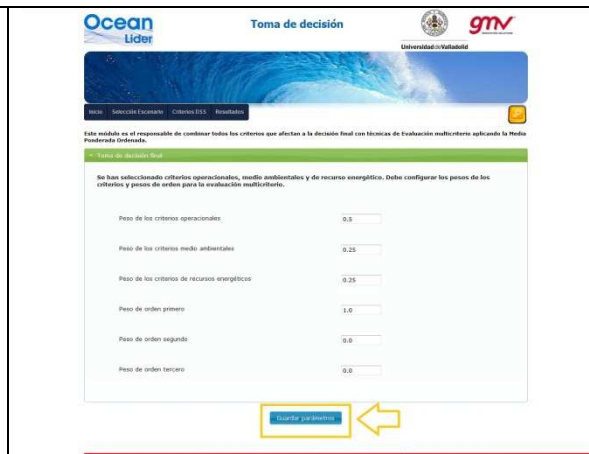


Paso 4: Introducimos los parámetros que se nos pide: los pesos de las ramas de los criterios y pesos de orden para la toma de decisión final



Paso 5: Presionamos el botón de ‘Guardar Parámetros’

Para almacenar los pesos y pesos de orden de la toma de decisión final, presionamos sobre el botón de ‘*Guardar Parámetros*’



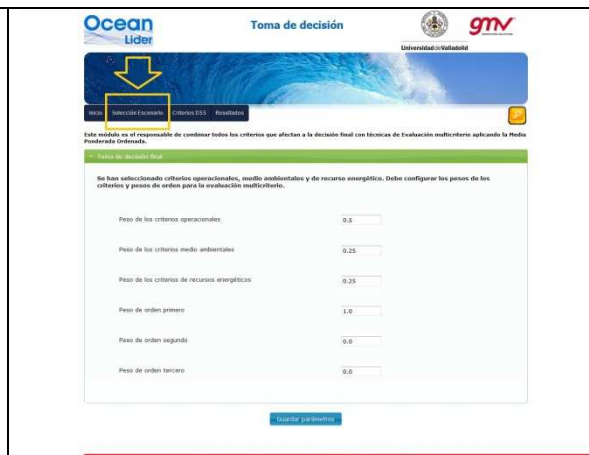
6.2.11. Ejecutar escenario


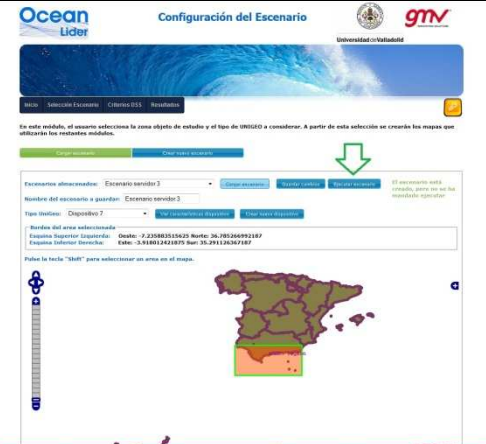
Una vez se hayan configurado los criterios del escenario y la toma de decisión en caso que se precise, el usuario puede mandar ejecutar el escenario.

Paso 1: Cargar el escenario

Una vez configurados los criterios del escenario y la toma de decisión final (en caso de precisarlo), el usuario tiene que cargar el escenario para mandar la acción de ejecución.

[Ver acción de cargar un escenario creado]



	
<p>Paso 2: Presionamos el botón de ‘Ejecutar escenario’</p> <p>Una vez cargado el escenario, el usuario puede mandar a ejecutar el escenario. El sistema le pondrá en cola de ejecución en espera de ser ejecutado.</p>	

6.2.12. Visualización de resultados

Una vez se haya ejecutado el escenario, se puede proceder a visualizar los resultados obtenidos por la herramienta.

Paso 1: Cargar el escenario

Una vez ejecutado el escenario, estará disponible la visualización de los resultados de dicho escenario



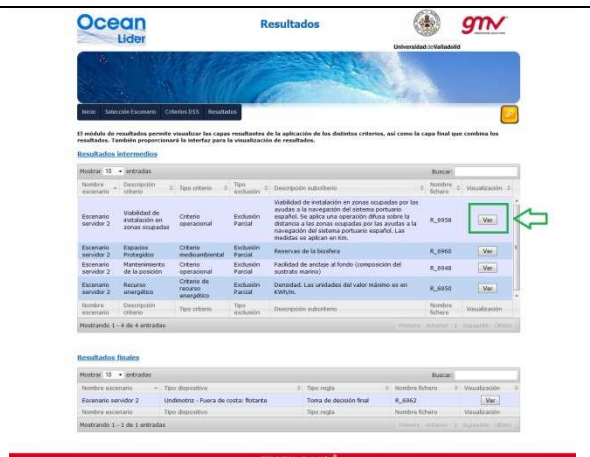
Paso 2: Ir al apartado de 'Resultados'

El sistema mostrará una lista con la información ligada a los resultados intermedios y finales que se han ejecutado en el escenario



Paso 3: Seleccionar resultado

Para visualizar el mapa resultado, se debe presionar el botón 'Ver' del resultado.



Paso 4: Visor de resultados en vista global

El sistema muestra el resultado seleccionado en un mapa desde una vista global junto con la línea de costa de Península, Baleares y Canarias



Paso 5: Visor de resultados en vista concreta

Al presionar la lupa de ampliación, el sistema muestra el resultado seleccionado en detalle en la región especificada en el escenario



Paso 6: Volver a lista de resultados

Si presionamos el botón de volver que se representa con una flecha volteada, volveremos a la página de resultados

Visualización del resultado

Ocean Lider Universidad de Valladolid gmv

Inicio Selección Escenarios Criterios EDS Resultados

Leyenda de colores

0.0 1.0

Paso 6: Volver a lista de resultados

Si presionamos el botón de volver que se representa con una flecha volteada, volveremos a la página de resultados

Visualización del resultado

Ocean Lider Universidad de Valladolid gmv

Inicio Selección Escenarios Criterios EDS Resultados

Leyenda de colores

0.0 1.0

Paso 6: Volver a lista de resultados

Si presionamos el botón de volver que se representa con una flecha volteada, volveremos a la página de resultados

Resultados

Ocean Lider Universidad de Valladolid gmv

Inicio Selección Escenarios Criterios EDS Resultados

El módulo de resultados permite visualizar las capas resultantes de la aplicación de los distintos criterios, así como la capa final que combina los resultados. También proporcionará la interfaz para la visualización de resultados.

Resultados Intermedios

Nombre escenario	Descripción criterio	Tipo escenario	Tipo evaluación	Descripción subcriterio	Nombre criterio	Visualización
Escenario servidor 2	Viabilidad de establos en zonas ocupadas	Criterio operacional	Exclusión Parcial	Viabilidad de evaluación en zonas ocupadas por los establos a la navegación del sistema portuario español. Se define una operación eficaz sobre la distancia a las zonas ocupadas por las ayudas a la navegación del sistema portuario español. Las medidas se aplican en km.	R_8958	[Ver]
Escenario servidor 3	España Postgigas	Criterio medioambiental	Exclusión Parcial	Reserven de la biología	R_8960	[Ver]
Escenario servidor 2	Mantenimiento de la posición	Criterio operacional	Exclusión Parcial	Realidad de acción al fondo (composición del sustrato marino)	R_8948	[Ver]
Escenario servidor 2	Reserva energética	Criterio de recurso energético	Exclusión Parcial	Demanda. Las unidades del valor máximo es en kWh/m2	R_8950	[Ver]
Nombre escenario	Descripción criterio	Tipo escenario	Tipo evaluación	Descripción subcriterio	Nombre criterio	Visualización

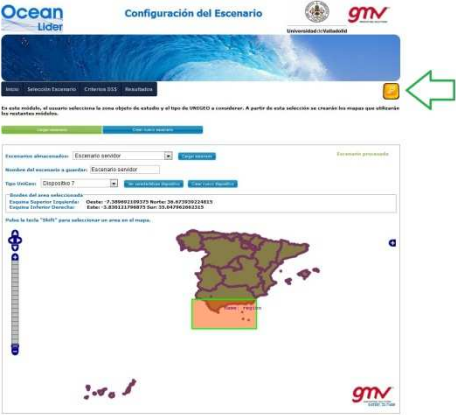

Mostrando: 1 - 4 de 4 entradas

Resultados finales

Nombre escenario	Tipo dispositivo	Tipo regla	Nombre criterio	Visualización
Escenario servidor 2	Undredoz - Fuera de costa. Rotante	Toma de decisión final	R_8962	[Ver]
Nombre escenario	Tipo dispositivo	Tipo regla	Nombre criterio	Visualización

Mostrando: 1 - 1 de 1 entradas

6.2.13. Salir del sistema

<p>Paso 1: Presionar botón de salir del sistema</p> <p>El sistema redirigirá a la página de inicio si presionamos sobre el botón de salir del sistema que se representa con una llave con fondo amarillo</p>	
<p>Paso 2: Página de inicio</p>	

6.3. Herramientas para el desarrollador

6.3.1. Configuración para la ejecución de comandos y scripts de GRASS en CMD de Windows

A continuación se indica la configuración de las variables de entorno para permitir lanzar la ejecución de comandos y scripts de GRASS en la ventana de símbolos del sistema¹³ de Windows. Se ha realizado la configuración de GRASS 6.4.1 en un sistema operativo Windows 7.

¹³ **CMD** o **símbolo del sistema** (en inglés *Command prompt*) es el intérprete de comandos de Windows.

Se puede proceder a la configuración desde Panel de control o desde la ventana de símbolos del sistema de Windows.

- **Configuración desde Panel de Control**

Para la realización de la configuración de las variables de entorno desde el panel de control de Windows, debemos ubicarnos en la pestaña de opciones avanzadas de la configuración del sistema e ir a variables de entorno (*Panel de Control->Sistema y Seguridad->Sistema->Cambiar configuración->pestaña Opciones Avanzadas->Variables de entorno*).

Añadimos las variables “GISBASE” y “GISRC”, con la ruta del directorio donde se instaló GRASS y la ruta al fichero sobre la información de la Base de datos de GRASS, respectivamente.

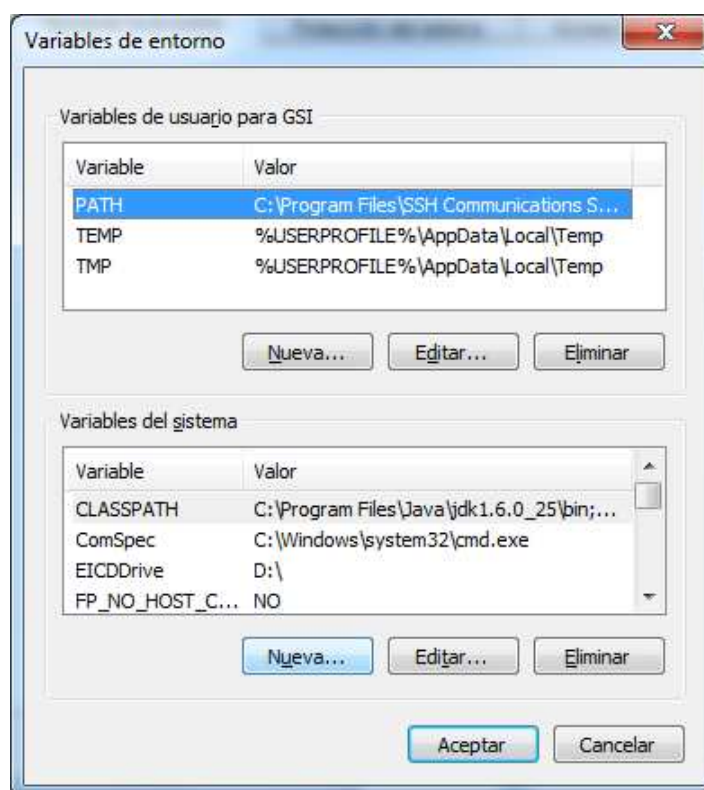


Figura 168: Ventana de variables de entorno de Windows 7

Debemos presionar el botón de “Nueva” y añadimos:

- Nombre de variable: **GISBASE**
Valor de variable: C:\Program Files\GRASS6.4.1;

Nombre de variable: **GISRC**

Valor de variable: C:\Users\Nombre Usuario\.grassrc6;

El archivo *.grassrc6* se ubica en el directorio del usuario de Windows por defecto en la instalación de GRASS. Este archivo puede ser editado para indicar dónde se encuentra la base de datos que gestiona GRASS de forma interna, la localización, el *mapset* de trabajo (directorio de mapas) y el modo en que se va a ejecutar GRASS. Este archivo será modificado cada vez que iniciemos GRASS y configurará los valores según el último acceso.

Un ejemplo del contenido de *.grassrc6*:

```
GISDBASE: C:\Users\GSI\Documents\GISDataBase
LOCATION_NAME: Peninsula
MAPSET: Pruebas
GRASS_GUI: text
```

Si queremos mantener una independencia de este archivo (que es el que utiliza GRASS en cada ejecución del programa) con la política de ejecución por línea de consola, haremos un duplicado de este archivo con las variables configuradas según nuestra conformidad y configuraremos la variable GISRC con la ruta al archivo duplicado.

Posteriormente añadimos las siguientes rutas a la variable “**Path**”¹⁴ que encontraremos en el panel de variables del sistema¹⁵ seleccionando la variable y presionamos en editar variable.

- La ubicación de los binarios ejecutables (comandos de GRASS) en las carpetas *bin* y *scripts*. Añadimos la siguiente línea de texto:

```
C:\Program Files\GRASS6.4.1\bin; C:\Program Files\GRASS6.4.1\scripts;
```

- La ubicación de las bibliotecas de GRASS. Añadimos la siguiente línea de texto:

```
C:\Program Files\GRASS6.4.1\lib; C:\Program Files\GRASS6.4.1\extralib;
```

- **Configuración desde la ventana de símbolos del sistema de Windows**

Se hará una configuración similar de las variables del apartado anterior. Se introducirá las siguientes secuencias por línea de comandos de la ventana de símbolos del sistema de Windows:

- Configuración de GISBASE:
set GISRC= C:\Program Files\GRASS6.4.1;
- Configuración de GISRC:

¹⁴ Es importante aclarar que debemos realizarlo sobre la variable del sistema y no del usuario, es decir, la variable “Path” y no “PATH”

¹⁵ **PATH** es la cadena de texto con las rutas donde el interprete debe buscar comandos o programas a ejecutar

```
set GISRC= C:\Users\Nombre Usuario\.grassrc6;
```

- Configuración de la variable Path¹⁶:
set Path=%Path%; C:\Program Files\GRASS6.4.1\bin; C:\Program Files\GRASS6.4.1\scripts; C:\Program Files\GRASS6.4.1\lib; C:\Program Files\GRASS6.4.1\extralib;

Esta configuración es válida para el sistema operativo Windows. Si queremos efectuar la configuración en un entorno Linux, las rutas de las bibliotecas deben estar definidas en una nueva variable llamada LD_LIBRARY_PATH, y exclusivamente las rutas de los binarios y comandos de GRASS en la variable PATH. El resto de variables no varía.

En este momento, podremos ejecutar cualquier comando de GRASS en la ventana de símbolos del sistema de Windows o ejecutar un script (en formato .BAT¹⁷) con secuencia de comandos de GRASS. El archivo de lotes ejecutable tendrá la siguiente estructura:

```
@echo off

Comando GRASS 1

Comando GRASS 2

_____
_____

Comando GRASS n
```

Figura 169: Ejemplo de script operacional con comandos de GRASS en formato .BAT para Windows

Notas de ejecución

Si nos pide la librería *libintl3.dll*, tenemos que hacer la instalación de la biblioteca de Windows de GNU [GNUWIN32, 2012] y copiar los archivos “*libiconv2.dll*” y “*libintl3.dll*” al directorio “*extralib*” de GRASS.

¹⁶ El intérprete de comandos de Windows representa la variable Path y PATH de forma unánime. Se podría aplicar para la variable PATH.

¹⁷ Es un **archivo batch**: un archivo de procesamiento por lotes de **Windows**

Si observamos que las rutas no se han configurado tras haber hecho la creación y modificación de las variables de entorno, debemos reiniciar el sistema.

6.3.2. Importación de una tabla Excel en PostgreSQL

Este estudio se ha realizado tras obtener los documentos Excel con los valores de idoneidad de las diferentes capas por parte de las entidades vinculantes al proyecto. Se van a contemplar diferentes vías para importar esta información en la base de datos con la intención de mantener de forma independiente los valores de idoneidad con las capas informativas del proyecto.

Según cómo se quieran almacenar los datos de una tabla Excel en PostgreSQL podemos actuar de diferentes maneras: almacenar los datos contenidos en la tabla del Excel como una tabla en la BBDD dentro de PostgreSQL (cada campo del Excel es un campo de la tabla) o almacenar los datos contenidos en la tabla Excel en un único campo de una tabla de la BBDD dentro de PostgreSQL, esto es posible gracias al tipo XML que soporta PostgreSQL.

Cada una de estas importaciones tiene sus propias características a la hora de tratar los datos. En este documento se va a mencionar cómo hacer la importación mediante estas dos vías y las características que nos puede aportar.

6.3.2.1. Importación de una tabla Excel campo a campo

En la importación de una tabla Excel campo a campo en la BBDD utilizamos **Microsoft Excel** para la traducción del fichero Excel a texto con delimitadores (tabuladores) y posteriormente utilizando la función **COPY** de PostgreSQL.

El primer paso es traducir el fichero Excel a texto con delimitadores. Para ello abrimos el Excel con Microsoft Excel, y nos ubicamos '*Archivo->Guardar como*'. En este cuadro de dialogo, desplegamos la pestaña de '*Tipo*' y seleccionamos '*Texto (delimitado por tabulaciones)*'.

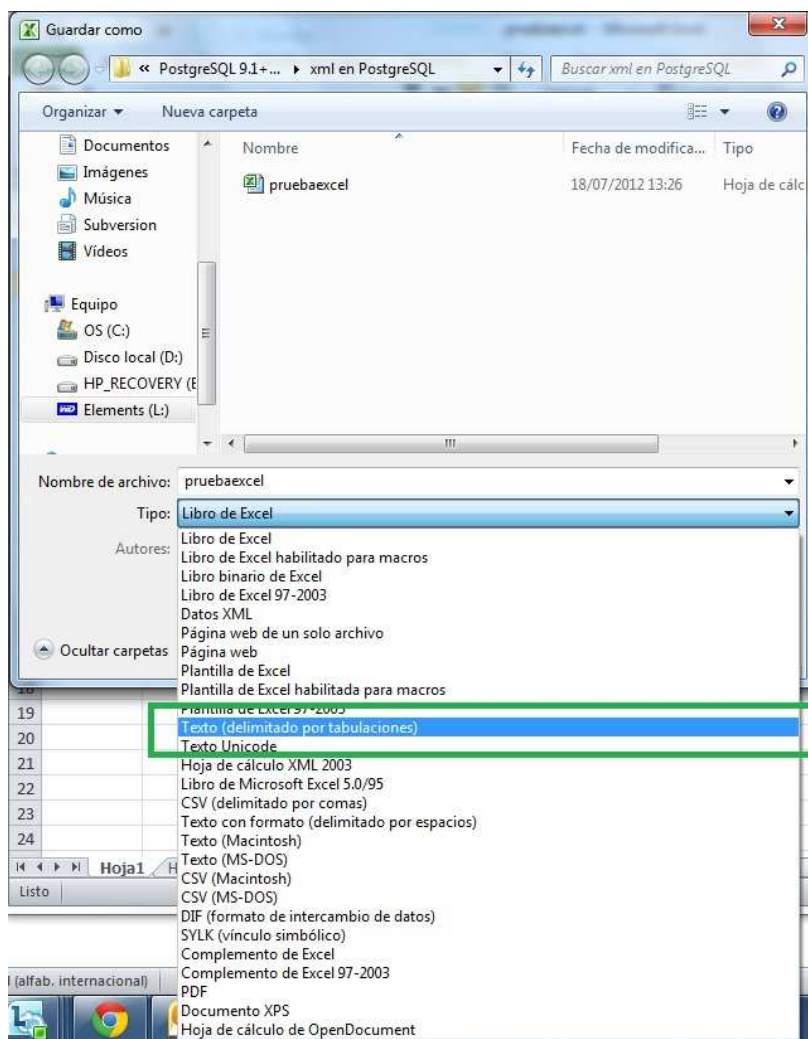


Figura 170: Cuadro de dialogo de ‘Guardar como’ de Microsoft Excel para seleccionar tipo de fichero

En este paso hemos traducido nuestra tabla Excel a texto delimitado con tabuladores que tiene extensión .TXT.

El siguiente paso es crear una tabla en nuestra BBDD de PostgreSQL que contenga tantas columnas como atributos contenga nuestra tabla Excel (no hace falta especificar una clave primaria). A continuación se muestra un ejemplo de una tabla Excel y la creación de la tabla en la BBDD.

	A	B
1	pruebaA	1
2	pruebaB	2

Figura 171: Ejemplo de una tabla Excel

```
CREATE TABLE prueba_xml
(
  columna1 character varying(100),
  columna2 character varying(100)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE prueba_xml
OWNER TO postgres;
```

El paso siguiente es ejecutar la siguiente consulta en la consola de ejecución de sentencias SQL de pgAdminIII utilizando la función *COPY* de PostgreSQL [PostgreSQL f, 2012]:

```
COPY prueba_xml FROM 'L:\GMV\PostgreSQL 9.1+PostGIS 2.0_soporte
raster y vectorial\xml en PostgreSQL\pruebaexcel.txt'
```

Por defecto esta función toma como delimitadores los tabuladores, si nuestro archivo de texto contiene delimitadores con el carácter ';' deberíamos introducir la siguiente sentencia:

```
COPY prueba_xml FROM 'L:\GMV\PostgreSQL 9.1+PostGIS 2.0_soporte
raster y vectorial\xml en PostgreSQL\pruebaexcel.txt' USING
DELIMITERS ';' ;'
```

Si comprobamos el resultado, obtenemos la siguiente tabla poblada acorde con el contenido del Excel original.

	columna1 character vai	columna2 character vai
1	pruebaA	1
2	pruebaB	2

Figura 172: Resultado de la importación de una tabla Excel a PostgreSQL

De esta manera hemos introducido los datos como cadena de caracteres. Si nuestra tabla contiene campos que son numéricos (integer, double...) y queremos tratarlos como tal, hay que especificarlo explícitamente al crear la tabla en la BBDD introduciendo el tipo numérico en el campo deseado. La importación hace la conversión de tipo a numérico.

Por ejemplo, creamos la tabla 'prueba_xml2' que contiene el segundo campo numérico de tipo entero:

```
CREATE TABLE prueba_xml2
(
  columna1 character varying(100),
  columna2 integer
)
WITH (
  OIDS=FALSE
);
ALTER TABLE prueba_xml2
OWNER TO postgres;
```

Introducimos en la consola de ejecución de SQL de pgAdminIII la sentencia de importación:

```
COPY prueba_xml2 FROM 'L:\GMV\PostgreSQL 9.1+PostGIS 2.0_soporte
raster y vectorial\xml en PostgreSQL\pruebaexcel.txt'
```

Nos dará como resultado la población de la tabla 'prueba_xml2' con la segunda columna de tipo entero.

	columna1 character va	columna2 integer
1	pruebaA	1
2	pruebaB	2

Figura 173: Resultado al poblar una tabla con un campo numérico

En caso que hayamos definido el tipo erróneamente a numérico y queramos introducir una cadena de texto, PostgreSQL nos avisará con un error en la importación.

```
CREATE TABLE prueba_xml3
(
  columna1 integer,
  columna2 character varying(100)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE prueba_xml3
OWNER TO postgres;
```

Ejecutamos la sentencia de importación:

```
COPY prueba_xml3 FROM 'L:\GMV\PostgreSQL 9.1+PostGIS 2.0_soporte
raster y vectorial\xml en PostgreSQL\pruebaexcel.txt'
```

Y nos indicará el error en la consola de ejecución de sentencias SQL de pgAdminIII:

```
ERROR: la sintaxis de entrada no es válida para integer:
«pruebaA»
CONTEXT: COPY prueba_xml3, línea 1, columna columnal: «pruebaA»
```

Una vez realizada la importación de los datos del Excel a una tabla en la BBDD de PostgreSQL podemos tratar los datos como si fuera una tabla tradicional en una BBDD relacional. Esta opción nos brinda la oportunidad de realizar las consultas de tratamiento y gestión de una BBDD relacional.

6.3.2.2. Alternativas teniendo en cuenta la variación en el tiempo de las tablas Excel

En este apartado se van a citar dos propuestas teniendo en cuenta la variación en el tiempo de las tablas Excel: utilizando el modelo EAV y utilizando el tipo XML de PostgreSQL. Se contempla algunos criterios que no se han tenido en cuenta en la propuesta anterior (soporte para tablas Excel de tipo dinámico), y se expone las ventajas y desventajas del tratamiento para estas nuevas propuestas.

6.3.2.2.1. Modelo EAV

El **modelo Entidad-Atributo-Valor** (EAV, Entity-Attribute-Value) también conocido como *Objeto-Atributo-Valor* o *esquema abierto* se usa en casos donde el número de atributos utilizados para describir una entidad es mayor que al ser aplicados individualmente a una entidad concreta. Se suele aplicar cuando:

- Entidades con atributos heterogéneos:
 - Entidades dinámicas en el tiempo, donde los atributos de una entidad son fijos pero puedan crecer o decrecer con el tiempo.
 - Entidades conceptualmente dinámicas cuyo número de atributos cambia según el contexto.
- Entidades con atributos homogéneos muy poco densos. Es decir, entidades con una desproporción muy grande entre el número de atributos posibles y el número de atributos con valor (no nulos).

Estos aspectos son interesantes cuando se prevé que una tabla Excel tenga tendencia a cambiar en el tiempo, se añadan o quiten atributos o la mayoría de las entradas a los atributos tengan valores nulos.

Este modelo se suele resolver sobre el modelo relacional con tres tablas: la de entidad, la de atributos posibles de dicha entidad y la de valores. La información se registra conceptualmente en la tabla de valores, que relaciona la información a través de tres columnas: clave ajena de entidad, clave ajena de atributo y valor.

Incidencia		Datos		DatosIncidencia		
Id	Fecha	Id	NombreAtributo	IdInc	IdDato	Valor
1	17/05/19 22:10	1	Longitud	1	1	5
2	18/05/19 10:13	2	Temperatura	1	4	A
3	18/05/19 17:05	3	Volumen	2	2	6
		4	Sector	2	3	4
				2	4	A

Figura 174: Aplicación del modelo EAV sobre el modelo relacional de BBDD [Balteus b, 2012]

Esta forma de importación permite modificar los datos de las tablas en un futuro de forma más eficiente y clara; y no se introduce valores nulos en la BBDD para atributos sin valores.

La problemática que surge al aplicar este modelo son:

- Consultas SQL complejas
- Tipo de datos único para todos los atributos, sea numérico, cadena de caracteres... Posteriormente tendrán que convertirse o complicar el modelo con distintas tablas de atributos en función del tipo de dato.

Una alternativa que surge para solventar la problemática sobre esta aplicación manteniendo el dinamismo es utilizar un atributo de tipo XML de una entidad de PostgreSQL. Se explica a continuación.

6.3.2.2.2. XML Y Xpath

El tipo de datos XML de PostgreSQL se utiliza para almacenar datos XML [PostgreSQL d, 2012]. La ventaja que proporciona es que podemos almacenar datos XML en un campo donde al hacer la importación PostgreSQL comprueba los valores de entrada para XML bien formados (well-formed), y soporta operaciones específicas para tratar los datos XML. En la versión 9.1 de PostgreSQL está instalado este nuevo tipo de datos.

El tipo XML puede almacenar documentos bien formados como define el estándar XML [W3C, 2012] o contenidos (son fragmentos de documentos XML). Esta característica nos permite evaluar si un valor XML es un *documento* o un *contenido* (fragmento). Podemos utilizar este tipo de datos para albergar datos XML estáticos o dinámicos.

Se puede contar con un número de atributos variable encapsulados en un documento XML en un solo campo de una entidad y evitar tener que usar el modelo EAV utilizando el *tipo XML* y procesarlo con la función *XPath*.

En el siguiente ejemplo se va a ilustrar la creación de una tabla con un campo tipo XML donde se albergará la misma tabla que el ejemplo anterior en formato XML.

```
CREATE TABLE prueba_tipoxml
(
  id serial NOT NULL,
  columna_xml xml
)
WITH (
  OIDS=FALSE
);
ALTER TABLE prueba_tipoxml
  OWNER TO postgres;
```

Para transformar la tabla Excel a formato XML, se necesita hacer la instalación previa del componente **XmlTools** de Excel [Ver *anexo H*]

Tras ejecutar los pasos para la transformación de la tabla Excel a XML obtenemos el siguiente esquema XML junto con sus datos:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Row>
    <Column1>pruebaA</Column1>
    <Column2>1</Column2>
  </Row>
  <Row>
    <Column1>pruebaB</Column1>
    <Column2>2</Column2>
  </Row>
</Root>
```

Insertamos en nuestra tabla una tupla que contiene el esquema XML que hemos creado. Se puede insertar sin utilizar la función XMLPARSE que pasa de un esquema en cadena de caracteres a un esquema XML nativo si el XML está bien formado. El intérprete de PostgreSQL lo hace automáticamente.

```
INSERT INTO prueba_tipoxml ("columna_xml") VALUES ('<?xml
version="1.0" encoding="UTF-8" standalone="yes"?><Root
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"><Row><Column1>pruebaA</Column1><Column2>1</Column2></Ro
w><Row><Column1>pruebaB</Column1><Column2>2</Column2></Row></Root
>')
```

ó

```
INSERT INTO prueba_tipoxml ("columna_xml") VALUES
(XMLPARSE(DOCUMENT'<?xml version="1.0" encoding="UTF-8"
standalone="yes"?><Root
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"><Row><Column1>pruebaA</Column1><Column2>1</Column2></Ro
w><Row><Column1>pruebaB</Column1><Column2>2</Column2></Row></Root
>' ) )
```

Para consultar el contenido del campo de la BBDD que contiene el XML se puede realizar la siguiente consulta:

```
SELECT columna_xml::xml FROM prueba_tipoxml where columna_xml is
not null
```

Para visualizar el contenido de una columna específica de nuestro XML utilizamos la función **xpath** de PostgreSQL:

```
xpath (xpath_expr, xml_value[, nsarray])
```

La función **xpath** evalúa la expresión *xpath_expr* contra el valor *xml_value* (debe ser un documento XML bien formado), devolviendo un vector de valores XML correspondiente al conjunto de nodos producidos por la expresión XPath. El tercer argumento es opcional, es el vector bidimensional de espacios de nombres (nombre espacio de nombres, URI espacio de nombres) que use el documento XML *xml_value*.

Por ejemplo, para visualizar el contenido de la columna 1 de nuestro XML realizamos la siguiente consulta:

```
SELECT xpath('//Column1',columna_xml::xml) as Resultados FROM
prueba_tipoxml where columna_xml is not null
```

Resultado: "{<Column1>pruebaA</Column1>,<Column1>pruebaB</Column1>}"

Para consultar el contenido XML que corresponda con el valor y nombre de una columna, se utiliza la sentencia:

```
SELECT xpath('//Row/Column1 [text() =
'pruebaA']',columna_xml::xml) as Resultados FROM
prueba_tipoxml where columna_xml is not null
```

Resultado: "<Column1>pruebaA</Column1>"

Es importante mencionar que la función *xpath* devuelve un vector de valores XML. Por ejemplo, si deseamos consultar el contenido de una columna específica de una tupla XML:

```
INSERT INTO prueba_tipoxml3 ("columna_xml") VALUES
(XMLPARSE(DOCUMENT '<?xml version="1.0" encoding="UTF-8"
standalone="yes"?><Root
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"><Row><motivo>tiempo</motivo></Row><Row><motivo>hardware
</motivo></Row><Row><cantidad>2</cantidad></Row><Row><cantidad>3<
/cantidad></Row></Root>' ) ) ;
```

```
SELECT xpath('//motivo/text()',columna_xml::xml) as Resultados
FROM prueba_tipoxml4 where columna_xml is not null
```

Resultado: "{tiempo,hardware}"

```
SELECT xpath('//cantidad/text()',columna_xml::xml) as Resultados
FROM prueba_tipoxml4 where columna_xml is not null
```

Resultado: "{2,3}"

Se pueden añadir atributos a las etiquetas (nodos). Un nodo puede tener tantos atributos como se desee. No obstante, dichos atributos no se consideran como hijos del nodo, sino más bien como etiquetas añadidas al nodo elemento. Esto es importante a la hora de utilizar la función *xpath*. Por ejemplo:

```
INSERT INTO prueba_tipoxml ("columna_xml") VALUES ('<?xml
version="1.0" encoding="UTF-8" standalone="yes"?><Root
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><Row><data
name="column1" value="pruebaA">pruebaA</data><data name="column2"
value="1">1</data></Row><Row><data name="column1"
value="pruebaB">pruebaB</data><data name="column2"
value="2">2</data></Row></Root>')
SELECT
xpath('//data[@name=''column1'']/@value',columna_xml::xml) as
Resultados FROM prueba_tipoxml where columna_xml is not null
```

Resultado: "{pruebaA,pruebaB}"

En la referencia [Rivas, 2012] se puede encontrar un tutorial sobre Xpath y algunos ejemplos de utilidad.

Las ventajas que proporciona esta forma de implementación es el dinamismo a la hora de insertar nuevos contenidos (fragmentos XML) donde se añaden nuevos campos a los documentos XML y proporciona un mecanismo de consulta más sencillo a través de la función *xpath*.

También proporciona un ahorro en el almacenaje, cada fila del XML contiene una serie de campos propios para cada fila (sólo los campos que contienen datos), en caso que lo quisiéramos almacenar en forma de tabla en la BBDD de la manera tradicional tendríamos celdas sin datos para completar toda la tabla.

La dificultad de implementarlo con el tipo XML estriba en la traducción de la tabla Excel a formato XML y la creación de las consultas para la búsqueda en el XML con la función *xpath*. Aunque la mayoría de estos inconvenientes ya se han solventado en este documento.

6.3.2.3. Conclusiones

Aunque este estudio se ha realizado para importar a la base de datos la información proporcionada por las entidades vinculadas sobre los valores de idoneidad de forma

independiente con las capas informativas del proyecto, se ha determinado incluir dichos valores a la información de las capas vectoriales. Esta información por tanto queda incluida dentro de cada polígono que contienen, en la que alojará un valor de idoneidad por cada tipo de dispositivo. Las capas con los valores de idoneidad pertenecen a la base de datos *BBDDOceanLider_restricciones*.

6.3.3. Importar capas vectoriales y raster en PostgreSQL 9.1 + PostGIS 2.0

6.3.3.1. Importar capas raster

Si queremos importar una capa raster dentro de la BBDD utilizaremos los comandos *raster2pgsql* y *psql* de PostGIS 2.0. Para hacer la importación de una forma correcta, tenemos que indicar el sistema de referencia espacial (SRID) de la capa raster, la ubicación en el sistema de ficheros de la capa raster, el esquema donde queremos que se importe, el nombre que se asignará a esta capa en la BBDD y el nombre del fichero SQL que genera el comando *raster2pgsql* para la importación de una capa raster:

```
raster2pgsql -s 23030 -I -C -M
"C:\Users\tregmm\Desktop\BBDD_Ocean_Lider\P\02_Capas\02_IT\RE\VI\
Viento_Anual.img"
-F public.Viento_Anual >
"C:\Users\tregmm\Desktop\vientoanual.sql"
psql -U postgres -d BBDDOceanLider -h localhost -p 5432 -f
"C:\Users\tregmm\Desktop\vientoanual.sql"
```

Si queremos realizar este procedimiento en un único paso, utilizaremos tuberías para concatenar la salida de *raster2pgsql* con la entrada de *psql*. Este proceso es útil a la hora de crear un script de importación de capas de forma automática:

```
raster2pgsql -s 23030 -I -C -M
"C:\Users\tregmm\Desktop\UnionHsMedia.tif" -F public.UnionHsMedia
| psql -U postgres -d BBDDOceanLider -h localhost -p 5432
```

Nota: Se ha desestimado el almacenaje de las capas raster en PostGIS por su inaccesibilidad desde GRASS 6.4. En la versión 6.4 de GRASS no se puede hacer la importación de capas raster alojadas en PostGIS.

6.3.3.2. Importar capas vectoriales

Se puede hacer la importación de capas vectoriales tipo Shape a la versión PostGIS 2.0 mediante tres vías: el **gestor de importación soportado por QGIS**, mediante los comandos *shp2pgsql* y *psql* similar a la importación de capas raster, y con la aplicación **ogr2gui** de licencia GPL. A continuación se van a exponer los pasos para la importación de shapfiles a PostGIS 2.0 que se han probado sin obtención de errores.

Utilizando el gestor de importación de capas vectoriales tipo Shape de QuantumGIS a PostgreSQL, seleccionamos en el menú de herramientas “Base de Datos->Importar (Spit)->Importar archivos shapes a PostgreSQL”.

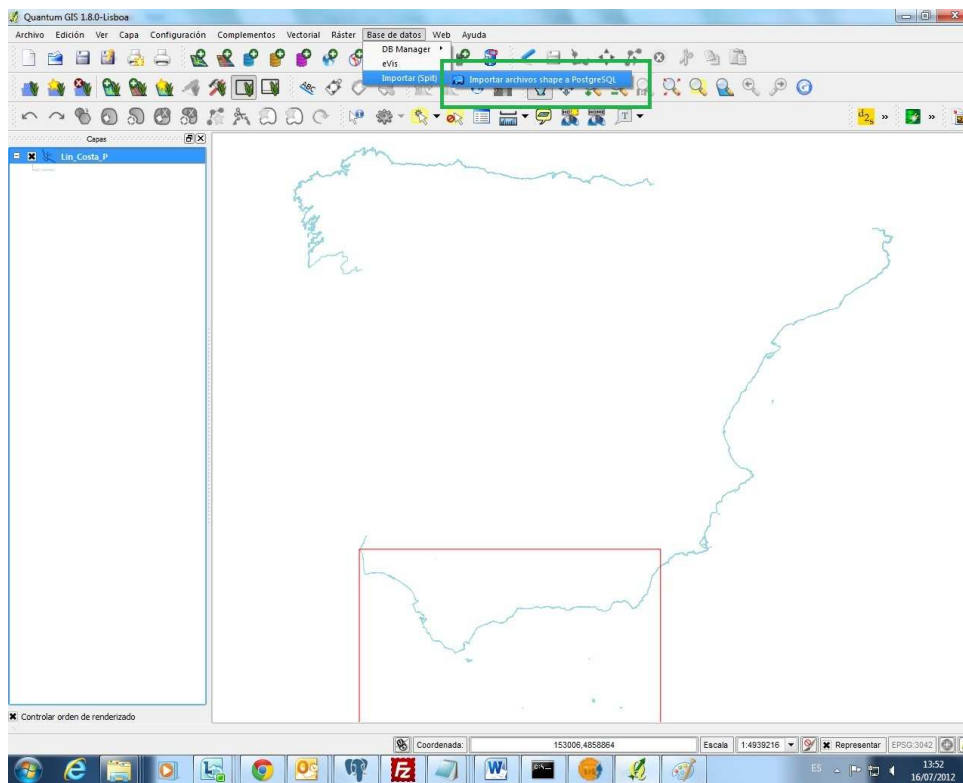


Figura 175: Importación de capas vectoriales tipo Shape a PostgreSQL desde QuantumGIS

Posteriormente nos aparece un cuadro de diálogo para conectarnos a la BBDD PostgreSQL. Para hacer la importación tenemos que configurar una nueva conexión a PostgreSQL si no la tenemos configurada previamente. Se tiene que indicar los datos de la conexión a PostgreSQL y la BBDD destino a donde se importará el Shape.

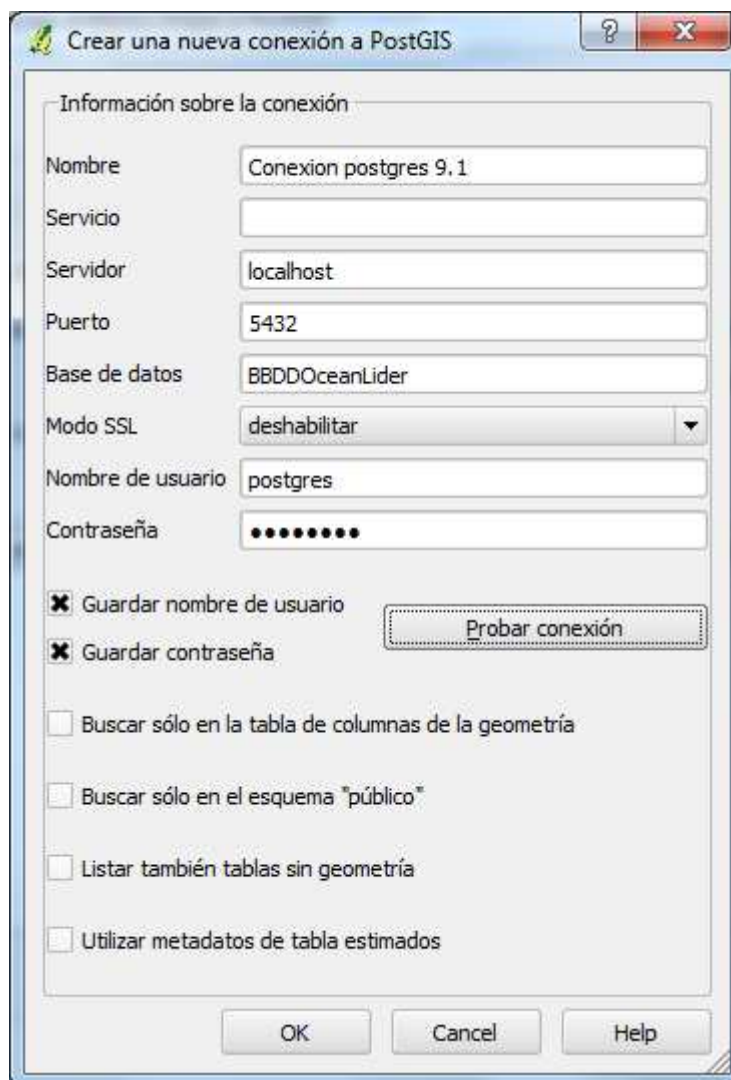


Figura 176: Cuadro de diálogo para la conexión a PostgreSQL

Una vez configurada la conexión, en el cuadro de diálogo principal, indicaremos la ruta del fichero shape que queremos importar, el esquema donde queremos que se importe la capa vectorial en la BBDD (se selecciona el esquema "public" por defecto) y seleccionamos la conexión que hemos creado en el paso anterior.

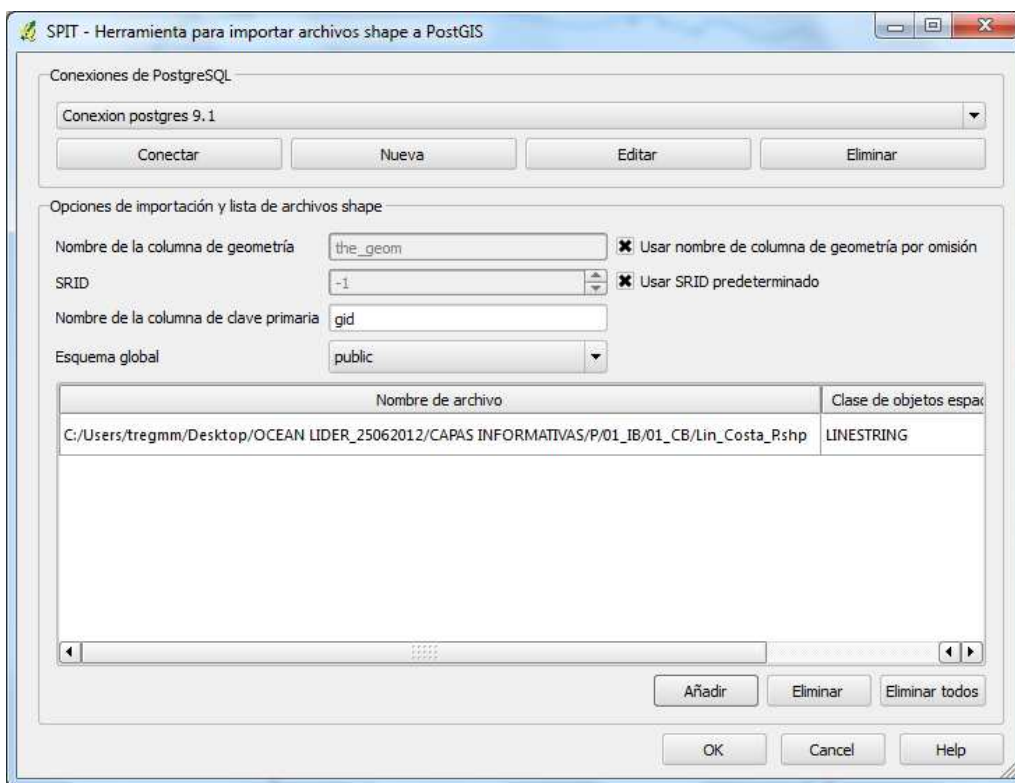


Figura 177: Cuadro de dialogo principal para la importación de shaples a PostreSQL en QGIS

Utilizando los **comandos shp2pgsql y psql** se recomienda indicar la codificación de las tablas asociadas al shape ya que sino obtendremos un error en la importación.

```
shp2pgsql -W LATIN1 -I -s 23030  
"C:\Users\tregmm\Desktop\BBDD_Ocean_Lider\P\02_Capas\01_IB\01_CB\  
Lin_Costa_P.shp" public.Lin_Costa_P | psql -U postgres -d  
BBDDOceanLider -h localhost -p 5432
```

Utilizando la **aplicación ogr2gui**, tendremos que en primer lugar disponer de esta herramienta disponible en el siguiente enlace:

<http://www.ogr2gui.ca/en/index.php>

Esta herramienta es una interfaz gráfica que utiliza el comando *ogr2ogr* para convertir formatos de GDAL (Geospatial Data Abstraction Library). Entre otras funcionalidades, nos permite importar/exportar archivos tipo shapefile a otros formatos y viceversa, realizar transformaciones de las capas de datum y proyección, unir varios archivos shapefile en un único archivo o incluso subir una carta náutica S57 dentro de PostgreSQL.

Para hacer la importación de una capa vectorial tipo Shape a PostgreSQL, en el cuadro de dialogo que nos aparece tenemos que indicar en la parte de los datos de destino: el formato de la capa vectorial, la ruta y nombre del fichero que queremos importar a PostgreSQL, y en la parte de los datos del destino indicamos la información de la conexión a la BBDD a la que queremos que se importe el Shape y la proyección (SRID).

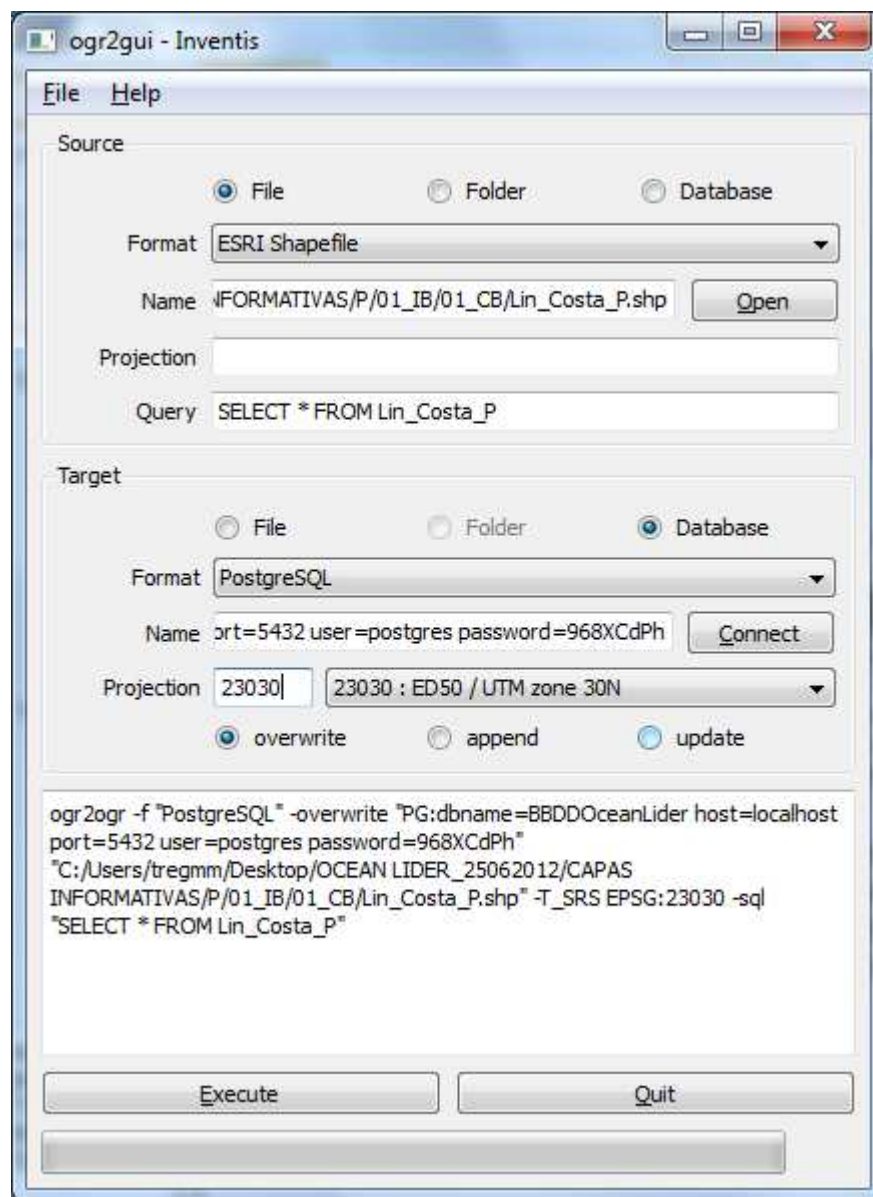


Figura 178: Cuadro de dialogo de ogr2gui para la importación de una capa vectorial tipo Shape a PostgreSQL

Capítulo

7

CONCLUSIONES

Se ha construido un **sistema inteligente de apoyo a la toma de decisión** mediante **técnicas de evaluación multicriterio** sustentando por operaciones de **Sistemas de Información Geográficos** y aplicado a **tecnologías web**.

El objetivo de la herramienta es la identificación de las zonas factibles para el emplazamiento de unidades de generación de energía oceánica tras la configuración y evaluación de diferentes criterios de índole físico, operacional, recurso energético y medioambiental.

Para realizar tal herramienta se ha partido de un prototipo realizado en el proyecto fin de carrera con título “*Estudio de herramientas SIG de libre distribución para la selección de emplazamientos de unidades de generación de energía oceánica*”, en el que se estudiaron las técnicas y tecnologías que se iban a usar. El prototipo estaba basado en la creación de escenarios y configuración de algunos criterios de ensayo basados en otros sistemas como WindSpeed.

En la actual herramienta DSS se han programado los criterios consensuados por los expertos con operaciones parametrizables en la que se usa la sintaxis de GRASS para ser ejecutada en este SIG. Como en GRASS carece de operaciones pertenecientes a las técnicas de evaluación multicriterio éstas han sido implementadas e integradas dentro de la herramienta DSS como son la operación fuzzy, operación MLP y la operación OWA.

También ha sido necesario el estudio de la configuración del entorno para ejecutar operaciones fuera de un SIG y un planificador para lanzar la ejecución en un servidor, que en este caso carecía el prototipo del que se partía.

Ha sido un proyecto de bastante envergadura no sólo por las técnicas, tecnologías empleadas y los conceptos aplicados, sino también por la presencia de varios procesos intermedios que han sido necesarios para su funcionamiento como han sido: la importación múltiple de capas vectoriales a la base de datos, la generación del mapa profundidad como un modelo digital de elevaciones a partir de isolíneas batimétricas o la creación de los mapas vectoriales y raster de unión en la proyección EPSG 4326.

Por lo tanto, la primera conclusión es que ha sido posible cumplir con los objetivos marcados a cambio de integrar más tecnologías y datos de los inicialmente esperados. En segundo lugar, la presencia de un grupo de desarrollo (la autora integrada en el Grupo GSI de la UVA o en el

grupo de Marítimo de GMV) y tres potenciales clientes (GMV, Iberdrola y PROES) ha requerido de más iteraciones de las inicialmente consideradas por los continuos cambios en las especificaciones de los criterios.

Además, la integración del trabajo en el proyecto OCEANLIDER ha constatado la dificultad del desarrollo de aplicaciones en grandes proyectos por las tremendas interdependencias entre las fuentes de información y la disponibilidad de éstas.

Todas estas circunstancias han permitido a la autora desarrollar múltiples tareas con muy distintos grados de responsabilidad en distintos entornos de trabajo, lo cual ha sido una experiencia enriquecedora a nivel personal y profesional.

Finalmente, el desarrollo de esta herramienta significa una apuesta por las energías renovables, una búsqueda de la optimización en su empleo y aprovechamiento, una ampliación a otros sistemas de apoyo a la decisión existentes (introducción en la producción de energía generada por el movimiento de las olas o las mareas) y una apertura para utilizar este tipo de técnicas de apoyo a la toma de decisión para algún otro fin específico.

BIBLIOGRAFÍA

- [Ascough, 2002] “Multicriteria Spatial Decision Support Systems: Overview, Applications, and Future Research Directions”, Ascough II, J.C., Rector, H.D., Hoag, D.L., McMaster, G.S., Vandenberg, B.C., Shaffer, M.J., Wertz, M.A., Ahuja, L.R. Online Proceedings of the International Environmental Modelling and Software Society Conference on Integrated Assessment and Decision Support [3] 175–180. <http://www.iemss.org/iemss2002/>. 2002.
- [Balteus a, 2012] “XML con PosgreSQL”. <http://balteus.blogspot.com.es/2010/06/xml-con-postgresql.html>, visitado 20/07/2012.
- [Balteus b, 2012] “Alternativas EAV con XML (en PostgreSQL 8.3)”. <http://balteus.blogspot.com.es/2009/05/alternativas-eav-con-xml-en-postgresql.html>, visitado 20/07/2012.
- [BAR09] Barrio Solorzano M. “Apuntes de Ingeniería de Software I”. Universidad de Valladolid. 2008-2009.
- [Calderón, Álvarez, Axpuc, 2011] “Utilización de los sistemas de información geográficos para la propuesta del manejo de los recursos naturales renovables de la parcialidad Chipuac del municipio de Totonicapán”. E.E. Calderón Oliva, E.A. Álvarez Castillo, J.R. Axpuc Corado. Universidad de San Carlos de Guatemala. Mayo 2011. http://biblioteca.usac.edu.gt/tesis/08/08_0213_MT.pdf, visitado 07/05/2013.
- [Carver, 1991] “Integrating multicriteria evaluation with SIG” S.J. Carver. Int. J. Geographics Information Systems. 1991.
- [Carver y Openshaw, 1996] “Using SIG to explore the technical and social aspects of site selection for radioactive waste disposal facilities”. S. Carver, S. Openshaw. Informe. 1996.
- [ComputerHope, 2013] “Microsoft DOS set command”. <http://www.computerhope.com/sethlp.htm>, visitado 07/05/2013.
- [Eastman y cols. 1993] “A procedure for Multi Objective Decision Making in SIG under Conditions of Conflicting Objectives”. Eastman, J.R., Kyem, P.A.K and Tolendano, J. Actas de la Fourth ESIG Conference, Genoa Italy. 1993.
- [Fernández-Coppel, 2001] “La proyección UTM (Universal Transversa Mercator)”. I.A. Fernández-Coppel. Escuela Técnica Superior de las Ingenierías Agrarias de Palencia. Febrero 2001. <http://www.cartesia.org/data/apuntes/cartografia/cartografia-utm.pdf>, visitado 07/05/2013.
- [ForosWeb a, 2012] “Insertar XML en Posgres”. <http://www.forosdelweb.com/f99/insertar-xml-postgres-621343/>, visitado 20/07/2012.
- [ForosWeb b, 2012] “Importar datos Posgres”. <http://www.forosdelweb.com/f99/importar-datos-postgres-436363/>, visitado 20/07/2012.
- [Geocomuna, 2012] “Generar un MDE con Software libre”. <http://geocomuna.wordpress.com/2010/10/14/generar-mde-con-software-libre/>, visitado 27/07/2012.
- [GNUWIN32, 2012] “LibIntl for Windows”. <http://gnuwin32.sourceforge.net/packages/libintl.htm>, visitado 08/04/2012.
- [Gómez y Barredo, 2005] “Sistemas de información geográfica y evaluación multicriterio en la ordenación del territorio”. M. Gómez Delgado y JI Barredo Cano. Ed. RA-MA, Madrid. 2ª Edición. 2005.
- [GRASS] Geographic Resources Analysis Support System. <http://www.phygeo.uni-hannover.de/grass/index.php>

- [GRASS, 2012] “Command v.surf.bspline”.
http://grass.fbk.eu/gdp/html_grass64/v.surf.bspline.html, visitado 27/07/2012.
- [GRASS, 2013] “GRASS Home Page”. OSSEO. <http://grass.osgeo.org/>, visitado el 08/02/2013.
- [Malczewski2006] Jacek Malczewski. “Integrating multicriteria analysis and geographic information systems: the ordered weighted averaging (OWA) approach”.
<http://inderscience.metapress.com/content/5n7u38n4mq45npau/>, visitado 08/02/2013.
- [Hansen, 2005] “SIG-based MultiCriteria Analysis of Wind Farm Development.” H.S. Hansen. Proceedings of the 10th Scandinavian Research Conference on Geographical Information Science (ScanSIG), pp. 75-85, ISBN: 91-7323-126-6, Stockholm, Sweden, 13-15 June 2005, Department of Planning and Environment, Stockholm.
- [Hexacta, 2013] “GIS- Visualizando un mapa con MapServer”. <http://blog.hexacta.com/hat/gis-mapserver/>, visitado 09/05/2013.
- [IDRISI TAIGA] IDRISI Taiga. Clark Laboratories. <http://www.clarklabs.org/products/idrisi-taiga.cfm> 2011
- [Jiang e Eastman, 2000] “Application of fuzzy measures in multicriteria evaluation in SIG”, H. Jiang. y J.R. Eastman. International Journal of Geographical Information Sciences, Vol. 14, pp.173–184. (2000).
- [JMT, 2013] “Lenguaje de Comandos- Scripts .CMD .BAT- Consideraciones sobre ámbito de variables”. <http://multingles.net/docs/jmt/comandos/comandos5.html>, visitado 07/05/2013.
- [Luger 2009] Artificial Intelligence : structures and Strategies for Complex Problem Solving. G. F. Luger. Addison-Wesley, 2009.
- [Malczewski, 2006] “Integrating multicriteria analysis and geographic information systems: the ordered weighted averaging, OWA, approach.” J. Malczewski. Int. J. Environmental Technology and Management, Vol. 6, Nos. 1/2, 2006 7. 2006.
- [MapServer a, 2013] “Raster Data”.<http://mapserver.org/input/raster.html#raster>, visitado 09/05/2013.
- [MapServer b, 2013] “Vector Data”.<http://mapserver.org/input/vector/index.html#vector>, visitado 09/05/2013.
- [MapServer c, 2013] “About MapServer”. <http://mapserver.org/about.html>, visitado 09/05/2013.
- [MapServer d, 2013] “An introduction to MapServer”
<http://mapserver.org/introduction.html#introduction>, visitado 09/05/2013.
- [MapServer e, 2013] “Layer”.<http://mapserver.org/mapfile/layer.html>, visitado 09/05/2013.
- [MapServer f, 2013] “Redesign of LOG/DEBUG output mechanisms”.<http://mapserver.org/development/rfc/ms-rfc-28.html#rfc28>, visitado 09/05/2013.
- [MapServer g, 2013] “Mapfile”. <http://mapserver.org/mapfile/index.html>, visitado 09/05/2013.
- [Microsoft a, 2012] “Crear un archivo de datos XML y un archivo de esquema XML a partir de datos de la hoja de cálculo”. <http://office.microsoft.com/es-hn/excel-help/crear-un-archivo-de-datos-xml-y-un-archivo-de-esquema-xml-a-partir-de-datos-de-la-hoja-de-calculo-HA010342365.aspx>, visitado el 20/07/2012.
- [Microsoft b, 2012] “Microsoft Answer: Cannot create xml table in Excel 2010 with 2003 xml tolos add-in”. http://answers.microsoft.com/en-us/office/forum/office_2010-customize/cannot-create-xml-table-in-excel-2010-with-2003/bc89e78b-00e9-44ff-b7fe-d91378a4af58, visitado 20/07/2012.
- [Nogués, 2000] “Delimitación Espacial de la Probabilidad de Uso Residencial del Suelo en el Entorno de Zaragoza con Análisis Multivariable” David Nogués Bravo. En “Tecnologías Geográficas para el Desarrollo Sostenible”, Departamento de Geografía. Universidad de Alcalá, pgs. 115-135. 2000.

- [OpenLayers, 2013] "OpenLayers: free maps for the web". <http://openlayers.org/>, visitado 09/05/2013.
- [OSGEO, 2012] "Contour lines to DEM". http://grass.osgeo.org/wiki/Contour_lines_to_DEM, visitado 27/07/2012.
- [OWFLOW] "OFFSHORE WIND FARM LAYOUT OPTIMIZATION OWFLOW PROJECT: AN INTRODUCTION", C.N. Elkinton, J.F. Manwell, J.G. McGowan.
- [PostGIS a, 2013] "PostGIS 1.5.2 Manual". <http://postgis.refractions.net/docs/>, visitado 08/02/2013.
- [PostGIS b, 2013] "PostGIS Home Page". <http://postgis.refractions.net/>, visitado 08/02/2013.
- [PostGIS c, 2013] "PostGIS 2.0.0 SVN Manual". OSGEO. <http://postgis.refractions.net/documentation/manual-svn>, visitado 08/02/2013.
- [PostGIS d, 2013] "PostGIS Raster Home Page". OSGEO. <http://trac.osgeo.org/postgis/wiki/WKTRaster>, visitado 08/02/2013.
- [PostGIS e, 2013] "PostGIS Raster. Raster Reference". OSGEO. http://postgis.refractions.net/documentation/manual-svn/RT_reference.html, visitado 08/02/2013.
- [PostgreSQL a, 2013] "PostgreSQL 9.1.8 Documentation. Psq command". <http://www.postgresql.org/docs/9.1/static/app-psql.html>, visitado el 20/02/2013.
- [PostgreSQL b, 2013] "PostgreSQL 9.2.3 Documentation". <http://www.postgresql.org/docs/current/interactive/multibyte.html>, visitado 20/02/2013.
- [PostgreSQL c, 2013] "Documentation- Data Types". <http://www.postgresql.org/docs/8.3/static/datatype-numeric.html>, visitado 07/05/2013.
- [PostgreSQL d, 2012] "XML Type". <http://www.postgresql.org/docs/8.3/interactive/datatype-xml.html>, visitado 20/07/2012.
- [PostgreSQL e, 2012] "XML Functions". <http://www.postgresql.org/docs/devel/static/functions-xml.html>, visitado 20/07/2012.
- [PostgreSQL f, 2012] "COPY". <http://www.postgresql.org/docs/8.1/static/sql-copy.html>, visitado 20/07/2012.
- [Quartz, 2013] "Enterprise Job Scheduler". <http://quartz-scheduler.org/>, visitado 15/05/2013.
- [Ramírez 2007] "El proceso de Análisis Jerárquico con base en funciones de producción para planear la siembra de maíz de temporal". AM Ramírez. Tesis Universidad Montecillo. México. 2008.
- [Rinner y Malczewski 2002] "Web-enabled spatial decision analysis using Ordered Weighted averaging (OWA)". J. Geographical Systems, 4:285-403. 2002.
- [Rivas, 2012] "Tutorial de XPath". <http://geneura.ugr.es/~victor/cursillos/xml/XPath/>, visitado 20/07/2012.
- [Rubio, 2012] "Estudio comparativo de herramientas SIG y SGBD espaciales". David Rubio. ETS de Ingeniería Informática. Universidad de Valladolid. 2012.
- [Saaty, 1980] T.L. Saaty. The Analytic Hierarchy Process, McGraw-Hill, New York. 1980
- [Snaider, 2010] "Proyecciones cartográficas y sistemas de referencia". Patricia Snaider. UNNE. Junio 2010. <http://hum.unne.edu.ar/revistas/geoweb/Geo13/archivos/snaider10.pdf>, visitado 07/05/2013.
- [Santos, 1997] "El planteamiento teórico multiobjetivo/multicriterio y su aplicación a la resolución de problemas medioambientales y territoriales, mediante los S.I.G. Ráster". JM Santos Preciado. Espacio, Tiempo y Forma, Serie VI, Geografía, t. 10, 1997, págs. 129-151
- [Scribd, 2012] "Tutorial rápido para crear un MDE a partir de cotas con GvSIG". <http://es.scribd.com/doc/22790044/Tutorial-Mde-Gvsig>, visitado 27/07/2012.
- [UPEDU, 2013] Unified Process for EDUcation. <http://www.yoopeedoo.org/upedu/>, visitado 19/06/2013.

[**vanderMerwe, 1997**] “SIG-aided land evaluation and decision-making for regulating urban expansion: A South African case study”. Van der Merwe, J. H. *GeoJournal* 43: 135–151. Kluwer Academic Publishers. 1997

[**Zadeh, 1965**] Fuzzy Sets. L. A. ZADEH. *INFORMATION AND CONTROL*. 8, 338-353 [1965]

[**W3C, 2012**] “Extensible Markup Language (XML) 1.1.”. <http://www.w3.org/TR/2006/REC-xml11-20060816/#NT-document>, visitado 20/07/2012.

[**WikiPostgreSQL, 2012**] “XML Support”. http://wiki.postgresql.org/wiki/XML_Support, visitado 20/07/2012.

Apéndice

A

CONTENIDO DEL CD-ROM

Acompañando a esta memoria se encuentra un CD – ROM que contiene la herramienta DSS con el código fuente, la memoria en un fichero PDF, unos ficheros de restauración de las bases de datos *BBDDOceanLider* y *BBDDOceanLider_restricciones*, los ficheros de creación e inserción de la base de datos *DSS_nueva*, bibliotecas GNU añadidas en caso de errores en la ejecución de comandos de GRASS, los ejecutables de instalación de los programas que necesita para funcionar, y la estructura de ficheros creada para el funcionamiento de la herramienta DSS.

Los directorios del CD – ROM son los siguientes:

- **Backup_BBDDOceanLider:** contiene los archivos de restauración de las bases de datos *BBDDOceanLider* y *BBDDOceanLider_restricciones*. Estas bases de datos contienen las capas vectoriales informativas y de restricciones que intervienen en el proyecto. Es el resultado de las actividades de la importación múltiple de capas a PostgreSQL 9.1 + PostGIS 2.0 y la unión de capas a una única proyección (WGS84).
- **BBDD_DSS_nueva:** contiene los archivos SQL de creación, inserción y borrado de la base de datos *DSS_nueva*. Esta base de datos es donde se almacena los datos de la herramienta DSS. Contiene las secuencias de creación del modelo estático y dinámico, y la inserción de los datos iniciales que necesita la herramienta para funcionar reflejados en la presente memoria.
- **Bibliotecas_GRASS:** contiene bibliotecas de Windows de GNU para solventar los errores de lanzamiento de operaciones de GRASS fuera del entorno. Contiene los archivos *libinconv2.dll* y *libintl3.dll*.
- **Ejecutables_instalación:** contiene los ficheros ejecutables de instalación de los programas que necesita la herramienta DSS para funcionar. Se proporcionan los ficheros de instalación de: *GRASS 6.4*, *PostgreSQL 9.1*, *PostGIS 2.0* y *NetBeans 7.1*.
- **Estructura_ficheros_servidor:** contiene la estructura de directorios y ficheros que necesita la herramienta DSS para funcionar. La carpeta raíz se llama *OceanLider* y debe ser albergada dentro de nuestro directorio de usuario en Windows. Contiene:

- **Resultados:** directorio que almacena los resultados de los escenarios que se hayan ejecutado. En él se crearán carpetas correspondientes a cada escenario ejecutado que contendrán el script operacional, las capas raster generadas del resultado final y resultados intermedios, ficheros de seguimiento de errores, y los ficheros .MAP para la visualización de las capas raster.
- **Union:** directorio que almacena las capas raster perteneciente a capas de entrada en la herramienta DSS. También contiene los ficheros .MAP de algunos mapas de visualización en la herramienta DSS.
- **.grassrc6_wgs84:** fichero de configuración del entorno de trabajo de GRASS para establecer la localización (LOCATION) y directorio de mapas (MAPSET).
- **Herramienta_DSS:** contiene el código fuente de la herramienta DSS. Contiene los cuatro directorios de las cuatro componentes: datos, entities, motor e interfaz.
- **Memoria:** contiene la memoria del proyecto en un fichero PDF, en versión extendida y reducida.

Apéndice

B

DICCIONARIO

- **ADF- ArcInfo Binary Coverages:** son un conjunto de archivos binarios internos de ARC/INFO para almacenar datos vectoriales.
- **ArcGIS:** conjunto de productos software en el campo de los SIG producidos y comercializados por ESRI. Cabe destacar *ArcGIS Desktop* (ArcGIS de escritorio) que son una familia de aplicaciones SIG de escritorio de pago.
- **ARC/INFO o ArcInfo:** es un completo SIG de escritorio producido y comercializado por ESRI a partir del año 1982. Fue originalmente un sistema basado en línea de comandos. Actualmente las habilidades de procesamiento por línea de comandos se desarrolla a través de la interfaz gráfica del SIG de escritorio ArcGIS.
- **ArcIMS- Arc Internet Map Server:** es un servidor web de mapas producidos por ESRI. Se trata de un SIG diseñado para servir mapas a través de Internet.
- **ArcSDE ó SDE- Spatial Database Engine:** es un servidor de datos producido y comercializado por ESRI. Tiene por objetivo permitir el uso de sistemas de gestión de base de datos relacionales (RDBMS) para datos espaciales.
- **BMP- Bitmap Image File:** es también conocido como Device Independent Bitmap (DIB). Es un formato de imagen gráfico raster. Se usa para almacenar imágenes digitales de mapa de bits independientemente del dispositivo de visualización.
- **CMD o símbolo del sistema** (en inglés *Command prompt*) es el intérprete de comandos de Windows.
- **DGN- Microstation Design Files:** son archivos de diseño/dibujo de Microstation (producto software de CAD).
- **DXF- Drawing Interchange Format:** es un formato de ficheros de diseño (vectorial) desarrollado por Autodesk para permitir la interoperabilidad entre AutoCAD y otros programas.
- **DWG- Drawing Format:** formato de archivo vectorial utilizado principalmente por el programa AutoCAD (de AutoDesk).
- **ECW- Enhanced Compression Wavelet:** es un formato de archivo para imágenes raster desarrollado por la empresa Earth Resource Mapping. Presenta unos ratios muy altos de compresión mediante el uso de técnicas wavelet. Preserva la georreferenciación de la imagen mediante un archivo de cabecera con extensión .ERS
- **ERDAS:** empresa que ha desarrollado diferentes productos para el procesamiento de imágenes de satélite y dirigido principalmente a procesamiento de datos raster con intención de permitir al usuario preparar, mostrar y mejorar imágenes digitales para su

- uso en un SIG o en un programa de software CAD (diseño asistido por ordenador). Actualmente trabajando con extensiones para ArcGIS.
- **ESRI- Environmental Systems Research Institute:** empresa fundada por Jack Dagermond en 1969. En sus inicios se dedicó a trabajos de consultoría del territorio y actualmente desarrolla y comercializa software para Sistemas de Información Geográfica.
 - **FWTools:** es un conjunto de binarios de código abierto para Windows (win32) y Linux (x86 32 bit) producido por Frank Warmerdam. Incluye: OpenEV, GDAL, MapServer, PROJ4 y OGDl.
 - **GDAL- Geospatial Data Abstraction Library:** es una biblioteca de software para la lectura y escritura de formatos de datos raster publicada por la OSG (Open Source Geospatial). Presenta un único modelo abstracto de datos que llama para todos los formatos soportados. Incluye utilidades en línea de comando para la traducción y el proceso de datos geospaciales.
 - **GeoServer:** es un servidor de código abierto escrito en Java basado en la biblioteca GeoTools. Permite a los usuarios compartir y editar datos geospaciales. Diseñado para la interoperabilidad, publica datos de diversas fuentes de entrada usando estándares abiertos. Incluye un cliente integrado OpenLayers para previsualizar las capas de datos y soporta la publicación de datos geospaciales para Google Earth.
 - **GeoTools:** es una biblioteca SIG de código libre que permite desarrollar soluciones adaptadas a los estándares. Proporciona una implementación de las especificaciones del OGC. Escrito en lenguaje Java siendo su diseño en estructuración modular. Se encuentra actualmente en desarrollo activo.
 - **GIF- Graphics Interchange Format:** es un format de imagen de mapa de bits introducido por CompuServe en 1987.
 - **GML- Geography Markup Language** o *Lenguaje de Marcado Geográfico:* es un sublenguaje XML descrito como una gramática en esquema XML para el modelaje, transporte y almacenamiento de información geográfica (vectorial).
 - **IDRISI:** es un integrado SIG desarrollado por los Laboratorios Clark para la Universidad Clark para el análisis y visualización de información digital geoespacial. Basado en un sistema de malla para imágenes raster.
 - **JDBC- Java Database Connectivity:** es una interfaz de programación de aplicaciones que permite la ejecución de operaciones sobre base de datos desde el lenguaje de programación Java independiente del sistema operativo que se ejecute o de la base de datos a la cual se accede.
 - **JPEG2000- Joint Photographic Experts Group committee in 2000:** es un estándar de compresión de imagen y sistema de codificación. Fue creado por el Comité del Grupo Conjunto de Expertos en Fotografía (Joint Photographic Experts Group) en el año 2000 con intención de sustituir la transformación del original JPEG (del año 1992)
 - **JPEG- Joint Photographic Experts Group** (Grupo Conjunto de Expertos en Fotografía): es un formato de ficheros para imágenes. El nombre hace referencia a un comité de expertos que creó un estándar de compresión y codificación de archivos de imágenes fijas. Utiliza un sistema de compresión especificado por el usuario consiguiendo elevados ratios de compresión.
 - **jQuery** es una biblioteca de JavaScript de software libre y de código abierto, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

- **JSP (JavaServer Pages)** es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.
- **KML- Keyhole Markup Language:** es un lenguaje de marcado basado en XML para representar datos geográficos en tres dimensiones. Fue desarrollado para ser manejado con Keyhole LT, precursor de Google Earth.
- **MapFish:** es un framework flexible y completo de código abierto para crear aplicaciones de mapeo web. Está basado en el framework web Pylons Python. Extiende Pylons con funcionalidades específicas geoespaciales, proporcionando herramientas para crear servicios web que permitan la consulta y edición de objetos espaciales. Proporciona también una completa caja de herramientas para JavaScript.
- **MapServer** es un entorno de desarrollo en código abierto para la creación de aplicaciones SIG en internet. Su funcionamiento básico está configurado en un fichero de texto, que tiene la extensión .MAP.
- **MID:** formato propio de MapInfo para almacenar datos vectoriales.
- **MIF:** formato propio de MapInfo para almacenar datos vectoriales.
- **MrSID- Multi-resolution Seamless Image Database:** es un formato de ficheros con extensión .SID desarrollado y patentado por LizardTech para la codificación de gráficos raster georreferenciados, tales como ortofotos.
- **NTF- National Transfer Format:** es un formato diseñado en 1988 para transferir información espacial (vectorial).
- **ODBC- Open DataBase Connectivity:** es un estándar de acceso a base de datos desarrollado por SQL Access Group en 1992. El objetivo de ODBC es hacer posible el acceso a cualquier dato desde cualquier aplicación sin importar qué sistema de gestión de base de datos (DBMS) se use. Se logra al insertar una capa intermedia (CLI) denominada nivel de Interfaz de Cliente SQL entre la aplicación y el DBMS.
- **OGC- Open Geospatial Consortium:** es una fundación creada en 1994. Su objetivo es la definición de estándares abiertos e interoperables dentro e los Sistemas de Información Geográfica y de la World Wide Web. Persigue acuerdos entre las diferentes empresas que posibiliten la interoperación de sus sistemas de geoprocetamiento y facilitar el intercambio de información geográfica en beneficio de los usuarios. Anteriormente conocido como OpenGIS Consortium.
- **OGR:** es una biblioteca software que proporciona las funcionalidades para soporte vectorial en la biblioteca GDAL.
- **OpenLayers:** es una biblioteca de JavaScript de código abierto para mostrar mapas interactivos en los navegadores web. Ofrece una API para acceder a diferentes fuentes de información cartográfica en la red: WMS, WFS, mapas comerciales, distintos formatos vectoriales, etc.
- **ORACLE ó ORDBMS (Object-Relational Data Base Management System):** es un sistema de gestión de base de datos objeto-relacional desarrollado y comercializado por Oracle Corporation. Cuenta con una extensión para dar soporte espacial denominada Oracle Spatial. La única edición gratuita es Oracle Database Express Edition (XE). Recientemente, adquirió a Sun Microsystem (empresa encargada comercial de MySQL).
- **OSGeo- Open Source Geospatial Foundation:** es la fundación para el código abierto geoespacial creada para animar el uso y desarrollo colaborativo de proyectos liderados por la comunidad sobre software geoespacial.

- **OVF- Open Virtualization Format:** es un estándar abierto para empaquetar y distribuir servicios virtualizados o de forma más general software a ejecutar en máquinas virtuales.
- **PNG- Portable Network Graphics:** es un formato gráfico tipo raster basado en un algoritmo de compresión sin pérdida para mapa de bits.
- **PROJ4:** es una biblioteca para dar soporte a las proyecciones más utilizadas en los proyectos del código abierto (como GRASS, MapServer, PostGIS, biblioteca OGR-GDAL, etc). Realizada en su mayor parte por Gerald Evenden. Inicialmente desarrollada para la plataforma Unix. Actualmente existe además una versión precompilada para Windows, una versión para Debian GNU/Linux y una migración al entorno Delphi, Borland C++ Builder.
- **SHP- ESRI Shapefile:** es un formato de archivo de datos vectoriales desarrollado por la compañía ESRI. Originalmente se creó para la utilización de su producto ArcView GIS, pero actualmente se ha convertido en un formato estándar.
- **SQL/MM- SQL Multimedia:** es una extensión del lenguaje SQL para tratar de forma apropiada archivos grandes, complejos y de transferencia, tales como video, audio y datos espaciales.
- **TAB:** es un formato para almacenar datos vectoriales creado por MapInfo. Contiene código ASCII y es el vínculo entre los demás archivos para almacenar dicha información vectorial.
- **TIFF o TIF- Tagged Image File Format:** es un formato de ficheros para imágenes. Permite almacenar más de una imagen en el mismo archivo.
- **TIGER files- Topologically Integrated Geographic Encoding and Referencing system:** son archivos vectoriales extraídos de la base de datos MAF/TIGER del Negociado del Censo (the Census Bureau's MAF/TIGER). El más reciente es el denominado TIGER/Line[®]Shapefiles.
- **UPEDU (Unified Process for Education):** Es un proceso de desarrollo software especializado para la educación.
- **Virtual Spatial Data (ODBC/OVF):** es una extensión OGR de MapServer. Permite conectar bases de datos que no mantienen explícitamente datos espaciales. La base de datos debe disponer de una columna X y una columna Y. Los datos se acceden a través de una conexión ODBC.
- **WCS- Web Coverage Service:** es una interfaz definida por el OGC. Permite realizar peticiones de cobertura geográfica a través de la web utilizando llamadas independientes de la plataforma. Las coberturas son objetos (o imágenes) en un área geográfica mientras que la interfaz WMS o portales de mapas online como Google Maps devuelven sólo una imagen, que los usuarios no pueden editar o analizar espacialmente. Las peticiones del usuario se envían al servidor web utilizando el protocolo HTTP.
- **WFS- Web Feature Service:** definido por el OGC. Es un servicio estándar que ofrece una interfaz de comunicación que permite interactuar con los mapas servidos por el estándar WMS, como por ejemplo, editar o analizar la imagen que nos ofrece el servicio WMS. Para realizar estas operaciones se utiliza el lenguaje GML.
- **WGS84** es un sistema de coordenadas geográficas mundial que permite localizar cualquier punto de la Tierra (sin necesitar otro de referencia) por medio de tres unidades dadas. WGS84 son las siglas en inglés de World Geodetic System 84 (que significa Sistema Geodésico Mundial 1984). Se trata de un estándar en geodesia, cartografía, y navegación, que data de 1984. Corresponde con el código EPSG: 4326

- **WMF- Windows Metafile:** es un formato de fichero gráfico desarrollado por Microsoft en 1990.
- **WMS- Web Map Service:** servicio definido por el OGC. Produce mapas de datos referenciados espacialmente, de forma dinámica a partir de información geográfica. Define un mapa como una representación de la información geográfica en forma de archivo de imagen digital. Se genera normalmente en un formato de imagen como PNG, GIF o JPEG y opcionalmente como gráficos vectoriales en formato SVG (Scalable Vector Graphics) o WebCGM (Web Computer Graphics Metafile).

Apéndice

C

TOMA DE DECISIÓN

1. Tipos de reglas de decisión.

En el proceso de toma de decisión del emplazamiento óptimo de un UNIGEO existirán distintos intereses de los distintos actores participantes:

- empresa que quiere realizar el emplazamiento (que buscará minimizar los costes de creación y explotación, para optimizar el beneficio económico),
- las autoridades (que deben proporcionar los permisos o determinan la regulación existente a nivel nacional o regional)
- otros organismos (por ejemplo, ecologistas que estarán interesados en un mínimo impacto ambiental, o instituciones internacionales que vigilen el cumplimiento de los tratados sobre zonas de protección de especies en peligro de extinción), etc.

Parece evidente que pueden tomarse las decisiones en función de múltiples criterios y que pueden existir distintas visiones sobre el problema.

Este es el procedimiento habitual en un Sistema de Apoyo a la Toma de Decisión, DSS, donde habrá intereses contrapuestos y será imposible satisfacer todas las preferencias de distintos actores.

1.1. Toma de decisión mediante procedimiento de Evaluación MultiCriterio.

En este ámbito utilizaremos las siguientes definiciones (traducidas de [Eastman y cols. 1993]):

- **Decisión:** Elección entre distintas alternativas. Las alternativas pueden representar distintas vías de actuación, diferentes hipótesis sobre una característica, etc.

- **Criterio:** Un criterio es la base de una decisión, que puede medirse y evaluarse. Es la evidencia sobre la que se basará la decisión.

Existen criterios de dos tipos [Eastman y cols. 1993] [Gómez y Barredo, 2005]:

- las **restricciones** (llamadas criterios duros, hard, crisp o booleanos: que se corresponden con operaciones que devuelven cierto / falso), y
- los **factores** (criterios o restricciones blandas o soft): asociadas a algún tipo de incertidumbre en el criterio de decisión. El empleo de este tipo de factores lleva asociado poder usar técnicas como lógica fuzzy o borrosa [Zadeh, 1965], redes de creencias, redes bayesianas (dentro del campo de la Inteligencia Artificial [Luger 2009]) o algún otro modelo probabilístico (por ejemplo conocer la distribución de probabilidad de un factor, como puede ser la distancia de un emplazamiento a un centro urbano).

Las restricciones impondrán criterios de exclusión (una propiedad se cumple o no se cumple) y generalmente limitarán los análisis (por ejemplo, en el caso de los SIG delimitarán regiones, excluirán zonas de un mapa en función de distancias, profundidades, etc.).

Los factores indicarán mediante sus funciones de pertenencia un posible grado de cumplimiento de un criterio. Para poder combinar estos factores será preciso realizar un proceso de normalización de esos grados de certidumbre (hasta qué punto un criterio se cumple, siempre en el intervalo [0,1], por poner un ejemplo).

En el ámbito de los SIG para el emplazamiento de dispositivos de generación de energía tanto en tierra como en el océano algunos ejemplos son:

- De Restricciones: zonas de naturaleza protegida, hábitats UE, zonas de protección de aves UE (ZEPA), áreas RAMSAR (humedales),...
- De Factores: proximidad a la costa, presencia de corrientes o asentamientos, carreteras, proximidad a líneas de potencia (conectividad al grid), presencia de aeropuertos, oleogaseoductos, iglesias o potencial eólico.

Puesto que generalmente los criterios que se buscan no se pueden cumplir en su totalidad, es necesario suavizar las restricciones y buscar soluciones de compromiso entre criterios contrapuestos. Para buscar estas soluciones de compromiso se suele utilizar la Evaluación MultiCriterio, EMC (MCE en sus siglas en inglés).

La **Evaluación MultiCriterio** [Gómez y Barredo, 2005] consiste en decidir en presencia de múltiples criterios y objetivos contradictorios.

Los algoritmos EMC funcionan permitiendo a los usuarios especificar pesos de preferencias para cada criterio considerado en el proceso de toma de decisión. Se genera una tabla donde se cruzan todos los criterios, para las distintas alternativas, generando una matriz o tabla de puntuación o evaluación, y después utiliza el valor de la tabla para elegir o sugerir.

Ejemplo: Matriz de evaluación. La puntuación del Criterio, X_{ij} , representa el nivel de deseabilidad que ha obtenido cada alternativa i para cada criterio j .

		Alternativas (i)		
		A1	...	Ai
Objetivos	Obj. 1			
	...		Puntuación Criterio X_{ij}	
	Obj. J			

Será necesario, por lo tanto, disponer de algún criterio de normalización para poder hacer comparaciones.

Además, suele existir una **ponderación o peso** de los criterios, que mide su importancia relativa (w_j) en la toma de decisión. Si w_j no es número, sino que indica una relación de orden o una preferencia cualitativa, se suele denominar **jerarquía**.

A partir de estos pesos, se puede construir un vector de prioridades, donde se relacionan los distintos pesos de los criterios para ese punto de vista. Si nuestra aplicación fuese capaz de manejar distintos puntos de vista, en lugar de un vector, tendríamos una **matriz o tabla de prioridades**.

A partir de estas dos tablas, en función de la técnica EMC, se puede construir la **Tabla de Valoración**, en la cual cada alternativa recibe un **valor** en función del criterio y del peso. Si este proceso se repite para distintos puntos de vista, tendremos una matriz de valoración.

Ejemplo: Matriz de Prioridades. Cada criterio j desde un punto de vista v , tendrá un cierto peso w_j .

		Criterios(j)		
		1	...	J
Puntos de vista	1			
	...		Peso (w_j)	
	V			

En la toma de decisión, conforme a Eastman y cols. [Eastman y cols. 1993], se tienen **objetivos** (que deben reflejar la motivación de la toma de decisión) que fijan cómo deben ser o cómo deben organizarse las reglas de decisión.

En este contexto, una **regla de decisión** es el procedimiento por el cual se combinan los (valores de los) criterios para llegar a una decisión. Las reglas describirán procedimientos de combinación de criterios para generar un valor ó índice y qué debe hacerse con ese índice. Estas reglas de decisión pueden ser tan sencillas como aplicar un umbral sobre un criterio, y descartar los valores que lo sobrepasen, o tan complicadas como resolver un problema de optimización multicriterio.

Este proceso puede realizarse definiendo una **función de selección** (también llamada función objetivo) que implique un procedimiento de optimización (maximizar o minimizar un índice), o bien mediante una **función heurística** (que indique un procedimiento para obtener la solución, que puede no ser la óptima, sino una aproximación, pero que realizará muchos menos cálculos).

El proceso concreto de aplicar la regla de decisión se denomina **evaluación**. Si para decidir sobre un objetivo tenemos que evaluar múltiples criterios, estaremos ante una **Evaluación MultiCriterio**. Pero puede ocurrir que para tomar una decisión tengamos que considerar distintos objetivos. En ese caso estamos ante una **Evaluación MultiObjetivo**. En estos casos, lo que se suele hacer es recurrir a una EMC para cada objetivo y establecer algún tipo de jerarquía para evaluar los objetivos.

Entre las técnicas EMC más utilizadas en los entornos SIG tenemos la Combinación Lineal Ponderada y la Media Ponderada Ordenada. Vamos a describirlas a continuación.

1.1.1. Combinación lineal ponderada, WLC

La Combinación Lineal Ponderada (WLC por sus siglas en inglés Weighted Linear Combination) busca una suma ponderada de todos los factores que hay en la decisión y calcula para cada punto cuál es el valor del criterio, en nuestro caso el coste [Hansen, 2005; Ramírez 2007].

Esta técnica, WLC, calcula el nivel de adecuación (Suitability, en inglés) que proporciona cada factor, S^k , mediante una fórmula: $S^k = \sum w_i x_i^k$ que se aplica a cada celda de un mapa y combinando varios mapas: donde w_i es el peso, x_i^k es el valor del criterio i en el pixel k -ésimo. Los pesos w_1, \dots, w_n reflejan la importancia relativa de cada criterio. Para que la técnica funcione adecuadamente la suma de los pesos tiene que ser 1.

En problemas reales estos pesos contendrán un alto grado de incertidumbre (si el ruido que generará un aero-generador debe estar entre 40 y 60dB, ¿debemos descartar valores de 61 dB?) Como se ha comentado antes, una técnica habitual para combinar distintos criterios son las reglas fuzzy [Hansen, 2005].

La ventaja de la aproximación, WLC, es la simplicidad de la regla, pero requiere estandarizar los criterios al intervalo $[0, 1]$. La forma más sencilla de realizar este paso es un escalado lineal entre los valores máximos y mínimos de cada valor: $v_i = (R_i - R_{\min}) / (R_{\max} - R_{\min})$, siendo R_i el valor inicial, R_{\min} el valor mínimo que se puede obtener y R_{\max} el máximo.

1.1.2. Media ponderada ordenada (OWA: Ordered Weighted Average)

La media ponderada ordenada es una generalización de la combinación lineal ponderada, WLC. Permite graduar los niveles de riesgo/compromiso en la toma de decisión de forma continua. Al igual que WLC requiere estandarizar los valores de los atributos, criterios o factores.

Utiliza un conjunto de pesos de atributos, W , para ponderar cada factor. Son los mismos pesos que define WCL, han de estar en el intervalo $[0, 1]$ y su suma debe de ser 1.

Adicionalmente, se introduce un segundo conjunto de pesos, V , denominados pesos de orden, por la forma en que trabaja el algoritmo.

Una vez definidos ambos vectores de pesos, el algoritmo trabaja de la siguiente forma:

1. Determinar, para cada localización, el valor ponderado de cada atributo (para localización i , $w_j a_{ij}$).
2. Ordenar los atributos, en cada localización, en orden decreciente.
3. Obtener la evaluación de cada localización mediante la combinación lineal de los valores de los atributos ordenados y los pesos de orden.

Según los valores del vector de pesos de orden, nos movemos en distintos puntos del espectro riesgo/compromiso, representado en la *figura 179*.

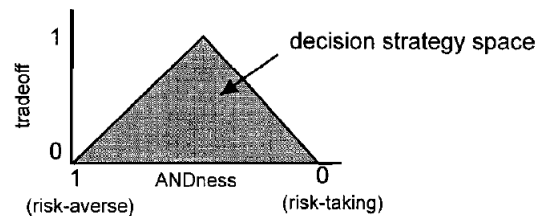


Figura 179: OWA: Espacio de estrategias de decisión [Jiang e Eastman 2000]

A título ilustrativo, suponiendo que manejamos cuatro atributos en la toma de decisión, tenemos los siguientes casos representativos:

- $\underline{V}=(0, 0, 0, 1)$: Mínimo riesgo, compromiso nulo. Corresponde a un AND de los criterios.
- $\underline{V}=(0.25, 0.25, 0.25, 0.25)$: Riesgo intermedio, compromiso máximo, dado por el vector de pesos \underline{W} . Corresponde al caso de la Media Lineal Ponderada (WLC)
- $\underline{V}=(1, 0, 0, 0)$: Máximo riesgo, compromiso nulo. Corresponde a un OR de los criterios.

1.1.3. Otros métodos EMC

- **Análisis de Puntos Ideales** (API ó IPA por sus siglas en inglés): define cuál sería el punto ideal -por ejemplo, menor coste- y busca aquellos puntos que minimicen la distancia de los distintos criterios a los criterios del punto ideal.
- **Análisis de concordancias / discordancias (CDA)**: Se basa en anotar puntos para las distintas alternativas por cada criterio que concuerda y penalizar aquellos en los que no concuerda. Ambos pesos, normalizados. Después ambos índices -de concordancia / discordancia- se estudian conjuntamente, beneficiando aquellos puntos que tienen más concordancias y menos discordancias a la vez.
- **Optimización Jerárquica (HO)**: consiste en aplicar por casos cada criterio de optimización, en función de los pesos: aquellas de mayor peso se optimizan primero. Después, paso a paso, se aplican nuevos algoritmos de optimización sobre los resultados de la etapa anterior. Después de cada fase de optimización, se eliminan los puntos que no cumplan unos criterios mínimos -establecidos por el usuario-.

En principio, este último método tiene la ventaja de que no necesita una normalización de todos los índices, ya que aquellos puntos que no superan un mínimo en cada iteración, son eliminados, si no superan un umbral de tolerancia en la etapa: $K_m \cdot S_{k-1}[i]^0$

Paso 1: $\max_{i \in S} S_1[i]$

Paso 2: $\max_{i \in S} S_2[i]$, para los $S_1[i] \geq K \cdot S_1[i]^0$

...

Paso m: $\max_{i \in S} S_k[i]$, para los $S_{k-1}[i] \geq K_m \cdot S_{k-1}[i]^0$

- **Método de Jerarquías Analíticas** (MJA ó Analytic Hierarchical Process, AHP), una técnica ideada por Saaty [Saaty, 1980; Ramírez, 2007] e incorporada en aplicaciones GIS como IDRISI [Eastman y cols. 1993].

Este procedimiento evita el proceso de selección de pesos normalizados que deban sumar 1. El procedimiento ideado por Saaty construye una matriz cuadrada que compara dos a dos todos los criterios. Esta matriz es recíproca, entendiéndose que si la preferencia para el par de criterios m_{ij} es c , para la pareja m_{ji} la preferencia será $1/c$. Obviamente, los valores de la diagonal principal serán $m_{ii} = 1$. Por este motivo, para hacer los cálculos sólo se necesita la matriz triangular inferior. Los valores de preferencia que pueden tomar deben estar en el intervalo $[1..9]$, dando lugar a dos escalas (menos y más importante) que pueden tomar valores: {igual=1, moderada=[3,5], fuerte [7]y extrema [9]}.

Los pesos de los criterios se obtienen como los auto valores principales de esa matriz cuadrada. Se puede evaluar además la consistencia de la matriz generada para evitar que se parezca a pesos generados de forma aleatoria.

Una vez generados los pesos, se puede aplicar la técnica WLC, por ejemplo.

Este método es el empleado en IDRISI en su módulo EMC y es el recomendado por la EPA (Environmental Protection Agency) en EEUU para la comparación de alternativas.

El problema asociado a este tipo de técnicas es que puede llevar asociado un proceso de optimización en el caso de necesitar evaluación multicriterio y multiatributo, como explicaremos a continuación en el entorno de un SIG.

1.2. Toma de Decisión mediante Evaluación MultiCriterio y su combinación con Sistemas de Información Geográfica.

En el caso de los SIG y la toma de decisión, la aplicación de reglas booleanas (restricciones o reglas hard) suele estar ligada a operaciones sobre mapas relativamente sencillas (superposición (overlay) de capas que cumplan un criterio –sobrepasar o no sobrepasar un umbral-), que pueden entenderse como operaciones básicas de teoría de conjuntos: intersección de áreas, unión, diferencia, etc.

Los otros tipos de reglas, los factores o reglas soft, necesitarán operaciones que suelen estar en los SIG, como operadores de lógica difusa [Zadeh, 1965][Jiang e Eastman, 2000], pero dada la

simplicidad de estos operadores es muy probable que haya que generar algoritmos propios (por ejemplo algoritmos de optimización como OWFLOW [OWFLOW]) que realicen estas operaciones ligadas a valores inciertos.

Las distintas propuestas EMC se diseñaron inicialmente en las áreas de planificación, análisis de decisiones y la investigación operativa para la toma de decisiones y resolución de conflictos. Su adaptación al ámbito de los SIG se propuso en distintos trabajos (inicialmente [Carver, 1991]) pero después existen múltiples ejemplos [Ascough, 2002; Hansen, 2005; Malczewski, 2006] sobre todo en el ámbito de planificación de uso de territorios/terrenos [vanderMerwe, 1997] [Nogués, 2000] o localización de complejos (plantas de tratamiento de residuos nucleares [Carver y Openshaw, 1996], por ejemplo).

Según nuestro conocimiento [Ascough y cols. 2002] [Malczewski, 2006] [Gómez y Barredo, 2006], pocos productos SIG tienen componentes que realicen directamente algún algoritmo tipo EMC:

- IDRISI [Eastman y cols., 1993], herramienta GIS que permite el análisis espacial en distintos tipos de tareas, desarrollado en los Laboratorios de la Universidad Clark, de EEUU, por el equipo de R. Eastman, fue de los primeros paquetes en incorporar un módulo específico ECM en 1993 con la técnica WLC, al que se añadió después el Método de Jerarquías Analíticas [Saaty, 1980; Ramírez, 2007], y después reglas heurísticas para realizar decisión Multiobjetivo participativo, MOLA [IDRISI Taiga].
- Otros SIG tienen capacidades EMC mediante rutinas “plug in”, como en el caso de ARC-INFO [Carver, 1991], IDRISI [vanderMerwe, 1997] o el uso de OWA en ARC Veiw [Malczewski, 2006], que permiten la construcción del algún tipo de EMC. Esta es la vía que se suele seguir en la mayor parte de los desarrollos no ligados a la extensión de productos existentes.

Gómez y Barredo [Gómez y Barredo, 2005] proponen un método general de integración de SIG y EMC para un único criterio como se puede ver en la *Figura 180*.

En el caso de la Combinación Lineal Ponderada, suponiendo que cada criterio genera una capa ráster, la aplicación de esta técnica generaría una nueva capa. Por ejemplo, si tenemos dos criterios A y B:

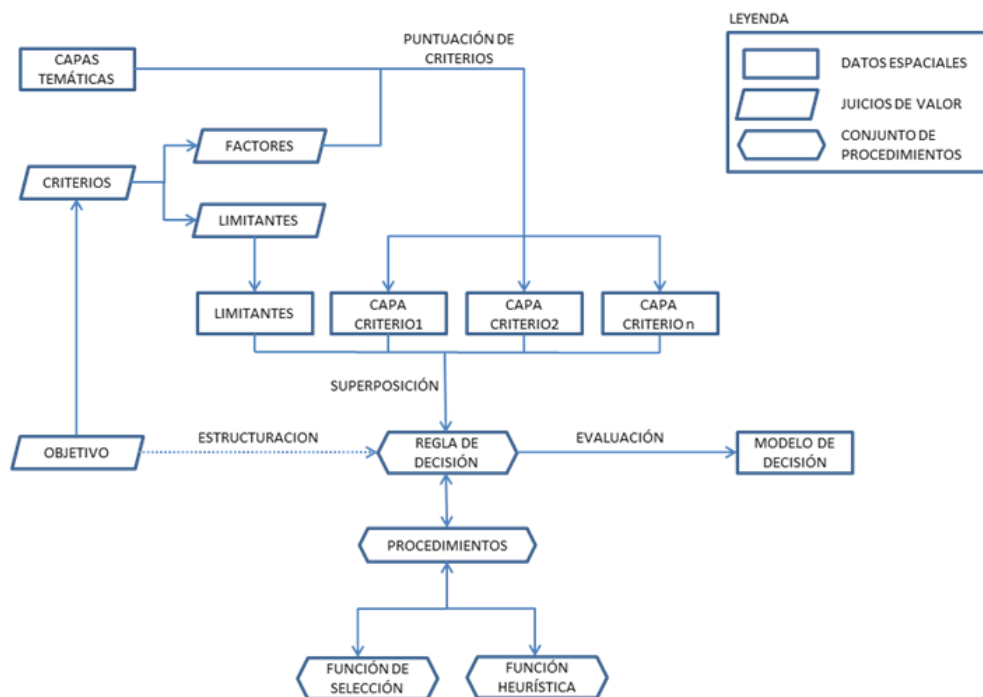


Figura 180: Sistema de integración entre SIG y EMC para un único objetivo [Gómez y Barredo, 2005]

Combinación lineal ponderada con 2 criterios (capas):

Capa A	<table border="1"> <tr><td>2</td><td>4</td><td>5</td></tr> <tr><td>3</td><td>2</td><td>3</td></tr> <tr><td>4</td><td>1</td><td>4</td></tr> </table>	2	4	5	3	2	3	4	1	4	Peso: 0,6
2	4	5									
3	2	3									
4	1	4									
Capa B	<table border="1"> <tr><td>1</td><td>3</td><td>4</td></tr> <tr><td>3</td><td>2</td><td>3</td></tr> <tr><td>5</td><td>3</td><td>2</td></tr> </table>	1	3	4	3	2	3	5	3	2	Peso: 0,4
1	3	4									
3	2	3									
5	3	2									
Capa final (WLC)	<table border="1"> <tr><td>1.6</td><td>3.6</td><td>4.6</td></tr> <tr><td>3.0</td><td>2.0</td><td>3.0</td></tr> <tr><td>4.4</td><td>1.8</td><td>3.2</td></tr> </table>	1.6	3.6	4.6	3.0	2.0	3.0	4.4	1.8	3.2	
1.6	3.6	4.6									
3.0	2.0	3.0									
4.4	1.8	3.2									

Si en lugar de tener un objetivo, tuviésemos dos o más, habría que seguir un proceso similar para cada uno y después combinarlos, como se puede ver en la Figura siguiente, obtenida de [Gómez y Barredo, 2005]. En este caso será necesaria una nueva regla de decisión, denominada regla

final o RF en la figura 181, que combine los valores obtenidos para los criterios en distintas capas. Aquí pueden de nuevo aplicarse distintas técnicas de decisión multicriterio: extensión jerárquica, solución priorizada o solución compromiso.

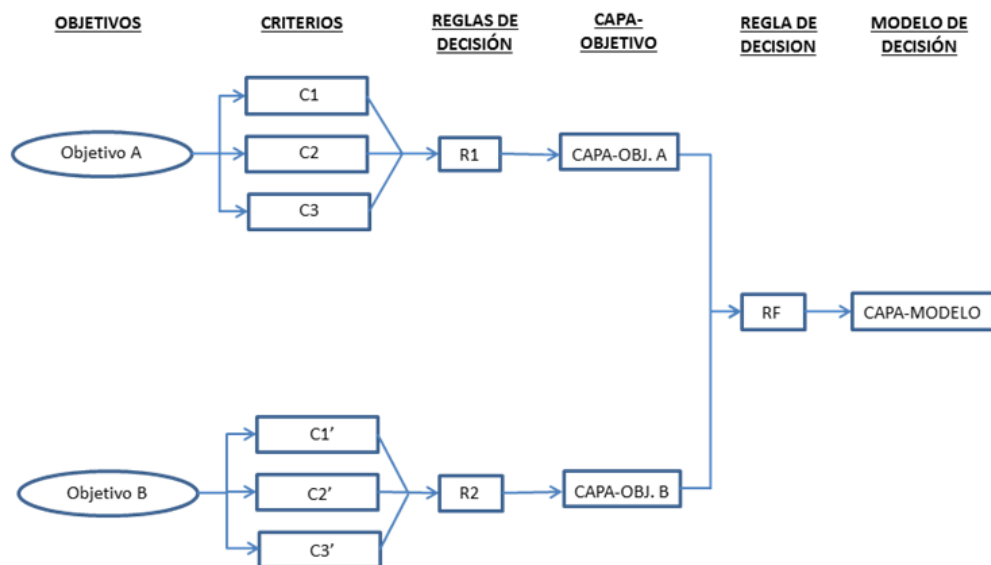


Figura 181: Esquema general del proceso de integración de las diversas fases que componen el procedimiento multicriterio/multiobjetivo en un SIG ráster

- La extensión jerárquica, donde se pueden asignar pesos a los distintos criterios no contradictorios en función de su jerarquía o preferencia y a continuación puede realizarse una suma ponderada.
- La solución priorizada suelen utilizarse cuando se conoce una jerarquía entre objetivos y éstos pueden ser contradictorios. En este caso se van generando capas por objetivos, según la jerarquía de mejor a peor, y cada nuevo objetivo sólo se puede aplicar sobre los puntos que hayan pasado el filtro del objetivo anterior. Esta decisión puede resultar drástica en algunos casos.
- La solución compromiso considera objetivos conflictivos de jerarquía desconocida. En estos casos se puede recurrir a un proceso de optimización como programación por metas, al Análisis del Punto Ideal o bien una solución heurística, como la que se utiliza en el paquete MOLA de IDRISI [IDRISI Taiga][Eastman y cols.] En los paquetes SIG comerciales, las dos últimas son las opciones más utilizadas por su facilidad de integración en SIG y por su menor coste computacional.

Aunque existen primitivas en algunos paquetes comerciales que facilitan la utilización de la aproximación EMC [IDRISI TAIGA, ARC-GIS, ARC-INFO, GRASS], en general el procedimiento para aplicar la técnica EMC ha sido la siguiente [Santos, 1997, vanderMerwe1997]:

- Decidir los objetivos y los criterios en función de lo que se quiere evaluar (necesitará esta fase la participación activa de expertos y usuarios finales), dando lugar a múltiples escenarios.
- Obtener mapas para los n criterios (por ejemplo, mediante overlays de ARC-INFO).
- Si es necesario, pasar a formato ráster las imágenes vectoriales.
- Normalizar las medidas (por ejemplo correlacionar los valores según la idoneidad).
- Fijar pesos para los criterios de cada objetivo (por ejemplo, la primitiva WEIGHT en IDRISI) y comprobar que sean consistentes.
- Realizar el procedimiento EMC elegido, creando tantas capas de idoneidad por objetivo (por ejemplo, esto es automático en IDRISI que tiene la primitiva EMC).
- Realizar la Evaluación MultiObjetivo si hay varios objetivos, por ejemplo mediante la aplicación MOLA de IDRISI, ponderando los objetivos si es posible.
- Crear mapas para los objetivos eligiendo los óptimos.

En principio las dos técnicas más son la Combinación Lineal Ponderada (esta es la más utilizada, siendo la opción por defecto en las últimas versiones del paquete IDRISI), y la Media Ponderada Ordenada. La primera técnica estaría indicada si se consiguen fijar de forma clara los pesos asociados a cada criterio. La segunda técnica podría usarse si el usuario final, quien debe tomar la decisión, quiere valorar distintas políticas de riesgo, teniendo en cuenta que dada decisión dará lugar a distintos resultados.

Apéndice

D

SCRIPTS OPERACIONALES DE LOS CASOS PRÁCTICOS

1. Script operacional de caso práctico de estudio con valores nulos

```
@echo off

time < time_reloj.txt >> Script_Grass\Hora_ini.txt

REM Configuración de la región: cogemos de referencia la región
del mapa DistanciasCosta.tif

g.region n=4166368 s=3854368 e=620578 w=99478 nsres=300 ewres=300

REM Aplicación de REGLAS FÍSICAS

REM Criterio: Zonas con distancia a la costa superior a 8 Km.
Primero se calcula el mapa de distancias a la costa

v.in.ogr -r "dsn=C:/Users/GSI/Desktop/bd GMV/BBDD
Oceanlider_actualizada/P/02_Capas/01_IB/01_CB/Lin_Costa_P.shp"
output=Lin_Costa min_area=0.0001 snap=-1

v.to.rast input=Lin_Costa layer=1 type=line output=Lin_Costa
use=val value=1 rows=4096

r.grow.distance input=Lin_Costa distance=Dist metric=euclidean

v.in.ogr -r "dsn=C:/Users/GSI/Desktop/bd GMV/BBDD
Oceanlider_actualizada/P/02_Capas/01_IB/03_LA/CCAA_P.shp"
output=CCAA_And min_area=0.0001 snap=-1
```

```
v.to.rast input=CCAA_And layer=1 type=line,area output=CCAA_And
use=val value=0 rows=4096

r.null map=CCAA_And null=1

REM r.null map=CCAA_And null=1

r.mapcalc "DistCosta = Dist * CCAA_And"

REM r.null map=DistCosta setnull=0 (se puede poner a null para
obviar el área de Andalucía)

r.reclass input=DistCosta output=RF_DistCosta
rules="E:\Ampliacion_PFC\Ejemplo_aplicacion_criterios_totales_y_p
arciales\reclass_dist_costa_completo.txt"

REM Criterio: Zonas que cumplan una altura de ola significativa
no superior a 14 m

r.in.gdal input=E:\Ampliacion_PFC\13288675008911.tif
output=UnionHsMedia_And

r.reclass input=UnionHsMedia_And output=RF_Hs
rules=E:\Ampliacion_PFC\reclass_Hs.txt

REM Operación: AND de las capas RF_DistCosta.tif y Capa RF_Hs.tif

r.mapcalc "RF_DCyHs = RF_DistCosta * RF_Hs"

REM Aplicación REGLAS OPERACIONALES

REM Criterio: Zonas excluyentes a un buffer de 2 millas náuticas
(3704 metros) a las rutas

v.in.ogr "dsn=C:\Users\GSI\Desktop\bd
GMV\02_CAPAS\02_INFORMACION_TEMATICA\USOS_MARINOS_Y_LITORALES\RUT
AS_TRANSPORTE_MARITIMO\Rutas_trans_maritimo.shp"
output=Rutas_Maritimas min_area=0.0001 type=line snap=-1

v.buffer input=Rutas_Maritimas output=Rutas_maritimas_Buffer
type=line,area layer=1 distance=3704 angle=0 scale=1.0
tolerance=0.01

v.to.rast input=Rutas_maritimas_Buffer layer=1 type=line,area
output=RO_CT_RutasNav use=val value=0 rows=4096

r.null map=RO_CT_RutasNav null=1

REM Operación: AND de RF_DCyHs.tif y RO_CT_RutasNav.tif
```

```
r.mapcalc "RF_RO_CT = RF_DCyHs * RO_CT_RutasNav"
```

REM A continuación se va aplicar MLP de las reglas parciales operacionales

REM Criterio: Aplicación de operación difusa sobre la potencia del viento

```
r.in.gdal "input=C:\Users\GSI\Desktop\bd GMV\BBDD
Oceanlider_actualizada\P\02_Capas\02_IT\RE\VI\Viento_Anual.img"
output=Viento_Anual
```

```
r.mapcalc "RO_CP_VelViento = if(Viento_Anual < 0.0, null(),
if((Viento_Anual <= 6.0 || Viento_Anual >= 15.0), 0.0,
if((Viento_Anual > 6.0 && Viento_Anual < 8.5), eval((Viento_Anual
- 6.0)/(8.5 - 6.0)), if((Viento_Anual > 14.0 && Viento_Anual <
15.0), eval((15.0 - Viento_Anual)/(15.0 - 14.0)), 1.0))))"
```

REM Criterio: Aplicar fuzzy sobre altura de la ola media

```
r.mapcalc "RO_CP_Hs = if(UnionHsMedia_And < 0.0, null(),
if((UnionHsMedia_And == 0), 1.0, if((UnionHsMedia_And > 0 &&
UnionHsMedia_And < 14.0), eval((14.0 - UnionHsMedia_And)/(14.0 -
0)), 0.0))))"
```

REM Criterio: Aplicación de operación difusa sobre las distancias a las rutas marítimas. Se calcula primero mapa de distancias a las rutas marítimas

```
v.to.rast input=Rutas_Maritimas layer=1 type=line
output=Rutas_Maritimas use=val value=1 rows=4096
```

```
r.grow.distance input=Rutas_Maritimas
distance=DistanciasRutasMaritimas metric=euclidean
```

```
r.mapcalc "RO_CP_DistRutasMarit = if(DistanciasRutasMaritimas <
0.0, null(), if((DistanciasRutasMaritimas <= 3704.0 ||
DistanciasRutasMaritimas >= 37040.0), 0.0,
if((DistanciasRutasMaritimas > 3704.0 && DistanciasRutasMaritimas
< 5556.0), eval((DistanciasRutasMaritimas - 3704.0)/(5556.0 -
3704.0)), if((DistanciasRutasMaritimas > 9260.0 &&
DistanciasRutasMaritimas < 37040.0), eval((37040.0 -
DistanciasRutasMaritimas)/(37040.0 - 9260.0)), 1.0))))"
```

REM Operación: MLP con pesos W1(RO_CP_VelViento.tif)=0.6,
 W2(RO_CP_Hs.tif)=0.3 y W3(RO_CP_DistRutasMarit.tif)=0.1
 ***** SE TOMAN LOS PESOS DE ORDEN COMO 1/3

```
r.mapcalc "RO_CP_MLP = if(isnull(RO_CP_VelViento) ||
isnull(RO_CP_Hs) || isnull(RO_CP_DistRutasMarit),
null(),if(RO_CP_VelViento >= RO_CP_Hs && RO_CP_Hs >=
RO_CP_DistRutasMarit, eval(((RO_CP_VelViento*0.6*(1.0/3.0)) +
(RO_CP_Hs*0.3*(1.0/3.0)) + (RO_CP_DistRutasMarit*0.1*(1.0/3.0)))
/ (0.6*(1.0/3.0)+0.3*(1.0/3.0)+0.1*(1.0/3.0))),
if(RO_CP_VelViento >= RO_CP_DistRutasMarit &&
RO_CP_DistRutasMarit > RO_CP_Hs,
eval(((RO_CP_VelViento*0.6*(1.0/3.0)) +
(RO_CP_DistRutasMarit*0.1*(1.0/3.0)) + (RO_CP_Hs*0.3*(1.0/3.0)))
/ (0.6*(1.0/3.0)+0.1*(1.0/3.0)+0.3*(1.0/3.0))), if(RO_CP_Hs >
RO_CP_VelViento && RO_CP_VelViento >= RO_CP_DistRutasMarit,
eval(((RO_CP_Hs*0.3*(1.0/3.0)) + (RO_CP_VelViento*0.6*(1.0/3.0))
+ (RO_CP_DistRutasMarit*0.1*(1.0/3.0))) /
(0.3*(1.0/3.0)+0.6*(1.0/3.0)+0.1*(1.0/3.0))), if(RO_CP_Hs >=
RO_CP_DistRutasMarit && RO_CP_DistRutasMarit > RO_CP_VelViento,
eval(((RO_CP_Hs*0.3*(1.0/3.0)) +
(RO_CP_DistRutasMarit*0.1*(1.0/3.0)) +
(RO_CP_VelViento*0.6*(1.0/3.0))) /
(0.3*(1.0/3.0)+0.1*(1.0/3.0)+0.6*(1.0/3.0))),if(RO_CP_DistRutasMa
rit > RO_CP_VelViento && RO_CP_VelViento >= RO_CP_Hs,
eval(((RO_CP_DistRutasMarit*0.1*(1.0/3.0)) +
(RO_CP_VelViento*0.6*(1.0/3.0)) + (RO_CP_Hs*0.3*(1.0/3.0))) /
(0.1*(1.0/3.0)+0.6*(1.0/3.0)+0.3*(1.0/3.0))),if(RO_CP_DistRutasMa
rit > RO_CP_Hs && RO_CP_Hs > RO_CP_VelViento,
eval(((RO_CP_DistRutasMarit*0.1*(1.0/3.0)) +
(RO_CP_Hs*0.3*(1.0/3.0)) + (RO_CP_VelViento*0.6*(1.0/3.0))) /
(0.1*(1.0/3.0)+0.3*(1.0/3.0)+0.6*(1.0/3.0))), null() ))))"
```

REM Operación:AND_fuzzy (mínimo) de RO_CP_MLP y RF_RO_CT

```
r.mapcalc "RF_RO_salida = if(isnull(RO_CP_MLP) ||
isnull(RF_RO_CT), null(), if(RO_CP_MLP <= RF_RO_CT, RO_CP_MLP,
RF_RO_CT))"
```

REM Aplicación de RESTRICCIONES DE IMPACTO MEDIO AMBIENTAL

REM Operación: Zonas excluyentes de las zonas de Almadrabas

```
v.in.ogr "dsn=C:\Users\GSI\Desktop\bd GMV\BBDD
Oceanlider_actualizada\P\02_Capas\02_IT\UMyL\UP\ALMADRABAS_p.shp"
output=Almadrabas min_area=0.0001 snap=-1
```


Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

```
v.to.rast input=Almadrabas layer=1 type=point,line,area  
output=REIA_Almadrabas use=val value=0 rows=4096
```

```
r.null map=REIA_Almadrabas null=1
```

REM Operación: Zonas excluyentes a las zonas MABs --- zonas de reserva de la biosfera

```
v.in.ogr "dsn=C:\Users\GSI\Desktop\bd GMV\BBDD  
Oceanlider_actualizada\P\02_Capas\02_IT\CyMA\EP\IN\MABs_P.shp"  
output=MABs  
min_area=0.0001 snap=-1
```

```
v.to.rast input=MABs layer=1 type=point,line,area  
output=REIA_MABs use=val value=0 rows=4096
```

```
r.null map=REIA_MABs null=1
```

REM Operación: MLP de las reglas medio ambientales, con pesos: W1 (RIA_Almadrabas) = 0.7 y W2 (RIA_MABs) = 0.3 CON PESOS DE ORDEN DE 0.5

```
r.mapcalc "REIA_MLP = if(isnull(REIA_Almadrabas) ||  
isnull(REIA_MABs), null(), if(REIA_Almadrabas >= REIA_MABs,  
eval(((REIA_Almadrabas*0.7*0.5) + (REIA_MABs*0.3*0.5)) /  
(0.7*0.5+0.3*0.5)), eval(((REIA_Almadrabas*0.7*0.5) +  
(REIA_MABs*0.3*0.5)) / (0.7*0.5+0.3*0.5))))"
```

REM Operación: : AND_fuzzy (mínimo) de reglas de impacto medio ambiental y reglas físicas (RF_DCyHs y REIA_MLP)

```
r.mapcalc "RF_REIA_salida = if(isnull(RF_DCyHs) ||  
isnull(REIA_MLP), null(), if(RF_DCyHs <= REIA_MLP, RF_DCyHs,  
REIA_MLP))"
```

REM Aplicación de REGLA DE DECISIÓN FINAL (OWA)

REM Criterio: Cálculo de la regla de decisión final mediante una Media Ponderada Ordenada (OWA) de las restricciones físicas y operacionales junto con las reglas físicas y reglas de impacto medio ambiental (RF_RO_salida y RF_REIA_salida) CON W1(RF_RO_salida)= 0.5 Y W2 (RF_REIA_salida) = 0.5

REM Cálculo de la CapaResultado con Z1=1 Y Z2=0

```
r.mapcalc "CapaResultado_1_0 = if(isnull(RF_RO_salida) ||  
isnull(RF_REIA_salida), null(),if(RF_RO_salida >= RF_REIA_salida,
```

```
eval(((RF_RO_salida*0.5*1) + (RF_REIA_salida*0.5*0)) /
(0.5*1+0.5*0)), eval(((RF_RO_salida*0.5*0) +
(RF_REIA_salida*0.5*1)) / (0.5*0+0.5*1))))"
```

REM Cálculo de la CapaResultado con Z1=0 y Z2=1

```
r.mapcalc "CapaResultado_0_1 = if(isnull(RF_RO_salida) ||
isnull(RF_REIA_salida), null(),if(RF_RO_salida >= RF_REIA_salida,
eval(((RF_RO_salida*0.5*0) + (RF_REIA_salida*0.5*1)) /
(0.5*0+0.5*1)), eval(((RF_RO_salida*0.5*1) +
(RF_REIA_salida*0.5*0)) / (0.5*1+0.5*0))))"
```

REM Cálculo de la CapaResultado con Z1=0.5 y Z2=0.5

```
r.mapcalc "CapaResultado_0.5_0.5 = if(isnull(RF_RO_salida) ||
isnull(RF_REIA_salida), null(),if(RF_RO_salida >= RF_REIA_salida,
eval(((RF_RO_salida*0.5*0.5) + (RF_REIA_salida*0.5*0.5)) /
(0.5*0.5+0.5*0.5)), eval(((RF_RO_salida*0.5*0.5) +
(RF_REIA_salida*0.5*0.5)) / (0.5*0.5+0.5*0.5))))"
```

REM Cálculo de la CapaResultado con Z1=0.25 y Z2=0.75

```
r.mapcalc "CapaResultado_0.25_0.75 = if(isnull(RF_RO_salida) ||
isnull(RF_REIA_salida), null(),if(RF_RO_salida >= RF_REIA_salida,
eval(((RF_RO_salida*0.5*0.25) + (RF_REIA_salida*0.5*0.75)) /
(0.5*0.25+0.5*0.75)), eval(((RF_RO_salida*0.5*0.75) +
(RF_REIA_salida*0.5*0.25)) / (0.5*0.75+0.5*0.25))))"
```

REM Cálculo de la CapaResultado con Z1=0.75 y Z2=0.25

```
r.mapcalc "CapaResultado_0.75_0.25 = if(isnull(RF_RO_salida) ||
isnull(RF_REIA_salida), null(), if(RF_RO_salida >=
RF_REIA_salida, eval(((RF_RO_salida*0.5*0.75) +
(RF_REIA_salida*0.5*0.25)) / (0.5*0.75+0.5*0.25)),
eval(((RF_RO_salida*0.5*0.25) + (RF_REIA_salida*0.5*0.75)) /
(0.5*0.25+0.5*0.75))))"
```

REM Cambio de paleta de colores en las Capas Resultado

```
r.colors map=CapaResultado_0_1
rules="E:\Ampliacion_PFC\Ejemplo_aplicacion_criterios_totales_y_p
arciales\colors_verde.txt"
```

```
r.colors map=CapaResultado_1_0
rules="E:\Ampliacion_PFC\Ejemplo_aplicacion_criterios_totales_y_p
arciales\colors_verde.txt"
```

```
r.colors map=CapaResultado_0.25_0.75
rules="E:\Ampliacion_PFC\Ejemplo_aplicacion_criterios_totales_y_p
arciales\colors_verde.txt"
```

```
r.colors map=CapaResultado_0.75_0.25
rules="E:\Ampliacion_PFC\Ejemplo_aplicacion_criterios_totales_y_p
arciales\colors_verde.txt"
```

```
r.colors map=CapaResultado_0.5_0.5
rules="E:\Ampliacion_PFC\Ejemplo_aplicacion_criterios_totales_y_p
arciales\colors_verde.txt"
```

```
time < time_reloj.txt >> Script_Grass\Hora_fin.txt
```

2. Script operacional de caso práctico de estudio sin valores nulos

```
@echo off
```

```
time < time_reloj.txt >> Script_Grass\Hora_ini.txt
```

```
REM Configuración de la región: cogemos de referencia la región
del mapa DistanciasCosta.tif
```

```
g.region n=4166368 s=3854368 e=620578 w=99478 nsres=300 ewres=300
```

```
REM Aplicación de REGLAS FÍSICAS
```

```
REM Criterio: Zonas con distancia a la costa superior a 8 Km.
Primero se calcula el mapa de distancias a la costa
```

```
v.in.ogr -r "dsn=C:/Users/GSI/Desktop/bd GMV/BBDD
Oceanlider_actualizada/P/02_Capas/01_IB/01_CB/Lin_Costa_P.shp"
output=Lin_Costa min_area=0.0001 snap=-1
```

```
v.to.rast input=Lin_Costa layer=1 type=line output=Lin_Costa
use=val value=1 rows=4096
```

```
r.grow.distance input=Lin_Costa distance=Dist metric=euclidean
```

```
v.in.ogr -r "dsn=C:/Users/GSI/Desktop/bd GMV/BBDD
Oceanlider_actualizada/P/02_Capas/01_IB/03_LA/CCAA_P.shp"
output=CCAA_And min_area=0.0001 snap=-1
```

```
v.to.rast input=CCAA_And layer=1 type=line,area output=CCAA_And
use=val value=0 rows=4096
```

```
r.null map=CCAA_And null=1

REM r.null map=CCAA_And null=1

r.mapcalc "DistCosta = Dist * CCAA_And"

REM r.null map=DistCosta setnull=0 (se puede poner a null para
obviar el área de Andalucía)

r.reclass input=DistCosta output=RF_DistCosta
rules="E:\Ampliacion_PFC\Ejemplo_aplicacion_criterios_totales_y_p
arciales\Resultados_finales_verde_V3\reclass_dist_costa.txt"

REM Criterio: Zonas que cumplan una altura de ola significativa
no superior a 14 m. El mapa de entrada procede de GvSIG con
valores nulos = -99999

r.in.gdal input=E:\Ampliacion_PFC\13288675008911.tif
output=UnionHsMedia_And

r.reclass input=UnionHsMedia_And output=RF_Hs
rules="E:\Ampliacion_PFC\Ejemplo_aplicacion_criterios_totales_y_p
arciales\Resultados_finales_verde_V3\reclass_Hs.txt"

REM Operación: AND de las capas RF_DistCosta.tif y Capa RF_Hs.tif

r.mapcalc "RF_DCyHs = RF_DistCosta * RF_Hs"

REM Aplicación REGLAS OPERACIONALES

REM Criterio: Zonas excluyentes a un buffer de 2 millas náuticas
(3704 metros) a las rutas

v.in.ogr "dsn=C:\Users\GSI\Desktop\bd
GMV\02_CAPAS\02_INFORMACION_TEMATICA\USOS_MARINOS_Y_LITORALES\RUT
AS_TRANSPORTE_MARITIMO\Rutas_trans_maritimo.shp"
output=Rutas_Maritimas min_area=0.0001 type=line snap=-1

v.buffer input=Rutas_Maritimas output=Rutas_maritimas_Buffer
type=line,area layer=1 distance=3704 angle=0 scale=1.0
tolerance=0.01

v.to.rast input=Rutas_maritimas_Buffer layer=1 type=line,area
output=RO_CT_RutasNav use=val value=0 rows=4096

r.null map=RO_CT_RutasNav null=1

REM Operación: AND de RF_DCyHs.tif y RO_CT_RutasNav.tif
```

Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

```
r.mapcalc "RF_RO_CT = RF_DCyHs * RO_CT_RutasNav"
```

REM A continuación se va aplicar MLP de las reglas parciales operacionales

REM Criterio: Aplicación de operación difusa sobre la potencia del viento. En la capa resultante del fuzzy se pone valor 0.0 a los valores nulos.

```
r.in.gdal "input=C:\Users\GSI\Desktop\bd GMV\BDD
Oceanlider_actualizada\P\02_Capas\02_IT\RE\VI\Viento_Anual.img"
output=Viento_Anual
```

```
r.mapcalc "RO_CP_VelViento = if(Viento_Anual < 0.0, null(),
if((Viento_Anual <= 6.0 || Viento_Anual >= 15.0), 0.0,
if((Viento_Anual > 6.0 && Viento_Anual < 8.5), eval((Viento_Anual
- 6.0)/(8.5 - 6.0)), if((Viento_Anual > 14.0 && Viento_Anual <
15.0), eval((15.0 - Viento_Anual)/(15.0 - 14.0)), 1.0)))"
```

```
r.null map=RO_CP_VelViento null=0.0
```

REM Criterio: Aplicar fuzzy sobre altura de la ola media. En la capa resultante del fuzzy se pone valor 0.0 a los valores nulos.

```
r.mapcalc "RO_CP_Hs = if(UnionHsMedia_And < 0.0, null(),
if((UnionHsMedia_And == 0), 1.0, if((UnionHsMedia_And > 0 &&
UnionHsMedia_And < 14.0), eval((14.0 - UnionHsMedia_And)/(14.0 -
0)), 0.0)))"
```

```
r.null map=RO_CP_Hs null=0.0
```

REM Criterio: Aplicación de operación difusa sobre las distancias a las rutas marítimas. Se calcula primero mapa de distancias a las rutas marítimas. En la capa resultante del fuzzy se pone valor 0.0 a los valores nulos (Distancias Rutas Maritimas no tiene valores nulos).

```
v.to.rast input=Rutas_Maritimas layer=1 type=line
output=Rutas_Maritimas use=val value=1 rows=4096
```

```
r.grow.distance input=Rutas_Maritimas
distance=DistanciasRutasMaritimas metric=euclidean
```

```
r.mapcalc "RO_CP_DistRutasMarit = if(DistanciasRutasMaritimas <
0.0, null(), if((DistanciasRutasMaritimas <= 3704.0 ||
```

```

DistanciasRutasMaritimas >= 37040.0), 0.0,
if((DistanciasRutasMaritimas > 3704.0 && DistanciasRutasMaritimas
< 5556.0), eval((DistanciasRutasMaritimas - 3704.0)/(5556.0 -
3704.0)), if((DistanciasRutasMaritimas > 9260.0 &&
DistanciasRutasMaritimas < 37040.0), eval((37040.0 -
DistanciasRutasMaritimas)/(37040.0 - 9260.0)), 1.0)))"

```

REM Operación: MLP con pesos W1(RO_CP_VelViento.tif)=0.6,
W2(RO_CP_Hs.tif)=0.3 y W3(RO_CP_DistRutasMarit.tif)=0.1
***** SE TOMAN LOS PESOS DE ORDEN COMO 1/3

```

r.mapcalc "RO_CP_MLP = if(isnull(RO_CP_VelViento) ||
isnull(RO_CP_Hs) || isnull(RO_CP_DistRutasMarit),
null(),if(RO_CP_VelViento >= RO_CP_Hs && RO_CP_Hs >=
RO_CP_DistRutasMarit, eval(((RO_CP_VelViento*0.6*(1.0/3.0)) +
(RO_CP_Hs*0.3*(1.0/3.0)) + (RO_CP_DistRutasMarit*0.1*(1.0/3.0)))
/ (0.6*(1.0/3.0)+0.3*(1.0/3.0)+0.1*(1.0/3.0))),
if(RO_CP_VelViento >= RO_CP_DistRutasMarit &&
RO_CP_DistRutasMarit > RO_CP_Hs,
eval(((RO_CP_VelViento*0.6*(1.0/3.0)) +
(RO_CP_DistRutasMarit*0.1*(1.0/3.0)) + (RO_CP_Hs*0.3*(1.0/3.0)))
/ (0.6*(1.0/3.0)+0.1*(1.0/3.0)+0.3*(1.0/3.0))), if(RO_CP_Hs >
RO_CP_VelViento && RO_CP_VelViento >= RO_CP_DistRutasMarit,
eval(((RO_CP_Hs*0.3*(1.0/3.0)) + (RO_CP_VelViento*0.6*(1.0/3.0))
+ (RO_CP_DistRutasMarit*0.1*(1.0/3.0))) /
(0.3*(1.0/3.0)+0.6*(1.0/3.0)+0.1*(1.0/3.0))), if(RO_CP_Hs >=
RO_CP_DistRutasMarit && RO_CP_DistRutasMarit > RO_CP_VelViento,
eval(((RO_CP_Hs*0.3*(1.0/3.0)) +
(RO_CP_DistRutasMarit*0.1*(1.0/3.0)) +
(RO_CP_VelViento*0.6*(1.0/3.0))) /
(0.3*(1.0/3.0)+0.1*(1.0/3.0)+0.6*(1.0/3.0))),if(RO_CP_DistRutasMa
rit > RO_CP_VelViento && RO_CP_VelViento >= RO_CP_Hs,
eval(((RO_CP_DistRutasMarit*0.1*(1.0/3.0)) +
(RO_CP_VelViento*0.6*(1.0/3.0)) + (RO_CP_Hs*0.3*(1.0/3.0))) /
(0.1*(1.0/3.0)+0.6*(1.0/3.0)+0.3*(1.0/3.0))),if(RO_CP_DistRutasMa
rit > RO_CP_Hs && RO_CP_Hs > RO_CP_VelViento,
eval(((RO_CP_DistRutasMarit*0.1*(1.0/3.0)) +
(RO_CP_Hs*0.3*(1.0/3.0)) + (RO_CP_VelViento*0.6*(1.0/3.0))) /
(0.1*(1.0/3.0)+0.3*(1.0/3.0)+0.6*(1.0/3.0))), null() ))))"

```

REM Operación:AND_fuzzy (mínimo) de RO_CP_MLP y RF_RO_CT

```

r.mapcalc "RF_RO_salida = if(isnull(RO_CP_MLP) ||
isnull(RF_RO_CT), null(), if(RO_CP_MLP <= RF_RO_CT, RO_CP_MLP,
RF_RO_CT))"

```

REM Aplicación de RESTRICCIONES DE IMPACTO MEDIO AMBIENTAL

REM Operación: Zonas excluyentes de las zonas de Almadrabas

```
v.in.ogr "dsn=C:\Users\GSI\Desktop\bd GMV\BBDD
Oceanlider_actualizada\P\02_Capas\02_IT\UMyL\UP\ALMADRABAS_p.shp"
output=Almadrabas min_area=0.0001 snap=-1
```

```
v.to.rast input=Almadrabas layer=1 type=point,line,area
output=REIA_Almadrabas use=val value=0 rows=4096
```

```
r.null map=REIA_Almadrabas null=1
```

REM Operación: Zonas excluyentes a las zonas MABs --- zonas de reserva de la biosfera

```
v.in.ogr "dsn=C:\Users\GSI\Desktop\bd GMV\BBDD
Oceanlider_actualizada\P\02_Capas\02_IT\CyMA\EP\IN\MABs_P.shp"
output=MABs
min_area=0.0001 snap=-1
```

```
v.to.rast input=MABs layer=1 type=point,line,area
output=REIA_MABs use=val value=0 rows=4096
```

```
r.null map=REIA_MABs null=1
```

REM Operación: MLP de las reglas medio ambientales, con pesos: W1 (RIA_Almadrabas) = 0.7 y W2 (RIA_MABs) = 0.3 CON PESOS DE ORDEN DE 0.5

```
r.mapcalc "REIA_MLP = if(isnull(REIA_Almadrabas) ||
isnull(REIA_MABs), null(), if(REIA_Almadrabas >= REIA_MABs,
eval(((REIA_Almadrabas*0.7*0.5) + (REIA_MABs*0.3*0.5)) /
(0.7*0.5+0.3*0.5)), eval(((REIA_Almadrabas*0.7*0.5) +
(REIA_MABs*0.3*0.5)) / (0.7*0.5+0.3*0.5))))"
```

REM Operación: : AND_fuzzy (mínimo) de reglas de impacto medio ambiental y reglas físicas (RF_DCyHs y REIA_MLP)

```
r.mapcalc "RF_REIA_salida = if(isnull(RF_DCyHs) ||
isnull(REIA_MLP), null(), if(RF_DCyHs <= REIA_MLP, RF_DCyHs,
REIA_MLP))"
```

REM Aplicación de REGLA DE DECISIÓN FINAL (OWA)

REM Criterio: Cálculo de la regla de decisión final mediante una Media Ponderada Ordenada (OWA) de las restricciones físicas y operacionales junto con las reglas físicas y reglas de impacto

medio ambiental (RF_RO_salida y RF_REIA_salida) CON
 W1(RF_RO_salida)= 0.5 Y W2 (RF_REIA_salida) = 0.5

REM Cálculo de la CapaResultado con Z1=1 Y Z2=0

```
r.mapcalc "CapaResultado_1_0 = if(isnull(RF_RO_salida) ||
isnull(RF_REIA_salida), null(),if(RF_RO_salida >= RF_REIA_salida,
eval(((RF_RO_salida*0.5*1) + (RF_REIA_salida*0.5*0)) /
(0.5*1+0.5*0)), eval(((RF_RO_salida*0.5*0) +
(RF_REIA_salida*0.5*1)) / (0.5*0+0.5*1))))"
```

REM Cálculo de la CapaResultado con Z1=0 y Z2=1

```
r.mapcalc "CapaResultado_0_1 = if(isnull(RF_RO_salida) ||
isnull(RF_REIA_salida), null(),if(RF_RO_salida >= RF_REIA_salida,
eval(((RF_RO_salida*0.5*0) + (RF_REIA_salida*0.5*1)) /
(0.5*0+0.5*1)), eval(((RF_RO_salida*0.5*1) +
(RF_REIA_salida*0.5*0)) / (0.5*1+0.5*0))))"
```

REM Cálculo de la CapaResultado con Z1=0.5 y Z2=0.5

```
r.mapcalc "CapaResultado_0.5_0.5 = if(isnull(RF_RO_salida) ||
isnull(RF_REIA_salida), null(),if(RF_RO_salida >= RF_REIA_salida,
eval(((RF_RO_salida*0.5*0.5) + (RF_REIA_salida*0.5*0.5)) /
(0.5*0.5+0.5*0.5)), eval(((RF_RO_salida*0.5*0.5) +
(RF_REIA_salida*0.5*0.5)) / (0.5*0.5+0.5*0.5))))"
```

REM Cálculo de la CapaResultado con Z1=0.25 y Z2=0.75

```
r.mapcalc "CapaResultado_0.25_0.75 = if(isnull(RF_RO_salida) ||
isnull(RF_REIA_salida), null(),if(RF_RO_salida >= RF_REIA_salida,
eval(((RF_RO_salida*0.5*0.25) + (RF_REIA_salida*0.5*0.75)) /
(0.5*0.25+0.5*0.75)), eval(((RF_RO_salida*0.5*0.75) +
(RF_REIA_salida*0.5*0.25)) / (0.5*0.75+0.5*0.25))))"
```

REM Cálculo de la CapaResultado con Z1=0.75 y Z2=0.25

```
r.mapcalc "CapaResultado_0.75_0.25 = if(isnull(RF_RO_salida) ||
isnull(RF_REIA_salida), null(), if(RF_RO_salida >=
RF_REIA_salida, eval(((RF_RO_salida*0.5*0.75) +
(RF_REIA_salida*0.5*0.25)) / (0.5*0.75+0.5*0.25)),
eval(((RF_RO_salida*0.5*0.25) + (RF_REIA_salida*0.5*0.75)) /
(0.5*0.25+0.5*0.75))))"
```

REM Cambio de paleta de colores en las Capas Resultado

```
r.colors map=CapaResultado_0_1
rules="E:\Ampliacion_PFC\Ejemplo_aplicacion_criterios_totales_y_p
arciales\Resultados_finales_verde_V3\colors_verde.txt"
```



```
r.colors map=CapaResultado_1_0  
rules="E:\Ampliacion_PFC\Ejemplo_aplicacion_criterios_totales_y_p  
arciales\Resultados_finales_verde_V3\colors_verde.txt"
```

```
r.colors map=CapaResultado_0.5_0.5  
rules="E:\Ampliacion_PFC\Ejemplo_aplicacion_criterios_totales_y_p  
arciales\Resultados_finales_verde_V3\colors_verde.txt"
```

```
r.colors map=CapaResultado_0.25_0.75  
rules="E:\Ampliacion_PFC\Ejemplo_aplicacion_criterios_totales_y_p  
arciales\Resultados_finales_verde_V3\colors_verde.txt"
```

```
r.colors map=CapaResultado_0.75_0.25  
rules="E:\Ampliacion_PFC\Ejemplo_aplicacion_criterios_totales_y_p  
arciales\Resultados_finales_verde_V3\colors_verde.txt"
```

```
time < time_reloj.txt >> Script_Grass\Hora_fin.txt
```


Apéndice

E

FICHERO SQL DE LA UNIÓN DE DIFERENTES CAPAS VECTORIALES INFORMATIVAS

```
-----  
-----  
--BDD: BDDOceanLider  
-----  
-----  
  
-----  
--Union Lin_costa de las tres entidades  
-----  
  
CREATE TABLE public.lin_costa (  
    gid serial NOT NULL,  
    linea character varying (5),  
    zona character varying(50),  
    CONSTRAINT lin_costa_pkey PRIMARY KEY (gid)  
);  
  
SELECT  
AddGeometryColumn('public','lin_costa','geom',4326,'MultiLineStri  
ng',2);  
  
INSERT INTO public.lin_costa ("linea","zona","geom") (  
  
SELECT "linea","zona",ST_Transform(ST_Transform(geom,23030),4326)  
FROM peninsula.lin_costa_p  
UNION  
SELECT "linea","zona",ST_Transform(ST_Transform(geom,23031),4326)  
FROM baleares.lin_costa_b  
UNION  
SELECT "linea","zona",ST_Transform(ST_Transform(geom,32628),4326)  
FROM canarias.lin_costa_can_wgs84h28);  
  
-----  
--Union de las CCAA de las tres entidades  
-----
```

```

CREATE TABLE public.ccaa (
    gid serial NOT NULL,
    comunidad character varying (30),
    CONSTRAINT ccaa_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','ccaas','geom',4326,'MultiPolygon',2);

INSERT INTO public.ccaas ("comunidad","geom") (
SELECT "comunidad",ST_Transform(ST_Transform(geom,23030),4326)
FROM peninsula.ccaas_p
UNION
SELECT "comunidad",ST_Transform(ST_Transform(geom,23031),4326)
FROM baleares.ccaas_b
UNION
SELECT "comunidad",ST_Transform(ST_Transform(geom,32628),4326)
FROM canarias.ccaas_can_wgs84h28);

-----
-- rutas transporte maritimo, sólo hay de Andalucía de una BBDD
-- antigua, la importamos y reprojectamos a 4326
-----

CREATE TABLE public.rutas_trans_maritimo (
    gid serial NOT NULL,
    id integer,
    ruta character varying(50),
    CONSTRAINT rutas_trans_maritimo_2pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','rutas_trans_maritimo','geom',4326,'MultiLineString',2);

INSERT INTO public.rutas_trans_maritimo ("id","ruta","geom") (
SELECT "id","ruta",ST_Transform(ST_Transform(geom,23030),4326)
FROM public.rutas);

-----
--Unión almadrabas
-----

CREATE TABLE public.almadrabas (
    gid serial NOT NULL,
    id integer,
    CONSTRAINT almadrabas_pkey PRIMARY KEY (gid)

```

```
);

SELECT
AddGeometryColumn('public','almadrabas','geom',4326,'MultiPolygon',2);

INSERT INTO public.almadrabas("id","geom") (
SELECT "id",ST_Transform(ST_Transform(geom,23030),4326) FROM
peninsula.almadrabas_p);

-----
-- Unión mabs
-----

CREATE TABLE public.mabs (
gid serial NOT NULL,
figura character varying(50),
nombre character varying(250),
CONSTRAINT mabs_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','mabs','geom',4326,'MultiPolygon',2);

INSERT INTO public.mabs ("figura","nombre","geom") (

SELECT
"figura","nombre",ST_Transform(ST_Transform(geom,23030),4326)
FROM peninsula.mabs_p
UNION
SELECT
"figura","nombre",ST_Transform(ST_Transform(geom,23031),4326)
FROM baleares.mabs_b
UNION
SELECT
"figura","nombre",ST_Transform(ST_Transform(geom,32628),4326)
FROM canarias.mabs_can_wgs84h28);

-----
-----
-- Unión batimetría
-----
--Cuidado para canarias zona tiene caracteres 100, para los demás
50
CREATE TABLE public.batimetria (
gid serial NOT NULL,
profundida integer,
zona character varying(100),
```

```

        CONSTRAINT batimetria_pkey PRIMARY KEY (gid)
    );

SELECT
AddGeometryColumn('public','batimetria','geom',4326,'MultiLineString',2);

INSERT INTO public.batimetria ("profundida","zona","geom") (

SELECT
"profundida","zona",ST_Transform(ST_Transform(geom,23030),4326)
FROM peninsula.batimetria_p
UNION
SELECT
"profundida","zona",ST_Transform(ST_Transform(geom,23031),4326)
FROM baleares.batimetria_b
UNION
SELECT
"profundida","zona",ST_Transform(ST_Transform(geom,32628),4326)
FROM canarias.batimetria_can_wgs84h28);

-----
-----
-- ATON Ayudas a la navegación
-----
-----

CREATE TABLE public.aton (
    gid serial NOT NULL,
    tipo character varying(100),
    zona_baliz character varying(254),
    notas character varying(254),
    alcance character varying(254),
    ritmo character varying(254),
    color character varying(254),
    periodo numeric,
    marca_de_t character varying(254),
    marca character varying(254),
    nombre character varying(254),
    n character varying(254),
    CONSTRAINT aton_pkey PRIMARY KEY (gid)
);

SELECT AddGeometryColumn('public','aton','geom',4326,'Point',2);

INSERT INTO public.aton
("tipo","zona_baliz","notas","alcance","ritmo","color","periodo",
"marca_de_t","marca","nombre","n","geom") (

```

```
SELECT
"tipo","zona_baliz","notas","alcance","ritmo","color","periodo","
marca_de_t","marca","nombre","n°",ST_Transform(ST_Transform(geom,
23030),4326) FROM peninsula.aton_p
UNION
SELECT
"tipo","zona_baliz","notas","alcance","ritmo","color","periodo","
marca_de_t","marca","nombre","n°",ST_Transform(ST_Transform(geom,
23031),4326) FROM baleares.aton_b
UNION
SELECT
"tipo","zona_baliz","notas","alcance","ritmo","color","periodo","
marca_de_t","marca","nombre","n°",ST_Transform(ST_Transform(geom,
32628),4326) FROM canarias.aton_can_wgs84h28);
```

```
-----
-----
-- DPP_PE Dominio Público Portuario
-----
-----
```

```
CREATE TABLE public.dpp (
gid serial NOT NULL,
origen_pol character varying(60),
aportuaria character varying(40),
puerto_est character varying(40),
uso character varying(3),
CONSTRAINT dpp_pkey PRIMARY KEY (gid)
);
```

--Cuidado: para Peninsula el atributo 'origen_pol', en Baleares es 'fuente'

```
SELECT
AddGeometryColumn('public','dpp','geom',4326,'MultiPolygon',2);
```

```
INSERT INTO public.dpp
("origen_pol","aportuaria","puerto_est","uso","geom") (
```

```
SELECT
"origen_pol","aportuaria","puerto_est","uso",ST_Transform(ST_Transform(geom,23030),4326) FROM peninsula.dpp_pe_p
UNION
SELECT
"fuente","aportuaria","puerto_est","uso",ST_Transform(ST_Transform(geom,32628),4326) FROM canarias.dpp_canarias_wgs84h28);
```

```
-----
-----
-- Conducciones - conducciones submarinas
```

```
-----  
-----  
--conducciones_p, conducciones_lineales_b, y  
conducciones_line_can_wgs84h28 son de tipo Multilinestring ZM (3  
dimensiones)  
-- Para forzar a insertar la geometría en dos dimensiones,  
utilizamos la función ST_Force_2D de PostGIS  
  
CREATE TABLE public.conducciones (  
    gid serial NOT NULL,  
    tipol character varying(50),  
    tipo character varying(40),  
    fuente character varying(50),  
    nombre character varying(100),  
    CONSTRAINT conducciones_pkey PRIMARY KEY (gid)  
);  
  
SELECT  
AddGeometryColumn('public','conducciones','geom',4326,'MultiLine  
string',2);  
  
INSERT INTO public.conducciones  
( "tipol", "tipo", "fuente", "nombre", "geom" ) (  
  
SELECT  
"tipol", "tipo", "fuente", "nombre", ST_Transform(ST_Transform(geom, 2  
3030), 4326) FROM peninsula.conducciones_p  
UNION  
SELECT  
"tipol", "tipo", null, "nombre", ST_Transform(ST_Transform(ST_Force_2  
D(geom), 23031), 4326) FROM baleares.conducciones_lineales_b  
UNION  
SELECT  
"tipol", "tipo", "fuente", "nombre", ST_Transform(ST_Transform(geom, 3  
2628), 4326) FROM canarias.conducciones_line_can_wgs84h28);  
  
-----  
-----  
-- ext_aridos - Extracción de áridos  
-----  
-----  
  
-- ext_aridos_p, ext_aridos_b y ext_aridos_can_wgs84h28 son de  
tipo multipolygon ZM  
-- Comprobado con:  
--SELECT ST_AsText(geom) FROM peninsula.ext_aridos_p;  
--SELECT ST_Zmflag(geom) FROM peninsula.ext_aridos_p;
```


--Para forzar a crear la tabla en dos dimensiones utilizamos la función ST_Force_2D de PostGIS cuando insertamos la geometría

```
CREATE TABLE public.ext_aridos (
    gid serial NOT NULL,
    fuente character varying(520),
    descriptio character varying(150),
    espesor character varying(10),
    d50_medio character varying(10),
    mat character varying(254),
    superficie numeric,
    "año" smallint,
    volumen integer,
    volumen_me character varying(254),
    volumen_to character varying(254),
    tipo character varying(50),
    CONSTRAINT aridos_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','ext_aridos','geom',4326,'MultiPolygon',2);

INSERT INTO public.ext_aridos
("fuente","descriptio","espesor","d50_medio","mat","superficie","año","volumen","volumen_me","volumen_to","tipo","geom") (

SELECT
"fuente","descriptio","espesor","d50_medio",null,null,null,null,null,null,ST_Transform(ST_Transform(ST_Force_2D(geom),23030),4326) FROM peninsula.ext_aridos_p
UNION
SELECT
"fuente",null,null,"d50","mat","superficie","año","volumen",null,null,null,ST_Transform(ST_Transform(ST_Force_2D(geom),23031),4326) FROM baleares.ext_aridos_b
UNION
SELECT
"fuente","descriptio","espesor_me","d50",null,null,null,null,"volumen_me","volumen_to","tipo",ST_Transform(ST_Transform(ST_Force_2D(geom),32628),4326) FROM canarias.ext_aridos_can_wgs84h28);

-----
-----
-- Vertidos - vertido material de dragado
-----
-----

CREATE TABLE public.vertidos (
```

```
        gid serial NOT NULL,
        valoracion numeric,
        fuente character varying(50),
        nombre character varying(100),
        tipo character varying(100),
        CONSTRAINT vertidos_pkey PRIMARY KEY (gid)
    );

--Baleares sólo tiene tipo
--Canarias no tiene valoracion

SELECT
AddGeometryColumn('public','vertidos','geom',4326,'Point',2);

INSERT INTO public.vertidos
("valoracion","fuente","nombre","tipo","geom") (

SELECT
"valoracion","fuente","nombre","tipo",ST_Transform(ST_Transform(geom,23030),4326) FROM peninsula.vertidos_p
UNION
SELECT
null,null,null,"tipo",ST_Transform(ST_Transform(geom,23031),4326)
FROM baleares.vertidos_b
UNION
SELECT
null,"fuente","nombre","tipo",ST_Transform(ST_Transform(geom,32628),4326) FROM canarias.vertidos_can_wgs84h28);

-----
-----
-- RMO - Sensores instrumentales de medición de parámetros
-----
-----

CREATE TABLE public.rmo (
    gid serial NOT NULL,
    cod character varying(50),
    model character varying(50),
    registros character varying(50),
    prof character varying(50),
    organismo character varying(50),
    tip_sensor character varying(50),
    tipo_reg character varying(50),
    informe character varying(254),
    red character varying(10),
    nombre character varying(50),
    CONSTRAINT rmo_pkey PRIMARY KEY (gid)
);
```

```
SELECT AddGeometryColumn('public','rmo','geom',4326,'Point',2);

INSERT INTO public.rmo
("cod","model","registros","prof","organismo","tip_sensor","tipo_
reg","informe","red","nombre","geom") (

SELECT
"cod","model","registros","prof","organismo","tip_sensor","tipo_r
eg","informe","red","nombre",ST_Transform(ST_Transform(geom,23030
),4326) FROM peninsula.rmo_p
UNION
SELECT
"cod","model","registros","prof","organismo","tip_sensor","tipo_r
eg","informe","red","nombre",ST_Transform(ST_Transform(geom,23031
),4326) FROM baleares.rmo_b
UNION
SELECT
"cod","model","registros","prof","organismo","tip_sensor","tipo_r
eg","informe","red","nombre",ST_Transform(ST_Transform(geom,32628
),4326) FROM canarias.rmo_can_wgs84h28);

-----
-----
-- DST - Dispositivos de separación d tráfico marítimo
-----
-----

CREATE TABLE public.dst (
    gid serial NOT NULL,
    valoracion numeric,
    dispositiv character varying(100),
    CONSTRAINT dst_pkey PRIMARY KEY (gid)
);

--Canarias no tiene valoracion
SELECT
AddGeometryColumn('public','dst','geom',4326,'MultiPolygon',2);

INSERT INTO public.dst ("valoracion","dispositiv","geom") (

SELECT
"valoracion","dispositiv",ST_Transform(ST_Transform(geom,23030),4
326) FROM peninsula.dst_p
UNION
SELECT
null,"dispositiv",ST_Transform(ST_Transform(geom,32628),4326)
FROM canarias.dst_ze_can_wgs84h28);
```

```
-----  
-----  
-- Militar - Zonas restringidas para uso militar  
-----  
-----  
  
CREATE TABLE public.militar (  
    gid serial NOT NULL,  
    nombre character varying(254),  
    CONSTRAINT militar_pkey PRIMARY KEY (gid)  
);  
  
SELECT  
AddGeometryColumn('public','militar','geom',4326,'MultiPolygon',2  
);  
  
INSERT INTO public.militar ("nombre","geom") (  
  
SELECT "nombre",ST_Transform(ST_Transform(geom,23030),4326) FROM  
peninsula.militar_p  
UNION  
SELECT "nombre",ST_Transform(ST_Transform(geom,32628),4326) FROM  
canarias.militar_can_wgs84h28);  
  
-----  
-----  
-- zona_cables - Zonas restringidas por la presencia de cables  
eléctricos submarinos  
-----  
-----  
  
CREATE TABLE public.zona_cables (  
    gid serial NOT NULL,  
    id integer,  
    CONSTRAINT zcables_pkey PRIMARY KEY (gid)  
);  
  
SELECT  
AddGeometryColumn('public','zona_cables','geom',4326,'MultiPolygo  
n',2);  
  
INSERT INTO public.zona_cables ("id","geom") (  
  
SELECT "id",ST_Transform(ST_Transform(geom,23030),4326) FROM  
peninsula.zona_cables_p  
UNION  
SELECT "id",ST_Transform(ST_Transform(geom,23031),4326) FROM  
balears.zona_cables_b);
```

```
-----  
-----  
-- Vertido dragado - Zonas restringidas por presentecia de  
vertidos de dragado  
-----  
-----
```

```
CREATE TABLE public.vertido_dragado(  
    gid serial NOT NULL,  
    "nombre" character varying(254),  
    "tipo" character varying(100),  
    "fuente" character varying(50),  
    CONSTRAINT vertido_dragado_pkey PRIMARY KEY (gid)  
);  
  
SELECT  
AddGeometryColumn('public','vertido_dragado','geom',4326,'Point',  
2);  
  
INSERT INTO public.vertido_dragado  
("nombre","tipo","fuente","geom") (  
SELECT  
"nombre","tipo","fuente",ST_Transform(ST_Transform(geom,23030),43  
26) FROM peninsula.vertido_dragado_p  
);
```

```
-----  
-----  
-- proteccionducciones- Zonas restringidas por ser áreas con  
prohibición de fondeo y pesca de arrastre  
-----  
-----
```

```
CREATE TABLE public.proteccionducciones(  
    gid serial NOT NULL,  
    "tipo" character varying(100),  
    "observac" character varying(250),  
    "localiza" character varying(50),  
    "fuente" character varying(50),  
    "nombre" character varying(60),  
    CONSTRAINT proteccionducciones_pkey PRIMARY KEY (gid)  
);  
  
--Peninsula no tiene fuente y nombre  
--Canarias no tiene observac y localiza  
  
SELECT  
AddGeometryColumn('public','proteccionducciones','geom',4326,'Mult  
iPolygon',2);
```

```

INSERT INTO public.proteccionducciones
("tipo","observac","localiza","fuente","nombre","geom") (

SELECT
"tipo","observac","localiza",null,null,ST_Transform(ST_Transform(
geom,23030),4326) FROM peninsula.proteccionducciones_p
UNION
SELECT
"tipo",null,null,"fuente","nombre",ST_Transform(ST_Transform(geom
,32628),4326) FROM canarias.proteccionducciones_can_wgs84h28);

-----
-----
-- vertido_material_militar- Zonas restringidas por la existencia
de vertidos de material militar
-----
-----

CREATE TABLE public.vertido_material_militar(
gid serial NOT NULL,
"tipo_verti" character varying(50),
"comunidad" character varying(50),
CONSTRAINT vertido_material_militar_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','vertido_material_militar','geom',4326
,'MultiPolygon',2);

INSERT INTO public.vertido_material_militar
("tipo_verti","comunidad","geom") (
SELECT
"tipo_verti","comunidad",ST_Transform(ST_Transform(geom,23030),43
26) FROM peninsula.vertido_material_militar_p
);

-----
-----
-- acuicultura - acuicultura marina
-----
-----

CREATE TABLE public.acuicultura(
gid serial NOT NULL,
"titular" character varying(70),
"nom_latin" character varying(50),
"nom_especi" character varying(50),
"fuente" character varying(50),

```

```
"tipo" character varying(50),
"ubicacio" character varying(50),
"codi_inst" integer,
CONSTRAINT acuicultura_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','acuicultura','geom',4326,'MultiPolygon',2);

INSERT INTO public.acuicultura
("titular","nom_latin","nom_especi","fuente","tipo","ubicacio","codi_inst","geom") (
SELECT
"titular","nom_latin","nom_especi","fuente","tipo","ubicacio","codi_inst",ST_Transform(ST_Transform(geom,23030),4326) FROM
peninsula.acuicultura_p
);

-----
-----
-- caladeros - caladeros de pesca
-----
-----

CREATE TABLE public.caladeros(
gid serial NOT NULL,
"fuente" character varying(50),
"nombre_cal" character varying(200),
"artes_pesc" character varying(200),
"especies" character varying(200),
CONSTRAINT caladeros_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','caladeros','geom',4326,'MultiPolygon',2);

INSERT INTO public.caladeros
("fuente","nombre_cal","artes_pesc","especies","geom") (
SELECT
"fuente","nombre_cal","artes_pesc","especies",ST_Transform(ST_Transform(geom,23030),4326) FROM peninsula.caladeros_p
);

-----
-----
-- viveros - viveros
```

```
-----  
-----  
  
CREATE TABLE public.viveros(  
    gid serial NOT NULL,  
    "id" integer,  
    CONSTRAINT viveros_pkey PRIMARY KEY (gid)  
);  
  
SELECT  
AddGeometryColumn('public','viveros','geom',4326,'MultiPolygon',2  
);  
  
INSERT INTO public.viveros ("id","geom") (  
  
SELECT "id",ST_Transform(ST_Transform(geom,23030),4326) FROM  
peninsula.viveros_p  
UNION  
SELECT "id",ST_Transform(ST_Transform(geom,23031),4326) FROM  
balears.viveros_refug_b);  
  
-----  
-----  
-- ZPIPS - Zonas protegidas para el interés pesquero  
-----  
-----  
  
CREATE TABLE public.zpips(  
    gid serial NOT NULL,  
    "nombre" character varying(200),  
    CONSTRAINT zpips_pkey PRIMARY KEY (gid)  
);  
  
SELECT  
AddGeometryColumn('public','zpips','geom',4326,'MultiPolygon',2);  
  
INSERT INTO public.zpips ("nombre","geom") (  
  
SELECT "nombre",ST_Transform(ST_Transform(geom,23030),4326) FROM  
peninsula.zpips_p  
);  
  
-----  
-----  
-- piscifactoria - piscifactorias  
-----  
-----  
  
CREATE TABLE public.piscifactoria(  

```



```
gid serial NOT NULL,
"fuente" character varying(50),
"id" integer,
"tipo" character varying(50),
CONSTRAINT piscifactoria_pkey PRIMARY KEY (gid)
);
--Canarias no tiene fuente

SELECT
AddGeometryColumn('public','piscifactoria','geom',4326,'MultiPolygon',2);

INSERT INTO public.piscifactoria ("fuente","id","tipo","geom") (

SELECT
"fuente","id",null,ST_Transform(ST_Transform(geom,23030),4326)
FROM peninsula.piscifactoria_p
UNION
SELECT
null,null,"tipo",ST_Transform(ST_Transform(geom,32628),4326) FROM
canarias.piscifactoria_can_wgs84h28);

-----
-----
-- cria_moluscos - cria de moluscos
-----
-----

CREATE TABLE public.cria_moluscos(
gid serial NOT NULL,
"especiel" character varying(254),
"especie" character varying(254),
"id_1" numeric,
"clave_1" character varying(50),
"id" smallint,
"clave" character varying(50),
"nombre" character varying(50),
"fuente" character varying(50),
"ubicacion" character varying(254),
"especies" character varying(254),
CONSTRAINT cria_moluscos_pkey PRIMARY KEY (gid)
);

--Baleares tiene especies, ubicación, clave

SELECT
AddGeometryColumn('public','cria_moluscos','geom',4326,'MultiPolygon',2);
```

```
INSERT INTO public.cria_moluscos
("especiel","especie","id_1","clave_1","id","clave","nombre","fuen
te","ubicacion","especies","geom") (

SELECT
"especiel","especie","id_1","clave_1","id","clave","nombre","fuen
te","ubicación",null,ST_Transform(ST_Transform(geom,23030),4326)
FROM peninsula.cria_moluscos_p
UNION
SELECT
null,null,null,null,null,"clave",null,null,"ubicación","especies"
,ST_Transform(ST_Transform(geom,23031),4326) FROM
baleares.cria_moluscos_b);
```

Apéndice

F

FICHERO SQL DE LA UNIÓN DE DIFERENTES CAPAS VECTORIALES DE RESTRICCIÓN

```
-----  
-- BBDD: BBDDOceanLider_restricciones  
-----  
  
-----  
--Union Geofis de las tres entidades  
-----  
  
CREATE TABLE public.geofis (  
    gid serial NOT NULL,  
    sustrato character varying (20),  
    descripcio character varying(150),  
    agrup character varying (50),  
    disp1 numeric (6,3),  
    disp2 numeric (6,3),  
    disp3 numeric (6,3),  
    disp4 numeric (6,3),  
    disp5 numeric (6,3),  
    disp6 numeric (6,3),  
    CONSTRAINT lin_costa_pkey PRIMARY KEY (gid)  
);  
  
SELECT  
AddGeometryColumn('public','geofis','geom',4326,'MultiPolygon',2)  
;  
  
INSERT INTO public.geofis  
( "sustrato", "descripcio", "agrup", "disp1", "disp2", "disp3", "disp4",  
  "disp5", "disp6", "geom" ) (  
  
SELECT  
"sustrato", "descripcio", "agrup", "disp1", "disp2", "disp3", "disp4", "
```

```

disp5", "disp6", ST_Transform(ST_Transform(geom, 23030), 4326) FROM
peninsula.geofis_p
UNION
SELECT
"sustrato", "descripcion", "agrup", "disp1", "disp2", "disp3", "disp4", "
disp5", "disp6", ST_Transform(ST_Transform(geom, 23031), 4326) FROM
balears.geofis_b
UNION
SELECT
"sustrato", "descripcion", "agrup", "disp1", "disp2", "disp3", "disp4", "
disp5", "disp6", ST_Transform(ST_Transform(geom, 32628), 4326) FROM
canarias.geofis_can_wgs84h28);

```

```

-----
--Union de las Formas de las tres entidades
-----

```

```

CREATE TABLE public.formas (
    gid serial NOT NULL,
    tipo character varying (50),
    fuente character varying(50),
    categoria character varying(50),
    disp1 numeric (6,3),
    disp2 numeric (6,3),
    disp3 numeric (6,3),
    disp4 numeric (6,3),
    disp5 numeric (6,3),
    disp6 numeric (6,3),
    CONSTRAINT formas_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public', 'formas', 'geom', 4326, 'MultiPolygon', 2)
;

INSERT INTO public.formas
("tipo", "fuente", "categoria", "disp1", "disp2", "disp3", "disp4", "dis
p5", "disp6", "geom") (

SELECT
"tipo", "fuente", "categor a", "disp1", "disp2", "disp3", "disp4", "disp
5", "disp6", ST_Transform(ST_Transform(geom, 23030), 4326) FROM
peninsula.formas_p );

```

```

-----
--Union Geoparques de las tres entidades
-----

```

```

CREATE TABLE public.geoparques (

```

```
gid serial NOT NULL,
habitat character varying (254),
porcentaje numeric,
area_total numeric,
documentac character varying(50),
area numeric,
nombre character varying(120),
figura character varying(50),
disp1 numeric (6,3),
disp2 numeric (6,3),
disp3 numeric (6,3),
disp4 numeric (6,3),
disp5 numeric (6,3),
disp6 numeric (6,3),
CONSTRAINT geoparques_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','geoparques','geom',4326,'MultiPolygon',2);

INSERT INTO public.geoparques
("habitat","porcentaje","area_total","documentac","area","nombre",
"figura","disp1","disp2","disp3","disp4","disp5","disp6","geom")
(

SELECT
"habitat","porcentaje","area_total","documentac","area","nombre",
"figura","disp1","disp2","disp3","disp4","disp5","disp6",ST_Transform(ST_Transform(geom,23030),4326) FROM peninsula.geopp_p_fin
);

-----
--Union MABs de las tres entidades
-----

CREATE TABLE public.mabs (
gid serial NOT NULL,
habitat character varying (254),
porcentaje numeric,
area_total numeric,
documentac character varying(50),
nombre character varying(250),
figura character varying(50),
disp1 numeric (6,3),
disp2 numeric (6,3),
disp3 numeric (6,3),
disp4 numeric (6,3),
disp5 numeric (6,3),
```

```

        disp6 numeric (6,3),
        CONSTRAINT marbs_pkey PRIMARY KEY (gid)
    );

SELECT
AddGeometryColumn('public','mabs','geom',4326,'MultiPolygon',2);

INSERT INTO public.mabs
("habitat","porcentaje","area_total","documentac","nombre","figura",
"disp1","disp2","disp3","disp4","disp5","disp6","geom") (

SELECT
"habitat","porcentaje","area_total","documentac","nombre","figura",
"disp1","disp2","disp3","disp4","disp5","disp6",ST_Transform(ST
_Transform(geom,23030),4326) FROM peninsula.mabs_p_fin
UNION
SELECT
"habitat","porcentaje","area_total",null,"nombre","figura","disp1",
"disp2","disp3","disp4","disp5","disp6",ST_Transform(ST_Transfo
rm(geom,32628),4326) FROM canarias.mabs_c_fin);

-----
--Union P_Humanidad de las tres entidades
-----

CREATE TABLE public.p_humanidad (
    gid serial NOT NULL,
    area numeric,
    figura character varying(50),
    nombre character varying(120),
    disp1 numeric (6,3),
    disp2 numeric (6,3),
    disp3 numeric (6,3),
    disp4 numeric (6,3),
    disp5 numeric (6,3),
    disp6 numeric (6,3),
    CONSTRAINT phumanidad_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','p_humanidad','geom',4326,'MultiPolygo
n',2);

INSERT INTO public.p_humanidad
("area","figura","nombre","disp1","disp2","disp3","disp4","disp5",
"disp6","geom") (

```

```
SELECT
"area", "figura", "nombre", "disp1", "disp2", "disp3", "disp4", "disp5",
"disp6", ST_Transform(ST_Transform(geom,23030),4326) FROM
peninsula.p_humanidad_p);
```

```
-----
--Union OSPAR_AMPS de las tres entidades
-----
```

```
-- Ha habido un error en la importación de ospar_amps_p, los
campos de los dispositivos se han importado como null
-- Se ha rellenado de forma manual con los valores 0.0
```

```
CREATE TABLE public.ospar_amps (
gid serial NOT NULL,
normativa character varying (50),
area numeric,
nombre character varying(100),
figura character varying(50),
disp1 numeric (6,3),
disp2 numeric (6,3),
disp3 numeric (6,3),
disp4 numeric (6,3),
disp5 numeric (6,3),
disp6 numeric(6,3),
CONSTRAINT ospar_amps_pkey PRIMARY KEY (gid)
);
```

```
SELECT
AddGeometryColumn('public','ospar_amps','geom',4326,'MultiPolygon',2);
```

```
INSERT INTO public.ospar_amps
("normativa","area","nombre","figura","disp1","disp2","disp3","disp4","disp5","disp6","geom") (
```

```
SELECT
"normativa","area","nombre","figura","disp1","disp2","disp3","disp4","disp5","disp6",ST_Transform(ST_Transform(geom,23030),4326)
FROM peninsula.ospar_amps_p
);
```

```
-----
--Union ZEPIM de las tres entidades
-----
```

```
CREATE TABLE public.zepims (
gid serial NOT NULL,
habitat character varying (254),
```

```

        porcentaje numeric,
        area_total numeric,
        documentac character varying(50),
        area numeric,
        figura character varying(100),
        nombre character varying(70),
        disp1 numeric (6,3),
        disp2 numeric (6,3),
        disp3 numeric (6,3),
        disp4 numeric (6,3),
        disp5 numeric (6,3),
        disp6 numeric (6,3),
        CONSTRAINT zepims_pkey PRIMARY KEY (gid)
    );

SELECT
AddGeometryColumn('public','zepims','geom',4326,'MultiPolygon',2)
;

INSERT INTO public.zepims
("habitat","porcentaje","area_total","documentac","area","figura"
,"nombre","disp1","disp2","disp3","disp4","disp5","disp6","geom")
(

SELECT
"habitat","porcentaje","area_total","documentac","area","figura",
"nombre","disp1","disp2","disp3","disp4","disp5","disp6",ST_Trans
form(ST_Transform(geom,23030),4326) FROM peninsula.zepims_p_fin
UNION
SELECT
"habitat","porcentaje","area_total",null,"area","figura","nombre"
,"disp1","disp2","disp3","disp4","disp5","disp6",ST_Transform(ST_
Transform(geom,23031),4326) FROM baleares.zepims_b_fin);

-----
--Union DipEuro de las tres entidades
-----

CREATE TABLE public.dipeuro (
    gid serial NOT NULL,
    fuente character varying(200),
    nombre character varying(120),
    area numeric,
    figura character varying(50),
    disp1 numeric (6,3),
    disp2 numeric (6,3),
    disp3 numeric (6,3),
    disp4 numeric (6,3),
    disp5 numeric (6,3),

```



```
        disp6 numeric (6,3),
        CONSTRAINT dipeuro_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','dipeuro','geom',4326,'MultiPolygon',2
);

INSERT INTO public.dipeuro
("fuente","nombre","area","figura","disp1","disp2","disp3","disp4
","disp5","disp6","geom") (

SELECT
"fuente","nombre","area","figura","disp1","disp2","disp3","disp4"
,"disp5","disp6",ST_Transform(ST_Transform(geom,23030),4326) FROM
peninsula.dipeuro_p
);

-----
--Union LICs de las tres entidades
-----

CREATE TABLE public.lics(
    gidserial NOT NULL,
    habitat character varying(254),
    porcentaje numeric,
    area_total numeric,
    normativa character varying(50),
    area numeric,
    lic_code character varying(20),
    nombre character varying(100),
    figura character varying(50),
    disp1 numeric (6,3),
    disp2 numeric (6,3),
    disp3 numeric (6,3),
    disp4 numeric (6,3),
    disp5 numeric (6,3),
    disp6 numeric (6,3),
    CONSTRAINT lics_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','lics','geom',4326,'MultiPolygon',2);

INSERT INTO public.lics
("habitat","porcentaje","area_total","normativa","area","lic_code
","nombre","figura","disp1","disp2","disp3","disp4","disp5","disp
6","geom") (
```

```

SELECT
"habitat", "porcentaje", "area_total", "normativa", "area", "lic_code"
, "nombre", "figura", "disp1", "disp2", "disp3", "disp4", "disp5", "disp6
", ST_Transform(ST_Transform(geom,23030),4326) FROM
peninsula.lics_p_fin
UNION
SELECT
"habitat", "porcentaje", "area_total", "normativa", "area", "lic_code"
, "nombre", "figura", "disp1", "disp2", "disp3", "disp4", "disp5", "disp6
", ST_Transform(ST_Transform(geom,23031),4326) FROM
baleares.lics_b_fin
UNION
SELECT
"habitat", "porcentaje", "area_total", "normativa", "area", "lic_code"
, "nombre", "figura", "disp1", "disp2", "disp3", "disp4", "disp5", "disp6
", ST_Transform(ST_Transform(geom,32628),4326) FROM
canarias.lics_c_fin);

```

```

-----
--Union ZEPAs de las tres entidades
-----

```

```

CREATE TABLE public.zepas(
gid serial NOT NULL,
habitat character varying(254),
porcentaje numeric,
area_total numeric,
normativa character varying(50),
area numeric,
zepa_code character varying(50),
nombre character varying(100),
figura character varying(50),
disp1 numeric (6,3),
disp2 numeric (6,3),
disp3 numeric (6,3),
disp4 numeric (6,3),
disp5 numeric (6,3),
disp6 numeric (6,3),
CONSTRAINT zepas_pkey PRIMARY KEY (gid)
);

```

```

SELECT
AddGeometryColumn('public','zepas','geom',4326,'MultiPolygon',2);

```

```

INSERT INTO public.zepas
("habitat", "porcentaje", "area_total", "normativa", "area", "zepa_cod
e", "nombre", "figura", "disp1", "disp2", "disp3", "disp4", "disp5", "dis
p6", "geom") (

```

```
SELECT
"habitat","porcentaje","area_total","normativa","area","zepa_code
","nombre","figura","disp1","disp2","disp3","disp4","disp5","disp
6",ST_Transform(ST_Transform(geom,23030),4326) FROM
peninsula.zepas_p_fin
UNION
SELECT
"habitat","porcentaje","area_total","normativa","area","zepa_code
","nombre","figura","disp1","disp2","disp3","disp4","disp5","disp
6",ST_Transform(ST_Transform(geom,23031),4326) FROM
balears.zepas_b_fin
);
```

```
-----
--Union PPNN de las tres entidades
-----
```

```
--Ha habido un error en la importación de ppnn, los valores de
los campos de los dispositivos se han almacenado como null, son
todos 0.0
-- Se introduce de forma manual los valores 0.0
```

```
CREATE TABLE public.ppnn(
gidserial NOT NULL,
normativa character varying(100),
area numeric,
figura character varying(50),
nombre character varying(50),
disp1 numeric (6,3),
disp2 numeric (6,3),
disp3 numeric (6,3),
disp4 numeric (6,3),
disp5 numeric (6,3),
disp6 numeric (6,3),
CONSTRAINT ppnn_pkey PRIMARY KEY (gid)
);
```

```
SELECT
AddGeometryColumn('public','ppnn','geom',4326,'MultiPolygon',2);
```

```
INSERT INTO public.ppnn
("normativa","area","figura","nombre","disp1","disp2","disp3","di
sp4","disp5","disp6","geom") (
```

```
SELECT
"normativa","area","figura","nombre","disp1","disp2","disp3","dis
p4","disp5","disp6",ST_Transform(ST_Transform(geom,23030),4326)
FROM peninsula.ppns_p
UNION
```

```

SELECT
"normativa","area","figura","nombre","displ","disp2","disp3","dis
p4","disp5","disp6",ST_Transform(ST_Transform(geom,23031),4326)
FROM balears.ppnns_b
);

```

```

-----
--Union BBPPro de las tres entidades
-----

```

```

CREATE TABLE public.bbppro(
gid serial NOT NULL,
normativa character varying(50),
area numeric,
figura character varying(50),
nombre character varying(100),
displ numeric (6,3),
disp2 numeric (6,3),
disp3 numeric (6,3),
disp4 numeric (6,3),
disp5 numeric (6,3),
disp6 numeric (6,3),
CONSTRAINT bbppro_pkey PRIMARY KEY (gid)
);

```

```

SELECT
AddGeometryColumn('public','bbppro','geom',4326,'MultiPolygon',2)
;

```

```

INSERT INTO public.bbppro
("normativa","area","figura","nombre","displ","disp2","disp3","di
sp4","disp5","disp6","geom") (

```

```

SELECT
"normativa","area","figura","nombre","displ","disp2","disp3","dis
p4","disp5","disp6",ST_Transform(ST_Transform(geom,23030),4326)
FROM peninsula.bbppro_p
);

```

```

-----
--Union Micror de las tres entidades
-----

```

```

CREATE TABLE public.micror(
gid serial NOT NULL,
habitat character varying(254),
porcentaje numeric,
area_total numeric,
normativa character varying(50),

```

```
        figura character varying(50),
        nombre character varying(100),
        disp1 numeric (6,3),
        disp2 numeric (6,3),
        disp3 numeric (6,3),
        disp4 numeric (6,3),
        disp5 numeric (6,3),
        disp6 numeric (6,3),
        CONSTRAINT micror_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','micror','geom',4326,'MultiPolygon',2)
;

INSERT INTO public.micror
("habitat","porcentaje","area_total","normativa","figura","nombre",
,"disp1","disp2","disp3","disp4","disp5","disp6","geom") (

SELECT
"habitat","porcentaje","area_total","normativa","figura","nombre",
,"disp1","disp2","disp3","disp4","disp5","disp6",ST_Transform(ST_
Transform(geom,23030),4326) FROM peninsula.micror_p_fin
);

-----
--Union MMNat de las tres entidades
-----

CREATE TABLE public.mmnat(
        gid serial NOT NULL,
        habitat character varying(254),
        porcentaje numeric,
        area_total numeric,
        normativa character varying(50),
        figura character varying(50),
        nombre character varying(240),
        disp1 numeric (6,3),
        disp2 numeric (6,3),
        disp3 numeric (6,3),
        disp4 numeric (6,3),
        disp5 numeric (6,3),
        disp6 numeric (6,3),
        CONSTRAINT mmnat_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','mmnat','geom',4326,'MultiPolygon',2);
```

```

INSERT INTO public.mmnat
("habitat","porcentaje","area_total","normativa","figura","nombre",
,"disp1","disp2","disp3","disp4","disp5","disp6","geom") (

SELECT
"habitat","porcentaje","area_total","normativa","figura","nombre",
,"disp1","disp2","disp3","disp4","disp5","disp6",ST_Transform(ST_
Transform(geom,23030),4326) FROM peninsula.mmnat_p_fin
);

-----
--Union PParNNaT de las tres entidades
-----

CREATE TABLE public.pparnnat(
gid serial NOT NULL,
habitat character varying(254),
porcentaje numeric,
area_total numeric,
normativa character varying(50),
figura character varying(50),
nombre character varying(100),
disp1 numeric (6,3),
disp2 numeric (6,3),
disp3 numeric (6,3),
disp4 numeric (6,3),
disp5 numeric (6,3),
disp6 numeric (6,3),
CONSTRAINT pparnnat_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','pparnnat','geom',4326,'MultiPolygon',
2);

INSERT INTO public.pparnnat
("habitat","porcentaje","area_total","normativa","figura","nombre",
,"disp1","disp2","disp3","disp4","disp5","disp6","geom") (

SELECT
"habitat","porcentaje","area_total","normativa","figura","nombre",
,"disp1","disp2","disp3","disp4","disp5","disp6",ST_Transform(ST_T
ransform(geom,23030),4326) FROM peninsula.pparnnat_p_fin
);

-----
--Union PPNat de las tres entidades
-----

```

```
CREATE TABLE public.ppnat(  
    gid serial NOT NULL,  
    habitat character varying(254),  
    porcentaje numeric,  
    area_total numeric,  
    normativa character varying(50),  
    nombre character varying(100),  
    figura character varying(50),  
    disp1 numeric (6,3),  
    disp2 numeric (6,3),  
    disp3 numeric (6,3),  
    disp4 numeric (6,3),  
    disp5 numeric (6,3),  
    disp6 numeric (6,3),  
    CONSTRAINT ppnat_pkey PRIMARY KEY (gid)  
);  
  
SELECT  
AddGeometryColumn('public','ppnat','geom',4326,'MultiPolygon',2);  
  
INSERT INTO public.ppnat  
("habitat","porcentaje","area_total","normativa","nombre","figura"  
,"disp1","disp2","disp3","disp4","disp5","disp6","geom") (  
  
SELECT  
"habitat","porcentaje","area_total","normativa","nombre","figura"  
,"disp1","disp2","disp3","disp4","disp5","disp6",ST_Transform(ST_  
Transform(geom,23030),4326) FROM peninsula.ppnat_p_fin  
UNION  
SELECT  
"habitat",null,null,"normativa","nombre","figura","disp1","disp2"  
,"disp3","disp4","disp5","disp6",ST_Transform(ST_Transform(geom,2  
3031),4326) FROM baleares.ppnat_b_fin  
UNION  
SELECT  
"habitat","porcentaje","area_total","normativa","nombre","figura"  
,"disp1","disp2","disp3","disp4","disp5","disp6",ST_Transform(ST_  
Transform(geom,32628),4326) FROM canarias.ppnat_c_fin);  
  
-----  
--Union rr_ley_prop de las tres entidades  
-----  
  
CREATE TABLE public.rr_ley_prop(  
    gid serial NOT NULL,  
    normativa character varying(50),  
    area numeric,  
    nombre character varying(100),  
    figura character varying(70),
```

```

        disp1 numeric (6,3),
        disp2 numeric (6,3),
        disp3 numeric (6,3),
        disp4 numeric (6,3),
        disp5 numeric (6,3),
        disp6 numeric (6,3),
        CONSTRAINT rr_ley_prop_pkey PRIMARY KEY (gid)
    );

SELECT
AddGeometryColumn('public','rr_ley_prop','geom',4326,'MultiPolygon',2);

INSERT INTO public.rr_ley_prop
("normativa","area","nombre","figura","disp1","disp2","disp3","disp4","disp5","disp6","geom") (

SELECT
"normativa","area","nombre","figura","disp1","disp2","disp3","disp4","disp5","disp6",ST_Transform(ST_Transform(geom,23030),4326)
FROM peninsula.rr_ley_prop_p
);

-----
--Union RRNN de las tres entidades
-----

CREATE TABLE public.rrnn(
    gid serial NOT NULL,
    habitat character varying(254),
    porcentaje numeric,
    area_total numeric,
    normativa character varying(50),
    nombre character varying(100),
    figura character varying(50),
    disp1 numeric (6,3),
    disp2 numeric (6,3),
    disp3 numeric (6,3),
    disp4 numeric (6,3),
    disp5 numeric (6,3),
    disp6 numeric (6,3),
    CONSTRAINT rrnn_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','rrnn','geom',4326,'MultiPolygon',2);

```



```
INSERT INTO public.rrnn
("habitat", "porcentaje", "area_total", "normativa", "nombre", "figura",
"disp1", "disp2", "disp3", "disp4", "disp5", "disp6", "geom") (

SELECT
"habitat", "porcentaje", "area_total", "normativa", "nombre", "figura",
"disp1", "disp2", "disp3", "disp4", "disp5", "disp6", ST_Transform(ST_
Transform(geom, 23030), 4326) FROM peninsula.ppnat_p_fin
UNION
SELECT
"habitat", null, null, "normativa", "nombre", "figura", "disp1", "disp2",
"disp3", "disp4", "disp5", "disp6", ST_Transform(ST_Transform(geom, 2
3031), 4326) FROM baleares.ppnat_b_fin
);
```

```
-----
--Union ZEPVN de las tres entidades
-----
```

```
CREATE TABLE public.zepvn(
gid serial NOT NULL,
habitat character varying(254),
porcentaje numeric,
area_total numeric,
normativa character varying(50),
area numeric,
nombre character varying(50),
figura character varying(50),
disp1 numeric (6,3),
disp2 numeric (6,3),
disp3 numeric (6,3),
disp4 numeric (6,3),
disp5 numeric (6,3),
disp6 numeric (6,3),
CONSTRAINT zepvn_pkey PRIMARY KEY (gid)
);
```

```
INSERT INTO public.zepvn
("habitat", "porcentaje", "area_total", "normativa", "area", "nombre",
"figura", "disp1", "disp2", "disp3", "disp4", "disp5", "disp6", "geom")
(

SELECT
"habitat", "porcentaje", "area_total", "normativa", "area", "nombre", "
figura", "disp1", "disp2", "disp3", "disp4", "disp5", "disp6", ST_Transf
orm(ST_Transform(geom, 23030), 4326) FROM peninsula.zepvn_p_fin
);
```

```
-----
```

--Union AICP de las tres entidades

```
-----  
  
CREATE TABLE public.aicp(  
    gid serial NOT NULL,  
    nom_habita character varying(50),  
    figura character varying(50),  
    nombre character varying(50),  
    disp1 numeric (6,3),  
    disp2 numeric (6,3),  
    disp3 numeric (6,3),  
    disp4 numeric (6,3),  
    disp5 numeric (6,3),  
    disp6 numeric (6,3),  
    CONSTRAINT aicp_pkey PRIMARY KEY (gid)  
);  
  
SELECT  
AddGeometryColumn('public','aicp','geom',4326,'MultiPolygon',2);  
  
INSERT INTO public.aicp  
("nom_habita","figura","nombre","disp1","disp2","disp3","disp4","d  
isp5","disp6","geom") (  
  
SELECT  
"nom_habita","figura","nombre","disp1","disp2","disp3","disp4","d  
isp5","disp6",ST_Transform(ST_Transform(geom,23030),4326) FROM  
peninsula.aicp_p_fin  
UNION  
SELECT  
"nom_habita","figura","nombre","disp1","disp2","disp3","disp4","d  
isp5","disp6",ST_Transform(ST_Transform(geom,23031),4326) FROM  
baleares.aicp_b_fin  
);
```

--Union AOPN de las tres entidades

```
-----  
  
CREATE TABLE public.aopn(  
    gid serial NOT NULL,  
    nom_habita character varying(50),  
    figura character varying(50),  
    nombre character varying(50),  
    disp1 numeric (6,3),  
    disp2 numeric (6,3),  
    disp3 numeric (6,3),  
    disp4 numeric (6,3),  
    disp5 numeric (6,3),
```

```
        disp6 numeric (6,3),
        CONSTRAINT aopn_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','aopn','geom',4326,'MultiPolygon',2);

INSERT INTO public.aopn
("nom_habita","figura","nombre","disp1","disp2","disp3","disp4","d
isp5","disp6","geom") (

SELECT
"nom_habita","figura","nombre","disp1","disp2","disp3","disp4","d
isp5","disp6",ST_Transform(ST_Transform(geom,23030),4326) FROM
peninsula.aopn_p_fin
UNION
SELECT
"nom_habita","figura","nombre","disp1","disp2","disp3","disp4","d
isp5","disp6",ST_Transform(ST_Transform(geom,23031),4326) FROM
baleares.aopn_b_fin
UNION
SELECT
"nom_habita","figura","nombre","disp1","disp2","disp3","disp4","d
isp5","disp6",ST_Transform(ST_Transform(geom,32628),4326) FROM
canarias.aopn_c_fin
);

-----
--Union IBAs de las tres entidades
-----

CREATE TABLE public.ibas(
    gid serial NOT NULL,
    nom_habita character varying(50),
    figura character varying(50),
    nombre character varying(70),
    disp1 numeric (6,3),
    disp2 numeric(6,3),
    disp3 numeric (6,3),
    disp4 numeric (6,3),
    disp5 numeric (6,3),
    disp6 numeric (6,3),
    CONSTRAINT ibas_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','ibas','geom',4326,'MultiPolygon',2);
```

```

INSERT INTO public.ibas
("nom_habita","figura","nombre","disp1","disp2","disp3","disp4","d
isp5","disp6","geom") (

SELECT
"nom_habita","figura","nombre","disp1","disp2","disp3","disp4","d
isp5","disp6",ST_Transform(ST_Transform(geom,23030),4326) FROM
peninsula.ibas_p_fin
UNION
SELECT
"nom_habita","figura","nombre","disp1","disp2","disp3","disp4","d
isp5","disp6",ST_Transform(ST_Transform(geom,23031),4326) FROM
baleares.ibas_b_fin
UNION
SELECT
"nom_habita","figura","nombre","disp1","disp2","disp3","disp4","d
isp5","disp6",ST_Transform(ST_Transform(geom,32628),4326) FROM
canarias.ibas_c_fin
);

-----
--Union IBM de las tres entidades
-----

CREATE TABLE public.ibm(
gid serial NOT NULL,
nom_habita character varying(50),
figura character varying(50),
nombre character varying(50),
disp1 numeric (6,3),
disp2 numeric (6,3),
disp3 numeric (6,3),
disp4 numeric (6,3),
disp5 numeric (6,3),
disp6 numeric (6,3),
CONSTRAINT ibm_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','ibm','geom',4326,'MultiPolygon',2);

INSERT INTO public.ibm
("nom_habita","figura","nombre","disp1","disp2","disp3","disp4","
disp5","disp6","geom") (

SELECT
"nom_habita","figura","nombre","disp1","disp2","disp3","disp4","d
isp5","disp6",ST_Transform(ST_Transform(geom,23030),4326) FROM
peninsula.ibm_p_fin

```

```
UNION
SELECT
null,"figura",null,"disp1","disp2","disp3","disp4","disp5","disp6
",ST_Transform(ST_Transform(geom,23031),4326) FROM baleares.ibm_b
UNION
SELECT
"nom_habita","figura",null,"disp1","disp2","disp3","disp4","disp5
","disp6",ST_Transform(ST_Transform(geom,32628),4326) FROM
canarias.ibm_c_fin
);
```

--Union ZMES de las tres entidades

```
CREATE TABLE public.zmes(
gid serial NOT NULL,
nom_habita character varying(50),
figura character varying(50),
nombre character varying(50),
disp1 numeric (6,3),
disp2 numeric (6,3),
disp3 numeric (6,3),
disp4 numeric (6,3),
disp5 numeric (6,3),
disp6 numeric (6,3),
CONSTRAINT zmes_pkey PRIMARY KEY (gid)
);
```

```
SELECT
AddGeometryColumn('public','zmes','geom',4326,'MultiPolygon',2);
```

```
INSERT INTO public.zmes
("nom_habita","figura","nombre","disp1","disp2","disp3","disp4","d
isp5","disp6","geom") (
```

```
SELECT
"nom_habita","figura","nombre","disp1","disp2","disp3","disp4","d
isp5","disp6",ST_Transform(ST_Transform(geom,23030),4326) FROM
peninsula.zmes_p_fin
UNION
SELECT
"nom_habita","figura","nombre","disp1","disp2","disp3","disp4","d
isp5","disp6",ST_Transform(ST_Transform(geom,32628),4326) FROM
canarias.zmes_c_fin
);
```

--Union Cetaceos de las tres entidades

```
-----  
  
CREATE TABLE public.cetaceos(  
    gid serial NOT NULL,  
    nom_ruta character varying(120),  
    tipo_area character varying(100),  
    disp1 numeric (6,3),  
    disp2 numeric (6,3),  
    disp3 numeric (6,3),  
    disp4 numeric (6,3),  
    disp5 numeric (6,3),  
    disp6 numeric (6,3),  
    CONSTRAINT cetaceos_pkey PRIMARY KEY (gid)  
);  
  
SELECT  
AddGeometryColumn('public','cetaceos','geom',4326,'MultiPolygon',  
2);  
  
INSERT INTO public.cetaceos  
("nom_ruta","tipo_area","disp1","disp2","disp3","disp4","disp5","  
disp6","geom") (  
  
SELECT  
"nom_ruta","tipo_area","disp1","disp2","disp3","disp4","disp5","d  
isp6",ST_Transform(ST_Transform(geom,23030),4326) FROM  
peninsula.cetaceos_p  
UNION  
SELECT  
"nom_ruta","tipo_area","disp1","disp2","disp3","disp4","disp5","d  
isp6",ST_Transform(ST_Transform(geom,23031),4326) FROM  
baleares.cetaceo_b  
);  
  
-----  
-- Unión Acuicultura_buf  
-----  
  
CREATE TABLE public.acuicultura_buf(  
    gid serial NOT NULL,  
    buff_dist numeric,  
    disp2 numeric (6,3),  
    disp3 numeric (6,3),  
    disp5 numeric (6,3),  
    disp6 numeric (6,3),  
    em_nombre character varying(254),  
    in_nombre character varying(200),  
    CONSTRAINT acuicultura_buf_pkey PRIMARY KEY (gid)  
);
```

```
SELECT
AddGeometryColumn('public','acuicultura_buf','geom',4326,'MultiPolygon',2);
```

```
INSERT INTO public.acuicultura_buf
("buff_dist","disp2","disp3","disp5","disp6","em_nombre","in_nombre","geom") (
```

```
SELECT
"buff_dist","disp2","disp3","disp5","disp6","em_nombre","in_nombre",ST_Transform(ST_Transform(geom,23030),4326) FROM
peninsula.acuicultura_buf_p
```

```
UNION
```

```
SELECT
"buff_dist","disp2","disp3","disp5","disp6","em_nombre","in_nombre",ST_Transform(ST_Transform(geom,23031),4326) FROM
baleares.acu_buf_b
```

```
UNION
```

```
SELECT
"buff_dist","disp2","disp3","disp5","disp6","em_nombre","in_nombre",ST_Transform(ST_Transform(geom,32628),4326) FROM
canarias.acuicultura_buf_c
```

```
);
```

```
-----
-- Unión Acuicultura (poligono)
-----
```

```
CREATE TABLE public.acuicultura(
gid serial NOT NULL,
disp2 numeric (6,3),
disp3 numeric (6,3),
disp5 numeric (6,3),
disp6 numeric (6,3),
titular character varying(70),
nom_latin character varying(50),
nom_especi character varying(50),
fuente character varying(50),
tipo character varying(50),
ubicacio character varying(50),
codi_inst integer,
CONSTRAINT acuicultura_pkey PRIMARY KEY (gid)
);
```

```
SELECT
AddGeometryColumn('public','acuicultura','geom',4326,'MultiPolygon',2);
```

```

INSERT INTO
public.acuicultura("disp2","disp3","disp5","disp6","titular","nom
_latin","nom_especi","fuente","tipo","ubicacio","codi_inst","geom
") (
SELECT
"disp2","disp3","disp5","disp6","titular","nom_latin","nom_especi
","fuente","tipo","ubicacio","codi_inst",ST_Transform(ST_Transfor
m(geom,23030),4326) FROM peninsula.acuicultura_p
);

-----
-- Unión Almadrabas_buf
-----

CREATE TABLE public.almadrabas_buf(
gid serial NOT NULL,
buff_dist numeric,
disp2 numeric (6,3),
disp3 numeric (6,3),
disp5 numeric (6,3),
disp6 numeric (6,3),
localiza character varying(50),
CONSTRAINT almadrabas_buf_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','almadrabas_buf','geom',4326,'MultiPol
ygon',2);

INSERT INTO
public.almadrabas_buf("buff_dist","disp2","disp3","disp5","disp6"
,"localiza","geom") (

SELECT
"buff_dist","disp2","disp3","disp5","disp6","localiza",ST_Transfo
rm(ST_Transform(geom,23030),4326) FROM peninsula.almadraba_buf_p
);

-----
-- Unión Almadrabas (poligono)
-----
-- Ha habido un error en la importación de almadrabas_p, los
valores de los campos se han importado como NULL
-- Se ha rellenado la tabla de forma manual porque ha sido
imposible hacer la imoportación correctamente

CREATE TABLE public.almadrabas(
gid serial NOT NULL,
disp2 numeric (6,3),

```



```
        disp3 numeric (6,3),
        disp5 numeric (6,3),
        disp6 numeric (6,3),
        id integer,
        CONSTRAINT almadrabas_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','almadrabas','geom',4326,'MultiPolygon'
,2);

INSERT INTO
public.almadrabas("disp2","disp3","disp5","disp6","id","geom") (

SELECT
"disp2","disp3","disp5","disp6","id",ST_Transform(ST_Transform(ge
om,23030),4326) FROM peninsula.almadrabas_p
);

-----
-- Unión Arrecifes
-----

CREATE TABLE public.arrecifes(
        gid serial NOT NULL,
        disp2 numeric (6,3),
        disp3 numeric (6,3),
        disp5 numeric (6,3),
        disp6 numeric (6,3),
        fuente character varying(50),
        titular character varying(100),
        nombre character varying(100),
        clave character varying(5),
        CONSTRAINT arrecifes_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','arrecifes','geom',4326,'MultiPolygon'
,2);

INSERT INTO
public.arrecifes("disp2","disp3","disp5","disp6","fuente","titula
r","nombre","clave","geom") (

SELECT
"disp2","disp3","disp5","disp6","fuente","titular","nombre","clav
e",ST_Transform(ST_Transform(geom,23030),4326) FROM
peninsula.arrecifes_p
UNION
```

```

SELECT
"disp2","disp3","disp5","disp6",null,"titular","nombre","clave",S
T_Transform(ST_Transform(geom,23031),4326) FROM
balears.arrecifes_b
UNION
SELECT
"disp2","disp3","disp5","disp6","fuente","titular","nombre","clav
e",ST_Transform(ST_Transform(geom,32628),4326) FROM
canarias.arrecifes_can_wgs84h28
);

```

```

-----
-- Unión Caladeros
-----

```

```

CREATE TABLE public.caladeros(
gid serial NOT NULL,
disp2 numeric (6,3),
disp3 numeric (6,3),
disp5 numeric (6,3),
disp6 numeric (6,3),
fuente character varying(50),
nombre_cal character varying(200),
artes_pesc character varying(200),
especies character varying(200),
CONSTRAINT caladeros_pkey PRIMARY KEY (gid)
);

```

```

SELECT
AddGeometryColumn('public','caladeros','geom',4326,'MultiPolygon'
,2);

```

```

INSERT INTO
public.caladeros("disp2","disp3","disp5","disp6","fuente","nombre
_cal","artes_pesc","especies","geom") (

```

```

SELECT
"disp2","disp3","disp5","disp6","fuente","nombre_cal","artes_pesc
","especies",ST_Transform(ST_Transform(geom,23030),4326) FROM
peninsula.caladeros_p
);

```

```

-----
-- Unión Viveros (poligono)
-----

```

```

CREATE TABLE public.viveros(
gid serial NOT NULL,
disp2 numeric (6,3),

```

```
        disp3 numeric (6,3),
        disp5 numeric (6,3),
        disp6 numeric (6,3),
        id integer,
        CONSTRAINT viveros_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','viveros','geom',4326,'MultiPolygon',2
);

INSERT INTO
public.viveros("disp2","disp3","disp5","disp6","id","geom") (

SELECT
"disp2","disp3","disp5","disp6","id",ST_Transform(ST_Transform(ge
om,23030),4326) FROM peninsula.viveros_p
UNION
SELECT
"disp2","disp3","disp5","disp6","id",ST_Transform(ST_Transform(ge
om,23031),4326) FROM baleares.viveros_refug_b
);

-----
-- Unión Viveros_buf
-----

CREATE TABLE public.viveros_buf(
    gid serial NOT NULL,
    buff_dist numeric,
    disp2 numeric (6,3),
    disp3 numeric (6,3),
    disp5 numeric (6,3),
    disp6 numeric (6,3),
    tipo character varying(50),
    CONSTRAINT viveros_buf_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','viveros_buf','geom',4326,'MultiPolygo
n',2);

INSERT INTO
public.viveros_buf("buff_dist","disp2","disp3","disp5","disp6","t
ipo","geom") (

SELECT
"buff_dist","disp2","disp3","disp5","disp6","tipo",ST_Transform(S
T_Transform(geom,23030),4326) FROM peninsula.viveros_buf_p
```

```
);

-----
-- Unión zpips - zonas protegidas de interés pesquero
-----

CREATE TABLE public.zpips(
    gid serial NOT NULL,
    disp2 numeric (6,3),
    disp3 numeric (6,3),
    disp5 numeric (6,3),
    disp6 numeric (6,3),
    nombre character varying(200),
    CONSTRAINT zpips_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','zpips','geom',4326,'MultiPolygon',2);

INSERT INTO
public.zpips("disp2","disp3","disp5","disp6","nombre","geom") (

SELECT
"disp2","disp3","disp5","disp6","nombre",ST_Transform(ST_Transform
m(geom,23030),4326) FROM peninsula.zpips_p
);

-----
-- Unión Piscifactoria (poligono)
-----

CREATE TABLE public.piscifactoria(
    gid serial NOT NULL,
    disp2 numeric (6,3),
    disp3 numeric (6,3),
    disp5 numeric (6,3),
    disp6 numeric (6,3),
    fuente character varying(50),
    id integer,
    CONSTRAINT piscifactoria_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','piscifactoria','geom',4326,'MultiPoly
gon',2);

INSERT INTO
public.piscifactoria("disp2","disp3","disp5","disp6","fuente","id
","geom") (
```

```
SELECT
"disp2","disp3","disp5","disp6","fuente","id",ST_Transform(ST_Tra
nsform(geom,23030),4326) FROM peninsula.piscifactoria_p
UNION
SELECT
"disp2","disp3","disp5","disp6",null,null,ST_Transform(ST_Transfo
rm(geom,32628),4326) FROM canarias.piscifactoria_can_wgs84h28
);
```

```
-----
-- Unión piscifactoria_buf
-----
```

```
CREATE TABLE public.piscifactoria_buf(
    gid serial NOT NULL,
    buff_dist numeric,
    disp2 numeric (6,3),
    disp3 numeric (6,3),
    disp5 numeric (6,3),
    disp6 numeric (6,3),
    observacio character varying(50),
    tipo character varying(100),
    nombrepres character varying(254),
    CONSTRAINT piscifactoria_buf_pkey PRIMARY KEY (gid)
);
```

```
SELECT
AddGeometryColumn('public','piscifactoria_buf','geom',4326,'Multi
Polygon',2);
```

```
INSERT INTO
public.piscifactoria_buf("buff_dist","disp2","disp3","disp5","dis
p6","observacio","tipo","nombrepres","geom") (
```

```
SELECT
"buff_dist","disp2","disp3","disp5","disp6","observacio","tipo","
nombrepres",ST_Transform(ST_Transform(geom,23030),4326) FROM
peninsula.piscifactoria_buf_p
);
```

```
-----
-- Unión RRMM - reservas marinas
-----
```

```
CREATE TABLE public.rrmm(
    gid serial NOT NULL,
    disp2 numeric (6,3),
    disp3 numeric (6,3),
```

```

        disp5 numeric (6,3),
        disp6 numeric (6,3),
        fuente character varying(50),
        reserva_ma character varying(254),
        reserva character varying(50),
        CONSTRAINT rrrmm_pkey PRIMARY KEY (gid)
    );

SELECT
AddGeometryColumn('public','rrmm','geom',4326,'MultiPolygon',2);

INSERT INTO
public.rrmm("disp2","disp3","disp5","disp6","fuente","reserva_ma"
,"reserva","geom") (

SELECT
"disp2","disp3","disp5","disp6","fuente","reserva_ma","reserva",S
T_Transform(ST_Transform(geom,23030),4326) FROM peninsula.rrmm_p
UNION
SELECT
"disp2","disp3","disp5","disp6",null,"reserva_ma","reserva",ST_Tr
ansform(ST_Transform(geom,23031),4326) FROM baleares.rrmm_b
UNION
SELECT
"disp2","disp3","disp5","disp6","fuente","reserva_ma","reserva",S
T_Transform(ST_Transform(geom,32628),4326) FROM
canarias.rrmm_can_wgs84h28
);

-----
-- Unión cria_moluscos
-----

CREATE TABLE public.cria_moluscos(
    gid serial NOT NULL,
    disp2 numeric (6,3),
    disp3 numeric (6,3),
    disp5 numeric (6,3),
    disp6 numeric (6,3),
    fuente character varying(50),
    especies character varying(254),
    ubicacion character varying(254),
    clave character varying(10),
    CONSTRAINT cria_moluscos_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','cria_moluscos','geom',4326,'MultiPoly
gon',2);

```

```
INSERT INTO
public.cria_moluscos("disp2","disp3","disp5","disp6","fuente","es
pecies","ubicacion","clave","geom") (

SELECT
"disp2","disp3","disp5","disp6","fuente","especies","ubicación","
clave",ST_Transform(ST_Transform(geom,23030),4326) FROM
peninsula.cria_molusc_p
UNION
SELECT
"disp2","disp3","disp5","disp6",null,"especies","ubicación","clav
e",ST_Transform(ST_Transform(geom,23031),4326) FROM
balears.cria_moluscos_b
);

-----
-- Unión iba_recom
-----

CREATE TABLE public.iba_recom(
gid serial NOT NULL,
nom_habita character varying(50),
disp2 numeric (6,3),
disp3 numeric (6,3),
disp5 numeric (6,3),
disp6 numeric (6,3),
figura character varying(50),
nombre character varying(45),
CONSTRAINT iba_recom_pkey PRIMARY KEY (gid)
);

SELECT
AddGeometryColumn('public','iba_recom','geom',4326,'MultiPolygon'
,2);

INSERT INTO
public.iba_recom("nom_habita","disp2","disp3","disp5","disp6","fi
gura","nombre","geom") (

SELECT
"nom_habita","disp2","disp3","disp5","disp6","figura","nombre",ST
_Transform(ST_Transform(geom,23030),4326) FROM
peninsula.iba_recom_p_fin
UNION
SELECT
"nom_habita","disp2","disp3","disp5","disp6","figura","nombre",ST
_Transform(ST_Transform(geom,32628),4326) FROM
canarias.iba_recom_c_fin
```

) ;

Apéndice

G

FICHERO .BAT DE LA UNIÓN DE CAPAS RASTER INFORMATIVAS

```
@echo off

REM unión y reproyección de capa raster HsMedia de las tres
entidades

SET GISRC=C:\Users\tregmm\.grassrc6_peninsula
g.region vect=lin_costa_p ewres=300 nsres=300
for %%f in ("C:\Users\tregmm\Desktop\OCEAN LIDER_25062012\CAPAS
INFORMATIVAS\P\02_IT\RE\OL\HsMedia*img") do (r.in.gdal -o
input="%%f" output=%%~nf)
setlocal enabledelayedexpansion
set var=
for %%f in ("C:\Users\tregmm\Desktop\OCEAN LIDER_25062012\CAPAS
INFORMATIVAS\P\02_IT\RE\OL\HsMedia*img") do (set var=%%~nf,!var!)
r.patch input=%var% output=HsMediaUnion_p

REM g.mlist type=%var% sep=, pat="HsMedia*"

SET GISRC=C:\Users\tregmm\.grassrc6_baleares
g.region vect=lin_costa_b ewres=300 nsres=300
for %%f in ("C:\Users\tregmm\Desktop\OCEAN LIDER_25062012\CAPAS
INFORMATIVAS\B\02_IT\RE\OL\HsMedia*img") do (r.in.gdal -o
input="%%f" output=%%~nf)
setlocal enabledelayedexpansion
set var=
for %%f in ("C:\Users\tregmm\Desktop\OCEAN LIDER_25062012\CAPAS
INFORMATIVAS\B\02_IT\RE\OL\HsMedia*img") do (set var=%%~nf,!var!)
r.patch input=%var% output=HsMediaUnion_b

SET GISRC=C:\Users\tregmm\.grassrc6_canarias
g.region vect=lin_costa_c ewres=300 nsres=300
```

```
for %%f in ("C:\Users\tregmm\Desktop\OCEAN LIDER_25062012\CAPAS
INFORMATIVAS\C\02_IT\RE\OL\HsMedia*img") do (r.in.gdal -o
input="%%f" output=%%~nf)
setlocal enabledelayedexpansion
set var=
for %%f in ("C:\Users\tregmm\Desktop\OCEAN LIDER_25062012\CAPAS
INFORMATIVAS\C\02_IT\RE\OL\HsMedia*img") do (set var=%%~nf,!var!)
r.patch input=%var% output=HsMediaUnion_c

SET GISRC=C:\Users\tregmm\.grassrc6_wgs84
REM important rows and cols
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300
rows=3224 cols=3453
r.proj input=HsMediaUnion_p location=Peninsula mapset=DSS
r.proj input=HsMediaUnion_b location=Baleares mapset=DSS
r.proj input=HsMediaUnion_c location=Canarias mapset=DSS

r.patch input=HsMediaUnion_p,HsMediaUnion_b,HsMediaUnion_c
output=HsMediaUnion

r.out.gdal input=HsMediaUnion
output="C:\Users\tregmm\Desktop\Union\unionhsmedia.tif"

REM viento anual

SET GISRC=C:\Users\tregmm\.grassrc6_peninsula
g.region vect=lin_costa_p ewres=300 nsres=300
r.in.gdal -o
input="C:\Users\tregmm\Desktop\BBDD_Ocean_Lider\P\02_Capas\02_IT\
RE\VI\Viento_Anual.img" output=Viento_Anual_p

SET GISRC=C:\Users\tregmm\.grassrc6_baleares
g.region vect=lin_costa_b ewres=300 nsres=300
r.in.gdal -o
input="C:\Users\tregmm\Desktop\BBDD_Ocean_Lider\B\02_Capas\02_IT\
RE\VI\Viento_Anual.img" output=Viento_Anual_b

SET GISRC=C:\Users\tregmm\.grassrc6_canarias
g.region vect=lin_costa_c ewres=300 nsres=300
r.in.gdal -o
input="C:\Users\tregmm\Desktop\BBDD_Ocean_Lider\C\02_Capas\02_IT\
RE\VI\viento_anual.img" output=Viento_Anual_c

SET GISRC=C:\Users\tregmm\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300
rows=3224 cols=3453
r.proj input=Viento_Anual_p location=Peninsula mapset=DSS
r.proj input=Viento_Anual_b location=Baleares mapset=DSS
r.proj input=Viento_Anual_c location=Canarias mapset=DSS
```

```
r.patch input=Viento_Anual_p,Viento_Anual_b,Viento_Anual_c
output=Viento_Anual
```

```
r.out.gdal input=Viento_Anual
output="C:\Users\tregmm\Desktop\Union\vientoanual.tif"
```

REM Ha habido un problema con el mapset DSS en la location Peninsula, se ha construido otro mapset DSS2 y es el que vamos a utilizar

REM raster marea media viva

```
SET GISRC=C:\Users\tregmm\.grassrc6_peninsula
g.region vect=lin_costa_p ewres=300 nsres=300
r.in.gdal -o input="C:\Users\tregmm\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\P\02_IT\RE\CO\medvivmed.img"
output=medvivmed_p
r.in.gdal -o input="C:\Users\tregmm\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\P\02_IT\RE\CO\medvivcant.img"
output=medvivcant_p
r.patch input=medvivmed_p,medvivcant_p output=medviv_p
```

```
SET GISRC=C:\Users\tregmm\.grassrc6_baleares
g.region vect=lin_costa_b ewres=300 nsres=300
r.in.gdal -o input="C:\Users\tregmm\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\B\02_IT\RE\CO\medvivmed.img"
output=medvivmed_b
```

```
SET GISRC=C:\Users\tregmm\.grassrc6_canarias
g.region vect=lin_costa_c ewres=300 nsres=300
r.in.gdal -o input="C:\Users\tregmm\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\C\02_IT\RE\CO\medvivcan.img"
output=medvivcan_c
```

```
SET GISRC=C:\Users\tregmm\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300
rows=3224 cols=3453
r.proj input=medviv_p location=Peninsula mapset=DSS2
r.proj input=medvivmed_b location=Baleares mapset=DSS
r.proj input=medvivcan_c location=Canarias mapset=DSS
```

```
r.patch input=medviv_p,medvivmed_b,medvivcan_c output=medviv
```

```
r.out.gdal input=medviv
output="C:\Users\tregmm\Desktop\Union\medviv.tif"
```

REM raster marea media muerta

```
SET GISRC=C:\Users\tregmm\.grassrc6_peninsula
g.region vect=lin_costa_p ewres=300 nsres=300
r.in.gdal -o input="C:\Users\tregmm\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\P\02_IT\RE\CO\medmuemed1.img"
output=medmuemed_p
r.in.gdal -o input="C:\Users\tregmm\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\P\02_IT\RE\CO\medmuecant1.img"
output=medmuecant_p
r.patch input=medmuemed_p,medmuecant_p output=medmue_p
```

```
SET GISRC=C:\Users\tregmm\.grassrc6_baleares
g.region vect=lin_costa_b ewres=300 nsres=300
r.in.gdal -o input="C:\Users\tregmm\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\B\02_IT\RE\CO\medmuemed1.img"
output=medmuemed_b
```

```
SET GISRC=C:\Users\tregmm\.grassrc6_canarias
g.region vect=lin_costa_c ewres=300 nsres=300
r.in.gdal -o input="C:\Users\tregmm\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\C\02_IT\RE\CO\medmuecan1.img"
output=medmuecan_c
```

```
SET GISRC=C:\Users\tregmm\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300
rows=3224 cols=3453
r.proj input=medmue_p location=Peninsula mapset=DSS2
r.proj input=medmuemed_b location=Baleares mapset=DSS
r.proj input=medmuecan_c location=Canarias mapset=DSS
```

```
r.patch input=medmue_p,medmuemed_b,medmuecan_c output=medmue
```

```
r.out.gdal input=medmue
output="C:\Users\tregmm\Desktop\Union\medmue.tif"
```

REM raster potencia media

```
SET GISRC=C:\Users\tregmm\.grassrc6_peninsula
g.region vect=lin_costa_p ewres=300 nsres=300
r.in.gdal -o input="C:\Users\tregmm\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\P\02_IT\RE\OL\PwMEDIA_Med.img"
output=PwMed_p
r.in.gdal -o input="C:\Users\tregmm\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\P\02_IT\RE\OL\PwMEDIA_Can.img"
output=PwCan_p
r.patch input=PwMed_p,PwCan_p output=Pw_p
```

```
SET GISRC=C:\Users\tregmm\.grassrc6_baleares
g.region vect=lin_costa_b ewres=300 nsres=300
```

Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

```
r.in.gdal -o input="C:\Users\tregmm\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\B\02_IT\RE\OL\PwMEDIA_B.img"
output=Pw_b

SET GISRC=C:\Users\tregmm\.grassrc6_canarias
g.region vect=lin_costa_c ewres=300 nsres=300
r.in.gdal -o input="C:\Users\tregmm\Desktop\OCEAN
LIDER_25062012\CAPAS INFORMATIVAS\C\02_IT\RE\OL\PwMEDIA_C.img"
output=Pw_c

SET GISRC=C:\Users\tregmm\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300
rows=3224 cols=3453
r.proj input=Pw_p location=Peninsula mapset=DSS2
r.proj input=Pw_b location=Baleares mapset=DSS
r.proj input=Pw_c location=Canarias mapset=DSS

r.patch input=Pw_p,Pw_b,Pw_c output=pw

r.out.gdal input=pw output="C:\Users\tregmm\Desktop\Union\pw.tif"

REM media lineal ponderada de medmue y medviv

SET GISRC=C:\Users\tregmm\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300
rows=3224 cols=3453
r.mapcalc "mareamedia = 0.5 * medmue + 0.5 * medviv"
r.out.gdal input=mareamedia
output="C:\Users\tregmm\Desktop\Union\mareamedia.tif"
```


Apéndice

H

INSTALACIÓN DEL COMPONENTE XMTOOLS DE EXCEL Y PASOS PARA LA TRANSFORMACIÓN DE UNA TABLA EXCEL A XML

En este apartado se van a mencionar los pasos para la instalación del componente **XmlTools**, lo que nos permitirá transformar un rango de valores de una tabla Excel a una lista XML.

En primer lugar, nos descargamos el complemento proporcionado por Microsoft en el siguiente enlace y hacemos la instalación del ejecutable:

<http://www.microsoft.com/en-us/download/details.aspx?id=3108>

Posteriormente, nos ubicamos en una hoja Excel de Microsoft Excel, y seleccionamos '*Archivo->Opciones*'. En el cuadro de diálogo de Opciones, nos situamos en '*Complementos*' y desglosamos el desplegable '*Administrar*'. En este desplegable seleccionamos '*Complementos de Excel*' y pulsamos el botón '*Ir*'. Comprobamos que la casilla de verificación '*XmlTools*' está seleccionada y damos a '*Aceptar*'.

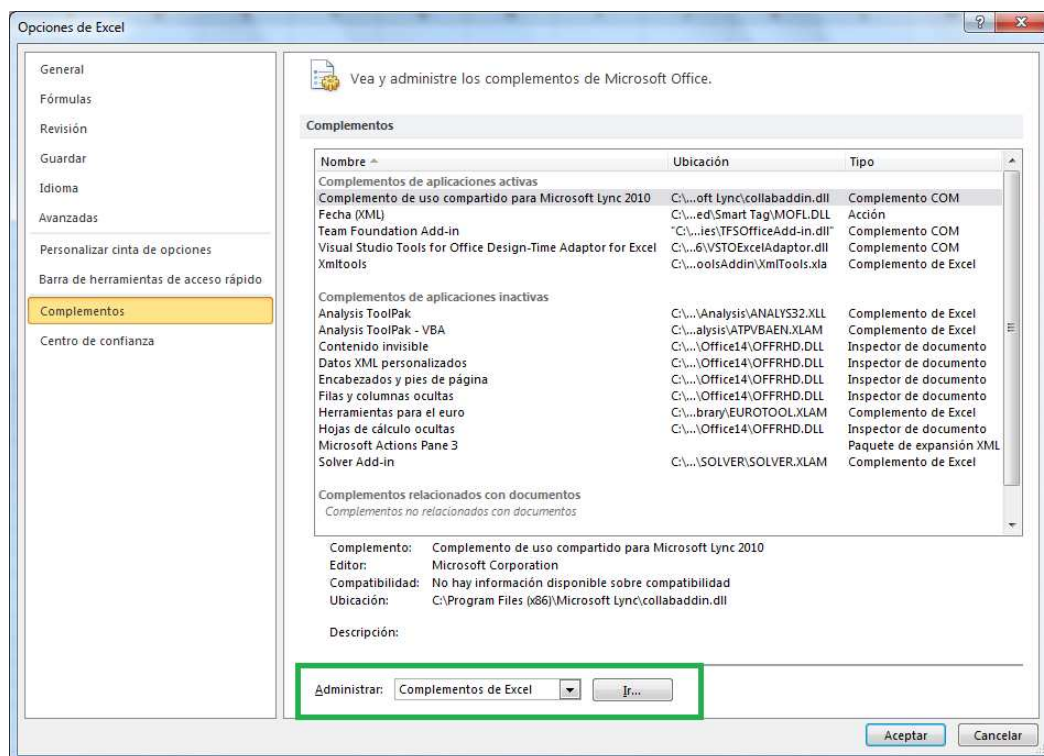


Figura 182: Cuadro de dialogo de Opciones de Microsoft Excel para la instalación de complementos

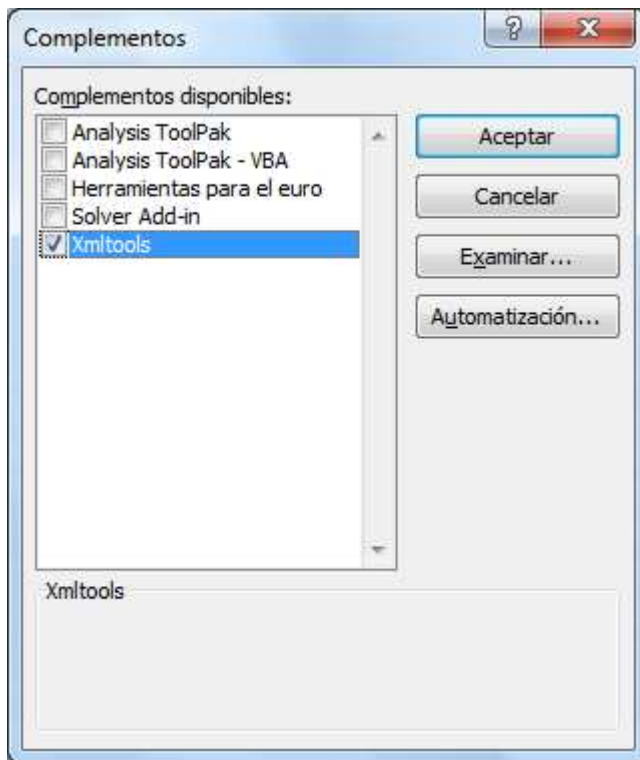


Figura 183: Cuadro de dialogo de Complementos Excel disponibles

De forma predeterminada, este archivo se almacena en la siguiente carpeta del disco duro: `\Office Samples\OfficeExcel2003XMLToolsAddin`.

En este momento veremos una nueva pestaña en el menú de comandos de Excel llamado 'Complementos'. Debe aparecer un nuevo comando del menú 'XML Tools'.

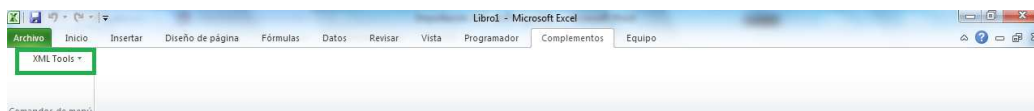


Figura 184: Menú de comandos de Microsoft Excel

Para convertir una tabla Excel a una tabla XML hacemos uso de este nuevo complemento, pulsamos en 'Convert range to XML List'.

La hoja Excel debe contener los datos para el archivo de datos XML y el archivo del esquema XML. Podemos asignar la primera fila a los nombres que tomarán estos atributos en el esquema y en las siguientes filas los datos del archivo de datos. Los datos deben estar en un formato

tabular de columnas y filas (denominados también datos sin formato). Esta herramienta también nos permite construir el archivo de datos XML y el archivo del esquema XML sin especificar la primera fila con el nombre de los atributos, se crearán nombres por defecto en el esquema.

Nos aparecerá un nuevo cuadro de dialogo donde tendremos que indicar el rango de valores de la tabla Excel para que construya el archivo de datos XML y el archivo del esquema XML. Indicamos también si la primera fila contiene nombre de columnas o no, y pulsamos 'Ok'.

Nos aparecerá una nueva ventana 'Microsoft Visual Basic para Aplicaciones' y nos indicará un error de compilación "No se ha definido el tipo definido por el usuario". Este error se ha solventado pulsando en 'Aceptar' y borrando el número "50" de todas las apariciones de "msxml2.DOMDocument50" en el código. Para solventarlo pulsamos F5 y borramos todas las apariciones de "50". Se ha consultado la página [Microsoft b, 2012].

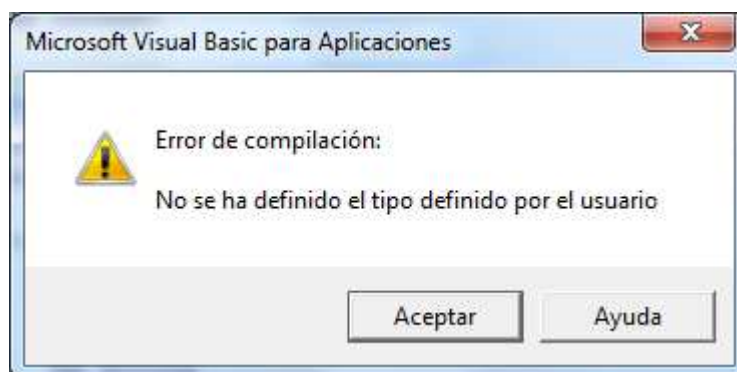


Figura 185: Error de compilación del complemento XmlTools

Una vez solventado los errores de compilación, tenemos a nuestra disposición el esquema XML de la tabla Excel. Si queremos hacer una previsualización de este esquema tenemos que activar la pestaña 'Programador' del menú de comandos. Para hacer visible esta pestaña volvemos a 'Archivo->Opciones', seleccionamos 'Personalizar cinta de opciones', en el desplegable de 'Comandos disponibles en' seleccionamos 'Comandos más utilizados' y activamos la ficha 'Programador'.

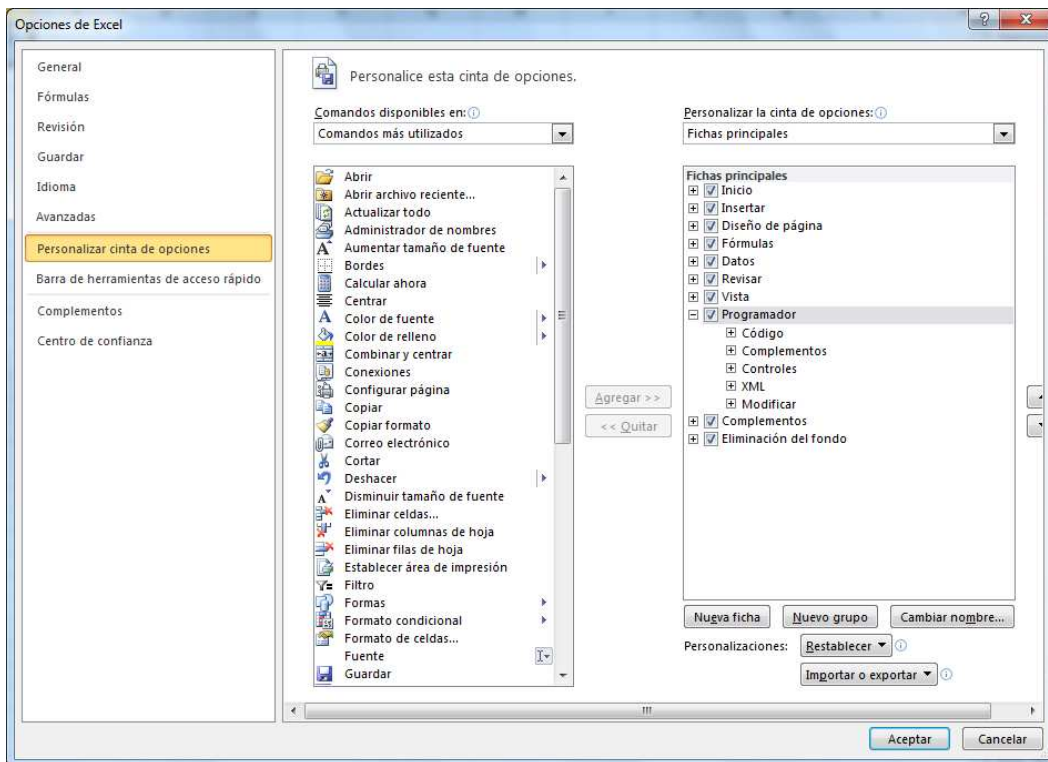


Figura 186: Cuadro de dialogo de Opciones de Microsoft Excel para activar la pestaña 'Programador'

Ahora podemos visualizar el esquema yendo a la pestaña 'Programador' del menú de comandos de Microsoft Excel, pulsamos en 'Origen' y en el menú lateral podemos visualizar el esquema que tendrá el XML de la tabla Excel. Se puede cambiar los nombres de las etiquetas pulsando el botón 'Asignaciones XML'.

Una vez tengamos el esquema y datos XML, podemos guardar el XML asociado a la tabla Excel. Pulsamos en 'Archivo->Guardar como', indicamos dónde queremos que se almacene, el nombre del archivo XML y seleccionamos el tipo a 'Datos XML'.

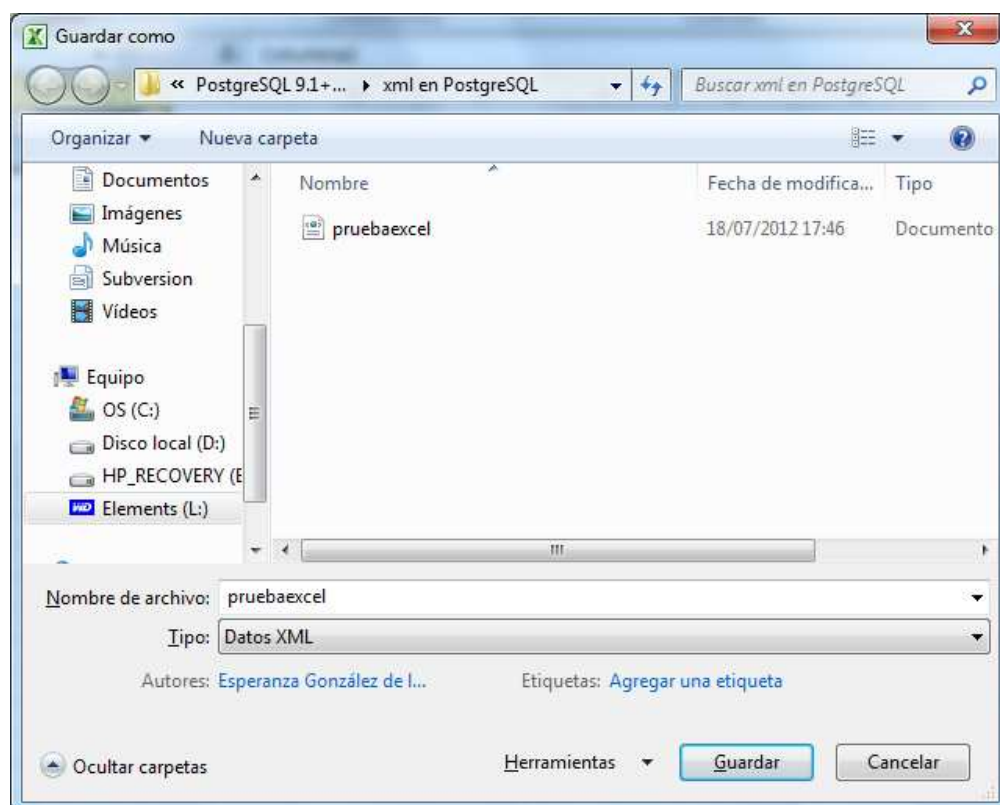


Figura 187: Cuadro de dialogo para almacenar una tabla Excel como archivo XML en Microsoft Excel

Apéndice

I









VALIDACIÓN Y PRUEBAS COMPLETA

1. Resultado de las operaciones de los criterios y subcriterios propuestos para la herramienta DSS

A continuación se muestran los resultados de las pruebas realizadas sobre los criterios/subcriterios propuestos para la herramienta OceanLider. Las pruebas se han realizado de forma unitaria para cada criterio/subcriterio y se han aplicado sobre la costa y zona marítima de Andalucía.

Se han utilizado dos paletas de colores para la visualización de resultados con el objetivo de seleccionar la más adecuada. Las paletas de colores utilizadas para mostrar los resultados son:

➤ **Paleta de colores 1:**

COLOR	RANGO	RGB	TEXTO
	0	255, 0, 0	No Apta
	{0, 0.1}	254, 235, 226	Alta Restricción
	{0.1, 0.2}	252, 219, 192	
	{0.2, 0.3}	249, 178, 119	
	{0.3, 0.4}	247, 150, 70	
	{0.4, 0.5}	255, 255, 159	Media Restricción
	{0.5, 0.6}	248, 242, 0	
	{0.6, 0.7}	220, 255, 209	Baja Restricción
	{0.7, 0.8}	162, 255, 133	
	{0.8, 0.9}	97, 255, 47	
	{0.9, 1}	0, 208, 134	
	1	0, 176, 80	Apta
	-1	204, 102, 0	Tierra

➤ **Paleta de colores 2:**

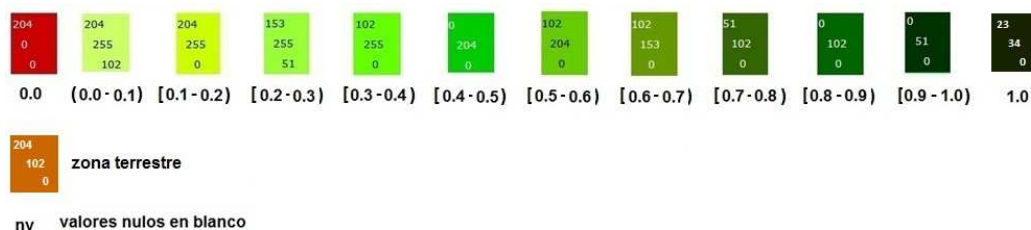


Figura 188: Paleta de colores 2

1.1. Criterios físicos

- **Distancia a la costa.** De exclusión parcial. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa (valor A, valor B, valor C y valor D)

Los valores de los parámetros introducidos en la interfaz:

Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km. ⓘ

Valor A	8
Valor B	9
Valor C	20
Valor D	25
Peso del subcriterio:	0.0

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el criterio.

```

/* ***** OPERACIONES DE CONFIGURACIÓN ***** */

//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Balears
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_272
//Rasterización por valor de una capa vectorial
v.to.rast input=R_272 layer=1 type=point,line,area output=R_273 use=val
value=1 rows=4096
    
```

```
//Creación de un mapa de distancias
r.grow.distance input=R_273 distance=R_274 metric=Euclidean
//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_275
//Rasterización por valor de una capa vectorial
v.to.rast input=R_275 layer=1 type=point,line,area output=R_276 use=val
value=0 rows=4096
//Establecimiento de un valor para los valores nulos
r.null map=R_276 null=1
//Multiplicación de dos capas raster
r.mapcalc "R_277 = R_274 * R_276 "
//Multiplicación de una capa raster por un escalar
r.mapcalc "R_278 = R_277 * 100"
//Operación difusa
r.mapcalc "R_279 = if(isnull( R_278 ) , null(), if((R_278 <= 8 || R_278
>= 25), 0.0, if((R_278 > 8 && R_278 < 9), eval((R_278 - 8)/(9 - 8 )),
if((R_278 > 20 && R_278 < 25 ), eval((25 - R_278 )/(25 - 20)), 1.0)))"
//Establecimiento de un valor para los valores nulos
r.null map=R_279 value=0

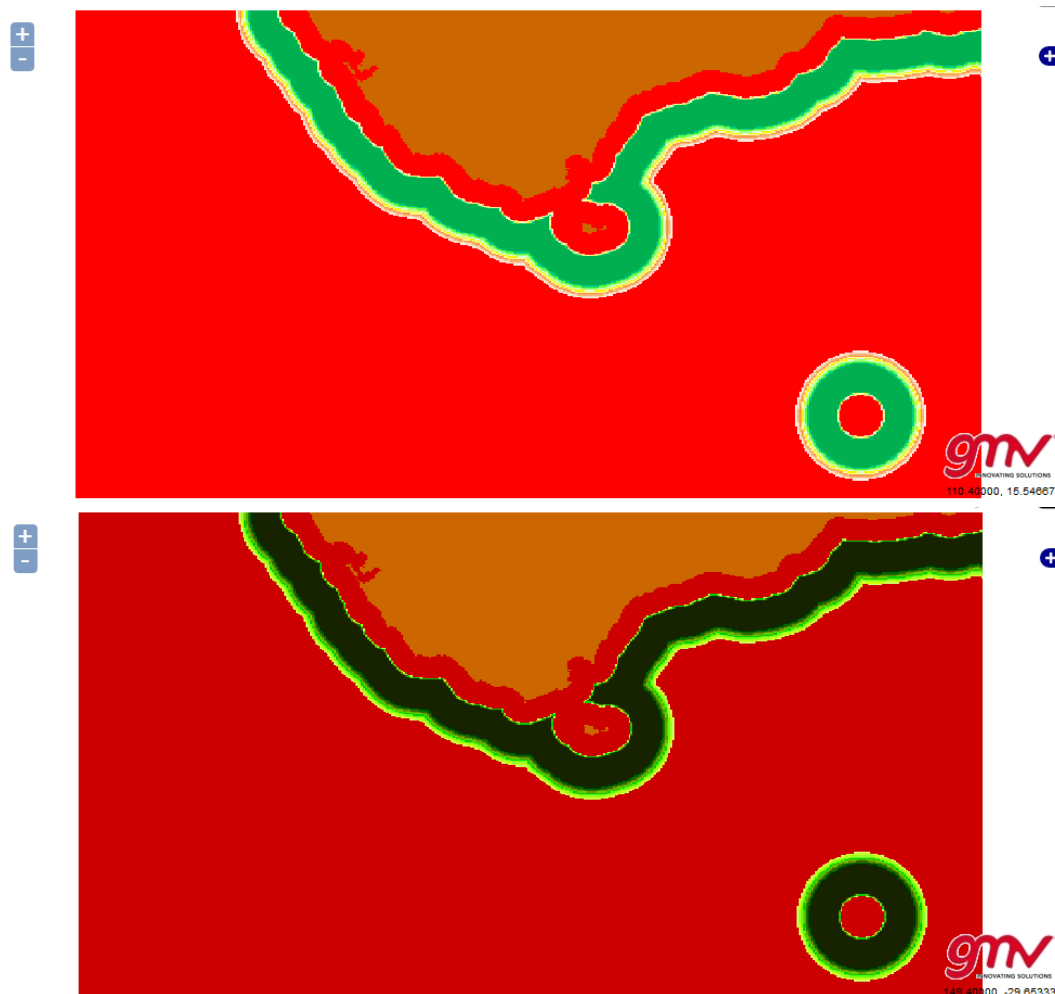
/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_280 = if(isnull(ccaa),R_279, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.736927215493 s=34.847278777993 e=-3.830121796875 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_279
output="C:\Users\GSI\OceanLider\Resultados\2\RI_R_279.tif"
r.out.gdal input=R_280
output="C:\Users\GSI\OceanLider\Resultados\2\RF_R_280.tif"
```

Visualización del resultado:



- **Profundidad.** De exclusión parcial. Se aplica una operación difusa sobre la profundidad extrayendo los parámetros del dispositivo (parámetros: Profundidad mínima, ProfMin, y profundidad máxima, ProfMax). Las medidas se aplican en metros. Se consideran los siguientes valores que procesa internamente:

valor A:= ProfMin-5m, valor B:= ProfMin, valor C:= ProfMax, valor D:= ProfMax+5m

Nota: Se puede proponer otra constante distinta a ± 5 metros para calcular los valores A y D.

Características del dispositivo

Nombre:

Tipo de dispositivo:

Tipo de captador:

Altura de buje: metros

Diámetro rotor: metros

Profundidad máxima: metros

Profundidad mínima: metros

Altura de ola máxima admitida: metros

Velocidad de la corriente máxima admitida: m/s

Potencia unitaria: vatios

Separación entre dispositivos de una misma línea: kilómetros

Separación entre líneas de dispositivos: kilómetros

Selección en interfaz:

Profundidad. Se aplica una operación difusa sobre la profundidad extrayendo los parámetros del dispositivo. Las medidas se aplican en metros.

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el criterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */  
  
//Configuración de la localización y Mapset del proyecto en el entorno  
de trabajo de GRASS  
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84  
//Configuración de la región que engloba a Península, Canarias y  
Balears  
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224  
cols=3453
```

```
/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa raster ubicada en el sistema de
// ficheros
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_281
//Operación difusa
r.mapcalc "R_282 = if(isnull( R_281 ) , null(), if((R_281 <= -55.0 ||
R_281 >= -10.0), 0.0, if((R_281 > -55.0 && R_281 < -50.0), eval((R_281 -
-55.0)/(-50.0 --55.0 )), if((R_281 > -15.0 && R_281 < -10.0 ), eval((-
10.0 - R_281 )/(-10.0 - -15.0)), 1.0)))"

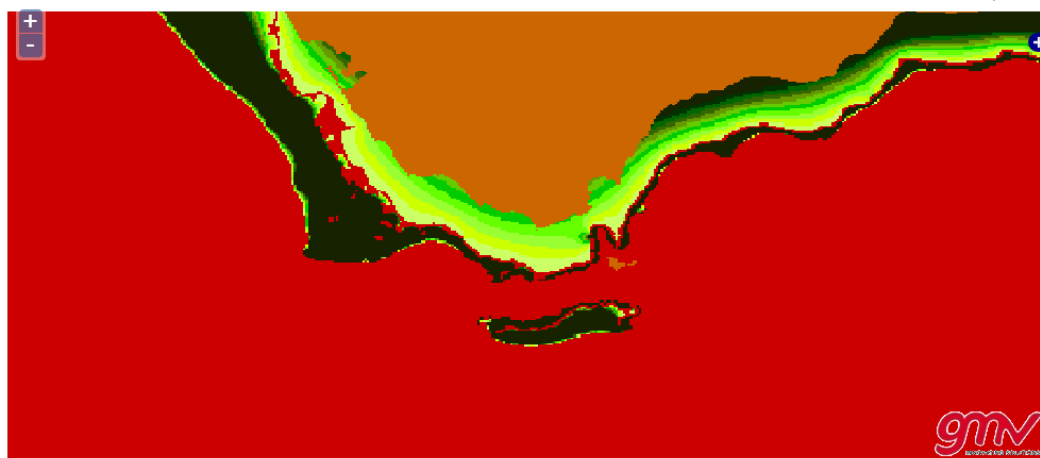
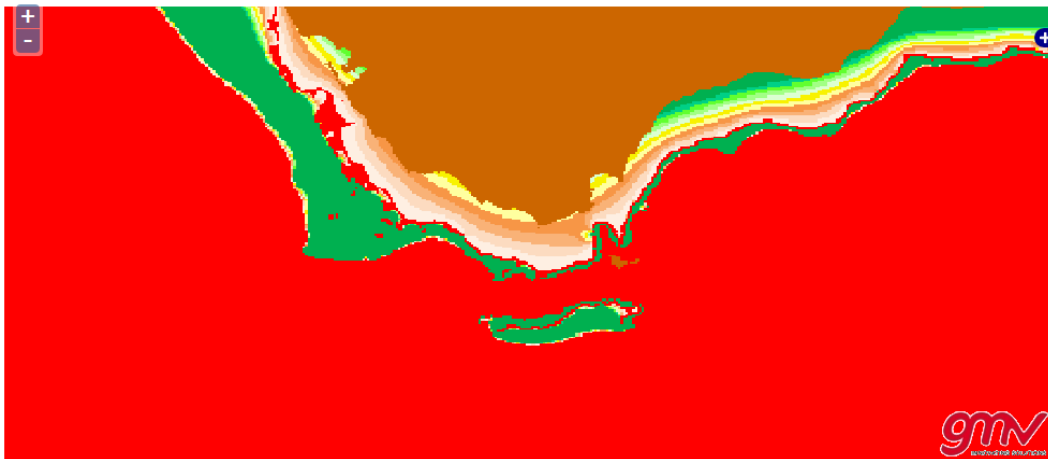
/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_283 = if(isnull(ccaa),R_282, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
// usuario
g.region n=36.716053259111 s=35.243885290361 e=-3.939985078125 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_282
output="C:\Users\GSI\OceanLider\Resultados\3\RI_R_282.tif"
r.out.gdal input=R_283
output="C:\Users\GSI\OceanLider\Resultados\3\RF_R_283.tif"
```

Visualización del resultado:



- **Supervivencia del dispositivo.** De exclusión total. El usuario puede seleccionar varios subcriterios. En caso que se seleccione más de un subcriterio se le da la opción al usuario de escoger el método de combinación de los subcriterios: método de razonamiento más que pesimista (multiplicación), pesimista (mínimo), optimista (máximo) y neutro (MLP).
- ✓ **Altura de ola significativa.** Se extrae el valor mayor de ola significativa admitido configurado en el dispositivo, HsMax. Se hace una reclasificación sobre el raster de altura de ola media.

Si $Altura_ola_media(Hs) \geq 0 \ \& \ Hs \leq HsMax$ entonces

$Altura_ola_significante := 1$

Sino

$Altura_ola_significante := 0$

Selección en la interfaz:

Altura de ola significativa. Se extrae el valor máximo de ola significativa admitido configurado en el dispositivo. 

Se muestran los resultados con una altura máxima admitida de 0.2 metros configurado en los parámetros del dispositivo:

Características del dispositivo

Nombre:	<input type="text" value="Dispositivo 2"/>	
Tipo de dispositivo:	<input type="text" value="De corrientes - Fuera de costa: flotante"/>	
Tipo de captador:	<input type="text" value="Eje horizontal"/>	
Altura de buje:	<input type="text" value="5"/>	metros
Diámetro rotor:	<input type="text" value="2"/>	metros
Profundidad máxima:	<input type="text" value="-50"/>	metros
Profundidad mínima:	<input type="text" value="-15"/>	metros
Altura de ola máxima admitida:	<input type="text" value="0.2"/>	metros
Velocidad de la corriente máxima admitida:	<input type="text" value="0.6"/>	m/s
Potencia unitaria:	<input type="text" value="10"/>	watios
Separación entre dispositivos de una misma línea:	<input type="text" value="2"/>	kilómetros
Separación entre líneas de dispositivos:	<input type="text" value="2"/>	kilómetros

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```

/* ***** OPERACIONES DE CONFIGURACIÓN ***** */

//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Baleares
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

```

```
/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa raster ubicada en el sistema de
//ficheros
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\unionhsmedia.tif"
output=R_284
//Reclasificación de la capa raster
r.reclass input=R_284 output=R_285
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_285.txt"
//Copia de una capa raster
r.mapcalc "R_286 = R_285"
//Establecimiento de un valor para los valores nulos
r.null map=R_286 null=0

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_287 = if(isnull(ccaa),R_286, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
//usuario
g.region n=36.758899871743 s=35.418567840493 e=-3.896039765625 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_286
output="C:\Users\GSI\OceanLider\Resultados\4\RI_R_286.tif"
r.out.gdal input=R_287
output="C:\Users\GSI\OceanLider\Resultados\4\RF_R_287.tif"
```

Fichero de reclasificación "C:\Users\tregmm\OceanLider\Resultados\reclass_R_285.txt":

```
0 thru 0.2 = 1
* = 0
```

Visualización del resultado:



Se muestran los resultados con una altura máxima admitida de 2.7 metros configurados en los parámetros del dispositivo:

Características del dispositivo

Nombre:	<input type="text" value="Dispositivo 3"/>
Tipo de dispositivo:	<input type="text" value="De corrientes - Fuera de costa: flotante"/>
Tipo de captador:	<input type="text" value="Eje horizontal"/>
Altura de buje:	<input type="text" value="5"/> metros
Diámetro rotor:	<input type="text" value="2"/> metros
Profundidad máxima:	<input type="text" value="-100"/> metros
Profundidad mínima:	<input type="text" value="-20"/> metros
Altura de ola máxima admitida:	<input type="text" value="2.7"/> metros
Velocidad de la corriente máxima admitida:	<input type="text" value="0.6"/> m/s
Potencia unitaria:	<input type="text" value="10"/> vatios
Separación entre dispositivos de una misma línea:	<input type="text" value="2"/> kilómetros
Separación entre líneas de dispositivos:	<input type="text" value="2"/> kilómetros

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */  
  
//Configuración de la localización y Mapset del proyecto en el entorno  
de trabajo de GRASS  
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84  
//Configuración de la región que engloba a Península, Canarias y  
Baleares  
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224  
cols=3453  
  
/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */
```

```
//Importación en GRASS de una capa raster ubicada en el sistema de
// ficheros
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\unionhsmmedia.tif"
output=R_292
//Reclasificación de la capa raster
r.reclass input=R_292 output=R_293
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_293.txt"
//Copia de una capa raster
r.mapcalc "R_294 = R_293"
//Establecimiento de un valor para los valores nulos
r.null map=R_294 null=0

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_295 = if(isnull(ccaa),R_294, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
// usuario
g.region n=36.758899536467 s=35.154895630217 e=-3.896039765625 w=-
7.367719453125 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_294
output="C:\Users\GSI\OceanLider\Resultados\6\RI_R_294.tif"
r.out.gdal input=R_295
output="C:\Users\GSI\OceanLider\Resultados\6\RF_R_295.tif"
```

El fichero de reclasificación "C:\Users\tregmm\OceanLider\Resultados\reclass_R_293.txt":

```
0 thru 2.7 = 1
* = 0
```

La visualización del resultado:



- ✓ **Velocidad de corriente.** Se extrae el valor máximo de corriente admitido en la configuración del dispositivo, CoMax. Se hace una reclasificación sobre el raster de la velocidad de mareas vivas.

Si *corriente_marea_viva* ≥ 0 & *corriente_marea_viva* \leq CoMax entonces

Velocidad_corriente := 1

Sino

Velocidad_corriente := 0

Selección en la interfaz:

Velocidad de corriente. Se extrae el valor máximo de corriente admitido en la configuración del dispositivo. 

Se muestran los resultados con una velocidad máxima de corriente admitida de 0.6 metros/segundo configurado en los parámetros del dispositivo:

Características del dispositivo

Nombre:	<input type="text" value="Dispositivo 2"/>	
Tipo de dispositivo:	<input type="text" value="De corrientes - Fuera de costa: flotante"/>	
Tipo de captador:	<input type="text" value="Eje horizontal"/>	
Altura de buje:	<input type="text" value="5"/>	metros
Diámetro rotor:	<input type="text" value="2"/>	metros
Profundidad máxima:	<input type="text" value="-50"/>	metros
Profundidad mínima:	<input type="text" value="-15"/>	metros
Altura de ola máxima admitida:	<input type="text" value="0.2"/>	metros
Velocidad de la corriente máxima admitida:	<input type="text" value="0.6"/>	m/s
Potencia unitaria:	<input type="text" value="10"/>	watios
Separación entre dispositivos de una misma línea:	<input type="text" value="2"/>	kilómetros
Separación entre líneas de dispositivos:	<input type="text" value="2"/>	kilómetros

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```

/* ***** OPERACIONES DE CONFIGURACIÓN ***** */

//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Baleares
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

```

Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

```
//Importación en GRASS de una capa raster ubicada en el sistema de
// ficheros
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\medviv.tif"
output=R_288
//Reclasificación de la capa raster
r.reclass input=R_288 output=R_289
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_289.txt"
//Copia de una capa raster
r.mapcalc "R_290 = R_289"
//Establecimiento de un valor para los valores nulos
r.null map=R_290 null=0

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BEDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_291 = if(isnull(ccaa),R_290, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
// usuario
g.region n=36.803943884111 s=35.309803259111 e=-3.720258515625 w=-
7.279828828125 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_290
output="C:\Users\GSI\OceanLider\Resultados\5\RI_R_290.tif"
r.out.gdal input=R_291
output="C:\Users\GSI\OceanLider\Resultados\5\RF_R_291.tif"
```

Archivo de reclasificación "C:\Users\tregmm\OceanLider\Resultados\reclass_R_289.txt":

```
0 thru 0.6 = 1
* = 0
```

Visualización del resultado:



1.2. Criterios operacionales

- **Mantenimiento de la posición.** De exclusión parcial. El usuario puede seleccionar varios subcriterios. En caso que se seleccione más de un subcriterio, el método de razonamiento para la combinación de los subcriterios es el más que pesimista (multiplicación).
- ✓ **Facilidad de anclaje al fondo (composición del sustrato marino).** Se extraen los valores de idoneidad de las zonas configuradas para la capa Geofis dependiendo del tipo de dispositivo elegido y se le deja la libertad al usuario para poder cambiar estos valores a través de la interfaz.

Los valores de idoneidad configurados para cada agrupación por defecto para cada tipo de dispositivo se reflejan en la siguiente tabla.

NOMBRE	NOMBRE CAPA	METADATOS	Tipo de Capa	CORRIENTES		OLEAJE		
				FUERA DE COSTA (Apoyados sobre base de gravedad o pilotados)	FUERA DE COSTA (Fondeados)	FUERA DE COSTA (Apoyados sobre base de gravedad o pilotados)	FUERA DE COSTA (Flotantes Fondeados)	
MORFOLOGÍA FONDOS MARINOS								
Sustrato de los fondos	Geofis_P.shp	Cartografía vectorial poligonal de los tipos de sustrato de los fondos marinos del litoral de la Península Ibérica, Ceuta y Melilla. Se encuentra en ETRS89 huso 30.	Roca	Excluyente parcial	0,5	0,75	0,5	0,75
			Vegetado		1	1	1	1
			Blando Cohesivo		0	0,5	0	0,5
			Blando no Cohesivo		1	1	1	1

Características del dispositivo

Nombre:

Tipo de dispositivo:

Largo: metros

Ancho: metros

Alto: metros

Profundidad máxima: metros

Profundidad mínima: metros

Altura de ola máxima admitida: metros

Velocidad de la corriente máxima admitida: m/s

Potencia unitaria: vatios

Separación entre dispositivos de una misma línea: kilómetros

Separación entre líneas de dispositivos: kilómetros

Selección en la interfaz:

Facilidad de anclaje al fondo (composición del sustrato marino) 

Roca	0.5
Vegetado	1
Blando cohesivo	0
Blando no cohesivo	1

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```

/* ***** OPERACIONES DE CONFIGURACIÓN ***** */

//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Balears
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributo
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_296 layer=geofis
where="Agrup='Roca'"
//Rasterización por valor de una capa vectorial
v.to.rast input=R_296 layer=1 type=point,line,area output=R_297 use=val
value=0.5 rows=4096
//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributo
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_298 layer=geofis
where="Agrup='Vegetado'"
//Rasterización por valor de una capa vectorial
v.to.rast input=R_298 layer=1 type=point,line,area output=R_299 use=val
value=1 rows=4096
//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributo
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_300 layer=geofis
where="Agrup='Blando cohesivo'"
//Rasterización por valor de una capa vectorial
v.to.rast input=R_300 layer=1 type=point,line,area output=R_301 use=val
value=0 rows=4096
//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributo

```

Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

```
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_302 layer=geofis
where="Agrup='Blando no cohesivo'"
//Rasterización por valor de una capa vectorial
v.to.rast input=R_302 layer=1 type=point,line,area output=R_303 use=val
value=1 rows=4096
//Operación de unión de dos capas raster
r.patch input=R_297,R_299 output=R_304
r.patch input=R_301,R_303 output=R_305
r.patch input=R_304,R_305 output=R_306
//Establecimiento de un valor para los valores nulos
r.null map=R_306 null=1

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_307 = if(isnull(ccaa),R_306, -1.0)"

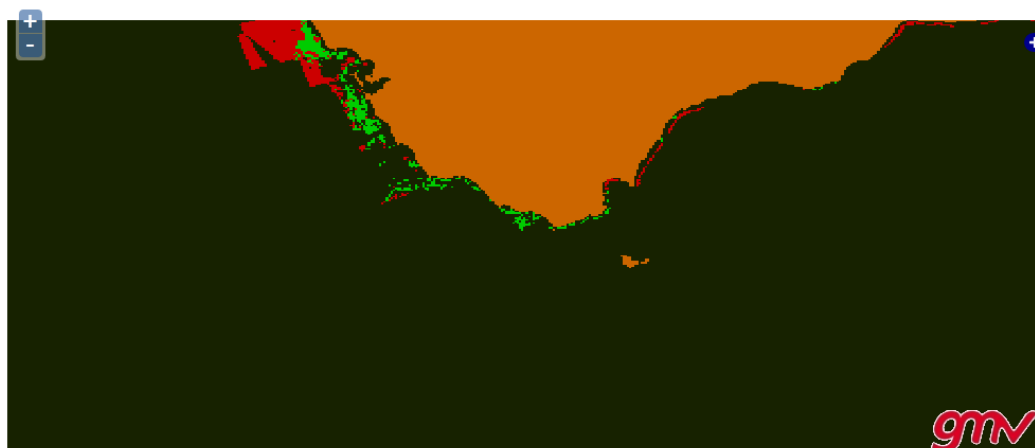
/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.716053259111 s=35.243885290361 e=-3.939985078125 w=-
7.477582734375 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_306
output="C:\Users\GSI\OceanLider\Resultados\7\RI_R_306.tif"
r.out.gdal input=R_307
output="C:\Users\GSI\OceanLider\Resultados\7\RF_R_307.tif"
```


Visualización del resultado:



135.60000, 13.14867



Cambiando los valores de idoneidad a través de la interfaz para cada agrupación.

Facilidad de anclaje al fondo (composición del sustrato marino) 

Roca	0.75
Vegetado	0.9
Blando cohesivo	0.25
Blando no cohesivo	0.8

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */
```



```
//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Balears
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributo
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_308 layer=geofis
where="Agrup='Roca'"
//Rasterización por valor de una capa vectorial
v.to.rast input=R_308 layer=1 type=point,line,area output=R_309 use=val
value=0.75 rows=4096
//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributo
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_310 layer=geofis
where="Agrup='Vegetado'"
//Rasterización por valor de una capa vectorial
v.to.rast input=R_310 layer=1 type=point,line,area output=R_311 use=val
value=0.9 rows=4096
//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributo
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_312 layer=geofis
where="Agrup='Blando cohesivo'"
//Rasterización por valor de una capa vectorial
v.to.rast input=R_312 layer=1 type=point,line,area output=R_313 use=val
value=0.25 rows=4096
//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributo
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_314 layer=geofis
where="Agrup='Blando no cohesivo'"
//Rasterización por valor de una capa vectorial
v.to.rast input=R_314 layer=1 type=point,line,area output=R_315 use=val
value=0.8 rows=4096
//Operación de unión de dos capas raster
r.patch input=R_309,R_311 output=R_316
r.patch input=R_313,R_315 output=R_317
r.patch input=R_316,R_317 output=R_318
//Establecimiento de un valor para los valores nulos
r.null map=R_318 null=1

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

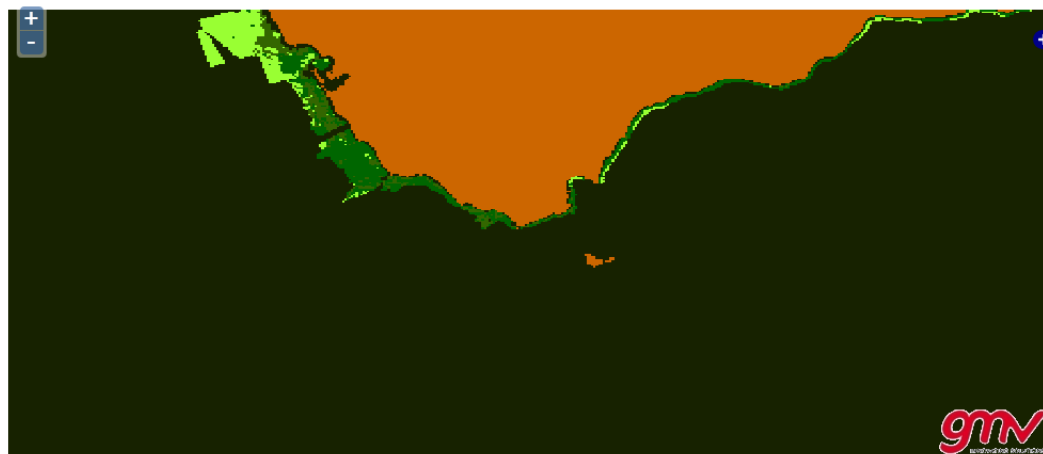
//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
```

```
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_319 = if(isnull(ccaa),R_318, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.736926880217 s=35.220813598967 e=-3.808149140625 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_318
output="C:\Users\GSI\OceanLider\Resultados\8\RI_R_318.tif"
r.out.gdal input=R_319
output="C:\Users\GSI\OceanLider\Resultados\8\RF_R_319.tif"
```

Visualización del resultado:



- ✓ **Facilidad de anclaje al fondo (morfología del fondo marino).** Se extraen los valores de idoneidad de las zonas configuradas en la capa Formas dependiendo del tipo de dispositivo elegido y se le deja la libertad al usuario de cambiar los valores según la agrupación para los diferentes tipos que se muestran en la siguiente tabla.

NOMBRE	NOMBRE CAPA	METADATOS	NOMBRE DE GRUPO	TIPO DE ACCIDENTE GEOGRÁFICO	CORRIENTES		OLEAJE	
					FUERA DE COSTA (Apoyados sobre base de gravedad o pilotados)	FUERA DE COSTA (Fondeados)	FUERA DE COSTA (Apoyados sobre base de gravedad o pilotados)	FUERA DE COSTA (Flotantes Fondeados)
Accidentes geográficos submarinos	Formas_P.shp	Cartografía vectorial lineal que se corresponde con los accidentes geográficos submarinos más reseñables de la península Ibérica, Ceuta y Melilla. Se encuentra en ETRS89 huso 30.	Grupo1	Afloramiento rocoso	1	1	1	1
			Grupo1	Marcas de arrastre	1	1	1	1
			Grupo1	Plataforma continental	1	1	1	1
			Grupo1	Playas sumergidas	1	1	1	1
			Grupo1	Ripples	1	1	1	1
			Grupo1	Talud continental	1	1	1	1
			Grupo0	Abanico deltaico	0	0	0	0
			Grupo0	Abanico deltaico profundo	0	0	0	0
			Grupo0	Cañon submarino	0	0	0	0
			Grupo0	Cañon submarino	0	0	0	0
			Grupo0	Llanuras abisales	0	0	0	0
			Grupo0	Megaripples	0	0	0	0
			Grupo0	Megaripples	0	0	0	0
			Grupo0	Morfología de crestas y valles	0	0	0	0
			Grupo0	Ondas de arena	0	0	0	0
			Grupo0	Relieve volcanico	0	0	0	0

Características del dispositivo

Nombre:	<input type="text" value="Dispositivo 1"/>
Tipo de dispositivo:	<input type="text" value="Undimotriz - Fuera de costa: flotante"/>
Largo:	<input type="text" value="5"/> metros
Ancho:	<input type="text" value="2"/> metros
Alto:	<input type="text" value="10"/> metros
Profundidad máxima:	<input type="text" value="-100"/> metros
Profundidad mínima:	<input type="text" value="-20"/> metros
Altura de ola máxima admitida:	<input type="text" value="1.6"/> metros
Velocidad de la corriente máxima admitida:	<input type="text" value="0.6"/> m/s
Potencia unitaria:	<input type="text" value="10"/> vatios
Separación entre dispositivos de una misma línea:	<input type="text" value="2"/> kilómetros
Separación entre líneas de dispositivos:	<input type="text" value="2"/> kilómetros

Valores de idoneidad configurados en la interfaz:

Facilidad de anclaje al fondo (morfología del fondo marino) 

Grupo0

Grupo1

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */

//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Balears
```

Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

```
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributos
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_338 layer=formas
where="(TIPO='Megaripples') OR (TIPO='Abanico deltaico') OR
(TIPO='Abanico deltaico profundo') OR (TIPO='Caynon submarino') OR
(TIPO='Caynon submarino') OR (TIPO='Llanuras abisales') OR
(TIPO='Morfología de crestas y valles') OR (TIPO='Megaripples') OR
(TIPO='Ondas de arena') OR (TIPO='Relieve volcanico')"
//Rasterización por valor de una capa vectorial
v.to.rast input=R_338 layer=1 type=point,line,area output=R_339 use=val
value=0 rows=4096
//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributos
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_340 layer=formas
where="(TIPO='Ripples') OR (TIPO='Marcas de arrastre') OR (TIPO='Talud
continental') OR (TIPO='Afloramiento rocoso') OR (TIPO='Playas
sumergidas') OR (TIPO='Plataforma continental')"
//Rasterización por valor de una capa vectorial
v.to.rast input=R_340 layer=1 type=point,line,area output=R_341 use=val
value=1 rows=4096
//Operación de unión de dos capas raster
r.patch input=R_339,R_341 output=R_342
//Establecimiento de un valor para los valores nulos
r.null map=R_342 null=1

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_343 = if(isnull(ccaa),R_342, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */


//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.736926880217 s=35.286731567717 e=-3.830121796875 w=-
7.257856171875 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_342
output="C:\Users\GSI\OceanLider\Resultados\9\RI_R_342.tif"
r.out.gdal input=R_343
output="C:\Users\GSI\OceanLider\Resultados\9\RF_R_343.tif"
```

Visualización del resultado:



- **Viabilidad de instalación en zonas ocupadas.** De exclusión parcial. El usuario puede seleccionar varios subcriterios. Si se selecciona más de un subcriterio el método de razonamiento para la combinación de subcriterios es el pesimista (mínimo de los valores).
 - ✓ *Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español.* Se aplica una operación difusa sobre la distancia a las zonas ocupadas por las ayudas a la navegación del sistema portuario español. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Los valores de los parámetros introducidos en la interfaz:

Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español. Se aplica una operación difusa sobre la distancia a las zonas ocupadas por las ayudas a la navegación del sistema portuario español. Las medidas se aplican en Km. 

De subida ▾	
Valor A	0.5
Valor B	1
Valor C	99999
Valor D	99999

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */

//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Baleares
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributos
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=aton output=R_332
//Rasterización por valor de una capa vectorial
v.to.rast input=R_332 layer=1 type=point,line,area output=R_333 use=val
value=0 rows=4096
//Cálculo de un mapa de distancias
r.grow.distance input=R_333 distance=R_334 metric=Euclidean
//Multiplicación de una capa raster por un escalar
r.mapcalc "R_335 = R_334 * 100"
//Operación difusa
r.mapcalc "R_336 = if(isnull( R_335 ), null(), if((R_335 <= 0.5 ||
R_335 >= 99999), 0.0, if((R_335 > 0.5 && R_335 < 1), eval((R_335 -
0.5)/(1 -0.5 )), if((R_335 > 99999 && R_335 < 99999 ), eval((99999 -
R_335)/(99999 - 99999)), 1.0))))"

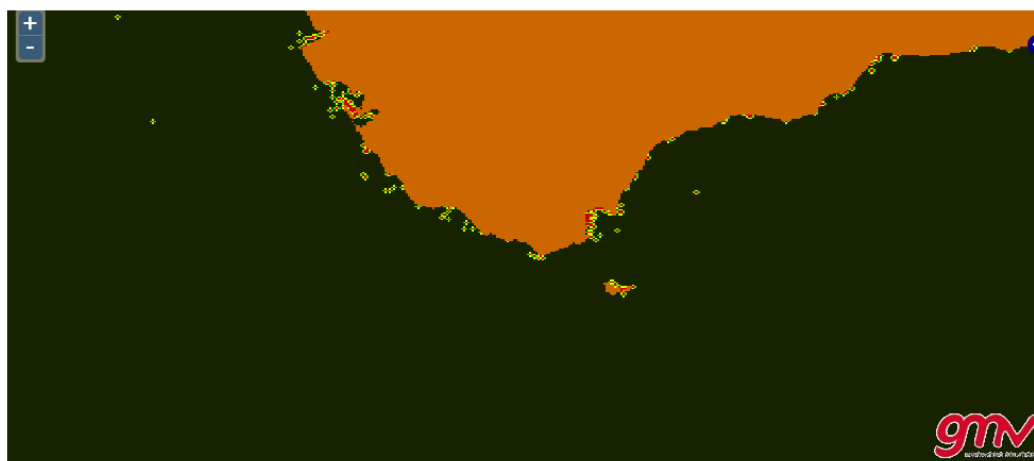
/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_337 = if(isnull(ccaa),R_336, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */
```

```
//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.868762817717 s=35.264758911467 e=-3.830121796875 w=-
7.477582734375 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_336
output="C:\Users\GSI\OceanLider\Resultados\10\RI_R_336.tif"
r.out.gdal input=R_337
output="C:\Users\GSI\OceanLider\Resultados\10\RF_R_337.tif"
```

Visualización del resultado:



- ✓ **Viabilidad de instalación en zonas designadas como Dominio Público Portuario** Se aplica una operación difusa sobre la distancia a las zonas designadas como Dominio Público Portuario. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Los valores de los parámetros introducidos en la interfaz:

Viabilidad de instalación en zonas designadas como Dominio Público Portuario. Se aplica una operación difusa sobre la distancia a las zonas designadas como Dominio Público Portuario. Las medidas se aplican en Km. 

De subida ▾

Valor A	<input type="text" value="0"/>
Valor B	<input type="text" value="1"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */
//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Baleares
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributos
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=dpp output=R_344
//Rasterización por valor de una capa vectorial
v.to.rast input=R_344 layer=1 type=point,line,area output=R_345 use=val
value=0 rows=4096
//Cálculo de un mapa de distancias
r.grow.distance input=R_345 distance=R_346 metric=Euclidean
//Multiplicación de una capa raster por un escalar
r.mapcalc "R_347 = R_346 * 100"
//Operación difusa
r.mapcalc "R_348 = if(isnull( R_347 ) , null(), if((R_347 <= 0 || R_347
>= 99999), 0.0, if((R_347 > 0 && R_347 < 1), eval((R_347 - 0)/(1 - 0)),
if((R_347 > 99999 && R_347 < 99999 ), eval((99999 - R_347)/(99999 -
99999)), 1.0)))"

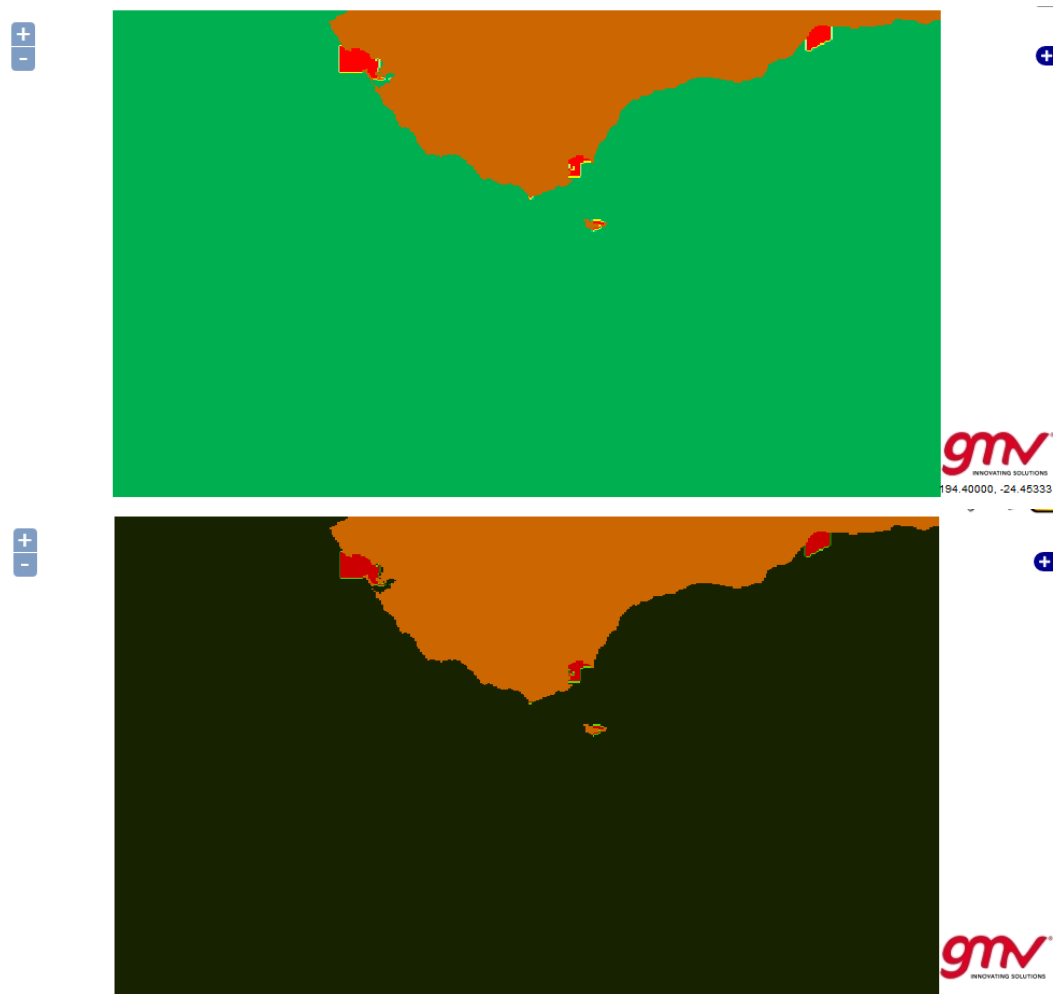
/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_349 = if(isnull(ccaa),R_348, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */
```

```
//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.781971227861 s=34.760486852861 e=-3.896039765625 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_348
output="C:\Users\GSI\OceanLider\Resultados\11\RI_R_348.tif"
r.out.gdal input=R_349
output="C:\Users\GSI\OceanLider\Resultados\11\RF_R_349.tif"
```


Visualización del resultado:



- ✓ *Viabilidad de instalación en zonas preseleccionadas para la extracción de áridos.* Se aplica una operación difusa sobre la distancia a las zonas preseleccionadas para la

extracción de áridos. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Los valores de los parámetros introducidos en la interfaz:

Viabilidad de instalación en zonas preseleccionadas para la extracción de áridos. Se aplica una operación difusa sobre la distancia a las zonas preseleccionadas para la extracción de áridos. Las medidas se aplican en Km. 

De subida ▾

Valor A	1
Valor B	2
Valor C	99999
Valor D	99999

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */
//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Baleares
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributos
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ext_aridos output=R_350
//Rasterización por valor de una capa vectorial
v.to.rast input=R_350 layer=1 type=point,line,area output=R_351 use=val
value=0 rows=4096
//Cálculo de un mapa de distancias
r.grow.distance input=R_351 distance=R_352 metric=Euclidean
//Multiplicación de una capa raster por un escalor
r.mapcalc "R_353 = R_352 * 100"
//Operación difusa
r.mapcalc "R_354 = if(isnull( R_353 ), null(), if((R_353 <= 1 || R_353
>= 99999), 0.0, if((R_353 > 1 && R_353 < 2), eval((R_353 - 1)/(2 -1 )),
if((R_353 > 99999 && R_353 < 99999 ), eval((99999 - R_353 )/(99999 -
99999)), 1.0)))"

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
```

```
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_355 = if(isnull(ccaa),R_354, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.781971227861 s=35.155994665361 e=-3.939985078125 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_354
output="C:\Users\GSI\OceanLider\Resultados\12\RI_R_354.tif"
r.out.gdal input=R_355
output="C:\Users\GSI\OceanLider\Resultados\12\RF_R_355.tif"
```

Visualización del resultado:



- ✓ **Viabilidad de instalación en zonas ocupadas por conducciones submarinas.** Se aplica una operación difusa sobre la distancia a las zonas ocupadas por conducciones submarinas. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Los valores de los parámetros introducidos en la interfaz:

Viabilidad de instalación en zonas ocupadas por conducciones submarinas. Se aplica una operación difusa sobre la distancia a las zonas ocupadas por conducciones submarinas. Las medidas se aplican en Km.

Normal

Valor A	1.5
Valor B	3
Valor C	6
Valor D	15

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */
//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Balears
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributos
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=conducciones output=R_356
//Rasterización por valor de una capa vectorial
v.to.rast input=R_356 layer=1 type=point,line,area output=R_357 use=val
value=0 rows=4096
//Cálculo de un mapa de distancias
r.grow.distance input=R_357 distance=R_358 metric=Euclidean
//Multiplicación de una capa raster por un escalar
r.mapcalc "R_359 = R_358 * 100"
//Operación difusa
r.mapcalc "R_360 = if(isnull( R_359 ) , null(), if((R_359 <= 1.5 ||
R_359 >= 15), 0.0, if((R_359 > 1.5 && R_359 < 3), eval((R_359 - 1.5)/(3
-1.5)), if((R_359 > 6 && R_359 < 15), eval((15 - R_359)/(15 - 6)),
1.0)))"

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
```

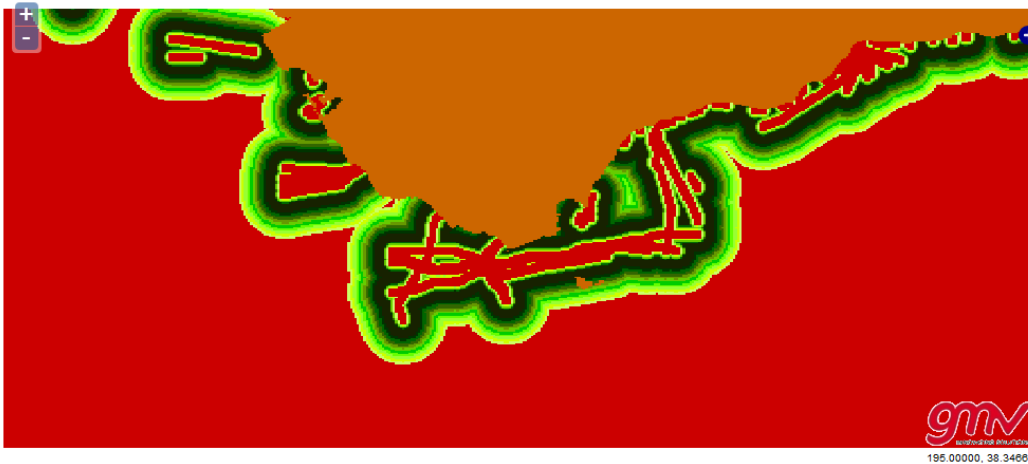
```
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_361 = if(isnull(ccaa),R_360, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.824817505217 s=35.330676880217 e=-3.808149140625 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_360
output="C:\Users\GSI\OceanLider\Resultados\13\RI_R_360.tif"
r.out.gdal input=R_361
output="C:\Users\GSI\OceanLider\Resultados\13\RF_R_361.tif"
```

Visualización del resultado:





- ✓ **Viabilidad de instalación en zonas preseleccionadas para el vertido de material de dragado.** Se aplica una operación difusa sobre la distancia a las zonas preseleccionadas para el vertido de material de dragado. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Los valores de los parámetros introducidos en la interfaz:

Viabilidad de instalación en zonas preseleccionadas para el vertido de material de dragado. Se aplica una operación difusa sobre la distancia a las zonas preseleccionadas para el vertido de material de dragado. Las medidas se aplican en Km.

De subida ▾

Valor A	1
Valor B	2
Valor C	99999
Valor D	99999

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */
//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Balears
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributos
```

```

v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=vertido_dragado output=R_392
//Rasterización por valor de una capa vectorial
v.to.rast input=R_392 layer=1 type=point,line,area output=R_393 use=val
value=0 rows=4096
//Cálculo de un mapa de distancias
r.grow.distance input=R_393 distance=R_394 metric=Euclidean
//Multiplicación de una capa raster por un escalar
r.mapcalc "R_395 = R_394 * 100"
//Operación difusa
r.mapcalc "R_396 = if(isnull( R_395 ), null(), if((R_395 <= 1 || R_395
>= 99999), 0.0, if((R_395 > 1 && R_395 < 2), eval((R_395 - 1)/(2 -1 )),
if((R_395 > 99999 && R_395 < 99999 ), eval((99999 - R_395 )/(99999 -
99999)), 1.0))))"

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_397 = if(isnull(ccaa),R_396, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.957752477861 s=35.177967321611 e=-3.764203828125 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_396
output="C:\Users\GSI\OceanLider\Resultados\19\RI_R_396.tif"
r.out.gdal input=R_397
output="C:\Users\GSI\OceanLider\Resultados\19\RF_R_397.tif"

```

Visualización del resultado:



- ✓ *Viabilidad de instalación en zonas ocupadas por sensores instrumentales de medición de parámetros oceanográficos.* Se aplica una operación difusa sobre la distancia a las zonas ocupadas por sensores instrumentales de medición de parámetros oceanográficos. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Los valores de los parámetros introducidos en la interfaz:

Viabilidad de instalación en zonas ocupadas por sensores instrumentales de medición de parámetros oceanográficos. Se aplica una operación difusa sobre la distancia a las zonas ocupadas por sensores instrumentales de medición de parámetros oceanográficos. Las medidas se aplican en Km. 

De subida ▾	
Valor A	0.5
Valor B	1
Valor C	99999
Valor D	99999

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```

/* ***** OPERACIONES DE CONFIGURACIÓN ***** */
//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Baleares
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributos
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=rmo output=R_368
//Rasterización por valor de una capa vectorial
v.to.rast input=R_368 layer=1 type=point,line,area output=R_369 use=val
value=0 rows=4096
//Cálculo de un mapa de distancias
r.grow.distance input=R_369 distance=R_370 metric=Euclidean
//Multiplicación de una capa raster por un escalar
r.mapcalc "R_371 = R_370 * 100"
//Operación difusa
r.mapcalc "R_372 = if(isnull( R_371 ), null(), if((R_371 <= 0.5 ||
R_371 >= 99999), 0.0, if((R_371 > 0.5 && R_371 < 1), eval((R_371 -
0.5)/(1 -0.5 )), if((R_371 > 99999 && R_371 < 99999 ), eval((99999 -
R_371)/(99999 - 99999)), 1.0)))"

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

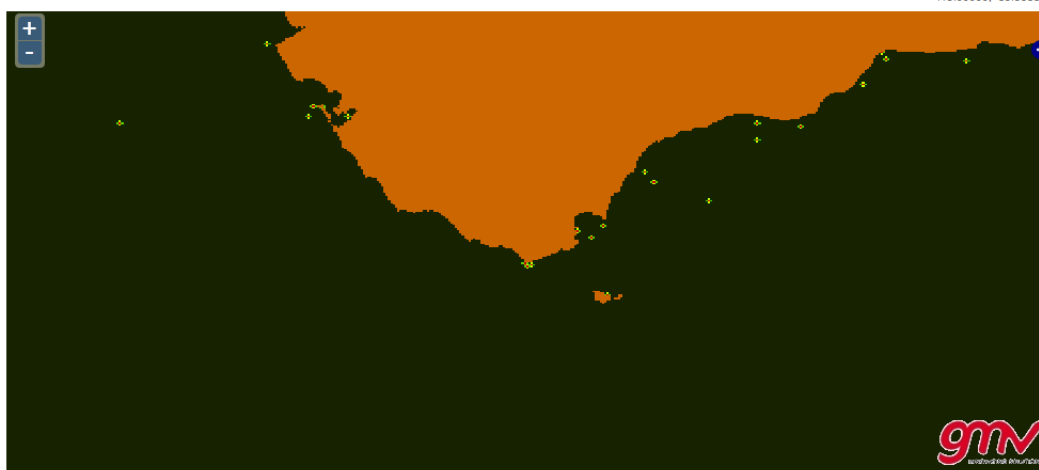
//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_373 = if(isnull(ccaa),R_372, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

```


```
//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.847889196611 s=35.287830602861 e=-3.852094453125 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_372
output="C:\Users\GSI\OceanLider\Resultados\15\RI_R_372.tif"
r.out.gdal input=R_373
output="C:\Users\GSI\OceanLider\Resultados\15\RF_R_373.tif"
```

Visualización del resultado:



- ✓ **Viabilidad de instalación en zonas ocupadas por dispositivos de separación del tráfico marítimo.** Se aplica una operación difusa sobre la distancia a las zonas ocupadas por dispositivos de separación del tráfico marítimo. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Los valores de los parámetros introducidos en la interfaz:

Viabilidad de instalación en zonas ocupadas por dispositivos de separación del tráfico marítimo. Se aplica una operación difusa sobre la distancia a las zonas ocupadas por dispositivos de separación del tráfico marítimo. Las medidas se aplican en Km. 

Normal	
Valor A	1.5
Valor B	3
Valor C	6
Valor D	15

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```

/* ***** OPERACIONES DE CONFIGURACIÓN ***** */
//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Baleares
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributos
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=dst output=R_374
//Rasterización por valor de una capa vectorial
v.to.rast input=R_374 layer=1 type=point,line,area output=R_375 use=val
value=0 rows=4096
//Cálculo de un mapa de distancias
r.grow.distance input=R_375 distance=R_376 metric=Euclidean
//Multiplicación de una capa raster por un escalar
r.mapcalc "R_377 = R_376 * 100"
//Operación difusa
r.mapcalc "R_378 = if(isnull( R_377 ), null(), if((R_377 <= 1.5 ||
R_377 >= 15), 0.0, if((R_377 > 1.5 && R_377 < 3), eval((R_377 - 1.5)/(3
-1.5 )), if((R_377 > 6 && R_377 < 15 ), eval((15 - R_377 )/(15 - 6)),
1.0)))"

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

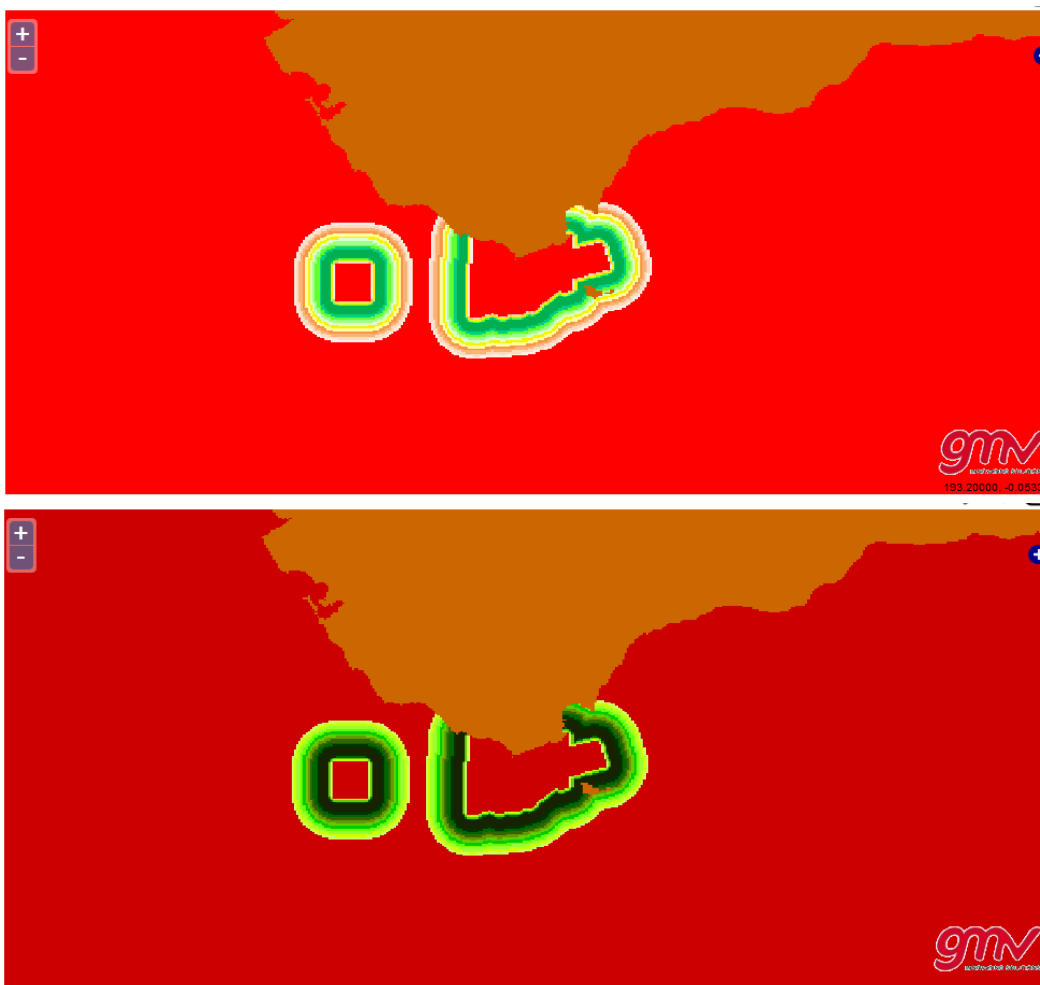
//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_379 = if(isnull(ccaa),R_378, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

```

```
//Configuración de la región a los parámetros configurados por el usuario
g.region n=36.825916540361 s=35.221912634111 e=-3.852094453125 w=-7.301801484375 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_378
output="C:\Users\GSI\OceanLider\Resultados\16\RI_R_378.tif"
r.out.gdal input=R_379
output="C:\Users\GSI\OceanLider\Resultados\16\RF_R_379.tif"
```


Visualización del resultado:



- **Viabilidad de instalación en zonas restringidas.** De exclusión parcial. El usuario puede escoger varios subcriterios. Si se selecciona más de un subcriterio, el método de razonamiento para la combinación de subcriterios es el pesimista (mínimo de los valores).

- ✓ **Viabilidad de instalación en zonas restringidas para uso militar.** Se aplica una operación difusa sobre la distancia a las zonas restringidas para uso militar. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Los valores de los parámetros introducidos en la interfaz:

Viabilidad de instalación en zonas restringidas para uso militar. Se aplica una operación difusa sobre la distancia a las zonas restringidas para uso militar. Las medidas se aplican en Km. 

De subida ▾

Valor A	1
Valor B	2
Valor C	99999
Valor D	99999

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```

/* ***** OPERACIONES DE CONFIGURACIÓN ***** */
//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Balears
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributos
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=militar output=R_380
//Rasterización por valor de una capa vectorial
v.to.rast input=R_380 layer=1 type=point,line,area output=R_381 use=val
value=0 rows=4096
//Cálculo de un mapa de distancias
r.grow.distance input=R_381 distance=R_382 metric=Euclidean
//Multiplicación de una capa raster por un escalar
r.mapcalc "R_383 = R_382 * 100"
//Operación difusa
r.mapcalc "R_384 = if(isnull( R_383 ), null(), if((R_383 <= 1 || R_383
>= 99999), 0.0, if((R_383 > 1 && R_383 < 2), eval((R_383 - 1)/(2 -1 )),
if((R_383 > 99999 && R_383 < 99999 ), eval((99999 - R_383 )/(99999 -
99999)), 1.0))))"

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial

```

Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

```
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_385 = if(isnull(ccaa),R_384, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */


//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.846790161467 s=35.286731567717 e=-3.786176484375 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_384
output="C:\Users\GSI\OceanLider\Resultados\17\RI_R_384.tif"
r.out.gdal input=R_385
output="C:\Users\GSI\OceanLider\Resultados\17\RF_R_385.tif"
```

Visualización del resultado:



- ✓ **Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías.** Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Los valores de los parámetros introducidos en la interfaz:

Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías.
 Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Las medidas se aplican en Km. 

Normal

Valor A	1.5
Valor B	3
Valor C	6
Valor D	15

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```

/* ***** OPERACIONES DE CONFIGURACIÓN ***** */
//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Baleares
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributos
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=zona_cables output=R_386
//Rasterización por valor de una capa vectorial
v.to.rast input=R_386 layer=1 type=point,line,area output=R_387 use=val
value=0 rows=4096
//Cálculo de un mapa de distancias
r.grow.distance input=R_387 distance=R_388 metric=Euclidean
//Multiplicación de una capa raster por un escalar
r.mapcalc "R_389 = R_388 * 100"
//Operación difusa
r.mapcalc "R_390 = if(isnull( R_389 ), null(), if((R_389 <= 1.5 ||
R_389 >= 15), 0.0, if((R_389 > 1.5 && R_389 < 3), eval((R_389 - 1.5)/(3
-1.5 )), if((R_389 > 6 && R_389 < 15 ), eval((15 - R_389)/(15 - 6)),
1.0)))"

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS

```


Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

```
v.in.ogr -o "dsn=PG:dbname=BDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_391 = if(isnull(ccaa),R_390, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.803943884111 s=35.068104040361 e=-3.786176484375 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_390
output="C:\Users\GSI\OceanLider\Resultados\18\RI_R_390.tif"
r.out.gdal input=R_391
output="C:\Users\GSI\OceanLider\Resultados\18\RF_R_391.tif"
```

Visualización del resultado:





- ✓ **Viabilidad de instalación en zonas restringidas por la presencia de vertidos.** Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de vertidos. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Los valores de los parámetros introducidos en la interfaz:

Viabilidad de instalación en zonas restringidas por la presencia de vertidos. Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de vertidos. Las medidas se aplican en Km.

De subida ▾

Valor A	1
Valor B	2
Valor C	99999
Valor D	99999

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */
//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Baleares
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributos
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=vertidos output=R_362
```

```
//Rasterización por valor de una capa vectorial
v.to.rast input=R_362 layer=1 type=point,line,area output=R_363 use=val
value=0 rows=4096
//Cálculo de un mapa de distancias
r.grow.distance input=R_363 distance=R_364 metric=Euclidean
//Multiplicación de una capa raster por un escalar
r.mapcalc "R_365 = R_364 * 100"
//Operación difusa
r.mapcalc "R_366 = if(isnull( R_365 ) , null(), if((R_365 <= 1 || R_365
>= 99999), 0.0, if((R_365 > 1 && R_365 < 2), eval((R_365 - 1)/(2 -1 )),
if((R_365 > 99999 && R_365 < 99999 ), eval((99999 - R_365 )/(99999 -
99999)), 1.0)))"

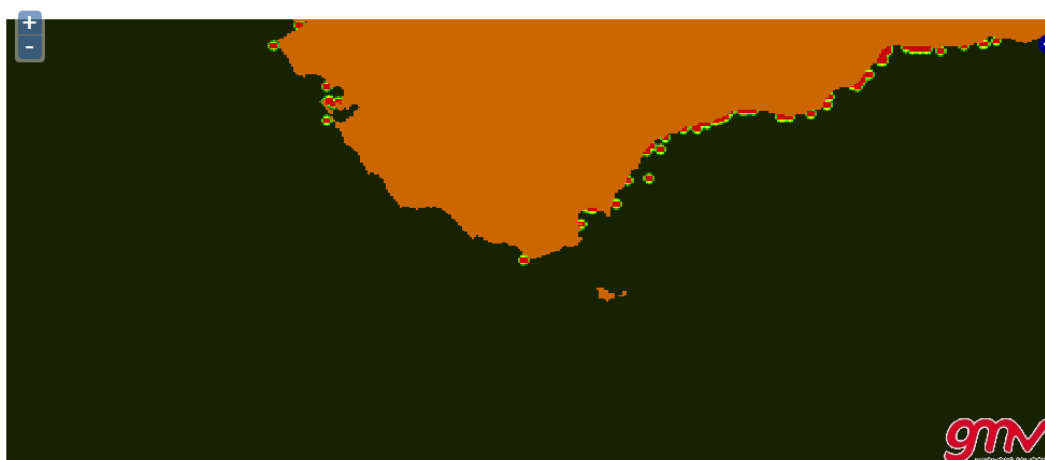
/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_367 = if(isnull(ccaa),R_366, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.803943884111 s=35.331775915361 e=-3.852094453125 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_366
output="C:\Users\GSI\OceanLider\Resultados\14\RI_R_366.tif"
r.out.gdal input=R_367
output="C:\Users\GSI\OceanLider\Resultados\14\RF_R_367.tif"
```

Visualización del resultado:



- ✓ **Viabilidad de instalación en zonas restringidas por la existencia de vertidos de material militar.** Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de vertidos de material militar. Las medidas se aplican en Km. Se debe introducir los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Los valores de los parámetros introducidos en la interfaz:

Viabilidad de instalación en zonas restringidas por la existencia de vertidos de material militar. Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de vertidos de material militar. Las medidas se aplican en Km.

De subida ▾

Valor A	1
Valor B	2
Valor C	99999
Valor D	99999

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el subcriterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */
//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Baleares
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS con condición
de atributos
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=vertido_material_militar
output=R_398
//Rasterización por valor de una capa vectorial
v.to.rast input=R_398 layer=1 type=point,line,area output=R_399 use=val
value=0 rows=4096
//Cálculo de un mapa de distancias
r.grow.distance input=R_399 distance=R_400 metric=Euclidean
//Multiplicación de una capa raster por un escalar
r.mapcalc "R_401 = R_400 * 100"
//Operación difusa
r.mapcalc "R_402 = if(isnull( R_401 ) , null(), if((R_401 <= 1 || R_401
>= 99999), 0.0, if((R_401 > 1 && R_401 < 2), eval((R_401 - 1)/(2 - 1)),
if((R_401 > 99999 && R_401 < 99999 ), eval((99999 - R_401)/(99999 -
99999)), 1.0)))"

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_403 = if(isnull(ccaa),R_402, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.802844848967 s=35.330676880217 e=-3.786176484375 w=-
7.257856171875 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_402
output="C:\Users\GSI\OceanLider\Resultados\20\RI_R_402.tif"
r.out.gdal input=R_403
output="C:\Users\GSI\OceanLider\Resultados\20\RF_R_403.tif"
```

Visualización del resultado:



1.3. Criterios recurso energético

La influencia del recurso energético en la toma de decisión se calcula mediante un único criterio. De exclusión parcial. Es dependiente del tipo de dispositivo: se aplica la densidad de potencia para tipos de dispositivo de oleaje, y la marea media para tipos de dispositivo de corrientes. Se aplica las siguientes fórmulas según los valores del raster de entrada y el parámetro valor_máximo configurado por el usuario.

- **Corrientes** (capa raster *de marea media*):
 - De 0 a VALOR MÁXIMO m/s aplicamos la ecuación: Valor Índice= (velocidad media de mareas vivas y mareas muertas /VALOR MÁXIMO) ^3
 - + de VALOR MÁXIMO m/s: Valor constante de 1

Parámetro	Valor	Unidad
Nombre:	Dispositivo 2	
Tipo de dispositivo:	De corrientes - Fuera de costa: flotante	
Tipo de captador:	Eje horizontal	
Altura de buje:	5	metros
Diámetro rotor:	2	metros
Profundidad máxima:	-50	metros
Profundidad mínima:	-15	metros
Altura de ola máxima admitida:	0.2	metros
Velocidad de la corriente máxima admitida:	0.6	m/s
Potencia unitaria:	10	vatios
Separación entre dispositivos de una misma línea:	2	kilómetros
Separación entre líneas de dispositivos:	2	kilómetros

Los valores de los parámetros introducidos en la interfaz:

Recurso energético

Peso criterio:

Media de las mareas vivas y muertas en condiciones de marea media. Las unidades del valor máximo es en m/s.

Valor Máximo

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el criterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */  
  
//Configuración de la localización y Mapset del proyecto en el entorno  
de trabajo de GRASS  
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84  
//Configuración de la región que engloba a Península, Canarias y  
Baleares
```

```
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa raster ubicada en el sistema de
ficheros
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\medviv.tif"
output=R_412
//Importación en GRASS de una capa raster ubicada en el sistema de
ficheros
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\medmue.tif"
output=R_413
//Media Lineal Ponderada de dos capas raster
r.mapcalc "R_414 = 0.5 * R_412 + 0.5 * R_413"
//Aplicación de una fórmula en un intervalo sobre una capa raster
r.mapcalc "R_415= if(R_414 >= 0 && R_414 < 1.2, eval((R_414/1.2)^3), 1)"
//Establecimiento de un valor para los valores nulos
r.null map=R_415 null=0

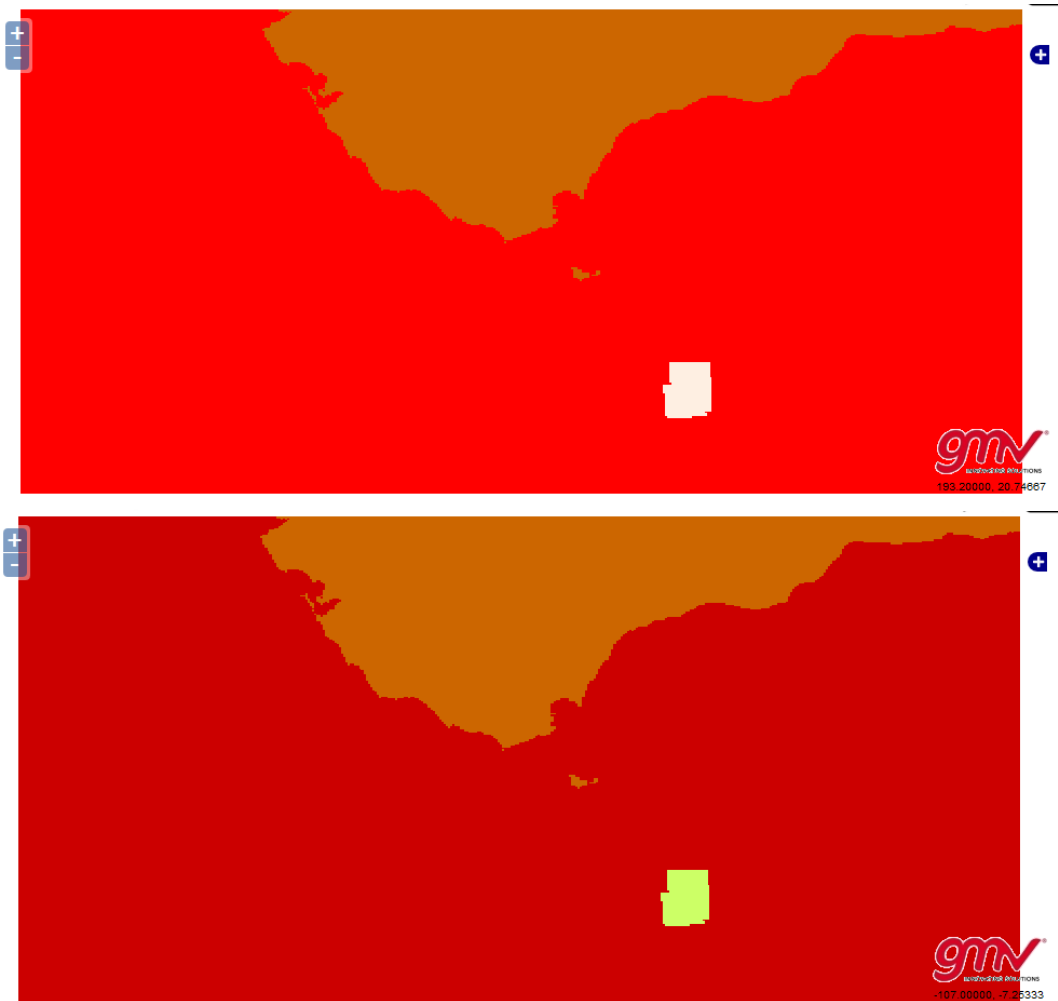
/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_416 = if(isnull(ccaa),R_415, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.803943884111 s=35.134022009111 e=-3.830121796875 w=-
7.279828828125 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_415
output="C:\Users\GSI\OceanLider\Resultados\21\RI_R_415.tif"
r.out.gdal input=R_416
output="C:\Users\GSI\OceanLider\Resultados\21\RF_R_416.tif"
```

Visualización del resultado:



Los valores de los parámetros introducidos en la interfaz:

Recurso energético

Peso criterio:

Media de las mareas vivas y muertas en condiciones de marea media. Las unidades del valor máximo es en m/s.

Valor Máximo

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el criterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */  
  
//Configuración de la localización y Mapset del proyecto en el entorno  
de trabajo de GRASS
```

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Balears
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa raster ubicada en el sistema de
ficheros
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\medviv.tif"
output=R_420
//Importación en GRASS de una capa raster ubicada en el sistema de
ficheros
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\medmue.tif"
output=R_421
//Media Lineal Ponderada de dos capas raster
r.mapcalc "R_422 = 0.5 * R_420 + 0.5 * R_421"
//Aplicación de una fórmula en un intervalo sobre una capa raster
r.mapcalc "R_423= if(R_422 >= 0 && R_422 < 0.6, eval((R_422/0.6)^3), 1)"
//Establecimiento de un valor para los valores nulos
r.null map=R_423 null=0

/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_424 = if(isnull(ccaa),R_423, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.803943884111 s=35.265857946611 e=-3.676313203125 w=-
7.587446015625 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_423
output="C:\Users\GSI\OceanLider\Resultados\23\RI_R_423.tif"
r.out.gdal input=R_424
output="C:\Users\GSI\OceanLider\Resultados\23\RF_R_424.tif"
```

Visualización del resultado:



- **Oleaje** (capa raster de *potencia*):
 - *De 0 a VALOR MÁXIMO KWh/m* aplicamos la ecuación: Valor índice= Valor del mapa raster/VALOR MÁXIMO
 - *+de VALOR MÁXIMO KWh/m*: Valor constante de 1


Características del dispositivo

Nombre:	<input type="text" value="Dispositivo 1"/>
Tipo de dispositivo:	<input type="text" value="Undimotriz - Fuera de costa: flotante"/>
Largo:	<input type="text" value="5"/> metros
Ancho:	<input type="text" value="2"/> metros
Alto:	<input type="text" value="10"/> metros
Profundidad máxima:	<input type="text" value="-100"/> metros
Profundidad mínima:	<input type="text" value="-20"/> metros
Altura de ola máxima admitida:	<input type="text" value="1.6"/> metros
Velocidad de la corriente máxima admitida:	<input type="text" value="0.6"/> m/s
Potencia unitaria:	<input type="text" value="10"/> vatios
Separación entre dispositivos de una misma línea:	<input type="text" value="2"/> kilómetros
Separación entre líneas de dispositivos:	<input type="text" value="2"/> kilómetros

Los valores de los parámetros introducidos en la interfaz:

Recurso energético

Peso criterio:

Densidad. Las unidades del valor máximo es en KWh/m. 

Valor Máximo

El script generado por la herramienta se muestra a continuación. Se han añadido comentarios para entender las operaciones realizadas en el criterio.

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */
```

Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

```
//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Balears
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa raster ubicada en el sistema de
ficheros
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\pw.tif" output=R_417
//Aplicación de una fórmula en un intervalo sobre una capa raster
r.mapcalc "R_418= if(R_417 >= 0 && R_417 < 0.3, eval(R_417/0.3), 1)"
//Establecimiento de un valor para los valores nulos
r.null map=R_418 null=0

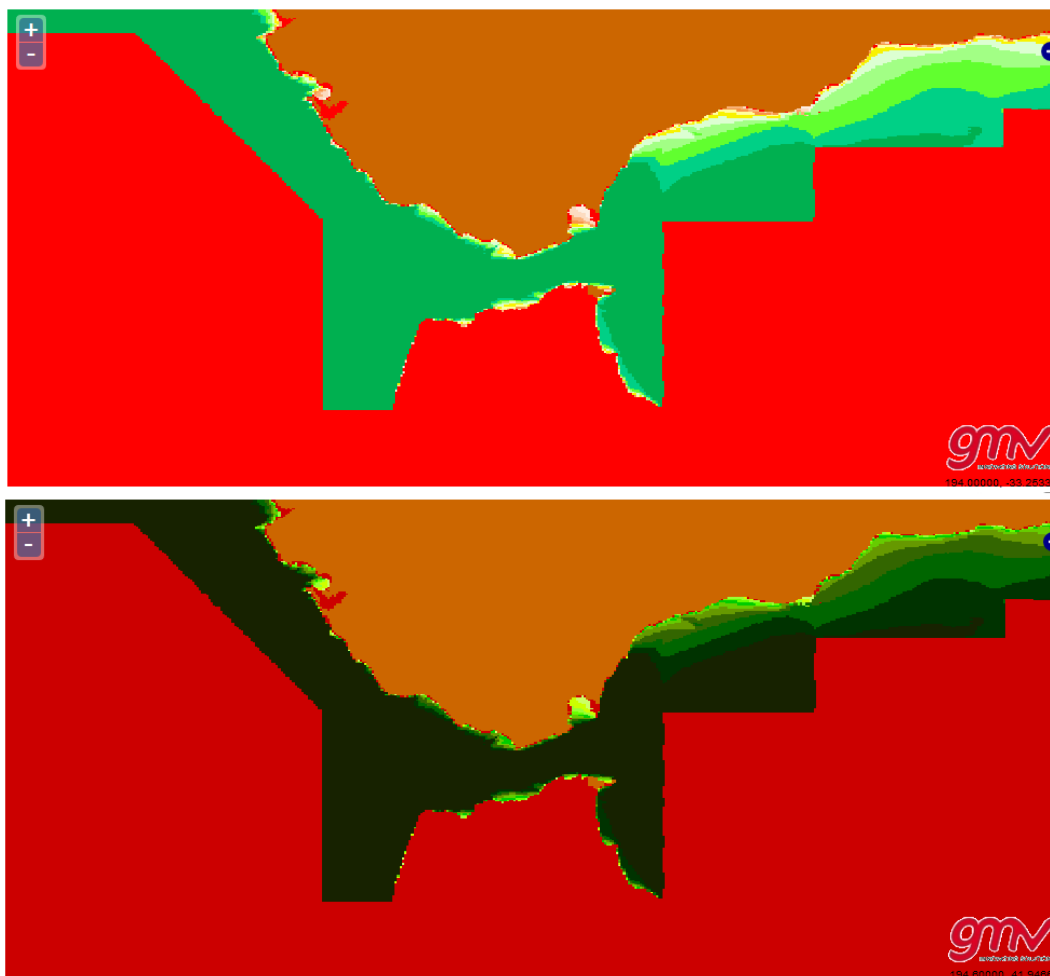
/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_419 = if(isnull(ccaa),R_418, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.825916540361 s=35.243885290361 e=-3.808149140625 w=-
7.301801484375 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_418
output="C:\Users\GSI\OceanLider\Resultados\22\RI_R_418.tif"
r.out.gdal input=R_419
output="C:\Users\GSI\OceanLider\Resultados\22\RF_R_419.tif"
```

Visualización del resultado:




1.4. Criterios medioambientales

Se ha creído conveniente realizar los comentarios de las operaciones del script operacional del primer subcriterio medioambiental, se aplica de forma similar para los scripts operacionales de los demás subcriterios.

- **Espacios protegidos.** De exclusión parcial. El usuario puede seleccionar varios subcriterios. Si se seleccionan más de uno se le da la opción al usuario de escoger el método de razonamiento para la combinación de los subcriterios: más que pesimista (multiplicación de valores), pesimista (mínimo), optimista (máximo) y neutro (MLP).
- ✓ **Geoparques.** Se aplican las áreas correspondientes a geoparques configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la

distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

<input checked="" type="checkbox"/> Geoparques 	
Valor de idoneidad	<input type="text" value="0.0"/>
Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="4"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

El script generado por la herramienta:

```
/* ***** OPERACIONES DE CONFIGURACIÓN ***** */

//Configuración de la localización y Mapset del proyecto en el entorno
de trabajo de GRASS
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
//Configuración de la región que engloba a Península, Canarias y
Baleares
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453

/* ***** OPERACIONES PERTENECIENTES AL SUBCRITERIO ***** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=geoparques output=R_762
//Rasterización por valor de una capa vectorial
v.to.rast input=R_762 layer=1 type=point,line,area output=R_763 use=val
value=0.0 rows=4096
//Creación de un mapa de distancias
r.grow.distance input=R_763 distance=R_764 metric=Euclidean
//Multiplicación de una capa raster por un escalar
r.mapcalc "R_765 = R_764 * 100"
//Operación difusa
r.mapcalc "R_766 = if(isnull( R_765 ), null(), if((R_765 <= 1 || R_765
>= 99999), 0.0, if((R_765 > 1 && R_765 < 4), eval((R_765 - 1)/(4 -1 )),
if((R_765 > 99999 && R_765 < 99999 ), eval((99999 - R_765)/(99999 -
99999)), 1.0)))"
//Operación de unión de dos capas raster
r.patch input=R_763,R_766 output=R_767
```

```
/* ** OPERACIONES DE EXTRACCIÓN DE ZONA TERRESTRE DEL MAPA FINAL ** */

//Importación en GRASS de una capa vectorial desde PostGIS
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
//Rasterización por valor de una capa vectorial
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
//Superposición de la zona terrestre sobre el mapa final
r.mapcalc "R_768 = if(isnull(ccaa),R_767, -1.0)"

/* ***** OPERACIONES DE EXPORTACIÓN DE MAPAS RASTER ***** */

//Configuración de la región a los parámetros configurados por el
usuario
g.region n=36.891834509111 s=35.134022009111 e=-3.808149140625 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
//Exportación de mapas raster final e intermedio
r.out.gdal input=R_767
output="C:\Users\GSI\OceanLider\Resultados\33\RI_R_767.tif"
r.out.gdal input=R_768
output="C:\Users\GSI\OceanLider\Resultados\33\RF_R_768.tif"
```

Visualización del resultado:





- ✓ **Reservas de la biosfera.** Se aplican las áreas correspondientes a las reservas de la biosfera configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

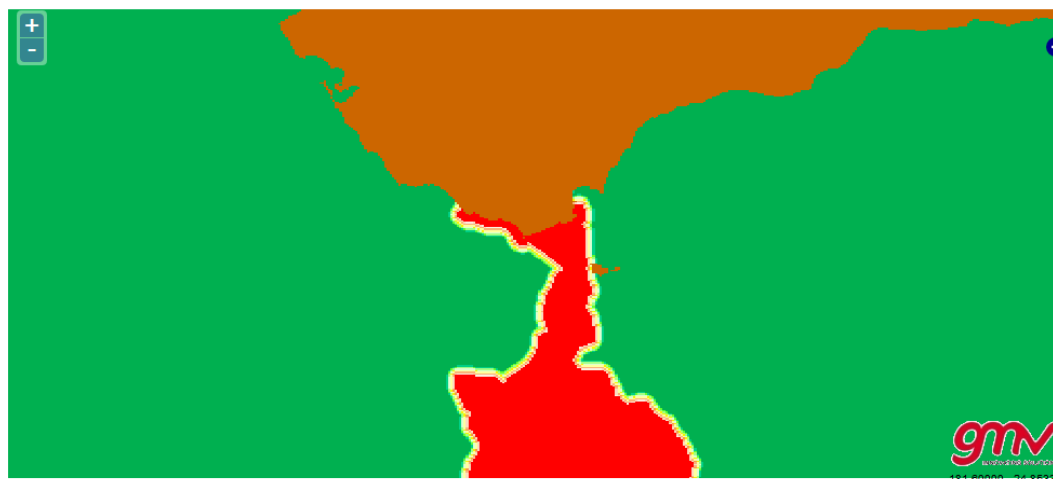
<input checked="" type="checkbox"/>	Reservas de la biosfera	
Valor de idoneidad	<input type="text" value="0.0"/>	
Valor A	<input type="text" value="1"/>	
Valor B	<input type="text" value="4"/>	
Valor C	<input type="text" value="99999"/>	
Valor D	<input type="text" value="99999"/>	
Peso del subcriterio:	<input type="text" value="0.0"/>	

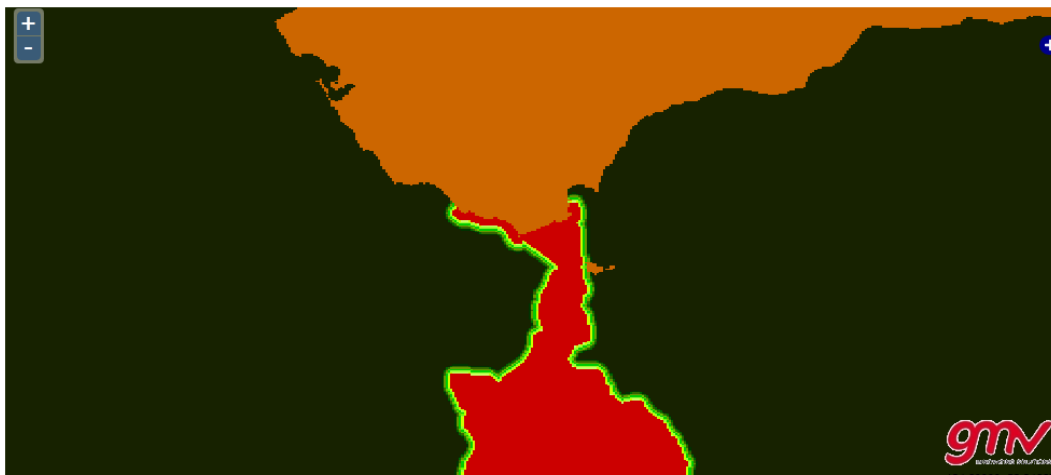
El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mabs output=R_453
v.to.rast input=R_453 layer=1 type=point,line,area output=R_454 use=val
value=0.0 rows=4096
```

```
r.grow.distance input=R_454 distance=R_455 metric=euclidean
r.mapcalc "R_456 = R_455 * 100"
r.mapcalc "R_457 = if(isnull( R_456 ), null(), if((R_456 <= 1 || R_456
>= 99999), 0.0, if((R_456 > 1 && R_456 < 4), eval((R_456 - 1)/(4 -1 )),
if((R_456 > 99999 && R_456 < 99999 ), eval((99999 - R_456 )/(99999 -
99999)), 1.0))))"
r.patch input=R_454,R_457 output=R_458
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_459 = if(isnull(ccaa),R_458, -1.0)"
g.region n=36.781971227861 s=35.177967321611 e=-3.764203828125 w=-
7.367719453125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_458
output="C:\Users\GSI\OceanLider\Resultados\26\RI_R_458.tif"
r.out.gdal input=R_459
output="C:\Users\GSI\OceanLider\Resultados\26\RF_R_459.tif"
```


Visualización del resultado:





- ✓ **Patrimonio de la humanidad.** Se aplican las áreas correspondientes a patrimonio de la humanidad configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Patrimonio de la humanidad 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=p_humanidad
output=R_467
v.to.rast input=R_467 layer=1 type=point,line,area output=R_468 use=val
value=0.5 rows=4096
```

```
r.grow.distance input=R_468 distance=R_469 metric=euclidean
r.mapcalc "R_470 = R_469 * 100"
r.mapcalc "R_471 = if(isnull( R_470 ), null(), if((R_470 <= 1 || R_470
>= 99999), 0.0, if((R_470 > 1 && R_470 < 4), eval((R_470 - 1)/(4 -1 )),
if((R_470 > 99999 && R_470 < 99999 ), eval((99999 - R_470 )/(99999 -
99999)), 1.0))))"
r.patch input=R_468,R_471 output=R_472
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_473 = if(isnull(ccaa),R_472, -1.0)"
g.region n=36.957752477861 s=35.068104040361 e=-3.830121796875 w=-
7.367719453125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_472
output="C:\Users\GSI\OceanLider\Resultados\27\RI_R_472.tif"
r.out.gdal input=R_473
output="C:\Users\GSI\OceanLider\Resultados\27\RF_R_473.tif"
```


Visualización del resultado:





- ✓ **Red de Áreas Marinas Protegidas del Convenio OSPAR.** Se aplican las áreas correspondientes a la red de áreas marinas protegidas del convenio OSPAR configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Red de Áreas Marinas Protegidas del Convenio OSPAR 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ospar_amps output=R_769
v.to.rast input=R_769 layer=1 type=point,line,area output=R_770 use=val
value=0.0 rows=4096
```

```
r.grow.distance input=R_770 distance=R_771 metric=euclidean
r.mapcalc "R_772 = R_771 * 100"
r.mapcalc "R_773 = if(isnull( R_772 ), null(), if((R_772 <= 1 || R_772
>= 99999), 0.0, if((R_772 > 1 && R_772 < 4), eval((R_772 - 1)/(4 -1 )),
if((R_772 > 99999 && R_772 < 99999 ), eval((99999 - R_772 )/(99999 -
99999)), 1.0))))"
r.patch input=R_770,R_773 output=R_774
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_775 = if(isnull(ccaa),R_774, -1.0)"
g.region n=36.846790161467 s=35.176868286467 e=-3.939985078125 w=-
7.257856171875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_774
output="C:\Users\GSI\OceanLider\Resultados\35\RI_R_774.tif"
r.out.gdal input=R_775
output="C:\Users\GSI\OceanLider\Resultados\35\RF_R_775.tif"
```


Visualización del resultado:





- ✓ **Zonas Especialmente Protegidas de Importancia para el Mediterráneo.** Se aplican las áreas correspondientes a las zonas especialmente protegidas de importancia para el Mediterráneo configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Zonas Especialmente Protegidas de Importancia para el Mediterráneo 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepims output=R_776
```

```
v.to.rast input=R_776 layer=1 type=point,line,area output=R_777 use=val
value=0.0 rows=4096
r.grow.distance input=R_777 distance=R_778 metric=euclidean
r.mapcalc "R_779 = R_778 * 100"
r.mapcalc "R_780 = if(isnull( R_779 ), null(), if((R_779 <= 1 || R_779
>= 99999), 0.0, if((R_779 > 1 && R_779 < 4), eval((R_779 - 1)/(4 -1 )),
if((R_779 > 99999 && R_779 < 99999 ), eval((99999 - R_779 )/(99999 -
99999)), 1.0))))"
r.patch input=R_777,R_780 output=R_781
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_782 = if(isnull(ccaa),R_781, -1.0)"
g.region n=36.869861852861 s=35.155994665361 e=-3.742231171875 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_781
output="C:\Users\GSI\OceanLider\Resultados\36\RI_R_781.tif"
r.out.gdal input=R_782
output="C:\Users\GSI\OceanLider\Resultados\36\RF_R_782.tif"
```

Visualización del resultado:





- ✓ **Sitios reconocidos con el Diploma Europeo de Calidad de Espacios Protegidos.** Se aplican las áreas correspondientes a los sitios reconocidos con el Diploma Europeo de Calidad de Espacios Protegidos configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Sitios reconocidos con el Diploma Europeo de Calidad de Espacios Protegidos 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=dipeuro output=R_783
v.to.rast input=R_783 layer=1 type=point,line,area output=R_784 use=val
value=0.0 rows=4096
```

```
r.grow.distance input=R_784 distance=R_785 metric=euclidean
r.mapcalc "R_786 = R_785 * 100"
r.mapcalc "R_787 = if(isnull( R_786 ), null(), if((R_786 <= 1 || R_786
>= 99999), 0.0, if((R_786 > 1 && R_786 < 4), eval((R_786 - 1)/(4 -1 )),
if((R_786 > 99999 && R_786 < 99999 ), eval((99999 - R_786 )/(99999 -
99999)), 1.0))))"
r.patch input=R_784,R_787 output=R_788
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_789 = if(isnull(ccaa),R_788, -1.0)"
g.region n=36.825916540361 s=35.221912634111 e=-3.764203828125 w=-
7.389692109375 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_788
output="C:\Users\GSI\OceanLider\Resultados\37\RI_R_788.tif"
r.out.gdal input=R_789
output="C:\Users\GSI\OceanLider\Resultados\37\RF_R_789.tif"
```


Visualización del resultado:





- ✓ **Lugares de Importancia Comunitaria.** Se aplican las áreas correspondientes a los lugares de importancia comunitaria configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Lugares de Importancia Comunitaria 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

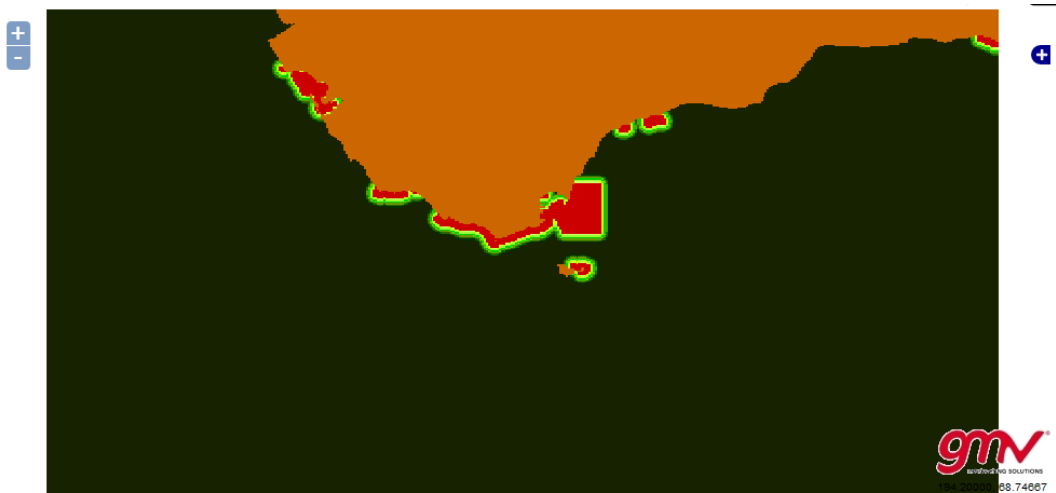
El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=lics output=R_790
v.to.rast input=R_790 layer=1 type=point,line,area output=R_791 use=val
value=0.0 rows=4096
r.grow.distance input=R_791 distance=R_792 metric=euclidean
r.mapcalc "R_793 = R_792 * 100"
```

```
r.mapcalc "R_794 = if(isnull( R_793 ) , null(), if((R_793 <= 1 || R_793
>= 99999), 0.0, if((R_793 > 1 && R_793 < 4), eval((R_793 - 1)/(4 -1 )),
if((R_793 > 99999 && R_793 < 99999 ), eval((99999 - R_793 )/(99999 -
99999)), 1.0))))"
r.patch input=R_791,R_794 output=R_795
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_796 = if(isnull(ccaa),R_795, -1.0)"
g.region n=36.847889196611 s=35.068104040361 e=-3.764203828125 w=-
7.257856171875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_795
output="C:\Users\GSI\OceanLider\Resultados\38\RI_R_795.tif"
r.out.gdal input=R_796
output="C:\Users\GSI\OceanLider\Resultados\38\RF_R_796.tif"
```


Visualización del resultado:





- ✓ **Zonas de Especial Protección para las Aves.** Se aplican las áreas correspondientes a las zonas de especial protección para las aves configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Zonas de Especial Protección para las Aves 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

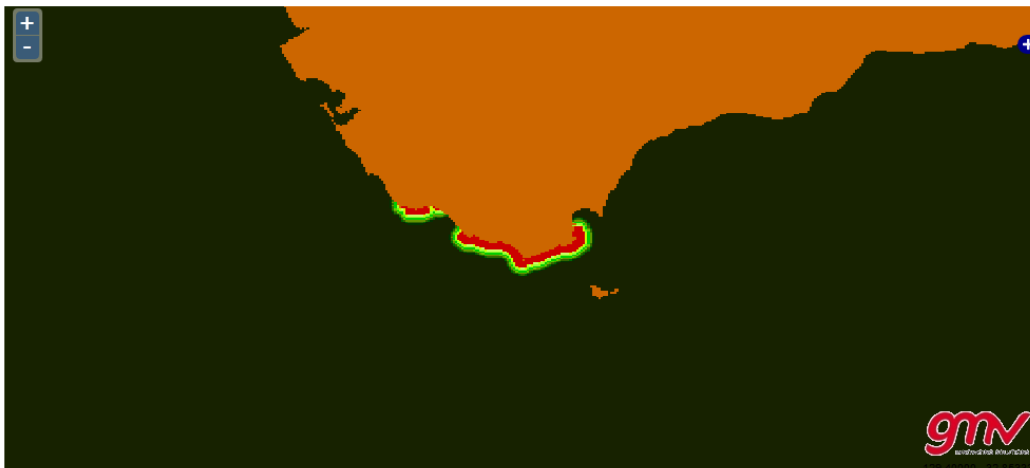
El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepas output=R_797
```

```
v.to.rast input=R_797 layer=1 type=point,line,area output=R_798 use=val
value=0.0 rows=4096
r.grow.distance input=R_798 distance=R_799 metric=euclidean
r.mapcalc "R_800 = R_799 * 100"
r.mapcalc "R_801 = if(isnull( R_800 ), null(), if((R_800 <= 1 || R_800
>= 99999), 0.0, if((R_800 > 1 && R_800 < 4), eval((R_800 - 1)/(4 -1 )),
if((R_800 > 99999 && R_800 < 99999 ), eval((99999 - R_800 )/(99999 -
99999)), 1.0))))"
r.patch input=R_798,R_801 output=R_802
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_803 = if(isnull(ccaa),R_802, -1.0)"
g.region n=36.869861852861 s=35.287830602861 e=-3.852094453125 w=-
7.389692109375 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_802
output="C:\Users\GSI\OceanLider\Resultados\39\RI_R_802.tif"
r.out.gdal input=R_803
output="C:\Users\GSI\OceanLider\Resultados\39\RF_R_803.tif"
```

Visualización del resultado:





- ✓ **Parques Nacionales.** Se aplican las áreas correspondientes a los parques nacionales configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

<input checked="" type="checkbox"/> Parques Nacionales	
Valor de idoneidad	<input type="text" value="0.0"/>
Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="4"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ppnn output=R_987
v.to.rast input=R_987 layer=1 type=point,line,area output=R_988 use=val
value=0.0 rows=4096
r.grow.distance input=R_988 distance=R_989 metric=euclidean
```

```
r.mapcalc "R_990 = R_989 * 100"
r.mapcalc "R_991 = if(isnull( R_990 ), null(), if((R_990 <= 1 || R_990
>= 99999), 0.0, if((R_990 > 1 && R_990 < 4), eval((R_990 - 1)/(4 - 1)),
if((R_990 > 99999 && R_990 < 99999 ), eval((99999 - R_990 )/(99999 -
99999)), 1.0)))"
r.patch input=R_988,R_991 output=R_992
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_993 = if(isnull(ccaa),R_992, -1.0)"
g.region n=36.890735473967 s=35.067005005217 e=-3.874067109375 w=-
7.301801484375 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_992
output="C:\Users\GSI\OceanLider\Resultados\40\RI_R_992.tif"
r.out.gdal input=R_993
output="C:\Users\GSI\OceanLider\Resultados\40\RF_R_993.tif"
```

Visualización del resultado:





- ✓ **Biotopos protegidos.** Se aplican las áreas correspondientes a los biotopos protegidos configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

<input checked="" type="checkbox"/> Biotopos protegidos	
Valor de idoneidad	<input type="text" value="0.0"/>
Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="4"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=bbppro output=R_811
```

```
v.to.rast input=R_811 layer=1 type=point,line,area output=R_812 use=val
value=0.0 rows=4096
r.grow.distance input=R_812 distance=R_813 metric=euclidean
r.mapcalc "R_814 = R_813 * 100"
r.mapcalc "R_815 = if(isnull( R_814 ), null(), if((R_814 <= 1 || R_814
>= 99999), 0.0, if((R_814 > 1 && R_814 < 4), eval((R_814 - 1)/(4 -1 )),
if((R_814 > 99999 && R_814 < 99999 ), eval((99999 - R_814 )/(99999 -
99999)), 1.0))))"
r.patch input=R_812,R_815 output=R_816
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_817 = if(isnull(ccaa),R_816, -1.0)"
g.region n=36.913807165361 s=35.221912634111 e=-3.676313203125 w=-
7.433637421875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_816
output="C:\Users\GSI\OceanLider\Resultados\41\RI_R_816.tif"
r.out.gdal input=R_817
output="C:\Users\GSI\OceanLider\Resultados\41\RF_R_817.tif"
```

Visualización del resultado:





- ✓ **Microrreservas.** Se aplican las áreas correspondientes a las microrreservas configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

<input checked="" type="checkbox"/> Microrreservas	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=micror output=R_994
v.to.rast input=R_994 layer=1 type=point,line,area output=R_995 use=val
value=0.0 rows=4096
r.grow.distance input=R_995 distance=R_996 metric=euclidean
r.mapcalc "R_997 = R_996 * 100"
```

```
r.mapcalc "R_998 = if(isnull( R_997 ) , null(), if((R_997 <= 1 || R_997
>= 99999), 0.0, if((R_997 > 1 && R_997 < 4), eval((R_997 - 1)/(4 -1 )),
if((R_997 > 99999 && R_997 < 99999 ), eval((99999 - R_997 )/(99999 -
99999)), 1.0)))"
r.patch input=R_995,R_998 output=R_999
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1000 = if(isnull(ccaa),R_999, -1.0)"
g.region n=36.847889196611 s=35.002186071611 e=-3.830121796875 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_999
output="C:\Users\GSI\OceanLider\Resultados\42\RI_R_999.tif"
r.out.gdal input=R_1000
output="C:\Users\GSI\OceanLider\Resultados\42\RF_R_1000.tif"
```

Visualización del resultado:





- ✓ **Monumentos Naturales.** Se aplican las áreas correspondientes a los monumentos naturales configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

<input checked="" type="checkbox"/> Monumentos Naturales	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mmnat output=R_825
v.to.rast input=R_825 layer=1 type=point,line,area output=R_826 use=val
value=0.0 rows=4096
r.grow.distance input=R_826 distance=R_827 metric=euclidean
r.mapcalc "R_828 = R_827 * 100"
```

```
r.mapcalc "R_829 = if(isnull( R_828 ) , null(), if((R_828 <= 1 || R_828
>= 99999), 0.0, if((R_828 > 1 && R_828 < 4), eval((R_828 - 1)/(4 -1 )),
if((R_828 > 99999 && R_828 < 99999 ), eval((99999 - R_828 )/(99999 -
99999)), 1.0)))"
r.patch input=R_826,R_829 output=R_830
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_831 = if(isnull(ccaa),R_830, -1.0)"
g.region n=36.890735473967 s=35.154895630217 e=-3.852094453125 w=-
7.235883515625 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_830
output="C:\Users\GSI\OceanLider\Resultados\43\RI_R_830.tif"
r.out.gdal input=R_831
output="C:\Users\GSI\OceanLider\Resultados\43\RF_R_831.tif"
```


Visualización del resultado:





- ✓ **Parajes Naturales.** Se aplican las áreas correspondientes a los parajes naturales configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Parajes Naturales 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=pparnnat output=R_832
v.to.rast input=R_832 layer=1 type=point,line,area output=R_833 use=val
value=0.0 rows=4096
r.grow.distance input=R_833 distance=R_834 metric=euclidean
```

```
r.mapcalc "R_835 = R_834 * 100"
r.mapcalc "R_836 = if(isnull( R_835 ), null(), if((R_835 <= 1 || R_835
>= 99999), 0.0, if((R_835 > 1 && R_835 < 4), eval((R_835 - 1)/(4 -1 )),
if((R_835 > 99999 && R_835 < 99999 ), eval((99999 - R_835 )/(99999 -
99999)), 1.0)))"
r.patch input=R_833,R_836 output=R_837
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_838 = if(isnull(ccaa),R_837, -1.0)"
g.region n=36.913807165361 s=35.287830602861 e=-3.698285859375 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_837
output="C:\Users\GSI\OceanLider\Resultados\44\RI_R_837.tif"
r.out.gdal input=R_838
output="C:\Users\GSI\OceanLider\Resultados\44\RF_R_838.tif"
```


Visualización del resultado:





- ✓ **Parques Naturales.** Se aplican las áreas correspondientes a los parques naturales configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Parques Naturales 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

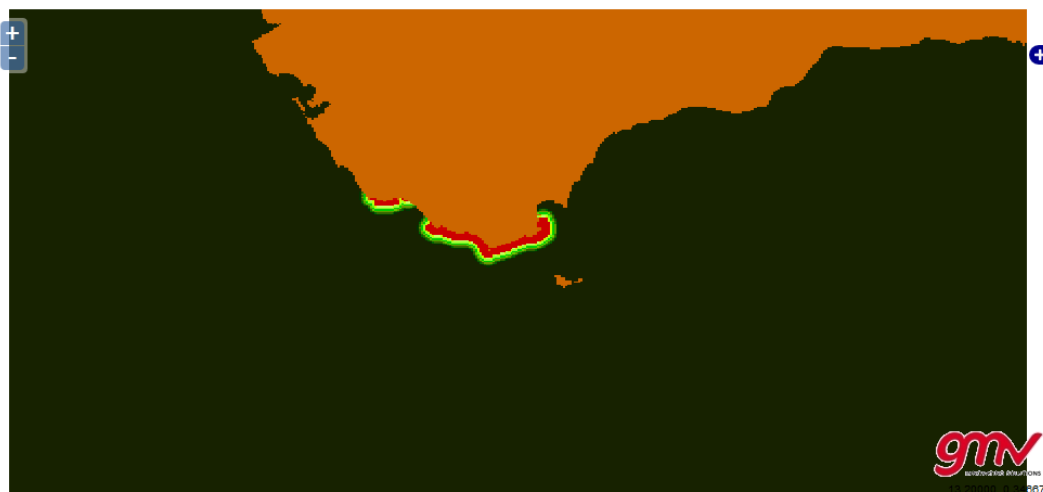
El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ppnat output=R_1001
v.to.rast input=R_1001 layer=1 type=point,line,area output=R_1002
use=val value=0.0 rows=4096
r.grow.distance input=R_1002 distance=R_1003 metric=euclidean
```

```
r.mapcalc "R_1004 = R_1003 * 100"
r.mapcalc "R_1005 = if(isnull( R_1004 ), null(), if((R_1004 <= 1 ||
R_1004 >= 99999), 0.0, if((R_1004 > 1 && R_1004 < 4), eval((R_1004 -
1)/(4 -1 )), if((R_1004 > 99999 && R_1004 < 99999 ), eval((99999 -
R_1004 )/(99999 - 99999)), 1.0))))"
r.patch input=R_1002,R_1005 output=R_1006
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1007 = if(isnull(ccaa),R_1006, -1.0)"
g.region n=36.846790161467 s=35.132922973967 e=-3.720258515625 w=-
7.301801484375 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1006
output="C:\Users\GSI\OceanLider\Resultados\45\RI_R_1006.tif"
r.out.gdal input=R_1007
output="C:\Users\GSI\OceanLider\Resultados\45\RF_R_1007.tif"
```


Visualización del resultado:





- ✓ **Reservas con ley de protección propia.** Se aplican las áreas correspondientes a las reservas con ley de protección propia configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Reservas con ley de protección propia 

Valor de idoneidad	<input type="text" value="0.0"/>
Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="4"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=rr_ley_prop
output=R_846
v.to.rast input=R_846 layer=1 type=point,line,area output=R_847 use=val
value=0.0 rows=4096
```

```
r.grow.distance input=R_847 distance=R_848 metric=euclidean
r.mapcalc "R_849 = R_848 * 100"
r.mapcalc "R_850 = if(isnull( R_849 ), null(), if((R_849 <= 1 || R_849
>= 99999), 0.0, if((R_849 > 1 && R_849 < 4), eval((R_849 - 1)/(4 - 1)),
if((R_849 > 99999 && R_849 < 99999 ), eval((99999 - R_849)/(99999 -
99999)), 1.0))))"
r.patch input=R_847,R_850 output=R_851
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_852 = if(isnull(ccaa),R_851, -1.0)"
g.region n=36.802844848967 s=35.220813598967 e=-3.874067109375 w=-
7.301801484375 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_851
output="C:\Users\GSI\OceanLider\Resultados\46\RI_R_851.tif"
r.out.gdal input=R_852
output="C:\Users\GSI\OceanLider\Resultados\46\RF_R_852.tif"
```


Visualización del resultado:





- ✓ **Reservas Naturales.** Se aplican las áreas correspondientes a las reservas naturales configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Reservas Naturales 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

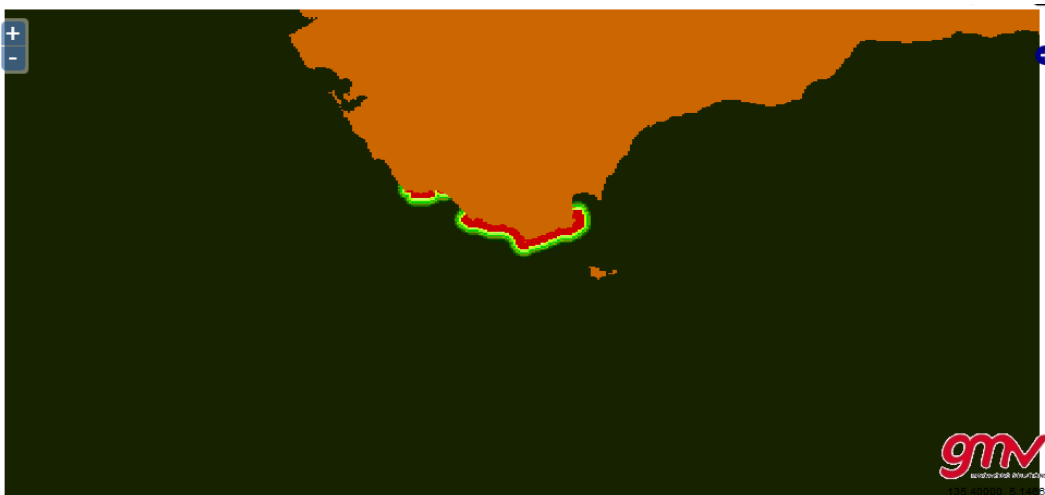
El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=rrnn output=R_853
v.to.rast input=R_853 layer=1 type=point,line,area output=R_854 use=val
value=0.0 rows=4096
r.grow.distance input=R_854 distance=R_855 metric=euclidean
```

```
r.mapcalc "R_856 = R_855 * 100"
r.mapcalc "R_857 = if(isnull( R_856 ), null(), if((R_856 <= 1 || R_856
>= 99999), 0.0, if((R_856 > 1 && R_856 < 4), eval((R_856 - 1)/(4 -1 )),
if((R_856 > 99999 && R_856 < 99999 ), eval((99999 - R_856 )/(99999 -
99999)), 1.0)))"
r.patch input=R_854,R_857 output=R_858
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_859 = if(isnull(ccaa),R_858, -1.0)"
g.region n=36.825916540361 s=35.090076696611 e=-3.786176484375 w=-
7.455610078125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_858
output="C:\Users\GSI\OceanLider\Resultados\47\RI_R_858.tif"
r.out.gdal input=R_859
output="C:\Users\GSI\OceanLider\Resultados\47\RF_R_859.tif"
```


Visualización del resultado:





- ✓ **Zonas de Especial Protección de los Valores Naturales.** Se aplican las áreas correspondientes a las zonas de especial protección de los valores naturales configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Zonas de Especial Protección de los Valores Naturales 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepvn output=R_1008
v.to.rast input=R_1008 layer=1 type=point,line,area output=R_1009
use=val value=0.0 rows=4096
```

```
r.grow.distance input=R_1009 distance=R_1010 metric=euclidean
r.mapcalc "R_1011 = R_1010 * 100"
r.mapcalc "R_1012 = if(isnull( R_1011 ) , null(), if((R_1011 <= 1 ||
R_1011 >= 99999), 0.0, if((R_1011 > 1 && R_1011 < 4), eval((R_1011 -
1)/(4 -1 )), if((R_1011 > 99999 && R_1011 < 99999 ), eval((99999 -
R_1011 )/(99999 - 99999)), 1.0))))"
r.patch input=R_1009,R_1012 output=R_1013
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1014 = if(isnull(ccaa),R_1013, -1.0)"
g.region n=36.825916540361 s=35.134022009111 e=-3.764203828125 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1013
output="C:\Users\GSI\OceanLider\Resultados\48\RI_R_1013.tif"
r.out.gdal input=R_1014
output="C:\Users\GSI\OceanLider\Resultados\48\RF_R_1014.tif"
```


Visualización del resultado:





- **Biodiversidad.** De exclusión parcial. El usuario puede seleccionar varios subcriterios. Si se seleccionan más de uno se le da la opción al usuario de escoger el método de razonamiento para la combinación de los subcriterios: más que pesimista (multiplicación de valores), pesimista (mínimo), optimista (máximo) y neutro (MLP).
 - ✓ *Áreas de interés para la conservación de los cetáceos.* Se aplican las áreas correspondientes a las áreas de interés para la conservación de los cetáceos configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Áreas de interés para la conservación de los cetáceos 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

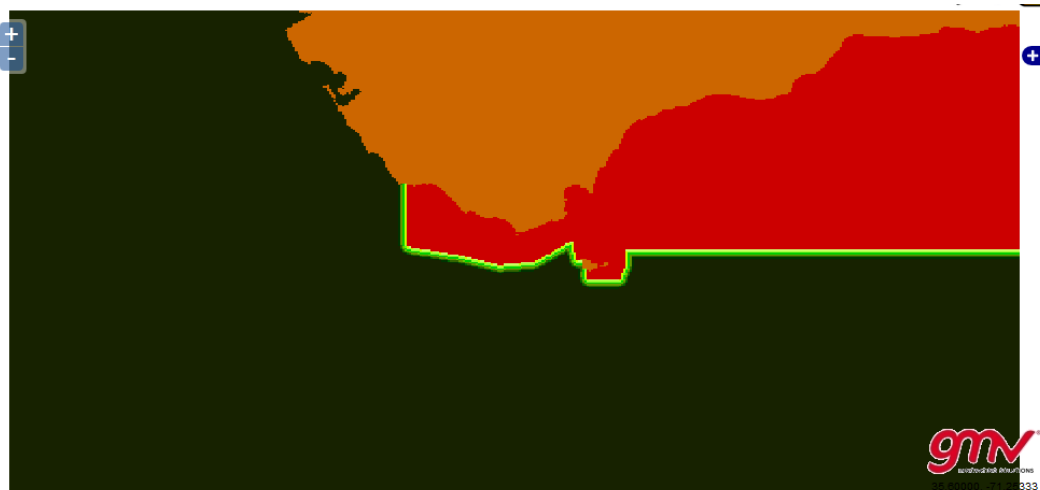
El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
```

```
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=aicp output=R_867
v.to.rast input=R_867 layer=1 type=point,line,area output=R_868 use=val
value=0.0 rows=4096
r.grow.distance input=R_868 distance=R_869 metric=euclidean
r.mapcalc "R_870 = R_869 * 100"
r.mapcalc "R_871 = if(isnull( R_870 ), null(), if((R_870 <= 1 || R_870
>= 99999), 0.0, if((R_870 > 1 && R_870 < 4), eval((R_870 - 1)/(4 -1 )),
if((R_870 > 99999 && R_870 < 99999 ), eval((99999 - R_870 )/(99999 -
99999)), 1.0)))"
r.patch input=R_868,R_871 output=R_872
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_873 = if(isnull(ccaa),R_872, -1.0)"
g.region n=36.803943884111 s=35.068104040361 e=-3.808149140625 w=-
7.433637421875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_872
output="C:\Users\GSI\OceanLider\Resultados\49\RI_R_872.tif"
r.out.gdal input=R_873
output="C:\Users\GSI\OceanLider\Resultados\49\RF_R_873.tif"
```

Visualización del resultado:





- ✓ **Áreas compatibles con la figura de Parque Nacional.** Se aplican las áreas correspondientes a las áreas compatibles con la figura de parque nacional configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Áreas compatibles con la figura de Parque Nacional ⓘ

Valor de idoneidad	<input type="text" value="0.0"/>
Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="4"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=aopn output=R_874
v.to.rast input=R_874 layer=1 type=point,line,area output=R_875 use=val
value=0.0 rows=4096
```

```
r.grow.distance input=R_875 distance=R_876 metric=euclidean
r.mapcalc "R_877 = R_876 * 100"
r.mapcalc "R_878 = if(isnull( R_877 ), null(), if((R_877 <= 1 || R_877
>= 99999), 0.0, if((R_877 > 1 && R_877 < 4), eval((R_877 - 1)/(4 -1 )),
if((R_877 > 99999 && R_877 < 99999 ), eval((99999 - R_877 )/(99999 -
99999)), 1.0))))"
r.patch input=R_875,R_878 output=R_879
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_880 = if(isnull(ccaa),R_879, -1.0)"
g.region n=36.979725134111 s=35.046131384111 e=-3.786176484375 w=-
7.367719453125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_879
output="C:\Users\GSI\OceanLider\Resultados\50\RI_R_879.tif"
r.out.gdal input=R_880
output="C:\Users\GSI\OceanLider\Resultados\50\RF_R_880.tif"
```

Visualización del resultado:





- ✓ **Áreas importantes para las aves.** Se aplican las áreas correspondientes a las áreas importantes para las aves configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

<input checked="" type="checkbox"/>	Áreas importantes para las aves	
Valor de idoneidad	<input type="text"/>	0.0
Valor A	<input type="text"/>	1
Valor B	<input type="text"/>	4
Valor C	<input type="text"/>	99999
Valor D	<input type="text"/>	99999
Peso del subcriterio:	<input type="text"/>	0.0

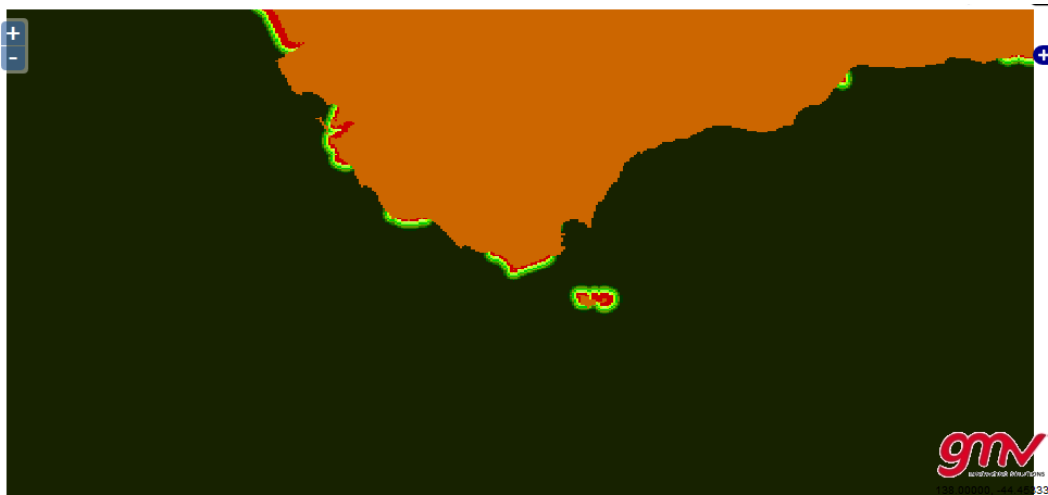
El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ibas output=R_881
v.to.rast input=R_881 layer=1 type=point,line,area output=R_882 use=val
value=0.0 rows=4096
r.grow.distance input=R_882 distance=R_883 metric=euclidean
```

```
r.mapcalc "R_884 = R_883 * 100"
r.mapcalc "R_885 = if(isnull( R_884 ) , null(), if((R_884 <= 1 || R_884
>= 99999), 0.0, if((R_884 > 1 && R_884 < 4), eval((R_884 - 1)/(4 -1 )),
if((R_884 > 99999 && R_884 < 99999 ), eval((99999 - R_884 )/(99999 -
99999)), 1.0)))"
r.patch input=R_882,R_885 output=R_886
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_887 = if(isnull(ccaa),R_886, -1.0)"
g.region n=36.913807165361 s=35.199939977861 e=-3.786176484375 w=-
7.389692109375 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_886
output="C:\Users\GSI\OceanLider\Resultados\51\RI_R_886.tif"
r.out.gdal input=R_887
output="C:\Users\GSI\OceanLider\Resultados\51\RF_R_887.tif"
```

Visualización del resultado:





- ✓ **Zonas propuestas para realizar el Inventario de la biodiversidad marina.** Se aplican las áreas correspondientes a las zonas propuestas para realizar el Inventario de la biodiversidad marina configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Zonas propuestas para realizar el Inventario de la biodiversidad marina

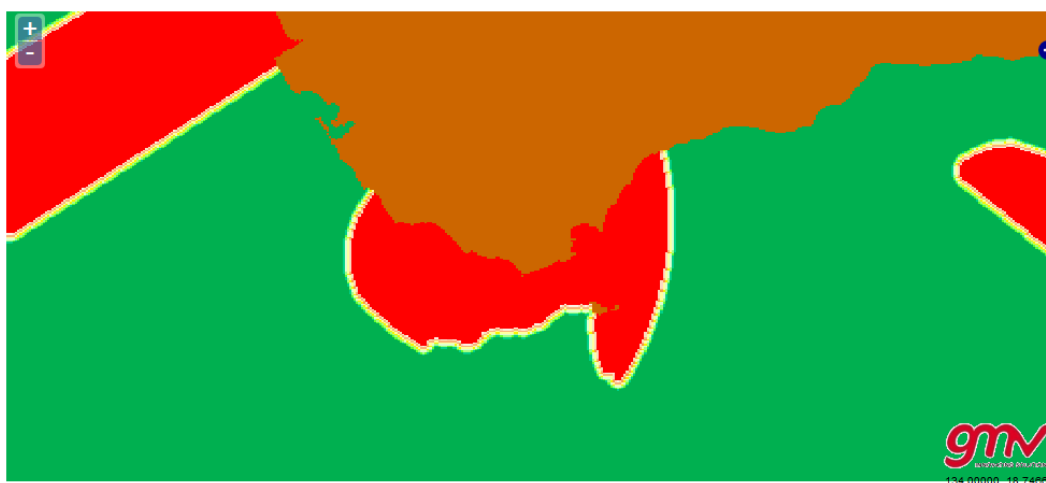
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

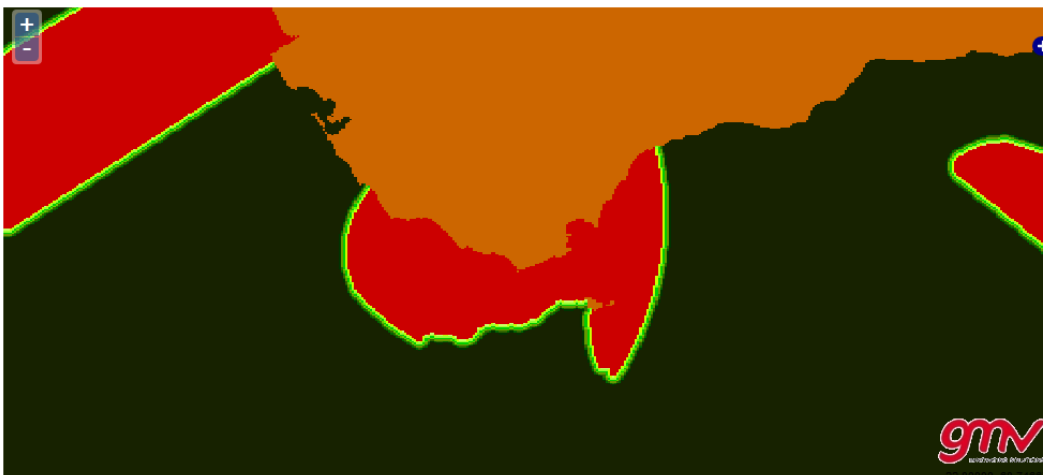
El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ibm output=R_888
v.to.rast input=R_888 layer=1 type=point,line,area output=R_889 use=val
value=0.0 rows=4096
r.grow.distance input=R_889 distance=R_890 metric=euclidean
```

```
r.mapcalc "R_891 = R_890 * 100"
r.mapcalc "R_892 = if(isnull( R_891 ) , null(), if((R_891 <= 1 || R_891
>= 99999), 0.0, if((R_891 > 1 && R_891 < 4), eval((R_891 - 1)/(4 -1 )),
if((R_891 > 99999 && R_891 < 99999 ), eval((99999 - R_891 )/(99999 -
99999)), 1.0)))"
r.patch input=R_889,R_892 output=R_893
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_894 = if(isnull(ccaa),R_893, -1.0)"
g.region n=36.891834509111 s=35.309803259111 e=-3.808149140625 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_893
output="C:\Users\GSI\OceanLider\Resultados\52\RI_R_893.tif"
r.out.gdal input=R_894
output="C:\Users\GSI\OceanLider\Resultados\52\RF_R_894.tif"
```


Visualización del resultado:





- ✓ **Áreas importantes para las aves marinas.** Se aplican las áreas correspondientes a las áreas importantes para las aves marinas configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Áreas importantes para las aves marinas 

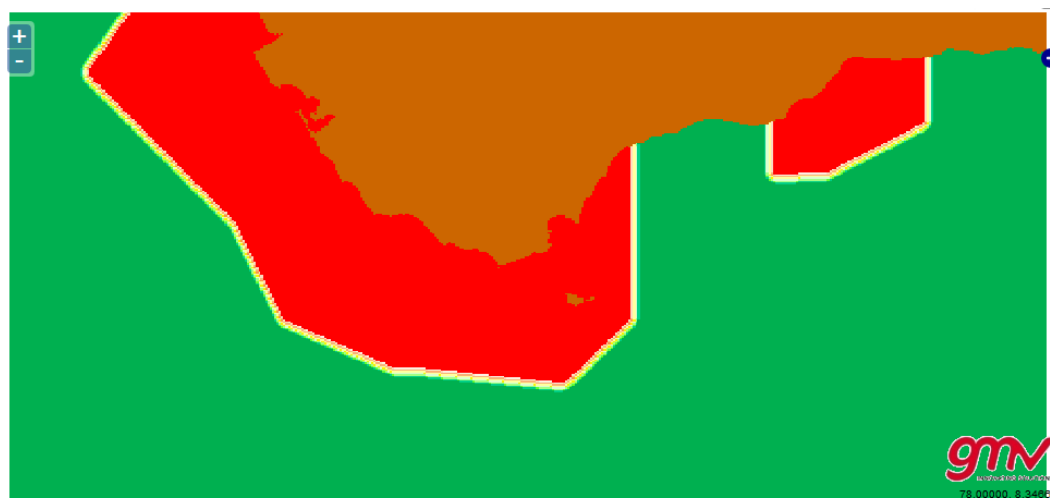
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

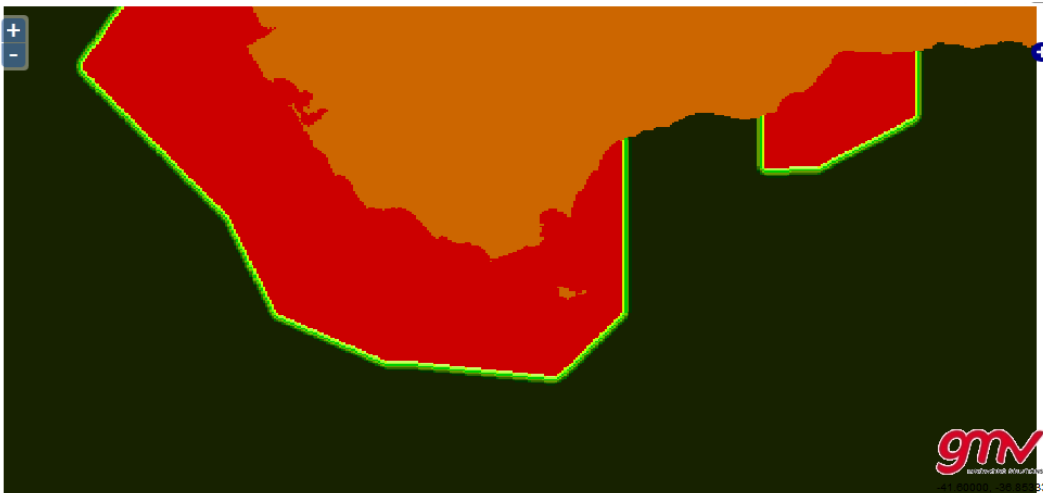
El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=iba_recom output=R_1015
v.to.rast input=R_1015 layer=1 type=point,line,area output=R_1016
use=val value=0.0 rows=4096
r.grow.distance input=R_1016 distance=R_1017 metric=euclidean
r.mapcalc "R_1018 = R_1017 * 100"
```

```
r.mapcalc "R_1019 = if(isnull( R_1018 ) , null(), if((R_1018 <= 1 ||
R_1018 >= 99999), 0.0, if((R_1018 > 1 && R_1018 < 4), eval((R_1018 -
1)/(4 -1 )), if((R_1018 > 99999 && R_1018 < 99999 ), eval((99999 -
R_1018 )/(99999 - 99999)), 1.0))))"
r.patch input=R_1016,R_1019 output=R_1020
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1021 = if(isnull(ccaa),R_1020, -1.0)"
g.region n=36.869861852861 s=35.199939977861 e=-3.742231171875 w=-
7.279828828125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1020
output="C:\Users\GSI\OceanLider\Resultados\53\RI_R_1020.tif"
r.out.gdal input=R_1021
output="C:\Users\GSI\OceanLider\Resultados\53\RF_R_1021.tif"
```


Visualización del resultado:





- ✓ **Zona marina especialmente sensible.** Se aplican las áreas correspondientes a las zonas marinas especialmente sensibles configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Zona marina especialmente sensible 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zmes output=R_5691
v.to.rast input=R_5691 layer=1 type=point,line,area output=R_5692
use=val value=0.0 rows=4096
r.grow.distance input=R_5692 distance=R_5693 metric=euclidean
```

```
r.mapcalc "R_5694 = R_5693 * 100"
r.mapcalc "R_5695 = if(isnull( R_5694 ), null(), if((R_5694 <= 1 ||
R_5694 >= 99999), 0.0, if((R_5694 > 1 && R_5694 < 4), eval((R_5694 -
1)/(4 -1 )), if((R_5694 > 99999 && R_5694 < 99999 ), eval((99999 -
R_5694 )/(99999 - 99999)), 1.0))))"
r.patch input=R_5692,R_5695 output=R_5696
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_5697 = if(isnull(ccaa),R_5696, -1.0)"
g.region n=36.824817505217 s=34.869251098967 e=-3.786176484375 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_5696
output="C:\Users\GSI\OceanLider\Resultados\54\RI_R_5696.tif"
r.out.gdal input=R_5697
output="C:\Users\GSI\OceanLider\Resultados\54\RF_R_5697.tif"
```


Visualización del resultado:





- ✓ **Rutas migratorias de cetáceos.** Se aplican las áreas correspondientes a las rutas migratorias de los cetáceos configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Rutas migratorias de cetáceos 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

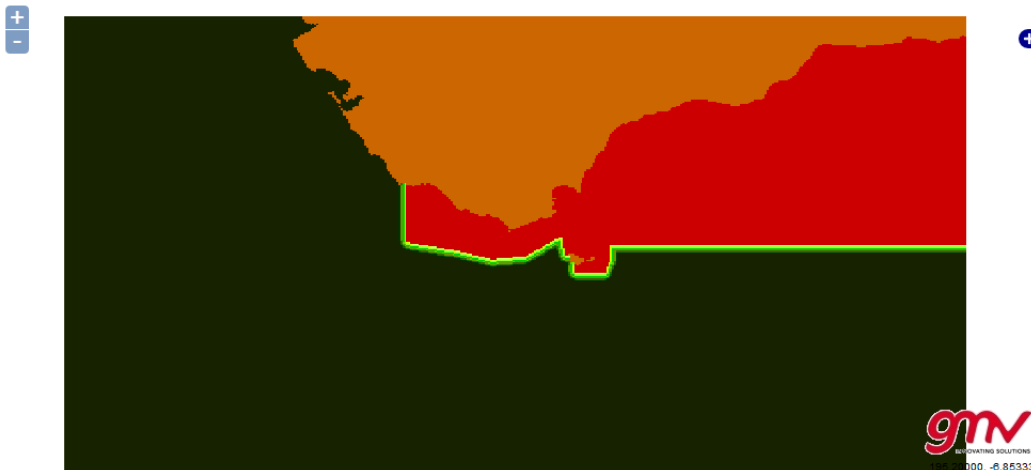
El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=cetaceos output=R_909
v.to.rast input=R_909 layer=1 type=point,line,area output=R_910 use=val
value=0.0 rows=4096
r.grow.distance input=R_910 distance=R_911 metric=euclidean
```

```
r.mapcalc "R_912 = R_911 * 100"
r.mapcalc "R_913 = if(isnull( R_912 ), null(), if((R_912 <= 1 || R_912
>= 99999), 0.0, if((R_912 > 1 && R_912 < 4), eval((R_912 - 1)/(4 -1 )),
if((R_912 > 99999 && R_912 < 99999 ), eval((99999 - R_912 )/(99999 -
99999)), 1.0)))"
r.patch input=R_910,R_913 output=R_914
  v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_915 = if(isnull(ccaa),R_914, -1.0)"
g.region n=36.912708130217 s=35.067005005217 e=-3.852094453125 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_914
output="C:\Users\GSI\OceanLider\Resultados\55\RI_R_914.tif"
r.out.gdal input=R_915
output="C:\Users\GSI\OceanLider\Resultados\55\RF_R_915.tif"
```


Visualización del resultado:





- **Usos pesqueros.** De exclusión parcial. El usuario puede seleccionar varios subcriterios. Si se seleccionan más de uno se le da la opción al usuario de escoger el método de razonamiento para la combinación de los subcriterios: más que pesimista (multiplicación de valores), pesimista (mínimo), optimista (máximo) y nuetro (MLP).
 - ✓ **Instalaciones de acuicultura.** Se aplican las áreas correspondientes a las instalaciones de acuicultura configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Instalaciones de acuicultura 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
```

```

g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=acuicultura
output=R_916
v.to.rast input=R_916 layer=1 type=point,line,area output=R_917 use=val
value=0.0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=acuicultura_buf
output=R_918
v.to.rast input=R_918 layer=1 type=point,line,area output=R_919 use=val
value=0.0 rows=4096
r.patch input=R_917,R_919 output=R_920
r.grow.distance input=R_920 distance=R_921 metric=euclidean
r.mapcalc "R_922 = R_921 * 100"
r.mapcalc "R_923 = if(isnull( R_922 ), null(), if((R_922 <= 1 || R_922
>= 99999), 0.0, if((R_922 > 1 && R_922 < 4), eval((R_922 - 1)/(4 - 1)),
if((R_922 > 99999 && R_922 < 99999 ), eval((99999 - R_922 )/(99999 -
99999)), 1.0))))"
r.patch input=R_920,R_923 output=R_924
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_925 = if(isnull(ccaa),R_924, -1.0)"
g.region n=36.890735473967 s=35.045032348967 e=-3.742231171875 w=-
7.455610078125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_924
output="C:\Users\GSI\OceanLider\Resultados\56\RI_R_924.tif"
r.out.gdal input=R_925
output="C:\Users\GSI\OceanLider\Resultados\56\RF_R_925.tif"

```


Visualización del resultado:





- ✓ **Almadrabas.** Se aplican las áreas correspondientes a las instalaciones de almadrabas configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Almadrabas 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=almadrabas output=R_926
v.to.rast input=R_926 layer=1 type=point,line,area output=R_927 use=val
value=0.0 rows=4096
```

```
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=almadrabas_buf
output=R_928
v.to.rast input=R_928 layer=1 type=point,line,area output=R_929 use=val
value=0.0 rows=4096
r.patch input=R_927,R_929 output=R_930
r.grow.distance input=R_930 distance=R_931 metric=euclidean
r.mapcalc "R_932 = R_931 * 100"
r.mapcalc "R_933 = if(isnull( R_932 ), null(), if((R_932 <= 1 || R_932
>= 99999), 0.0, if((R_932 > 1 && R_932 < 4), eval((R_932 - 1)/(4 -1 )),
if((R_932 > 99999 && R_932 < 99999 ), eval((99999 - R_932 )/(99999 -
99999)), 1.0)))"
r.patch input=R_930,R_933 output=R_934
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_935 = if(isnull(ccaa),R_934, -1.0)"
g.region n=36.825916540361 s=35.090076696611 e=-3.588422578125 w=-
7.411664765625 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_934
output="C:\Users\GSI\OceanLider\Resultados\57\RI_R_934.tif"
r.out.gdal input=R_935
output="C:\Users\GSI\OceanLider\Resultados\57\RF_R_935.tif"
```

Visualización del resultado:





- ✓ **Arrecifes.** Se aplican las áreas correspondientes a los arrecifes configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

<input checked="" type="checkbox"/> Arrecifes	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

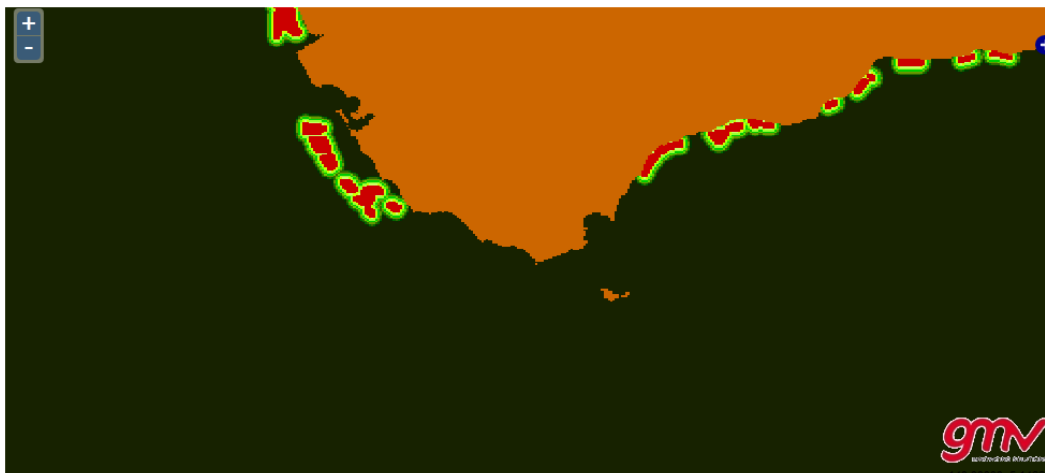
El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=arrecifes output=R_936
v.to.rast input=R_936 layer=1 type=point,line,area output=R_937 use=val
value=0.0 rows=4096
r.grow.distance input=R_937 distance=R_938 metric=euclidean
```

```
r.mapcalc "R_939 = R_938 * 100"
r.mapcalc "R_940 = if(isnull( R_939 ), null(), if((R_939 <= 1 || R_939
>= 99999), 0.0, if((R_939 > 1 && R_939 < 4), eval((R_939 - 1)/(4 -1 )),
if((R_939 > 99999 && R_939 < 99999 ), eval((99999 - R_939 )/(99999 -
99999)), 1.0)))"
r.patch input=R_937,R_940 output=R_941
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_942 = if(isnull(ccaa),R_941, -1.0)"
g.region n=36.891834509111 s=35.265857946611 e=-3.808149140625 w=-
7.455610078125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_941
output="C:\Users\GSI\OceanLider\Resultados\58\RI_R_941.tif"
r.out.gdal input=R_942
output="C:\Users\GSI\OceanLider\Resultados\58\RF_R_942.tif"
```

Visualización del resultado:





- ✓ **Caladeros.** Se aplican las áreas correspondientes a los caladeros configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

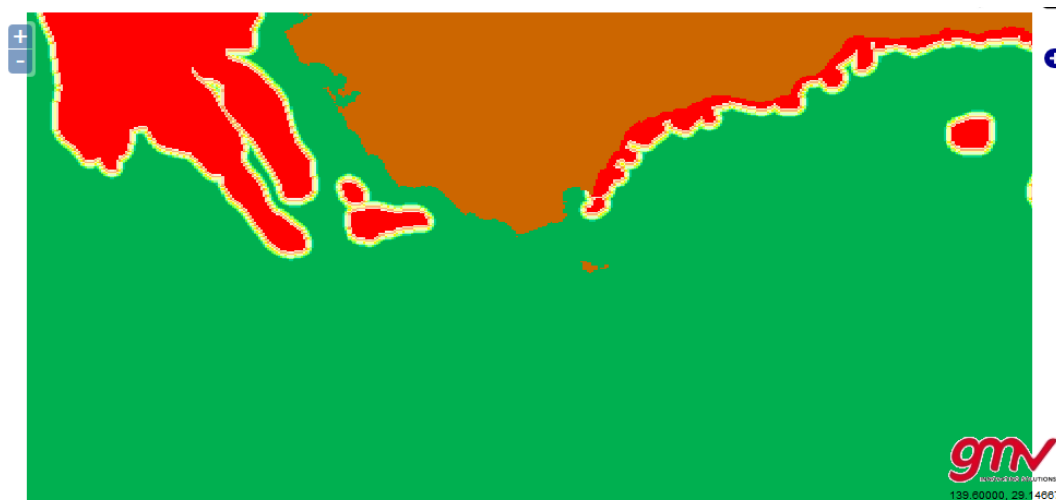
<input checked="" type="checkbox"/> Caladeros	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

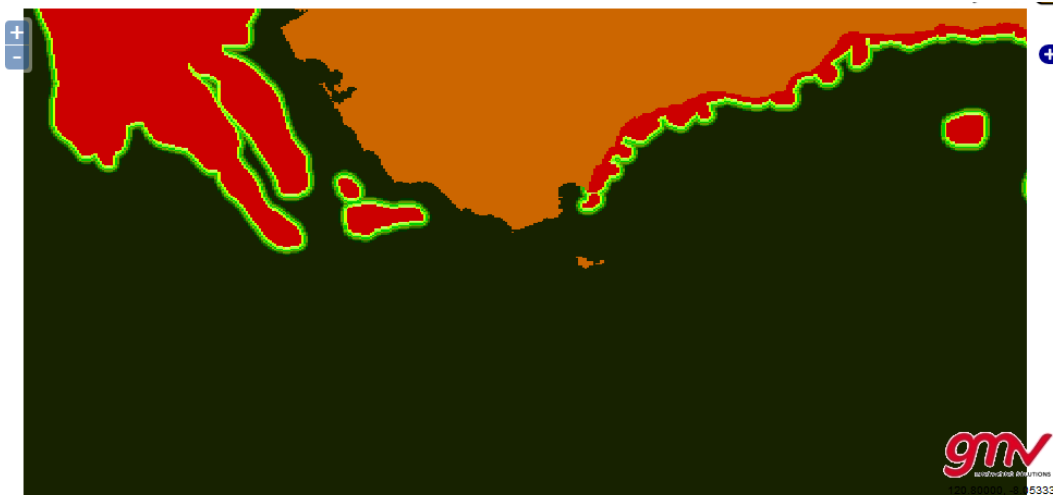
El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=caladeros output=R_943
v.to.rast input=R_943 layer=1 type=point,line,area output=R_944 use=val
value=0.0 rows=4096
r.grow.distance input=R_944 distance=R_945 metric=euclidean
r.mapcalc "R_946 = R_945 * 100"
```

```
r.mapcalc "R_947 = if(isnull( R_946 ), null(), if((R_946 <= 1 || R_946
>= 99999), 0.0, if((R_946 > 1 && R_946 < 4), eval((R_946 - 1)/(4 -1 )),
if((R_946 > 99999 && R_946 < 99999 ), eval((99999 - R_946 )/(99999 -
99999)), 1.0)))"
r.patch input=R_944,R_947 output=R_948
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_949 = if(isnull(ccaa),R_948, -1.0)"
g.region n=36.802844848967 s=35.045032348967 e=-3.764203828125 w=-
7.367719453125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_948
output="C:\Users\GSI\OceanLider\Resultados\59\RI_R_948.tif"
r.out.gdal input=R_949
output="C:\Users\GSI\OceanLider\Resultados\59\RF_R_949.tif"
```

Visualización del resultado:





- ✓ **Viveros.** Se aplican las áreas correspondientes a los viveros configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Viveros ⓘ

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

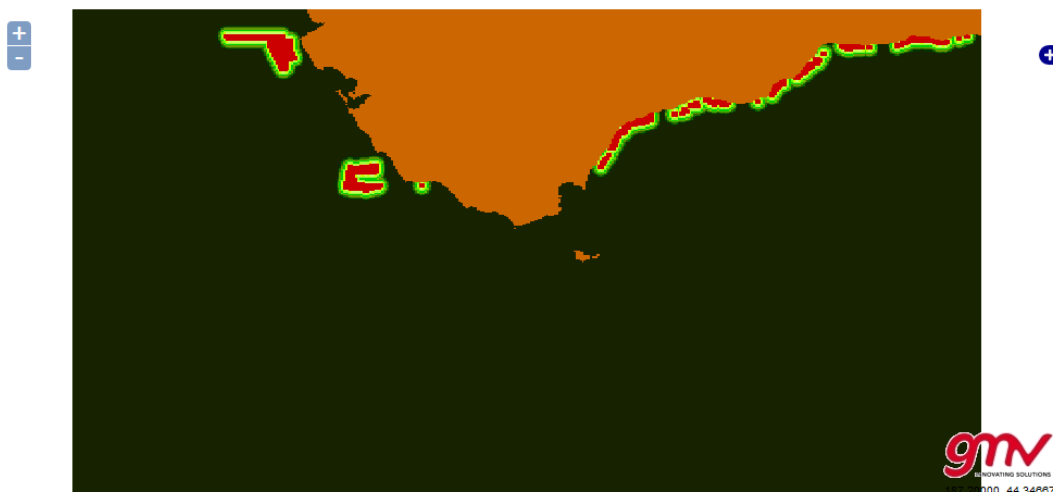
El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=viveros output=R_5725
v.to.rast input=R_5725 layer=1 type=point,line,area output=R_5726
use=val value=0.0 rows=4096
```

```
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=viveros_buf
output=R_5727
v.to.rast input=R_5727 layer=1 type=point,line,area output=R_5728
use=val value=0.0 rows=4096
r.patch input=R_5726,R_5728 output=R_5729
r.grow.distance input=R_5729 distance=R_5730 metric=euclidean
r.mapcalc "R_5731 = R_5730 * 100"
r.mapcalc "R_5732 = if(isnull( R_5731 ) , null(), if((R_5731 <= 1 ||
R_5731 >= 99999), 0.0, if((R_5731 > 1 && R_5731 < 4), eval((R_5731 -
1)/(4 -1 )), if((R_5731 > 99999 && R_5731 < 99999 ), eval((99999 -
R_5731 )/(99999 - 99999)), 1.0))))"
r.patch input=R_5729,R_5732 output=R_5733
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_5734 = if(isnull(ccaa),R_5733, -1.0)"
g.region n=36.847889196611 s=34.958240759111 e=-3.808149140625 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_5733
output="C:\Users\GSI\OceanLider\Resultados\60\RI_R_5733.tif"
r.out.gdal input=R_5734
output="C:\Users\GSI\OceanLider\Resultados\60\RF_R_5734.tif"
```


Visualización del resultado:





- ✓ **Zonas protegidas de interés pesquero.** Se aplican las áreas correspondientes a las zonas protegidas de interés pesquero configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Zonas protegidas de interés pesquero 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zpipes output=R_960
v.to.rast input=R_960 layer=1 type=point,line,area output=R_961 use=val
value=0.0 rows=4096
r.grow.distance input=R_961 distance=R_962 metric=euclidean
r.mapcalc "R_963 = R_962 * 100"
```

```
r.mapcalc "R_964 = if(isnull( R_963 ) , null(), if((R_963 <= 1 || R_963
>= 99999), 0.0, if((R_963 > 1 && R_963 < 4), eval((R_963 - 1)/(4 -1 )),
if((R_963 > 99999 && R_963 < 99999 ), eval((99999 - R_963 )/(99999 -
99999)), 1.0))))"
r.patch input=R_961,R_964 output=R_965
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_966 = if(isnull(ccaa),R_965, -1.0)"
g.region n=36.780872192717 s=34.979114380217 e=-3.852094453125 w=-
7.279828828125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_965
output="C:\Users\GSI\OceanLider\Resultados\61\RI_R_965.tif"
r.out.gdal input=R_966
output="C:\Users\GSI\OceanLider\Resultados\61\RF_R_966.tif"
```

Visualización del resultado:





- ✓ **Piscifactorías.** Se aplican las áreas correspondientes a las piscifactorías configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

<input checked="" type="checkbox"/> Piscifactorías	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=piscifactoria
output=R_967
v.to.rast input=R_967 layer=1 type=point,line,area output=R_968 use=val
value=0.0 rows=4096
```

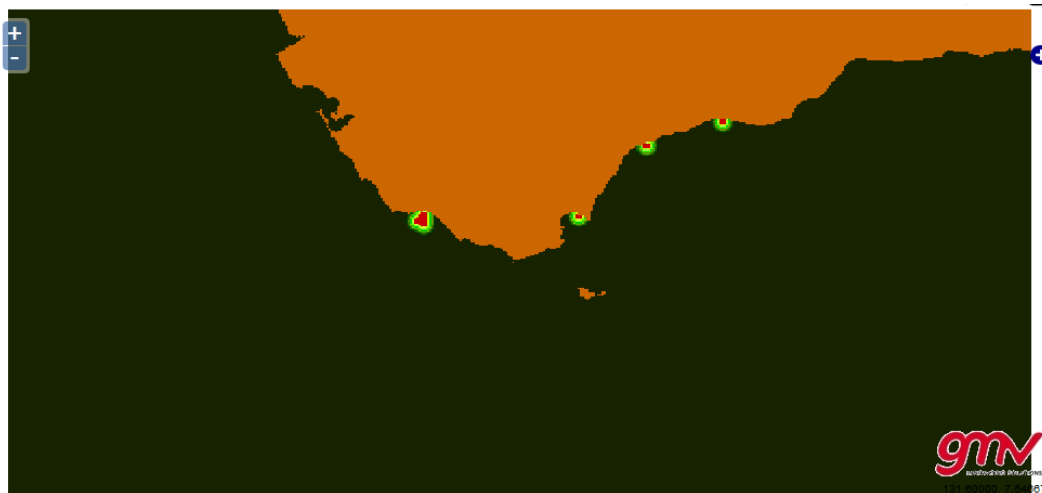
```

v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=piscifactoria_buf
output=R_969
v.to.rast input=R_969 layer=1 type=point,line,area output=R_970 use=val
value=0.0 rows=4096
r.patch input=R_968,R_970 output=R_971
r.grow.distance input=R_971 distance=R_972 metric=euclidean
r.mapcalc "R_973 = R_972 * 100"
r.mapcalc "R_974 = if(isnull( R_973 ), null(), if((R_973 <= 1 || R_973
>= 99999), 0.0, if((R_973 > 1 && R_973 < 4), eval((R_973 - 1)/(4 -1 )),
if((R_973 > 99999 && R_973 < 99999 ), eval((99999 - R_973 )/(99999 -
99999)), 1.0)))"
r.patch input=R_971,R_974 output=R_975
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_976 = if(isnull(ccaa),R_975, -1.0)"
g.region n=36.890735473967 s=35.176868286467 e=-3.786176484375 w=-
7.389692109375 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_975
output="C:\Users\GSI\OceanLider\Resultados\62\RI_R_975.tif"
r.out.gdal input=R_976
output="C:\Users\GSI\OceanLider\Resultados\62\RF_R_976.tif"

```

Visualización del resultado:





- ✓ **Reservas marinas.** Se aplican las áreas correspondientes a las reservas marinas configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

<input checked="" type="checkbox"/> Reservas marinas	
Valor de idoneidad	<input type="text" value="0.0"/>
Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="4"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=rrmm output=R_1029
v.to.rast input=R_1029 layer=1 type=point,line,area output=R_1030
use=val value=0.0 rows=4096
r.grow.distance input=R_1030 distance=R_1031 metric=euclidean
```

```
r.mapcalc "R_1032 = R_1031 * 100"
r.mapcalc "R_1033 = if(isnull( R_1032 ), null(), if((R_1032 <= 1 ||
R_1032 >= 99999), 0.0, if((R_1032 > 1 && R_1032 < 4), eval((R_1032 -
1)/(4 -1 )), if((R_1032 > 99999 && R_1032 < 99999 ), eval((99999 -
R_1032 )/(99999 - 99999)), 1.0))))"
r.patch input=R_1030,R_1033 output=R_1034
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1035 = if(isnull(ccaa),R_1034, -1.0)"
g.region n=36.846790161467 s=35.154895630217 e=-3.786176484375 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1034
output="C:\Users\GSI\OceanLider\Resultados\63\RI_R_1034.tif"
r.out.gdal input=R_1035
output="C:\Users\GSI\OceanLider\Resultados\63\RF_R_1035.tif"
```


Visualización del resultado:





- ✓ **Zonas de producción de moluscos y de otros invertebrados marinos.** Se aplican las áreas correspondientes a las zonas de producción de moluscos y de otros invertebrados marinos configurando un valor de idoneidad. Se le deja la libertad al usuario para configurar el valor de idoneidad y se le muestra uno por defecto. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

Selección en la interfaz:

Zonas de producción de moluscos y de otros invertebrados marinos 

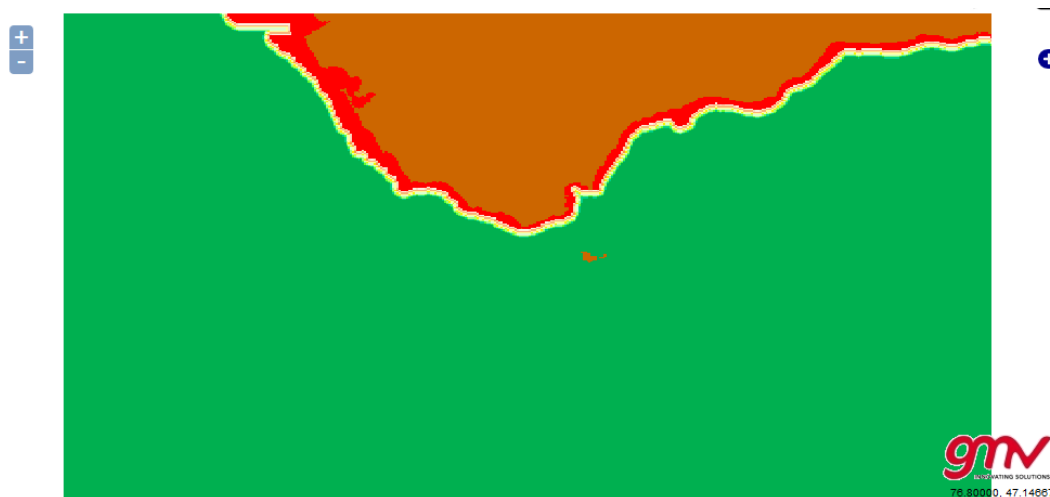
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

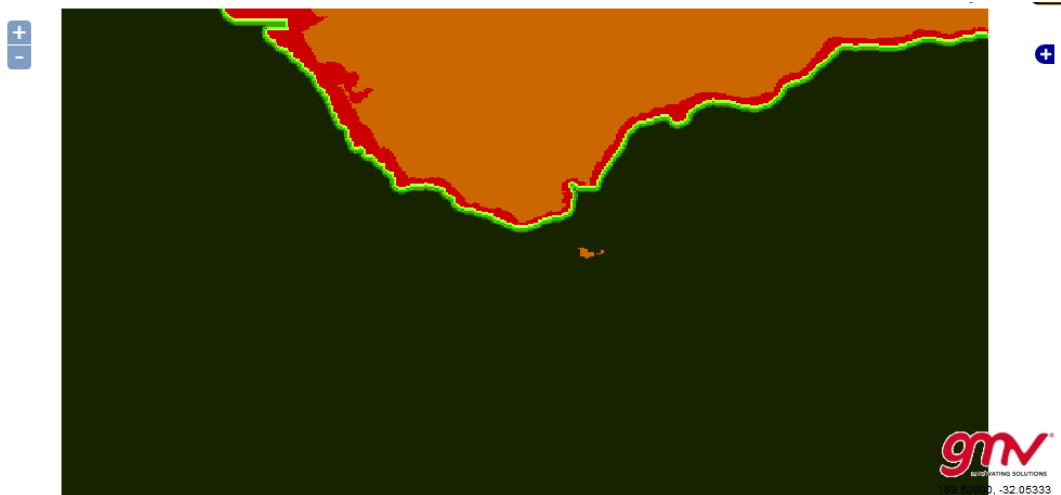
El script generado por la herramienta:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=cria_moluscos
output=R_1036
```

```
v.to.rast input=R_1036 layer=1 type=point,line,area output=R_1037
use=val value=0.0 rows=4096
r.grow.distance input=R_1037 distance=R_1038 metric=euclidean
r.mapcalc "R_1039 = R_1038 * 100"
r.mapcalc "R_1040 = if(isnull( R_1039 ) , null(), if((R_1039 <= 1 ||
R_1039 >= 99999), 0.0, if((R_1039 > 1 && R_1039 < 4), eval((R_1039 -
1)/(4 -1 )), if((R_1039 > 99999 && R_1039 < 99999 ), eval((99999 -
R_1039 )/(99999 - 99999)), 1.0))))"
r.patch input=R_1037,R_1040 output=R_1041
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1042 = if(isnull(ccaa),R_1041, -1.0)"
g.region n=36.824817505217 s=34.957141723967 e=-3.808149140625 w=-
7.367719453125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1041
output="C:\Users\GSI\OceanLider\Resultados\64\RI_R_1041.tif"
r.out.gdal input=R_1042
output="C:\Users\GSI\OceanLider\Resultados\64\RF_R_1042.tif"
```

Visualización del resultado:





2. Resultados de la combinación de criterios

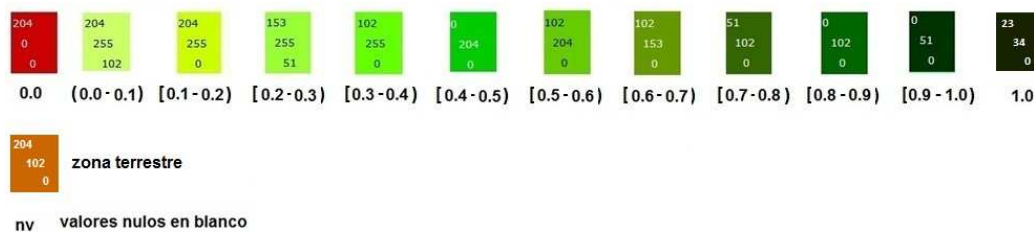
A continuación se muestran los resultados de las pruebas de combinación realizadas sobre los criterios propuestos para la herramienta OceanLider. Las pruebas se han realizado en la costa y zona marítima de Andalucía.

Se han utilizado dos paletas de colores para la visualización de resultados con el objetivo de seleccionar la más adecuada. Las paletas de colores utilizadas para mostrar los resultados son:

➤ **Paleta de colores 1:**

COLOR	RANGO	RGB	TEXTO
	0	255, 0, 0	No Apta
	[0, 0.1]	254, 235, 226	Alta Restricción
	[0.1, 0.2]	252, 219, 192	
	[0.2, 0.3]	249, 178, 119	
	[0.3, 0.4]	247, 150, 70	
	[0.4, 0.5]	255, 255, 159	Media Restricción
	[0.5, 0.6]	248, 242, 0	Baja Restricción
	[0.6, 0.7]	220, 255, 209	
	[0.7, 0.8]	162, 255, 133	
	[0.8, 0.9]	97, 255, 47	
	[0.9, 1]	0, 208, 134	Apta
	1	0, 176, 80	
	-1	204, 102, 0	Tierra

➤ **Paleta de colores 2:**



2.1. Combinación de criterios físicos

2.1.1. Combinación 1

Combinación de los dos subcriterios del criterio **Supervivencia del dispositivo** de exclusión total: *subcriterio altura de ola significativa y velocidad de corriente.*

Características del dispositivo

Nombre:

Tipo de dispositivo:

Largo: metros

Ancho: metros

Alto: metros

Profundidad máxima: metros

Profundidad mínima: metros

Altura de ola máxima admitida: metros

Velocidad de la corriente máxima admitida: m/s

Potencia unitaria: vatios

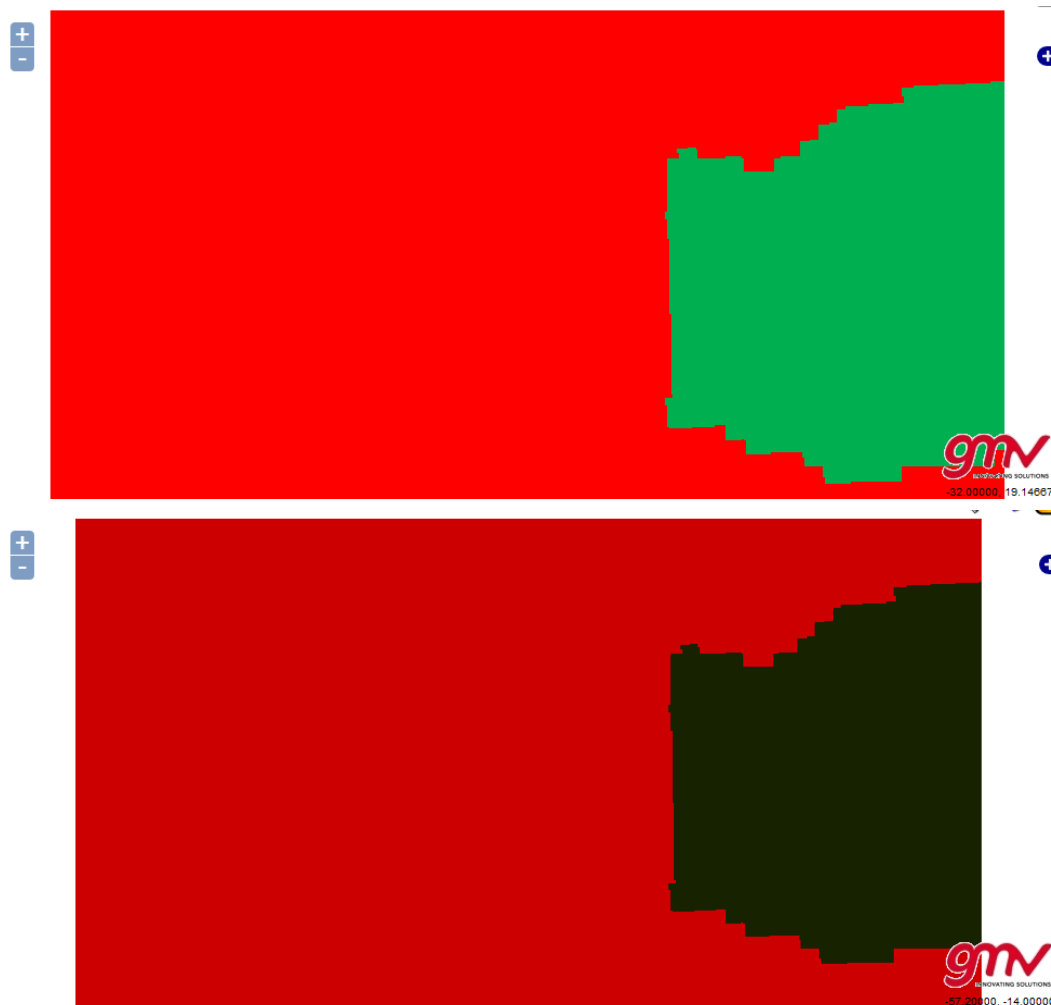
Separación entre dispositivos de una misma línea: kilómetros

Separación entre líneas de dispositivos: kilómetros

- **Subcriterio Altura de ola significativa** con el parámetro de dispositivo “altura de ola máxima admitida” de 2.7 metros.



- **Subcriterio Velocidad de corriente** con el parámetro de dispositivo “velocidad de la corriente máxima admitida” de 1.7 m/s.



Resultado con el método de razonamiento elegido para la combinación: **Pesimista (mínimo de los valores)**

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\medviv.tif"
output=R_1043
r.reclass input=R_1043 output=R_1044
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1044.txt"
r.mapcalc "R_1045 = R_1044"
r.null map=R_1045 null=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\unionhsmedia.tif"
output=R_1046
```

```
r.reclass input=R_1046 output=R_1047
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1047.txt"
r.mapcalc "R_1048 = R_1047"
r.null map=R_1048 null=0
r.mapcalc "R_1049 = if(isnull( R_1045 ) || isnull( R_1048 ), null(),
if(R_1045 <= R_1048 , R_1045 , R_1048 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1050 = if(isnull(ccaa),R_1049, -1.0)"
g.region n=36.935779821611 s=35.134022009111 e=-3.808149140625 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1045
output="C:\Users\GSI\OceanLider\Resultados\65\RI_R_1045.tif"
r.out.gdal input=R_1048
output="C:\Users\GSI\OceanLider\Resultados\65\RI_R_1048.tif"
r.out.gdal input=R_1050
output="C:\Users\GSI\OceanLider\Resultados\65\RF_R_1050.tif"
```



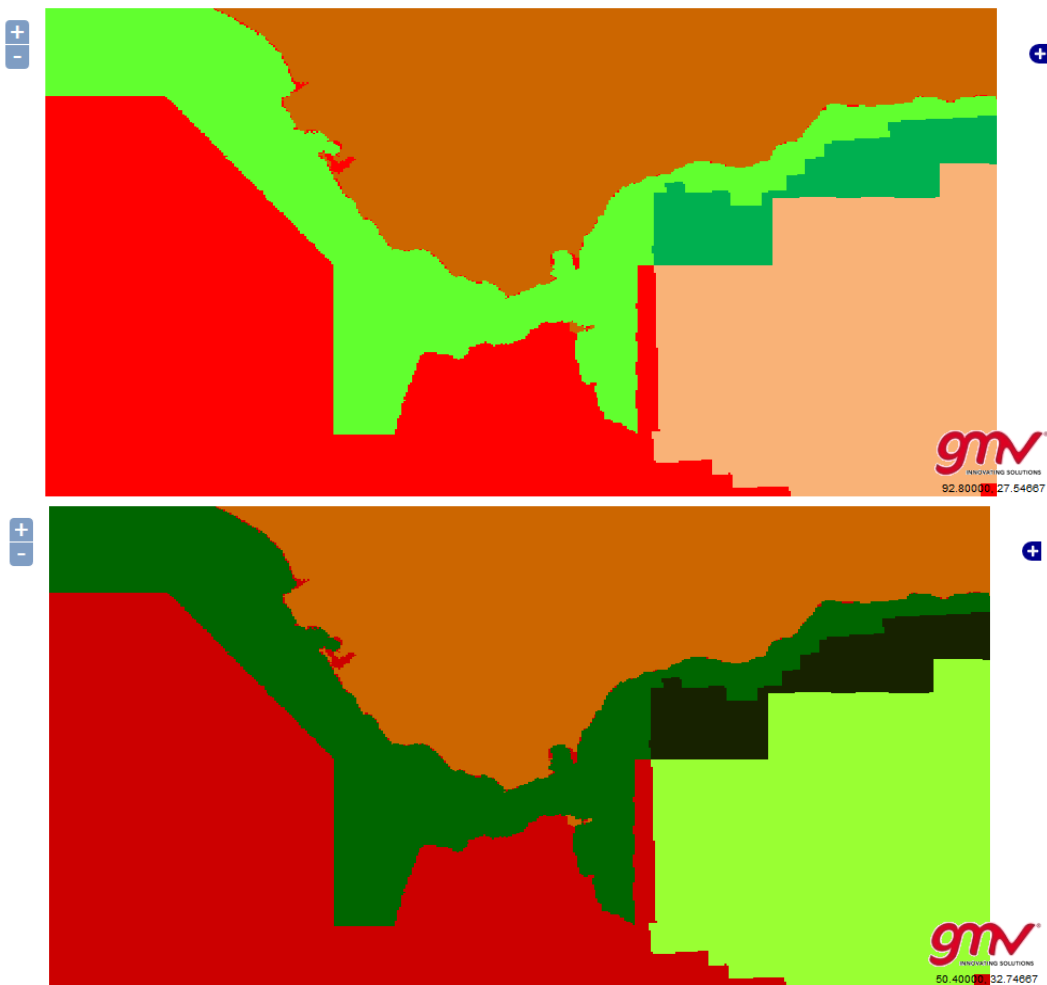


Resultado con el método de razonamiento elegido para la combinación: **Neutro (MLP) con pesos 0.8 y 0.2 respectivamente. (No sería de exclusión total propiamente dicho)**

```

SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\medviv.tif"
output=R_1051
r.reclass input=R_1051 output=R_1052
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1052.txt"
r.mapcalc "R_1053 = R_1052"
r.null map=R_1053 null=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\unionhsmedia.tif"
output=R_1054
r.reclass input=R_1054 output=R_1055
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1055.txt"
r.mapcalc "R_1056 = R_1055"
r.null map=R_1056 null=0
r.mapcalc "R_1057 = 0.2 * R_1053 + 0.8 * R_1056"
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1058 = if(isnull(ccaa),R_1057, -1.0)"
g.region n=37.070911523437 s=35.269153710937 e=-3.786176484375 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1053
output="C:\Users\GSI\OceanLider\Resultados\66\RI_R_1053.tif"
r.out.gdal input=R_1056
output="C:\Users\GSI\OceanLider\Resultados\66\RI_R_1056.tif"
r.out.gdal input=R_1058
output="C:\Users\GSI\OceanLider\Resultados\66\RF_R_1058.tif"

```



Resultado con el método de razonamiento elegido para la combinación: **Optimista (máximo de los valores).**

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\medviv.tif"
output=R_1059
r.reclass input=R_1059 output=R_1060
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1060.txt"
r.mapcalc "R_1061 = R_1060"
r.null map=R_1061 null=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\unionhsmedia.tif"
output=R_1062
r.reclass input=R_1062 output=R_1063
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1063.txt"
```

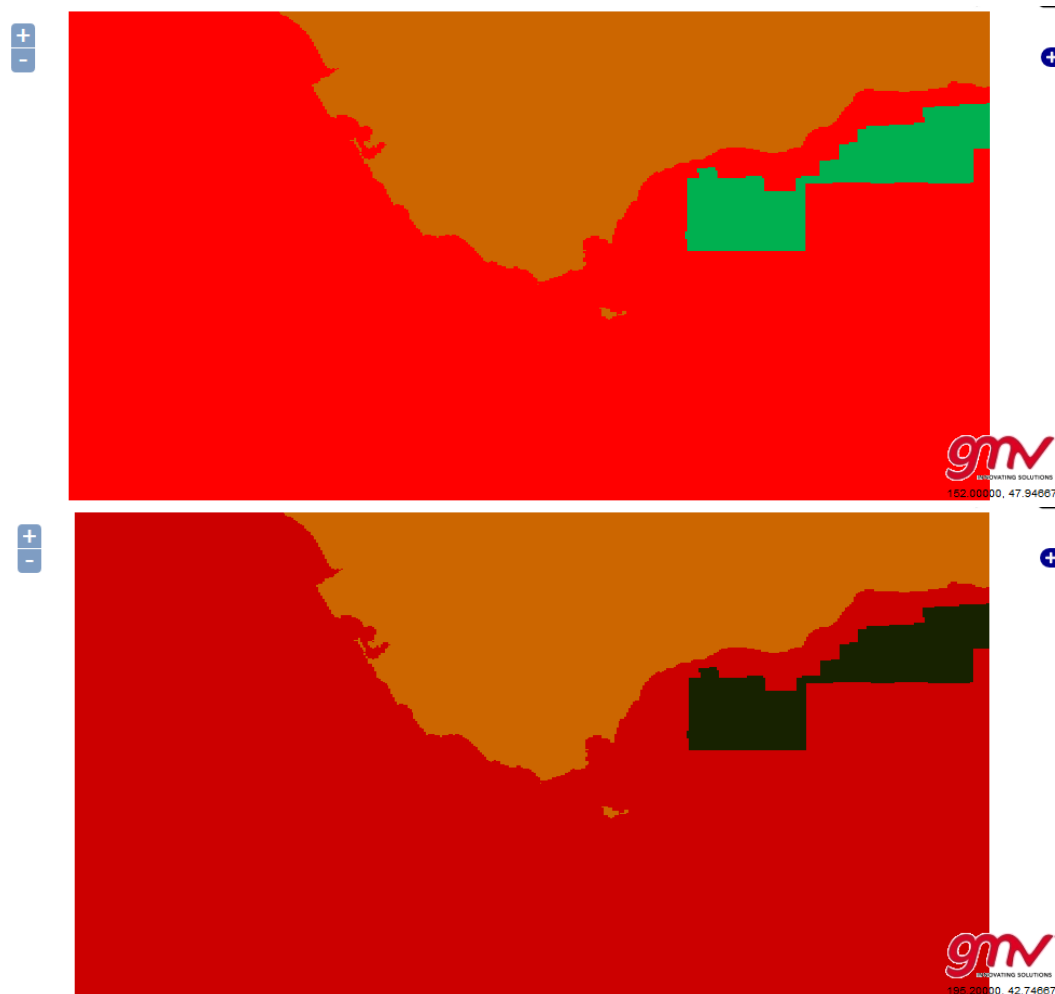
```
r.mapcalc "R_1064 = R_1063"
r.null map=R_1064 null=0
r.mapcalc "R_1065 = if(isnull( R_1061 ) || isnull( R_1064 ), null(),
if(R_1061 >= R_1064 , R_1061 , R_1064 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1066 = if(isnull(ccaa),R_1065, -1.0)"
g.region n=37.092884179687 s=35.422962304687 e=-3.786176484375 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1061
output="C:\Users\GSI\OceanLider\Resultados\67\RI_R_1061.tif"
r.out.gdal input=R_1064
output="C:\Users\GSI\OceanLider\Resultados\67\RI_R_1064.tif"
r.out.gdal input=R_1066
output="C:\Users\GSI\OceanLider\Resultados\67\RF_R_1066.tif"
```





Resultado con el método de razonamiento elegido para la combinación: **Más que pesimista (multiplicación de valores).**

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\medviv.tif"
output=R_1067
r.reclass input=R_1067 output=R_1068
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1068.txt"
r.mapcalc "R_1069 = R_1068"
r.null map=R_1069 null=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\unionhsmedia.tif"
output=R_1070
r.reclass input=R_1070 output=R_1071
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1071.txt"
r.mapcalc "R_1072 = R_1071"
r.null map=R_1072 null=0
r.mapcalc "R_1073 = R_1069 * R_1072 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1074 = if(isnull(ccaa),R_1073, -1.0)"
g.region n=37.004993554687 s=35.203235742187 e=-3.939985078125 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1069
output="C:\Users\GSI\OceanLider\Resultados\68\RI_R_1069.tif"
r.out.gdal input=R_1072
output="C:\Users\GSI\OceanLider\Resultados\68\RI_R_1072.tif"
r.out.gdal input=R_1074
output="C:\Users\GSI\OceanLider\Resultados\68\RF_R_1074.tif"
```



2.1.2. Combinación 2

Combinación del criterio **Profundidad** (de exclusión parcial) y **Supervivencia del dispositivo** (de exclusión total).

En el criterio **supervivencia del dispositivo** participan dos subcriterios eligiéndose como método para combinarse el método de razonamiento **optimista** (máximo de los valores): *altura de ola significativa* y *velocidad de corriente*.

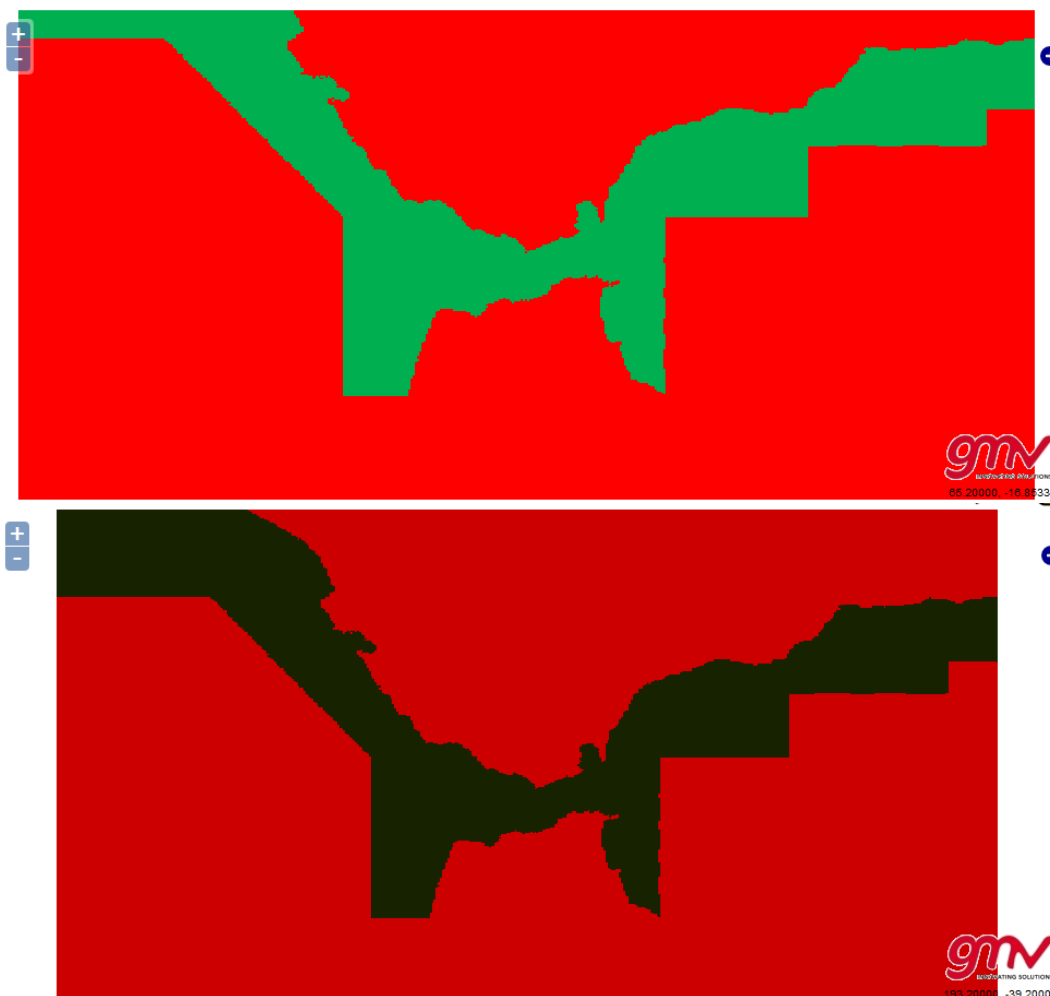
Como es la combinación de un criterio total (supervivencia del dispositivo) y otro parcial (profundidad) internamente en el motor de la herramienta se combinan con el método pesimista (mínimo de los valores).

Características del dispositivo

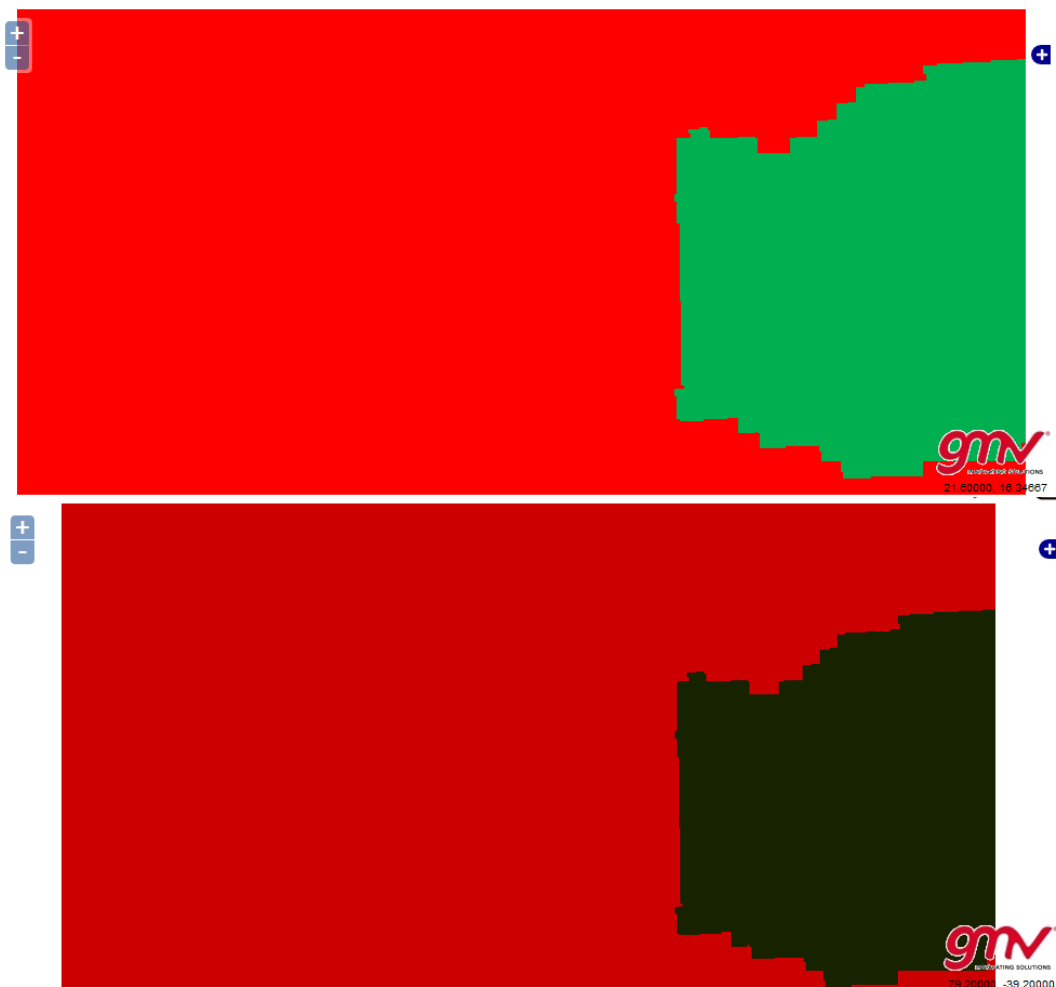
Nombre:	<input type="text" value="Dispositivo 7"/>
Tipo de dispositivo:	<input type="text" value="Undimotriz - Fuera de costa: flotante"/>
Largo:	<input type="text" value="8"/> metros
Ancho:	<input type="text" value="2"/> metros
Alto:	<input type="text" value="15"/> metros
Profundidad máxima:	<input type="text" value="-100"/> metros
Profundidad mínima:	<input type="text" value="-20"/> metros
Altura de ola máxima admitida:	<input type="text" value="2.7"/> metros
Velocidad de la corriente máxima admitida:	<input type="text" value="1.7"/> m/s
Potencia unitaria:	<input type="text" value="10"/> vatios
Separación entre dispositivos de una misma línea:	<input type="text" value="2"/> kilómetros
Separación entre líneas de dispositivos:	<input type="text" value="2"/> kilómetros

Los parámetros que intervienen se extraen del dispositivo al que está ligado:

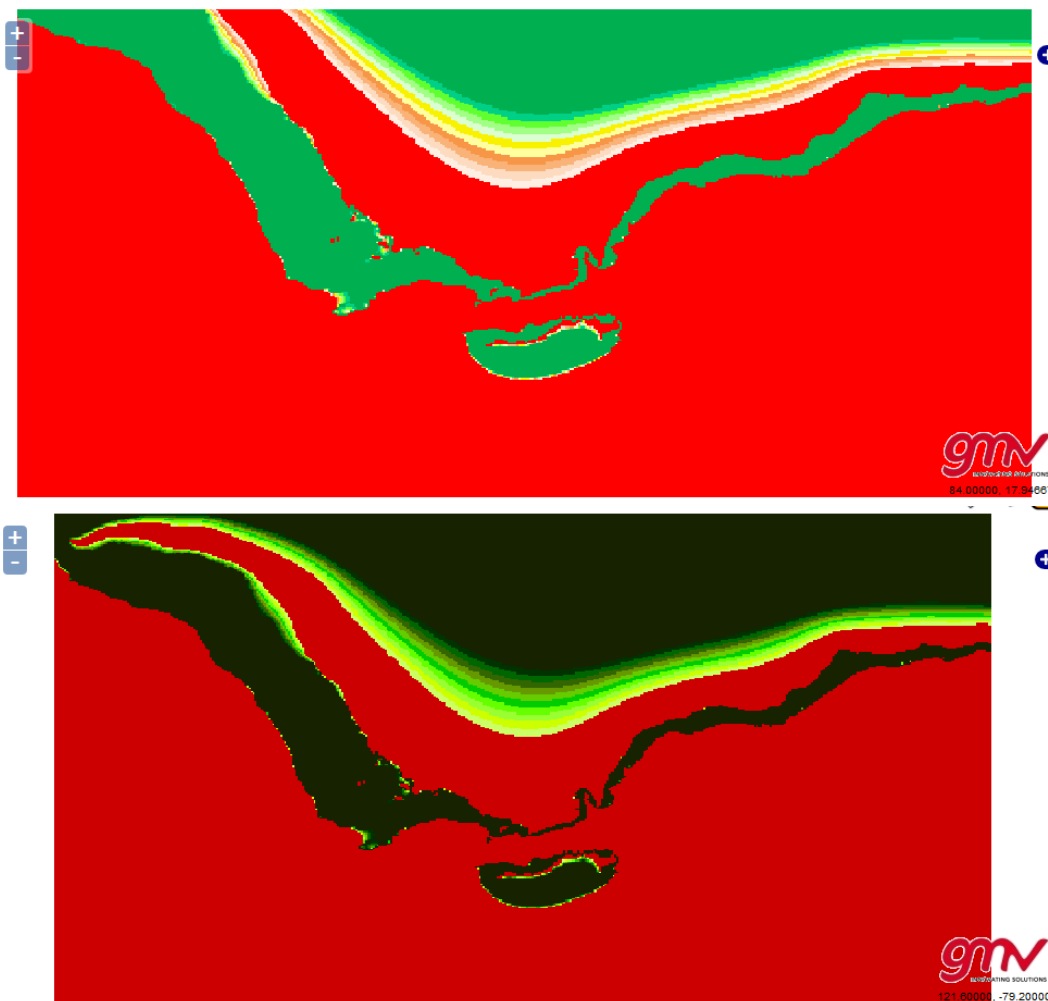
- ✓ Para el subcriterio *Altura de ola significativa* del criterio Supervivencia del dispositivo: parámetro altura de ola máxima admitida de 2.7 metros



- ✓ Para el subcriterio *Velocidad de corriente* del criterio Supervivencia del dispositivo: parámetro velocidad de la corriente máxima admitida de 1.7 m/s.



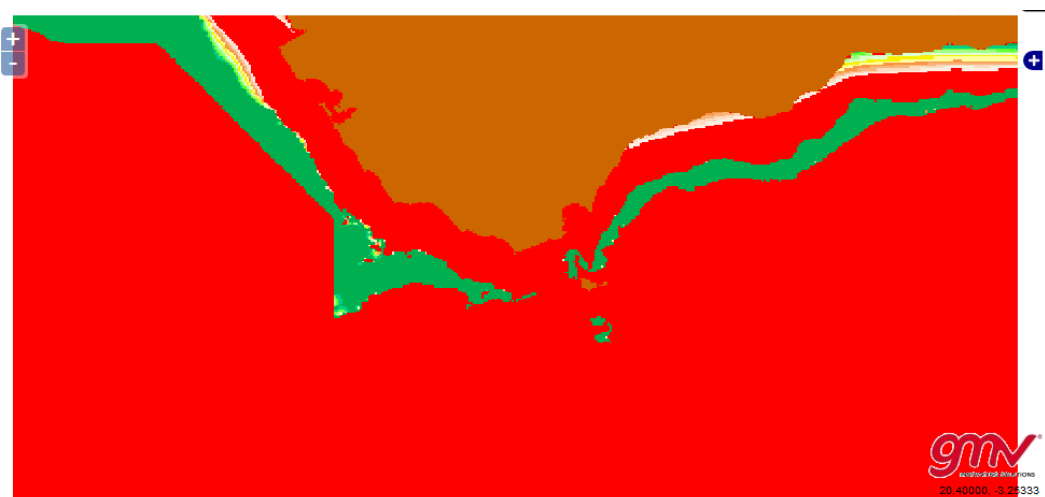
- ✓ Para criterio *profundidad*: los parámetros profundidad máxima y mínima: -100 metros y -20 metros

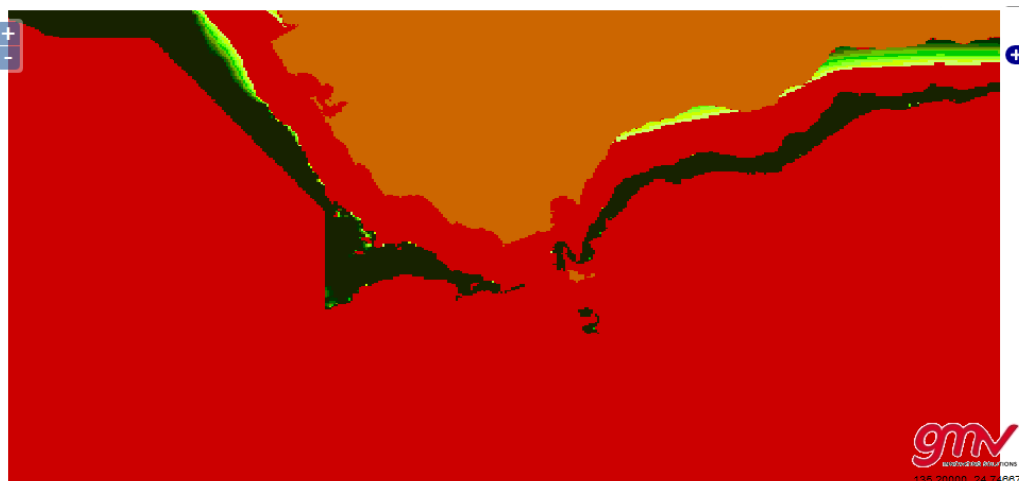


El resultado de la combinación:

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\medviv.tif"
output=R_1075
r.reclass input=R_1075 output=R_1076
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1076.txt"
r.mapcalc "R_1077 = R_1076"
r.null map=R_1077 null=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\unionhsmedia.tif"
output=R_1078
r.reclass input=R_1078 output=R_1079
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1079.txt"
r.mapcalc "R_1080 = R_1079"
```

```
r.null map=R_1080 null=0
r.mapcalc "R_1083 = if(isnull( R_1077 ) || isnull( R_1080 ), null(),
if(R_1077 >= R_1080 , R_1077 , R_1080 ))"
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_1081
r.mapcalc "R_1082 = if(isnull( R_1081 ) , null(), if((R_1081 <= -105.0
|| R_1081 >= -15.0), 0.0, if((R_1081 > -105.0 && R_1081 < -100.0),
eval((R_1081 - -105.0)/(-100.0 --105.0 )), if((R_1081 > -20.0 && R_1081
< -15.0 ), eval((-15.0 - R_1081 )/(-15.0 - -20.0)), 1.0))))"
r.mapcalc "R_1084 = if(isnull( R_1082 ) || isnull( R_1083 ), null(),
if(R_1082 <= R_1083 , R_1082 , R_1083 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1085 = if(isnull(ccaa),R_1084, -1.0)"
g.region n=36.847889196611 s=35.134022009111 e=-3.830121796875 w=-
7.389692109375 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1077
output="C:\Users\GSI\OceanLider\Resultados\69\RI_R_1077.tif"
r.out.gdal input=R_1080
output="C:\Users\GSI\OceanLider\Resultados\69\RI_R_1080.tif"
r.out.gdal input=R_1082
output="C:\Users\GSI\OceanLider\Resultados\69\RI_R_1082.tif"
r.out.gdal input=R_1085
output="C:\Users\GSI\OceanLider\Resultados\69\RF_R_1085.tif"
```





2.1.3. Combinación 3

Combinación de los criterios: **Distancia a la costa** (exclusión parcial), **profundidad** (exclusión parcial) y **supervivencia del dispositivo** (exclusión total).

En el criterio **supervivencia del dispositivo** intervienen dos subcriterios: *altura de ola significativa* y *velocidad de corriente*, eligiéndose para su combinación el método de razonamiento **optimista** (máximo de los valores).

A su vez, se puede escoger el método de razonamiento para la combinación de los dos criterios parciales: *más que pesimista*, *pesimista*, *optimista* y *MLP*.

Características del dispositivo

Nombre:

Tipo de dispositivo:

Largo: metros

Ancho: metros

Alto: metros

Profundidad máxima: metros

Profundidad mínima: metros

Altura de ola máxima admitida: metros

Velocidad de la corriente máxima admitida: m/s

Potencia unitaria: vatios

Separación entre dispositivos de una misma línea: kilómetros

Separación entre líneas de dispositivos: kilómetros

Los parámetros que intervienen:

- ✓ En el subcriterio *altura de ola significativa* del criterio supervivencia del dispositivo: el parámetro altura de ola máxima admitida del dispositivo al que está ligado y tiene un valor de 2.7 metros
- ✓ En el subcriterio *velocidad de corriente* del criterio supervivencia del dispositivo: el parámetro velocidad de la corriente máxima admitida del dispositivo al que está ligado y tiene un valor de 1.7 m/s.
- ✓ El criterio *profundidad*: los parámetros profundidad máxima y mínima del dispositivo al que está ligado y tiene los valores -100 y -20, respectivamente.
- ✓ El criterio *distancia a la costa*: los parámetros son pertenecientes al criterio con los valores: valor A->8 kilómetros, valor B-> 9 kilómetros, valor C-> 20 kilómetros y valor D-> 25 kilómetros.

Distancia a la costa
 Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km.

Valor A	8
Valor B	9
Valor C	20
Valor D	25
Peso del subcriterio:	0.0

Combinación de criterios parciales con método de razonamiento: **pesimista**. El método que combina el resultado de la combinación de los criterios parciales (criterio distancia a la costa y profundidad) y los criterios totales (criterio supervivencia del dispositivo) se realiza por defecto mediante el método pesimista (mínimo de los valores)

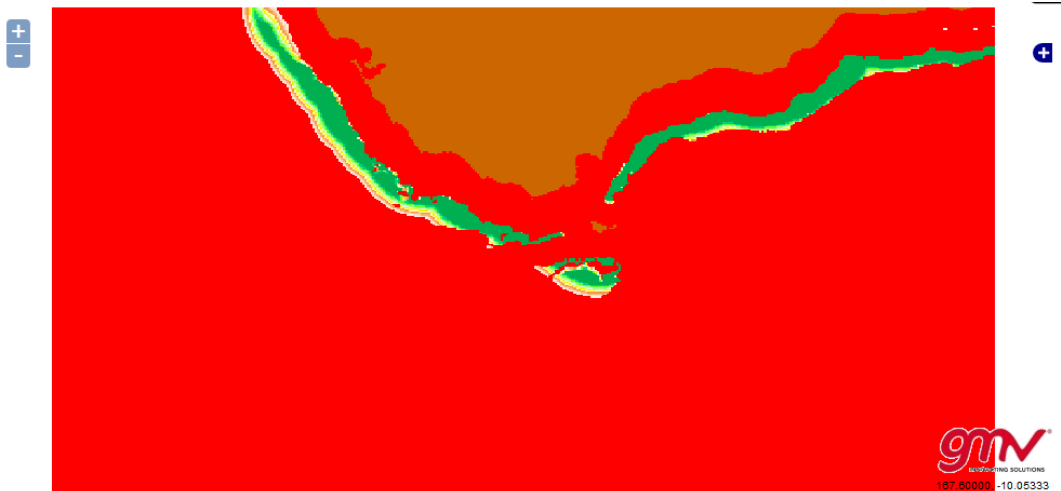
```

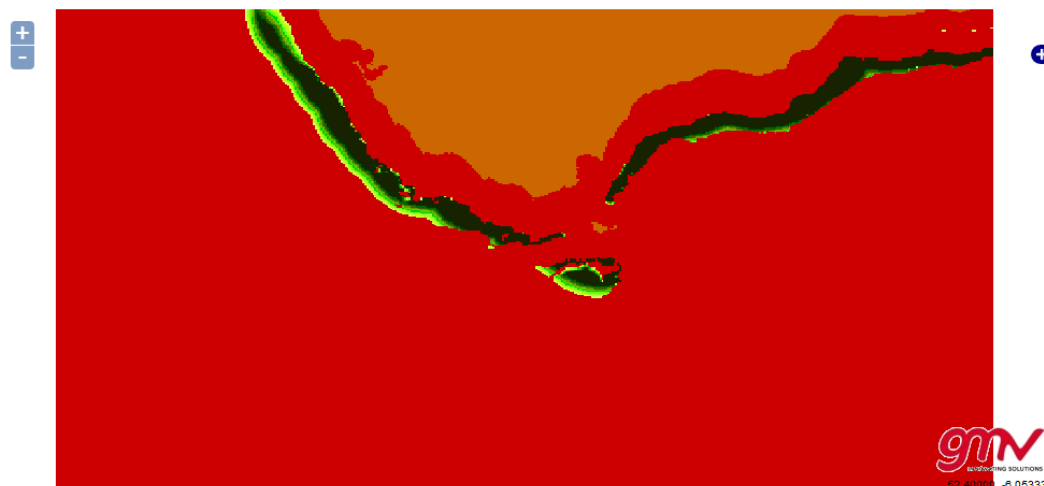
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\medviv.tif"
output=R_1086
r.reclass input=R_1086 output=R_1087
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1087.txt"
r.mapcalc "R_1088 = R_1087"
r.null map=R_1088 null=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\unionhsmedia.tif"
output=R_1097
r.reclass input=R_1097 output=R_1098
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1098.txt"
r.mapcalc "R_1099 = R_1098"
r.null map=R_1099 null=0
r.mapcalc "R_1102 = if(isnull( R_1088 ) || isnull( R_1099 ), null(),
if(R_1088 >= R_1099 , R_1088 , R_1099 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_1089
v.to.rast input=R_1089 layer=1 type=point,line,area output=R_1090
use=val value=1 rows=4096
r.grow.distance input=R_1090 distance=R_1091 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_1092
v.to.rast input=R_1092 layer=1 type=point,line,area output=R_1093
use=val value=0 rows=4096
r.null map=R_1093 null=1
r.mapcalc "R_1094 = R_1091 * R_1093 "
r.mapcalc "R_1095 = R_1094 * 100"
r.mapcalc "R_1096 = if(isnull( R_1095 ) , null(), if((R_1095 <= 8 ||
R_1095 >= 25), 0.0, if((R_1095 > 8 && R_1095 < 9), eval((R_1095 - 8)/(9
-8 )), if((R_1095 > 20 && R_1095 < 25 ), eval((25 - R_1095)/(25 - 20)),
1.0)))"
r.null map=R_1096 value=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_1100
r.mapcalc "R_1101 = if(isnull( R_1100 ) , null(), if((R_1100 <= -105.0
|| R_1100 >= -15.0), 0.0, if((R_1100 > -105.0 && R_1100 < -100.0),
eval((R_1100 - -105.0)/(-100.0 --105.0 )), if((R_1100 > -20.0 && R_1100
< -15.0 ), eval((-15.0 - R_1100 )/(-15.0 - -20.0)), 1.0)))"

```

Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

```
r.mapcalc "R_1103 = if(isnull( R_1096 ) || isnull( R_1101 ), null(),  
if(R_1096 <= R_1101 , R_1096 , R_1101 ))"  
r.mapcalc "R_1104 = if(isnull( R_1103 ) || isnull( R_1101 ), null(),  
if(R_1103 <= R_1101 , R_1103 , R_1101 ))"  
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432  
user=postgres password=968XCdPh" layer=ccaa output=ccaa  
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val  
value=-1.0 rows=4096  
r.mapcalc "R_1105 = if(isnull(ccaa),R_1104, -1.0)"  
g.region n=36.738025915361 s=34.870350134111 e=-3.830121796875 w=-  
7.455610078125 nsres=300 ewres=300 rows=3224 cols=3453  
r.out.gdal input=R_1088  
output="C:\Users\GSI\OceanLider\Resultados\70\RI_R_1088.tif"  
r.out.gdal input=R_1099  
output="C:\Users\GSI\OceanLider\Resultados\70\RI_R_1099.tif"  
r.out.gdal input=R_1096  
output="C:\Users\GSI\OceanLider\Resultados\70\RI_R_1096.tif"  
r.out.gdal input=R_1101  
output="C:\Users\GSI\OceanLider\Resultados\70\RI_R_1101.tif"  
r.out.gdal input=R_1105  
output="C:\Users\GSI\OceanLider\Resultados\70\RF_R_1105.tif"
```





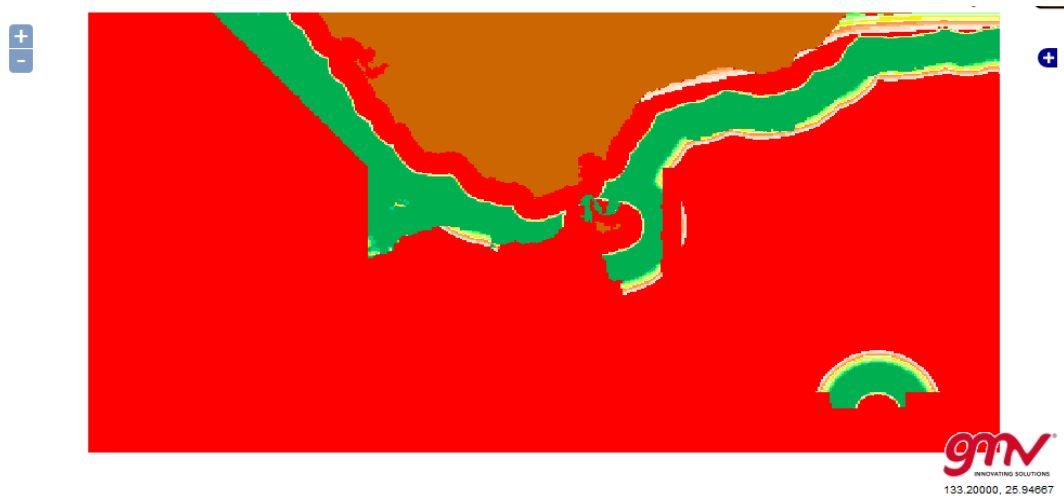
Combinación de criterios parciales con método de razonamiento: **optimista**. El método que combina el resultado de la combinación de los criterios parciales (criterio distancia a la costa y profundidad) y los criterios totales (criterio supervivencia del dispositivo) se realiza por defecto mediante el método pesimista (mínimo de los valores)

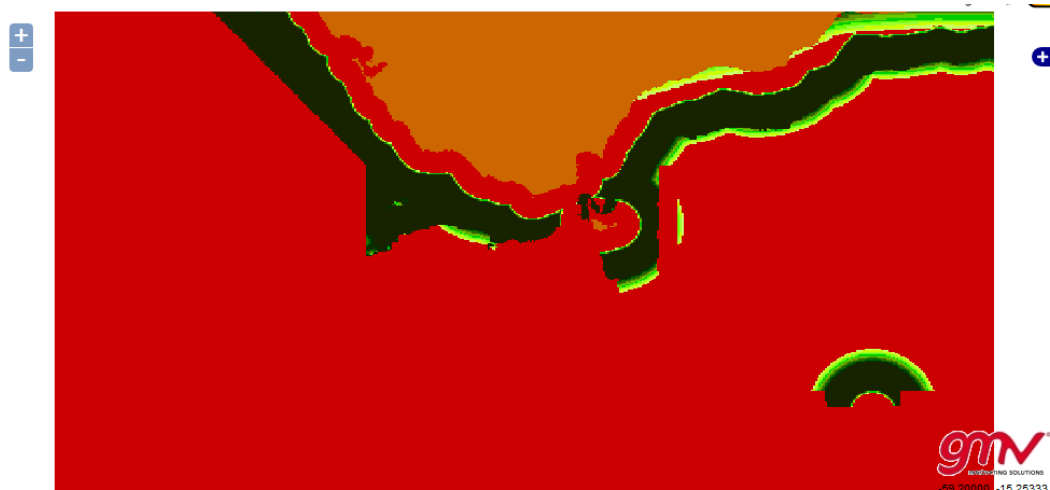
```

SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\medviv.tif"
output=R_1126
r.reclass input=R_1126 output=R_1127
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1127.txt"
r.mapcalc "R_1128 = R_1127"
r.null map=R_1128 null=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\unionhsmedia.tif"
output=R_1137
r.reclass input=R_1137 output=R_1138
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1138.txt"
r.mapcalc "R_1139 = R_1138"
r.null map=R_1139 null=0
r.mapcalc "R_1142 = if(isnull( R_1128 ) || isnull( R_1139 ), null(),
if(R_1128 >= R_1139 , R_1128 , R_1139 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_1129
v.to.rast input=R_1129 layer=1 type=point,line,area output=R_1130
use=val value=1 rows=4096
r.grow.distance input=R_1130 distance=R_1131 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_1132
v.to.rast input=R_1132 layer=1 type=point,line,area output=R_1133
use=val value=0 rows=4096
r.null map=R_1133 null=1
r.mapcalc "R_1134 = R_1131 * R_1133 "
r.mapcalc "R_1135 = R_1134 * 100"

```

```
r.mapcalc "R_1136 = if(isnull( R_1135 ), null(), if((R_1135 <= 8 || R_1135 >= 25), 0.0, if((R_1135 > 8 && R_1135 < 9), eval((R_1135 - 8)/(9 - 8)), if((R_1135 > 20 && R_1135 < 25 ), eval((25 - R_1135)/(25 - 20)), 1.0))))"
r.null map=R_1136 value=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_1140
r.mapcalc "R_1141 = if(isnull( R_1140 ), null(), if((R_1140 <= -105.0 || R_1140 >= -15.0), 0.0, if((R_1140 > -105.0 && R_1140 < -100.0), eval((R_1140 - -105.0)/(-100.0 - -105.0)), if((R_1140 > -20.0 && R_1140 < -15.0 ), eval((-15.0 - R_1140)/(-15.0 - -20.0)), 1.0))))"
r.mapcalc "R_1143 = if(isnull( R_1136 ) || isnull( R_1141 ), null(), if(R_1136 >= R_1141 , R_1136 , R_1141 ))"
r.mapcalc "R_1144 = if(isnull( R_1143 ) || isnull( R_1142 ), null(), if(R_1143 <= R_1142 , R_1143 , R_1142 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432 user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val value=-1.0 rows=4096
r.mapcalc "R_1145 = if(isnull(ccaa),R_1144, -1.0)"
g.region n=36.716053259111 s=34.848377477861 e=-3.830121796875 w=-7.455610078125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1128
output="C:\Users\GSI\OceanLider\Resultados\71\RI_R_1128.tif"
r.out.gdal input=R_1139
output="C:\Users\GSI\OceanLider\Resultados\71\RI_R_1139.tif"
r.out.gdal input=R_1136
output="C:\Users\GSI\OceanLider\Resultados\71\RI_R_1136.tif"
r.out.gdal input=R_1141
output="C:\Users\GSI\OceanLider\Resultados\71\RI_R_1141.tif"
r.out.gdal input=R_1145
output="C:\Users\GSI\OceanLider\Resultados\71\RF_R_1145.tif"
```





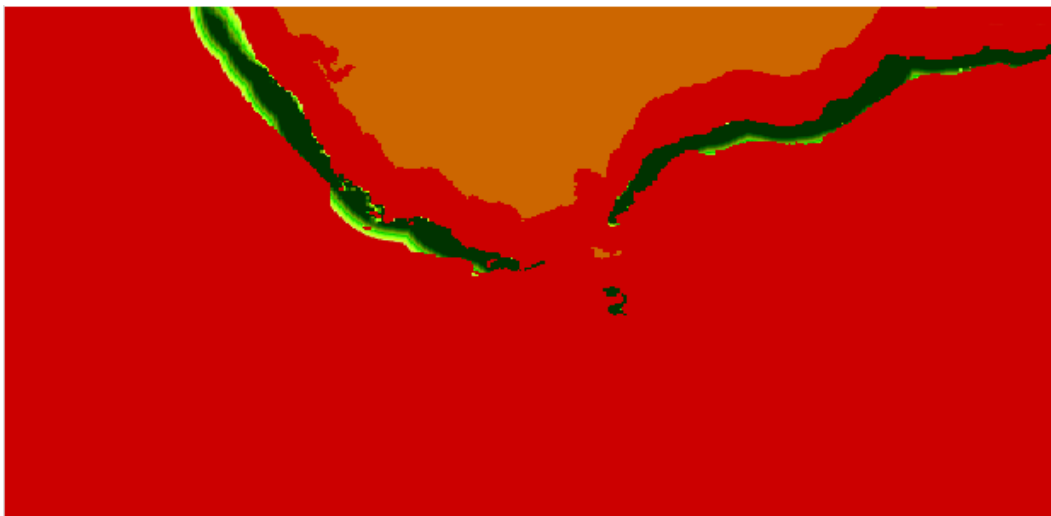
Combinación de criterios parciales con método de razonamiento: **más que pesimista**. El método que combina el resultado de la combinación de los criterios parciales (criterio distancia a la costa y profundidad) y los criterios totales (criterio supervivencia del dispositivo) se realiza por defecto mediante el método pesimista (mínimo de los valores)

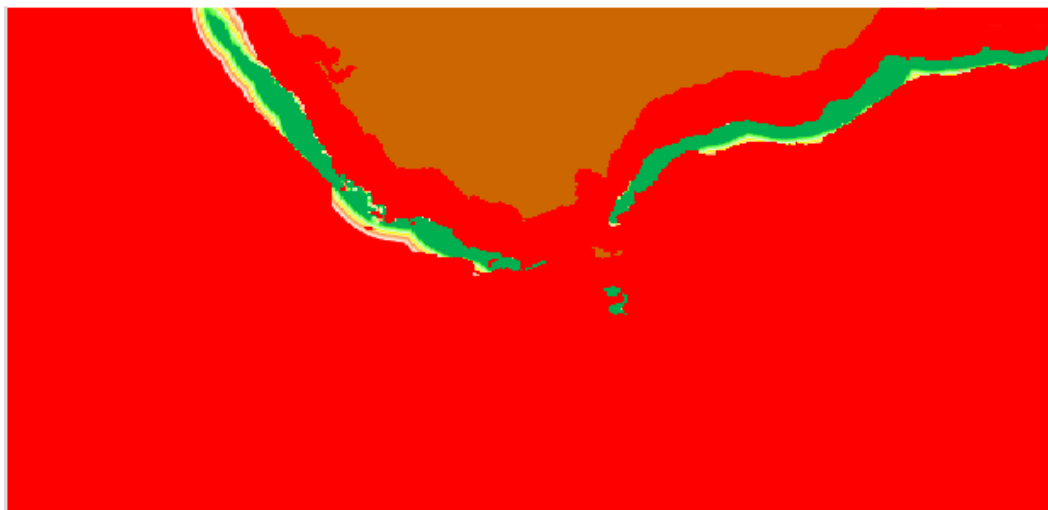
```

SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\medviv.tif"
output=R_1146
r.reclass input=R_1146 output=R_1147
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1147.txt"
r.mapcalc "R_1148 = R_1147"
r.null map=R_1148 null=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\unionhsmedia.tif"
output=R_1157
r.reclass input=R_1157 output=R_1158
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1158.txt"
r.mapcalc "R_1159 = R_1158"
r.null map=R_1159 null=0
r.mapcalc "R_1162 = if(isnull( R_1148 ) || isnull( R_1159 ), null(),
if(R_1148 >= R_1159 , R_1148 , R_1159 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_1149
v.to.rast input=R_1149 layer=1 type=point,line,area output=R_1150
use=val value=1 rows=4096
r.grow.distance input=R_1150 distance=R_1151 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_1152
v.to.rast input=R_1152 layer=1 type=point,line,area output=R_1153
use=val value=0 rows=4096
r.null map=R_1153 null=1
r.mapcalc "R_1154 = R_1151 * R_1153 "
r.mapcalc "R_1155 = R_1154 * 100"

```

```
r.mapcalc "R_1156 = if(isnull( R_1155 ) , null(), if((R_1155 <= 8 || R_1155 >= 25), 0.0, if((R_1155 > 8 && R_1155 < 9), eval((R_1155 - 8)/(9 - 8)), if((R_1155 > 20 && R_1155 < 25 ), eval((25 - R_1155)/(25 - 20)), 1.0))))"
r.null map=R_1156 value=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_1160
r.mapcalc "R_1161 = if(isnull( R_1160 ) , null(), if((R_1160 <= -105.0 || R_1160 >= -15.0), 0.0, if((R_1160 > -105.0 && R_1160 < -100.0), eval((R_1160 - -105.0)/(-100.0 --105.0 )), if((R_1160 > -20.0 && R_1160 < -15.0 ), eval((-15.0 - R_1160)/(-15.0 - -20.0)), 1.0))))"
r.mapcalc "R_1163 = R_1156 * R_1161 "
r.mapcalc "R_1164 = if(isnull( R_1163 ) || isnull( R_1162 ), null(), if(R_1163 <= R_1162 , R_1163 , R_1162 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432 user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val value=-1.0 rows=4096
r.mapcalc "R_1165 = if(isnull(ccaa),R_1164, -1.0)"
g.region n=36.803943884111 s=34.958240759111 e=-3.896039765625 w=-7.477582734375 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1148
output="C:\Users\GSI\OceanLider\Resultados\72\RI_R_1148.tif"
r.out.gdal input=R_1159
output="C:\Users\GSI\OceanLider\Resultados\72\RI_R_1159.tif"
r.out.gdal input=R_1156
output="C:\Users\GSI\OceanLider\Resultados\72\RI_R_1156.tif"
r.out.gdal input=R_1161
output="C:\Users\GSI\OceanLider\Resultados\72\RI_R_1161.tif"
r.out.gdal input=R_1165
output="C:\Users\GSI\OceanLider\Resultados\72\RF_R_1165.tif"
```



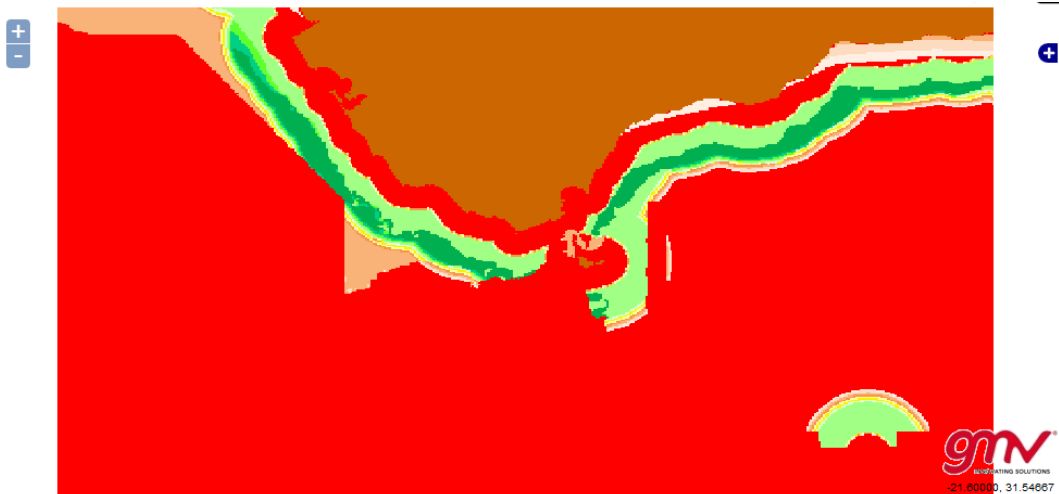


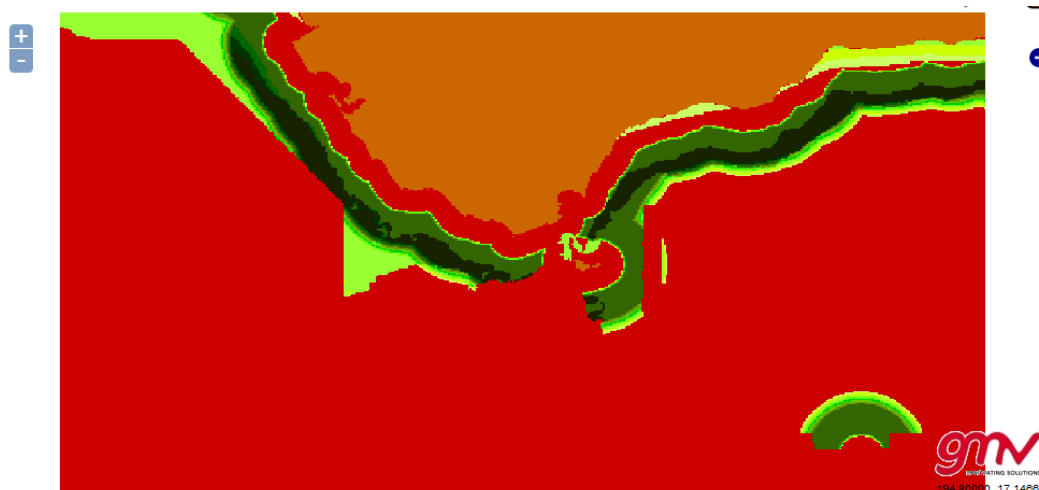
Combinación de criterios parciales con método de razonamiento: **MLP con los pesos 0.75 para el subcriterio distancia a la costa y 0.25 para profundidad**. El método que combina el resultado de la combinación de los criterios parciales (criterio distancia a la costa y profundidad) y los criterios totales (criterio supervivencia del dispositivo) se realiza por defecto mediante el método pesimista (mínimo de los valores)

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\medviv.tif"
output=R_1166
r.reclass input=R_1166 output=R_1167
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1167.txt"
r.mapcalc "R_1168 = R_1167"
r.null map=R_1168 null=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\unionhsmedia.tif"
output=R_1177
r.reclass input=R_1177 output=R_1178
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_1178.txt"
r.mapcalc "R_1179 = R_1178"
r.null map=R_1179 null=0
r.mapcalc "R_1182 = if(isnull( R_1168 ) || isnull( R_1179 ), null(),
if(R_1168 >= R_1179 , R_1168 , R_1179 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_1169
v.to.rast input=R_1169 layer=1 type=point,line,area output=R_1170
use=val value=1 rows=4096
r.grow.distance input=R_1170 distance=R_1171 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_1172
v.to.rast input=R_1172 layer=1 type=point,line,area output=R_1173
use=val value=0 rows=4096
r.null map=R_1173 null=1
r.mapcalc "R_1174 = R_1171 * R_1173 "
```



```
r.mapcalc "R_1175 = R_1174 * 100"  
r.mapcalc "R_1176 = if(isnull( R_1175 ), null(), if((R_1175 <= 8 ||  
R_1175 >= 25), 0.0, if((R_1175 > 8 && R_1175 < 9), eval((R_1175 - 8)/(9  
- 8 )), if((R_1175 > 20 && R_1175 < 25 ), eval((25 - R_1175 )/(25 - 20)),  
1.0))))"  
r.null map=R_1176 value=0  
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"  
output=R_1180  
r.mapcalc "R_1181 = if(isnull( R_1180 ), null(), if((R_1180 <= -105.0  
|| R_1180 >= -15.0), 0.0, if((R_1180 > -105.0 && R_1180 < -100.0),  
eval((R_1180 - -105.0)/(-100.0 --105.0 )), if((R_1180 > -20.0 && R_1180  
< -15.0 ), eval((-15.0 - R_1180 )/(-15.0 - -20.0)), 1.0))))"  
r.mapcalc "R_1183 = 0.75 * R_1176 + 0.25 * R_1181"  
r.mapcalc "R_1184 = if(isnull( R_1183 ) || isnull( R_1182 ), null(),  
if(R_1183 <= R_1182 , R_1183 , R_1182 ))"  
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432  
user=postgres password=968XCdPh" layer=ccaa output=ccaa  
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val  
value=-1.0 rows=4096  
r.mapcalc "R_1185 = if(isnull(ccaa),R_1184, -1.0)"  
g.region n=36.846790161467 s=35.023059692717 e=-3.830121796875 w=-  
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453  
r.out.gdal input=R_1168  
output="C:\Users\GSI\OceanLider\Resultados\73\RI_R_1168.tif"  
r.out.gdal input=R_1179  
output="C:\Users\GSI\OceanLider\Resultados\73\RI_R_1179.tif"  
r.out.gdal input=R_1176  
output="C:\Users\GSI\OceanLider\Resultados\73\RI_R_1176.tif"  
r.out.gdal input=R_1181  
output="C:\Users\GSI\OceanLider\Resultados\73\RI_R_1181.tif"  
r.out.gdal input=R_1185  
output="C:\Users\GSI\OceanLider\Resultados\73\RF_R_1185.tif"
```





2.1.4. Combinación 4

Combinación de dos criterios: **Distancia a la costa** (exclusión parcial) y **profundidad** (exclusión parcial). Se pueden combinar estos dos criterios parciales mediante los diferentes métodos de razonamiento: *más que pesimista*, *pesimista*, *optimista* y *MLP*.

Los parámetros que se han configurado para estos criterios son:

- ✓ Para el criterio distancia a la costa se han configurado los valores: valor A->8 kilómetros, valor B-> 9 kilómetros, valor C-> 20 kilómetros y valor D-> 25 kilómetros.

Distancia a la costa

Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km.

Valor A	8
Valor B	9
Valor C	20
Valor D	25
Peso del subcriterio:	0.0

- ✓ Para el criterio profundidad se extraen los parámetros profundidad máxima y mínima del dispositivo al que está asociado.

Características del dispositivo

Nombre:	<input type="text" value="Dispositivo 7"/>	
Tipo de dispositivo:	<input type="text" value="Undimotriz - Fuera de costa: flotante"/>	
Largo:	<input type="text" value="8"/>	metros
Ancho:	<input type="text" value="2"/>	metros
Alto:	<input type="text" value="15"/>	metros
Profundidad máxima:	<input type="text" value="-100"/>	metros
Profundidad mínima:	<input type="text" value="-20"/>	metros
Altura de ola máxima admitida:	<input type="text" value="2.7"/>	metros
Velocidad de la corriente máxima admitida:	<input type="text" value="1.7"/>	m/s
Potencia unitaria:	<input type="text" value="10"/>	vatios
Separación entre dispositivos de una misma línea:	<input type="text" value="2"/>	kilómetros
Separación entre líneas de dispositivos:	<input type="text" value="2"/>	kilómetros

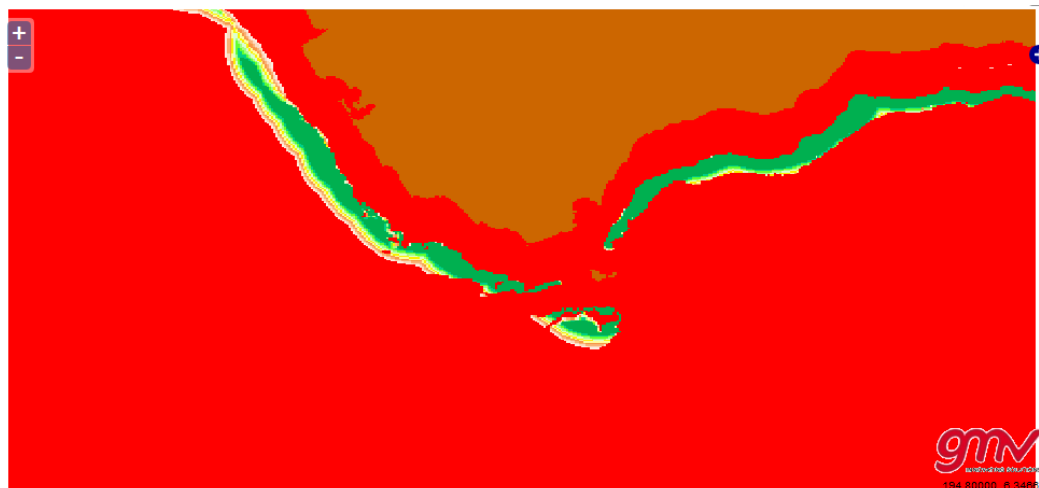
Para la combinación de los criterios distancia a la costa y profundidad utilizando el método de razonamiento: **pesimista** (mínimo de los valores)

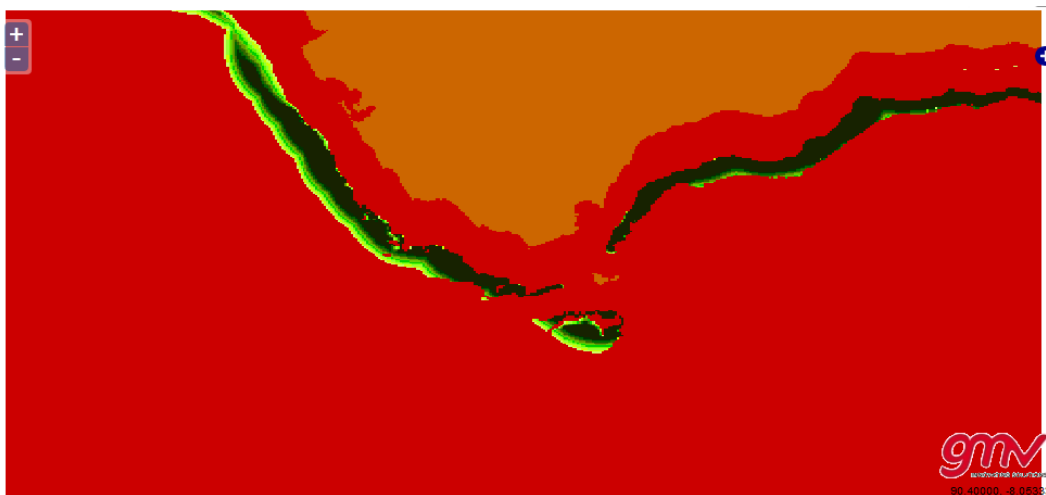
```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_1186
v.to.rast input=R_1186 layer=1 type=point,line,area output=R_1187
use=val value=1 rows=4096
r.grow.distance input=R_1187 distance=R_1188 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_1189
v.to.rast input=R_1189 layer=1 type=point,line,area output=R_1190
use=val value=0 rows=4096
r.null map=R_1190 null=1
r.mapcalc "R_1191 = R_1188 * R_1190 "
r.mapcalc "R_1192 = R_1191 * 100"
r.mapcalc "R_1193 = if(isnull( R_1192 ) , null(), if((R_1192 <= 8 ||
R_1192 >= 25), 0.0, if((R_1192 > 8 && R_1192 < 9), eval((R_1192 - 8)/(9
```

```

-8 )), if((R_1192 > 20 && R_1192 < 25 ), eval((25 - R_1192 )/(25 - 20)),
1.0))))"
r.null map=R_1193 value=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_1194
r.mapcalc "R_1195 = if(isnull( R_1194 ), null(), if((R_1194 <= -105.0
|| R_1194 >= -15.0), 0.0, if((R_1194 > -105.0 && R_1194 < -100.0),
eval((R_1194 - -105.0)/(-100.0 --105.0 )), if((R_1194 > -20.0 && R_1194
< -15.0 ), eval((-15.0 - R_1194 )/(-15.0 - -20.0)), 1.0))))"
r.mapcalc "R_1196 = if(isnull( R_1193 ) || isnull( R_1195 ), null(),
if(R_1193 <= R_1195 , R_1193 , R_1195 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1197 = if(isnull(ccaa),R_1196, -1.0)"
g.region n=36.868762817717 s=35.088977661467 e=-3.742231171875 w=-
7.521528046875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1193
output="C:\Users\GSI\OceanLider\Resultados\75\RI_R_1193.tif"
r.out.gdal input=R_1195
output="C:\Users\GSI\OceanLider\Resultados\75\RI_R_1195.tif"
r.out.gdal input=R_1197
output="C:\Users\GSI\OceanLider\Resultados\75\RF_R_1197.tif"

```

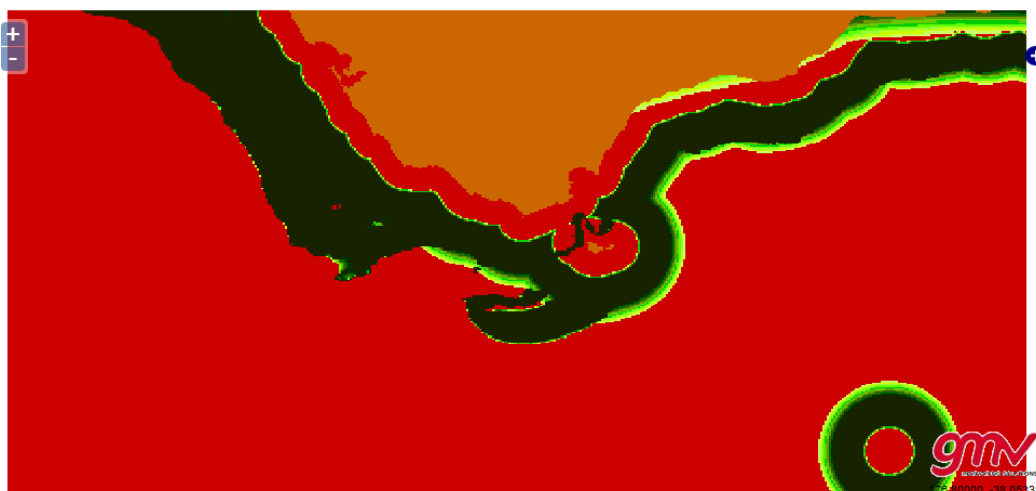
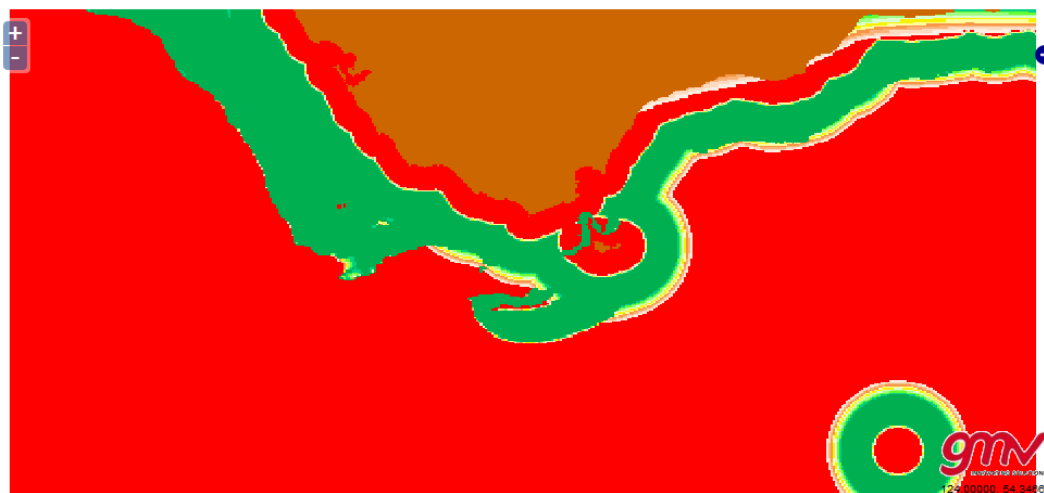




Para la combinación de los criterios distancia a la costa y profundidad utilizando el método de razonamiento: **optimista** (máximo de los valores)

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_1198
v.to.rast input=R_1198 layer=1 type=point,line,area output=R_1199
use=val value=1 rows=4096
r.grow.distance input=R_1199 distance=R_1200 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_1201
v.to.rast input=R_1201 layer=1 type=point,line,area output=R_1202
use=val value=0 rows=4096
r.null map=R_1202 null=1
r.mapcalc "R_1203 = R_1200 * R_1202 "
r.mapcalc "R_1204 = R_1203 * 100"
r.mapcalc "R_1205 = if(isnull( R_1204 ), null(), if((R_1204 <= 8 ||
R_1204 >= 25), 0.0, if((R_1204 > 8 && R_1204 < 9), eval((R_1204 - 8)/(9
-8 )), if((R_1204 > 20 && R_1204 < 25 ), eval((25 - R_1204)/(25 - 20)),
1.0)))"
r.null map=R_1205 value=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_1206
r.mapcalc "R_1207 = if(isnull( R_1206 ), null(), if((R_1206 <= -105.0
|| R_1206 >= -15.0), 0.0, if((R_1206 > -105.0 && R_1206 < -100.0),
eval((R_1206 - -105.0)/(-100.0 --105.0 )), if((R_1206 > -20.0 && R_1206
< -15.0 ), eval((-15.0 - R_1206 )/(-15.0 - -20.0)), 1.0)))"
r.mapcalc "R_1208 = if(isnull( R_1205 ) || isnull( R_1207 ), null(),
if(R_1205 >= R_1207 , R_1205 , R_1207 )"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
```

```
r.mapcalc "R_1209 = if(isnull(ccaa),R_1208, -1.0)"
g.region n=36.7369268880217 s=35.001087036467 e=-3.808149140625 w=-
7.455610078125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1205
output="C:\Users\GSI\OceanLider\Resultados\76\RI_R_1205.tif"
r.out.gdal input=R_1207
output="C:\Users\GSI\OceanLider\Resultados\76\RI_R_1207.tif"
r.out.gdal input=R_1209
output="C:\Users\GSI\OceanLider\Resultados\76\RF_R_1209.tif"
```

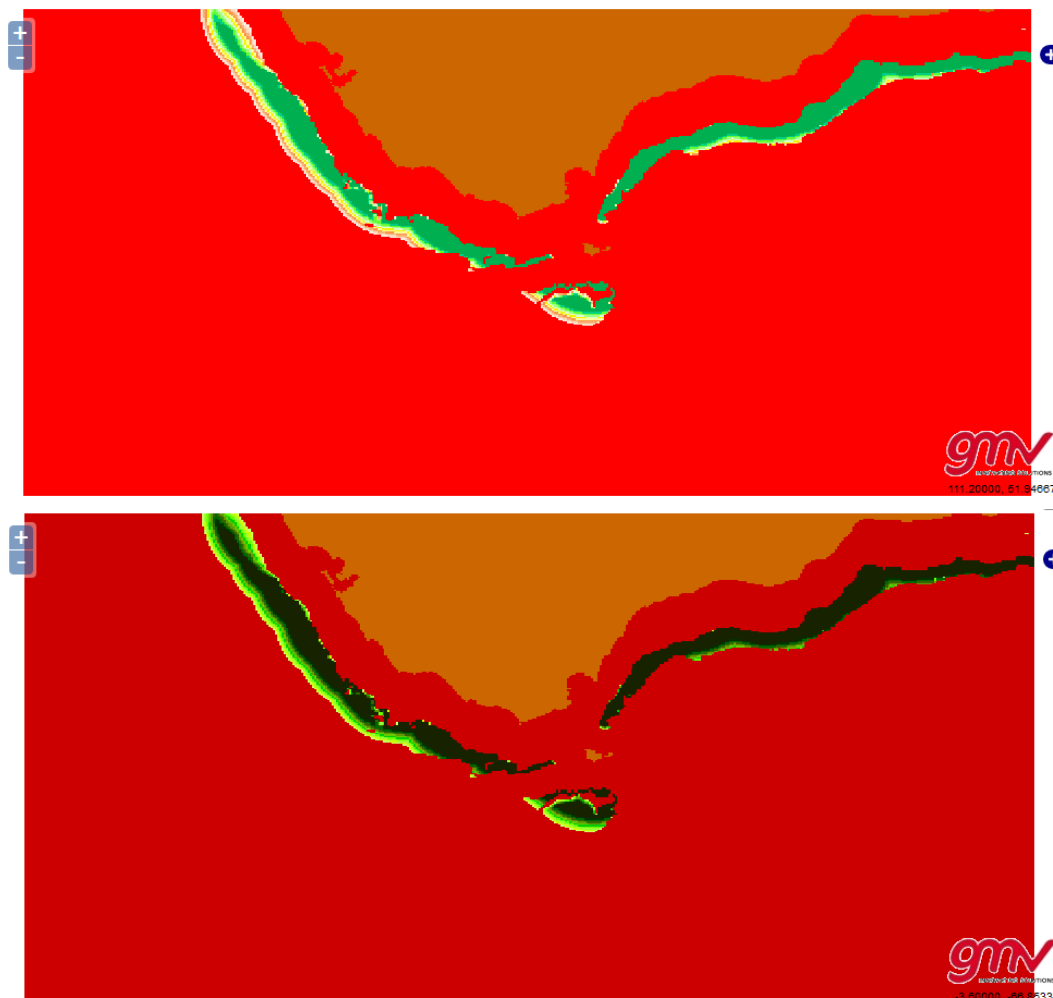


Para la combinación de los criterios distancia a la costa y profundidad utilizando el método de razonamiento: **más que pesimista** (multiplicación de valores)

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
```

Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

```
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_1252
v.to.rast input=R_1252 layer=1 type=point,line,area output=R_1253
use=val value=1 rows=4096
r.grow.distance input=R_1253 distance=R_1254 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_1255
v.to.rast input=R_1255 layer=1 type=point,line,area output=R_1256
use=val value=0 rows=4096
r.null map=R_1256 null=1
r.mapcalc "R_1257 = R_1254 * R_1256 "
r.mapcalc "R_1258 = R_1257 * 100"
r.mapcalc "R_1259 = if(isnull( R_1258 ) , null(), if((R_1258 <= 8 ||
R_1258 >= 25), 0.0, if((R_1258 > 8 && R_1258 < 9), eval((R_1258 - 8)/(9
-8 )), if((R_1258 > 20 && R_1258 < 25 ), eval((25 - R_1258)/(25 - 20)),
1.0))))"
r.null map=R_1259 value=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_1260
r.mapcalc "R_1261 = if(isnull( R_1260 ) , null(), if((R_1260 <= -105.0
|| R_1260 >= -15.0), 0.0, if((R_1260 > -105.0 && R_1260 < -100.0),
eval((R_1260 - -105.0)/(-100.0 --105.0 )), if((R_1260 > -20.0 && R_1260
< -15.0 ), eval((-15.0 - R_1260)/(-15.0 - -20.0)), 1.0))))"
r.mapcalc "R_1262 = R_1259 * R_1261 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1263 = if(isnull(ccaa),R_1262, -1.0)"
g.region n=36.736926880217 s=35.023059692717 e=-3.808149140625 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1259
output="C:\Users\GSI\OceanLider\Resultados\77\RI_R_1259.tif"
r.out.gdal input=R_1261
output="C:\Users\GSI\OceanLider\Resultados\77\RI_R_1261.tif"
r.out.gdal input=R_1263
output="C:\Users\GSI\OceanLider\Resultados\77\RF_R_1263.tif"
```

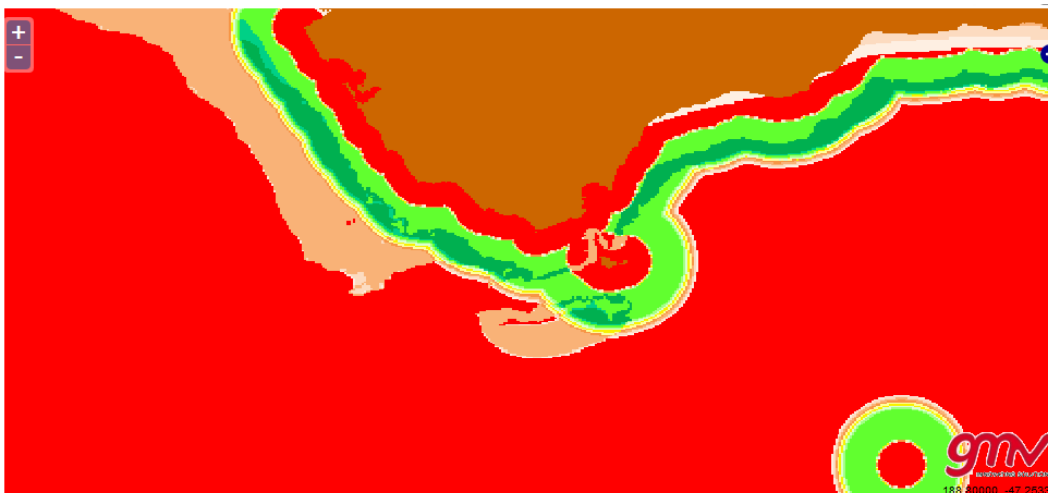


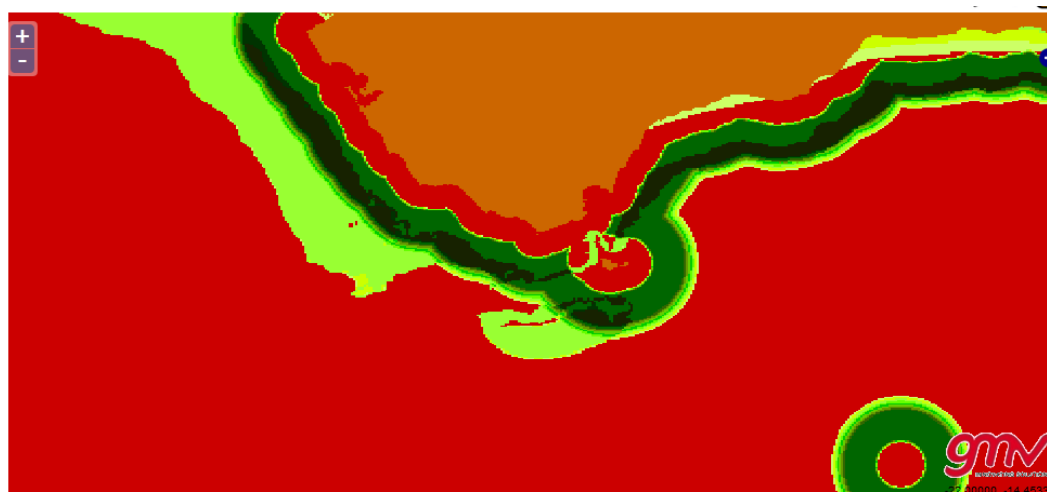
Para la combinación de los criterios distancia a la costa y profundidad utilizando el método de razonamiento: **MLP**, con ponderaciones: **0.8** para el criterio **distancia a la costa** y **0.2** para **profundidad**.

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_1222
v.to.rast input=R_1222 layer=1 type=point,line,area output=R_1223
use=val value=1 rows=4096
r.grow.distance input=R_1223 distance=R_1224 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_1225
v.to.rast input=R_1225 layer=1 type=point,line,area output=R_1226
use=val value=0 rows=4096
```



```
r.null map=R_1226 null=1
r.mapcalc "R_1227 = R_1224 * R_1226 "
r.mapcalc "R_1228 = R_1227 * 100"
r.mapcalc "R_1229 = if(isnull( R_1228 ) , null(), if((R_1228 <= 8 ||
R_1228 >= 25), 0.0, if((R_1228 > 8 && R_1228 < 9), eval((R_1228 - 8)/(9
-8 )), if((R_1228 > 20 && R_1228 < 25 ), eval((25 - R_1228 )/(25 - 20)),
1.0))))"
r.null map=R_1229 value=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_1230
r.mapcalc "R_1231 = if(isnull( R_1230 ) , null(), if((R_1230 <= -105.0
|| R_1230 >= -15.0), 0.0, if((R_1230 > -105.0 && R_1230 < -100.0),
eval((R_1230 - -105.0)/(-100.0 --105.0 )), if((R_1230 > -20.0 && R_1230
< -15.0 ), eval((-15.0 - R_1230 )/(-15.0 - -20.0)), 1.0))))"
r.mapcalc "R_1232 = 0.8 * R_1229 + 0.2 * R_1231"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1233 = if(isnull(ccaa),R_1232, -1.0)"
g.region n=36.802844848967 s=35.045032348967 e=-3.764203828125 w=-
7.521528046875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1229
output="C:\Users\GSI\OceanLider\Resultados\79\RI_R_1229.tif"
r.out.gdal input=R_1231
output="C:\Users\GSI\OceanLider\Resultados\79\RI_R_1231.tif"
r.out.gdal input=R_1233
output="C:\Users\GSI\OceanLider\Resultados\79\RF_R_1233.tif"
```





2.2. Combinación de criterios operacionales

2.2.1. Combinación 5

Combinación de los subcriterios del criterio **mantenimiento de la posición** (de exclusión parcial): *facilidad de anclaje al fondo (composición del sustrato marino)* y *facilidad de anclaje al fondo (morfología al fondo marino)*. El método de razonamiento para combinar dichos subcriterios es el **más que pesimista** (multiplicación de valores) que está configurado por defecto.

Se ha escogido un tipo de dispositivo: **undimotriz- fuera costa (flotante)**

Los parámetros que se han configurado para estos criterios son:

- ✓ Para el subcriterio de facilidad de anclaje al fondo (composición del sustrato marino): Se extraen los valores de idoneidad de las zonas configuradas para la capa Geofis dependiendo del tipo de dispositivo elegido y se le deja la libertad al usuario para poder cambiar estos valores a través de la interfaz. Los valores de idoneidad configurados en la interfaz:

<input checked="" type="checkbox"/> Facilidad de anclaje al fondo (composición del sustrato marino)	
Roca	0.5
Vegetado	1
Blando cohesivo	0
Blando no cohesivo	1

- ✓ Para el subcriterio de facilidad de anclaje al fondo (morfología al fondo marino): Se extraen los valores de idoneidad de las zonas configuradas en la capa Formas dependiendo del tipo de dispositivo elegido y se le deja la libertad al usuario de cambiar los valores según la agrupación para los diferentes tipos. Los valores de idoneidad configurados en la interfaz:

Facilidad de anclaje al fondo (morfología del fondo marino) 

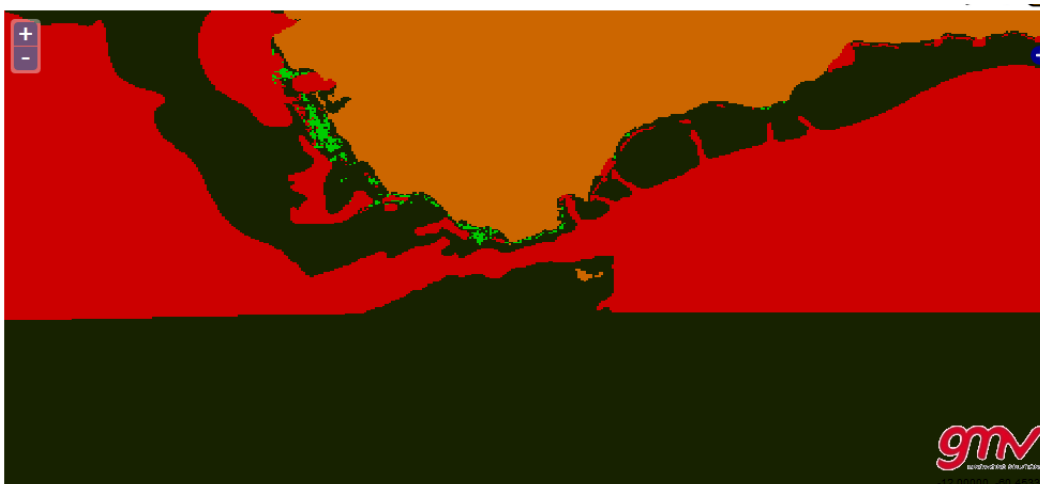
Grupo0	0
Grupo1	1

```

SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1234 layer=geofis
where="Agrup='Roca'"
v.to.rast input=R_1234 layer=1 type=point,line,area output=R_1235
use=val value=0.5 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1236 layer=geofis
where="Agrup='Vegetado'"
v.to.rast input=R_1236 layer=1 type=point,line,area output=R_1237
use=val value=1 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1238 layer=geofis
where="Agrup='Blando cohesivo'"
v.to.rast input=R_1238 layer=1 type=point,line,area output=R_1239
use=val value=0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1240 layer=geofis
where="Agrup='Blando no cohesivo'"
v.to.rast input=R_1240 layer=1 type=point,line,area output=R_1241
use=val value=1 rows=4096
r.patch input=R_1235,R_1237 output=R_1242
r.patch input=R_1239,R_1241 output=R_1243
r.patch input=R_1242,R_1243 output=R_1244
r.null map=R_1244 null=1
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1245 layer=formas
where="(TIPO='Megaripples') OR (TIPO='Abanico deltaico') OR
(TIPO='Abanico deltaico profundo') OR (TIPO='Caynon submarino') OR
(TIPO='Caynon submarino') OR (TIPO='Llanuras abisales') OR
(TIPO='Morfologia de crestas y valles') OR (TIPO='Megaripples') OR
(TIPO='Ondas de arena') OR (TIPO='Relieve volcanico')"
v.to.rast input=R_1245 layer=1 type=point,line,area output=R_1246
use=val value=0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1247 layer=formas
where="(TIPO='Ripples') OR (TIPO='Marcas de arrastre') OR (TIPO='Talud
    
```

```
continental') OR (TIPO='Afloramiento rocoso') OR (TIPO='Playas
sumergidas') OR (TIPO='Plataforma continental')"
v.to.rast input=R_1247 layer=1 type=point,line,area output=R_1248
use=val value=1 rows=4096
r.patch input=R_1246,R_1248 output=R_1249
r.null map=R_1249 null=1
r.mapcalc "R_1250 = R_1244 * R_1249 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1251 = if(isnull(ccaa),R_1250, -1.0)"
g.region n=36.825916540361 s=35.134022009111 e=-3.720258515625 w=-
7.389692109375 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1244
output="C:\Users\GSI\OceanLider\Resultados\80\RI_R_1244.tif"
r.out.gdal input=R_1249
output="C:\Users\GSI\OceanLider\Resultados\80\RI_R_1249.tif"
r.out.gdal input=R_1251
output="C:\Users\GSI\OceanLider\Resultados\80\RF_R_1251.tif"
```





2.2.2. Combinación 6

Combinación de los subcriterios del criterio **viabilidad de instalación en zonas ocupadas** (de exclusión parcial): *viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español, viabilidad de instalación en zonas designadas como Dominio Público Portuario, viabilidad de instalación en zonas preseleccionadas para la extracción de áridos, viabilidad de instalación en zonas ocupadas por conducciones submarinas, viabilidad de instalación en zonas preseleccionadas para el vertido de material de dragado, viabilidad de instalación en zonas ocupadas por sensores instrumentales de medición de parámetros oceanográficos y viabilidad de instalación en zonas ocupadas por dispositivos de separación del tráfico marítimo.*

Los parámetros que intervienen en cada subcriterio son:


- ✓ Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español.

Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español. Se aplica una operación difusa sobre la distancia a las zonas ocupadas por las ayudas a la navegación del sistema portuario español. Las medidas se aplican en Km.

De subida ▾

Valor A	0.5
Valor B	1
Valor C	99999
Valor D	99999


- ✓ Viabilidad de instalación en zonas desinadas como Dominio Público Portuario.

Viabilidad de instalación en zonas designadas como Dominio Público Portuario. Se aplica una operación difusa sobre la distancia a las zonas designadas como Dominio Público Portuario. Las medidas se aplican en Km. 

De subida ▾

Valor A	0
Valor B	1
Valor C	99999
Valor D	99999


✓ Viabilidad de instalación en zonas preseleccionadas para la extracción de áridos.

Viabilidad de instalación en zonas preseleccionadas para la extracción de áridos. Se aplica una operación difusa sobre la distancia a las zonas preseleccionadas para la extracción de áridos. Las medidas se aplican en Km. 

De subida ▾

Valor A	1
Valor B	2
Valor C	99999
Valor D	99999


✓ Viabilidad de instalación en zonas ocupadas por conducciones submarinas.

Viabilidad de instalación en zonas preseleccionadas para el vertido de material de dragado. Se aplica una operación difusa sobre la distancia a las zonas preseleccionadas para el vertido de material de dragado. Las medidas se aplican en Km. 

De subida ▾

Valor A	1
Valor B	2
Valor C	99999
Valor D	99999

✓ Viabilidad de instalación en zonas preseleccionadas para el vertido de material de dragado.

Viabilidad de instalación en zonas preseleccionadas para el vertido de material de dragado. Se aplica una operación difusa sobre la distancia a las zonas preseleccionadas para el vertido de material de dragado. Las medidas se aplican en Km. 

De subida ▾

Valor A	1
Valor B	2
Valor C	99999
Valor D	99999


✓ Viabilidad de instalación den zonas ocupadas por sensores instrumentales de medición de parámetros oceanográficos.

Viabilidad de instalación en zonas ocupadas por sensores instrumentales de medición de parámetros oceanográficos. Se aplica una operación difusa sobre la distancia a las zonas ocupadas por sensores instrumentales de medición de parámetros oceanográficos. Las medidas se aplican en Km. 

De subida ▼

Valor A	0.5
Valor B	1
Valor C	99999
Valor D	99999

✓ Viabilidad de instalación en zonas ocupadas por dispositivos de separación de tráfico marítimo.

Viabilidad de instalación en zonas ocupadas por dispositivos de separación del tráfico marítimo. Se aplica una operación difusa sobre la distancia a las zonas ocupadas por dispositivos de separación del tráfico marítimo. Las medidas se aplican en Km. 

De subida ▼

Valor A	1.5
Valor B	3
Valor C	99999
Valor D	99999

El método de razonamiento para su combinación viene establecido por defecto: **pesimista** (mínimo de los valores).

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=conducciones output=R_1264
v.to.rast input=R_1264 layer=1 type=point,line,area output=R_1265
use=val value=0 rows=4096
r.grow.distance input=R_1265 distance=R_1266 metric=euclidean
r.mapcalc "R_1267 = R_1266 * 100"
r.mapcalc "R_1268 = if(isnull( R_1267 ), null(), if((R_1267 <= 1.5 ||
R_1267 >= 99999), 0.0, if((R_1267 > 1.5 && R_1267 < 3), eval((R_1267 -
1.5)/(3 -1.5 )), if((R_1267 > 99999 && R_1267 < 99999 ), eval((99999 -
R_1267)/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=dst output=R_1269
v.to.rast input=R_1269 layer=1 type=point,line,area output=R_1270
use=val value=0 rows=4096
r.grow.distance input=R_1270 distance=R_1271 metric=euclidean
r.mapcalc "R_1272 = R_1271 * 100"
r.mapcalc "R_1273 = if(isnull( R_1272 ), null(), if((R_1272 <= 1.5 ||
R_1272 >= 99999), 0.0, if((R_1272 > 1.5 && R_1272 < 3), eval((R_1272 -
1.5)/(3 -1.5 )), if((R_1272 > 99999 && R_1272 < 99999 ), eval((99999 -
R_1272)/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=aton output=R_1274
v.to.rast input=R_1274 layer=1 type=point,line,area output=R_1275
use=val value=0 rows=4096
r.grow.distance input=R_1275 distance=R_1276 metric=euclidean
```

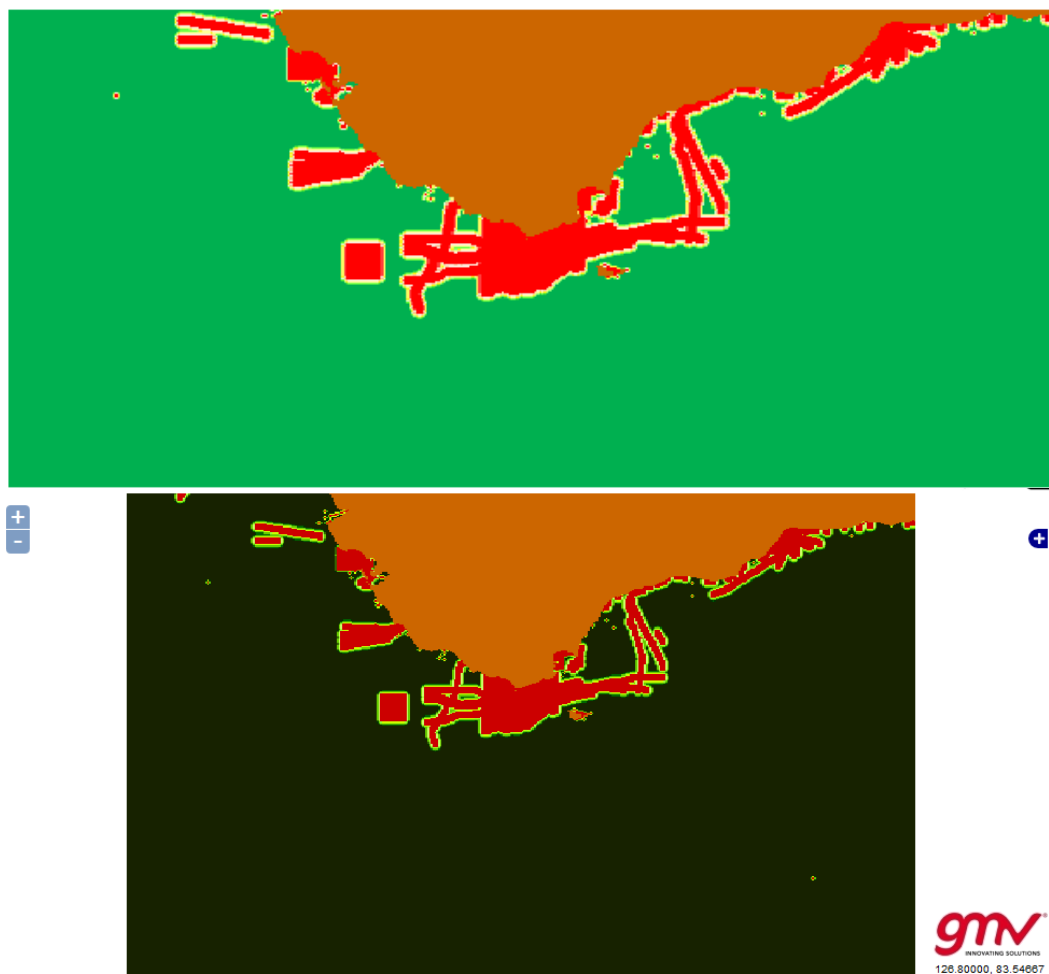
```

r.mapcalc "R_1277 = R_1276 * 100"
r.mapcalc "R_1278 = if(isnull( R_1277 ) , null(), if((R_1277 <= 0.5 ||
R_1277 >= 99999), 0.0, if((R_1277 > 0.5 && R_1277 < 1), eval((R_1277 -
0.5)/(1 -0.5 )), if((R_1277 > 99999 && R_1277 < 99999 ), eval((99999 -
R_1277)/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=vertido_dragado output=R_1279
v.to.rast input=R_1279 layer=1 type=point,line,area output=R_1280
use=val value=0 rows=4096
r.grow.distance input=R_1280 distance=R_1281 metric=euclidean
r.mapcalc "R_1282 = R_1281 * 100"
r.mapcalc "R_1283 = if(isnull( R_1282 ) , null(), if((R_1282 <= 1 ||
R_1282 >= 99999), 0.0, if((R_1282 > 1 && R_1282 < 2), eval((R_1282 -
1)/(2 -1 )), if((R_1282 > 99999 && R_1282 < 99999 ), eval((99999 -
R_1282)/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=rmo output=R_1284
v.to.rast input=R_1284 layer=1 type=point,line,area output=R_1285
use=val value=0 rows=4096
r.grow.distance input=R_1285 distance=R_1286 metric=euclidean
r.mapcalc "R_1287 = R_1286 * 100"
r.mapcalc "R_1288 = if(isnull( R_1287 ) , null(), if((R_1287 <= 0.5 ||
R_1287 >= 99999), 0.0, if((R_1287 > 0.5 && R_1287 < 1), eval((R_1287 -
0.5)/(1 -0.5 )), if((R_1287 > 99999 && R_1287 < 99999 ), eval((99999 -
R_1287)/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=dpp output=R_1289
v.to.rast input=R_1289 layer=1 type=point,line,area output=R_1290
use=val value=0 rows=4096
r.grow.distance input=R_1290 distance=R_1291 metric=euclidean
r.mapcalc "R_1292 = R_1291 * 100"
r.mapcalc "R_1293 = if(isnull( R_1292 ) , null(), if((R_1292 <= 0 ||
R_1292 >= 99999), 0.0, if((R_1292 > 0 && R_1292 < 1), eval((R_1292 -
0)/(1 -0 )), if((R_1292 > 99999 && R_1292 < 99999 ), eval((99999 -
R_1292)/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ext_aridos output=R_1294
v.to.rast input=R_1294 layer=1 type=point,line,area output=R_1295
use=val value=0 rows=4096
r.grow.distance input=R_1295 distance=R_1296 metric=euclidean
r.mapcalc "R_1297 = R_1296 * 100"
r.mapcalc "R_1298 = if(isnull( R_1297 ) , null(), if((R_1297 <= 1 ||
R_1297 >= 99999), 0.0, if((R_1297 > 1 && R_1297 < 2), eval((R_1297 -
1)/(2 -1 )), if((R_1297 > 99999 && R_1297 < 99999 ), eval((99999 -
R_1297)/(99999 - 99999)), 1.0))))"
r.mapcalc "R_1299 = if(isnull( R_1268 ) || isnull( R_1273 ), null(),
if(R_1268 <= R_1273 , R_1268 , R_1273 ))"
r.mapcalc "R_1300 = if(isnull( R_1299 ) || isnull( R_1273 ), null(),
if(R_1299 <= R_1273 , R_1299 , R_1273 ))"
r.mapcalc "R_1301 = if(isnull( R_1300 ) || isnull( R_1278 ), null(),
if(R_1300 <= R_1278 , R_1300 , R_1278 ))"
r.mapcalc "R_1302 = if(isnull( R_1301 ) || isnull( R_1283 ), null(),
if(R_1301 <= R_1283 , R_1301 , R_1283 ))"
r.mapcalc "R_1303 = if(isnull( R_1302 ) || isnull( R_1288 ), null(),
if(R_1302 <= R_1288 , R_1302 , R_1288 ))"

```



```
r.mapcalc "R_1304 = if(isnull( R_1303 ) || isnull( R_1293 ), null(),
if(R_1303 <= R_1293 , R_1303 , R_1293 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1305 = if(isnull(ccaa),R_1304, -1.0)"
g.region n=36.868762817717 s=34.737415161467 e=-3.852094453125 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1268
output="C:\Users\GSI\OceanLider\Resultados\81\RI_R_1268.tif"
r.out.gdal input=R_1273
output="C:\Users\GSI\OceanLider\Resultados\81\RI_R_1273.tif"
r.out.gdal input=R_1278
output="C:\Users\GSI\OceanLider\Resultados\81\RI_R_1278.tif"
r.out.gdal input=R_1283
output="C:\Users\GSI\OceanLider\Resultados\81\RI_R_1283.tif"
r.out.gdal input=R_1288
output="C:\Users\GSI\OceanLider\Resultados\81\RI_R_1288.tif"
r.out.gdal input=R_1293
output="C:\Users\GSI\OceanLider\Resultados\81\RI_R_1293.tif"
r.out.gdal input=R_1298
output="C:\Users\GSI\OceanLider\Resultados\81\RI_R_1298.tif"
r.out.gdal input=R_1305
output="C:\Users\GSI\OceanLider\Resultados\81\RF_R_1305.tif"
```



2.2.3. Combinación 7

Combinación de los subcriterios del criterio **viabilidad de instalación en zonas restringidas** (de exclusión parcial): *viabilidad de instalación en zonas restringidas para uso militar, viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías, viabilidad de instalación en zonas restringidas por la presencia de vertidos, y viabilidad de instalación en zonas restringidas por la existencia de vertidos de material militar.*

Los parámetros que intervienen en cada subcriterio son:

- ✓ Viabilidad de instalación en zonas restringidas para uso militar.

Viabilidad de instalación en zonas restringidas para uso militar. Se aplica una operación difusa sobre la distancia a las zonas restringidas para uso militar. Las medidas se aplican en Km. 

De subida ▾

Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="2"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>

✓ Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías.

Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Las medidas se aplican en Km. 

De subida ▾

Valor A	<input type="text" value="1.5"/>
Valor B	<input type="text" value="3"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>


✓ Viabilidad de instalación en zonas restringidas por la presencia de vertidos.

Viabilidad de instalación en zonas restringidas por la presencia de vertidos. Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de vertidos. Las medidas se aplican en Km. 

De subida ▾

Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="2"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>

✓ Viabilidad de instalación en zonas restringidas por la existencia de vertidos de material militar.

Viabilidad de instalación en zonas restringidas por la existencia de vertidos de material militar. Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de vertidos de material militar. Las medidas se aplican en Km. 

De subida ▾

Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="2"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>

El método de razonamiento para su combinación viene establecido por defecto: **pesimista** (mínimo de los valores).

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
```

```

v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=militar output=R_1306
v.to.rast input=R_1306 layer=1 type=point,line,area output=R_1307
use=val value=0 rows=4096
r.grow.distance input=R_1307 distance=R_1308 metric=euclidean
r.mapcalc "R_1309 = R_1308 * 100"
r.mapcalc "R_1310 = if(isnull( R_1309 ) , null(), if((R_1309 <= 1 ||
R_1309 >= 99999), 0.0, if((R_1309 > 1 && R_1309 < 2), eval((R_1309 -
1)/(2 -1 )), if((R_1309 > 99999 && R_1309 < 99999 ), eval((99999 -
R_1309 )/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=zona_cables output=R_1311
v.to.rast input=R_1311 layer=1 type=point,line,area output=R_1312
use=val value=0 rows=4096
r.grow.distance input=R_1312 distance=R_1313 metric=euclidean
r.mapcalc "R_1314 = R_1313 * 100"
r.mapcalc "R_1315 = if(isnull( R_1314 ) , null(), if((R_1314 <= 1.5 ||
R_1314 >= 99999), 0.0, if((R_1314 > 1.5 && R_1314 < 3), eval((R_1314 -
1.5)/(3 -1.5 )), if((R_1314 > 99999 && R_1314 < 99999 ), eval((99999 -
R_1314 )/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=vertido_material_militar
output=R_1316
v.to.rast input=R_1316 layer=1 type=point,line,area output=R_1317
use=val value=0 rows=4096
r.grow.distance input=R_1317 distance=R_1318 metric=euclidean
r.mapcalc "R_1319 = R_1318 * 100"
r.mapcalc "R_1320 = if(isnull( R_1319 ) , null(), if((R_1319 <= 1 ||
R_1319 >= 99999), 0.0, if((R_1319 > 1 && R_1319 < 2), eval((R_1319 -
1)/(2 -1 )), if((R_1319 > 99999 && R_1319 < 99999 ), eval((99999 -
R_1319 )/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=vertidos output=R_1321
v.to.rast input=R_1321 layer=1 type=point,line,area output=R_1322
use=val value=0 rows=4096
r.grow.distance input=R_1322 distance=R_1323 metric=euclidean
r.mapcalc "R_1324 = R_1323 * 100"
r.mapcalc "R_1325 = if(isnull( R_1324 ) , null(), if((R_1324 <= 1 ||
R_1324 >= 99999), 0.0, if((R_1324 > 1 && R_1324 < 2), eval((R_1324 -
1)/(2 -1 )), if((R_1324 > 99999 && R_1324 < 99999 ), eval((99999 -
R_1324 )/(99999 - 99999)), 1.0))))"
r.mapcalc "R_1326 = if(isnull( R_1310 ) || isnull( R_1315 ), null(),
if(R_1310 <= R_1315 , R_1310 , R_1315 )"
r.mapcalc "R_1327 = if(isnull( R_1326 ) || isnull( R_1315 ), null(),
if(R_1326 <= R_1315 , R_1326 , R_1315 )"
r.mapcalc "R_1328 = if(isnull( R_1327 ) || isnull( R_1320 ), null(),
if(R_1327 <= R_1320 , R_1327 , R_1320 )"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1329 = if(isnull(ccaa),R_1328, -1.0)"
g.region n=36.868762817717 s=35.176868286467 e=-3.852094453125 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453

```

Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

```
r.out.gdal input=R_1310  
output="C:\Users\GSI\OceanLider\Resultados\82\RI_R_1310.tif"  
r.out.gdal input=R_1315  
output="C:\Users\GSI\OceanLider\Resultados\82\RI_R_1315.tif"  
r.out.gdal input=R_1320  
output="C:\Users\GSI\OceanLider\Resultados\82\RI_R_1320.tif"  
r.out.gdal input=R_1325  
output="C:\Users\GSI\OceanLider\Resultados\82\RI_R_1325.tif"  
r.out.gdal input=R_1329  
output="C:\Users\GSI\OceanLider\Resultados\82\RF_R_1329.tif"
```




2.2.4. Combinación 8

Combinación de tres criterios operacionales: **mantenimiento de la posición** (de exclusión parcial), **viabilidad de instalación en zonas ocupadas** (de exclusión parcial) y **viabilidad de instalación en zonas restringidas** (de exclusión parcial).

- En el criterio **mantenimiento de la posición** han intervenido dos subcriterios que se combinan por defecto con el método de razonamiento: **más que pesimista** (multiplicación de valores): *facilidad de anclaje al fondo (composición del sustrato marino)* y *facilidad de anclaje al fondo (morfología del fondo marino)*.
- En el criterio **viabilidad de instalación en zonas ocupadas** han intervenido dos subcriterios que se combinan por defecto con el método **pesimista** (mínimo de los valores): *viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español* y *viabilidad de instalación en zonas designadas como Dominio Público Portuario*.
- En el criterio **viabilidad de instalación en zonas restringidas** han intervenido dos subcriterios que se combinan por defecto con el método de razonamiento **pesimista** (mínimo de los valores): *viabilidad de instalación en zonas restringidas para uso militar* y *viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías*.

Los parámetros establecidos en cada subcriterio:

- ✓ *Para el subcriterio de facilidad de anclaje al fondo (composición del sustrato marino):* Se extraen los valores de idoneidad de las zonas configuradas para la capa Geofis dependiendo del tipo de dispositivo elegido (en este escenario se ha configurado un tipo de dispositivo undimotriz- fuera costa (flotante)) y se le deja la libertad al usuario para poder cambiar estos valores a través de la interfaz. Los valores de idoneidad configurados en la interfaz:


<input checked="" type="checkbox"/> Facilidad de anclaje al fondo (composición del sustrato marino) 	
Roca	0.5
Vegetado	1
Blando cohesivo	0
Blando no cohesivo	1

- ✓ *Para el subcriterio de facilidad de anclaje al fondo (morfología al fondo marino):* Se extraen los valores de idoneidad de las zonas configuradas en la capa Formas dependiendo del tipo de dispositivo elegido (en este escenario se ha configurado un tipo de dispositivo undimotriz- fuera costa (flotante)) y se le deja la libertad al usuario de cambiar los valores según la agrupación para los diferentes tipos. Los valores de idoneidad configurados en la interfaz:

Facilidad de anclaje al fondo (morfología del fondo marino) 

Grupo0	<input type="text" value="0"/>
Grupo1	<input type="text" value="1"/>

✓ *Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español.*

Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español. Se aplica una operación difusa sobre la distancia a las zonas ocupadas por las ayudas a la navegación del sistema portuario español. Las medidas se aplican en Km. 

De subida ▾

Valor A	<input type="text" value="0.5"/>
Valor B	<input type="text" value="1"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>

✓ *Viabilidad de instalación en zonas desinadas como Dominio Público Portuario.*

Viabilidad de instalación en zonas designadas como Dominio Público Portuario. Se aplica una operación difusa sobre la distancia a las zonas designadas como Dominio Público Portuario. Las medidas se aplican en Km. 

De subida ▾

Valor A	<input type="text" value="0"/>
Valor B	<input type="text" value="1"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>

✓ *Viabilidad de instalación en zonas restringidas para uso militar.*

Viabilidad de instalación en zonas restringidas para uso militar. Se aplica una operación difusa sobre la distancia a las zonas restringidas para uso militar. Las medidas se aplican en Km. 

De subida ▾

Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="2"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>

✓ *Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías.*

Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Las medidas se aplican en Km. 

De subida ▾	
Valor A	1.5
Valor B	3
Valor C	99999
Valor D	99999

El método de razonamiento que se ha escogido para la combinación de los tres criterios: **más que pesimista (multiplicación de valores)**

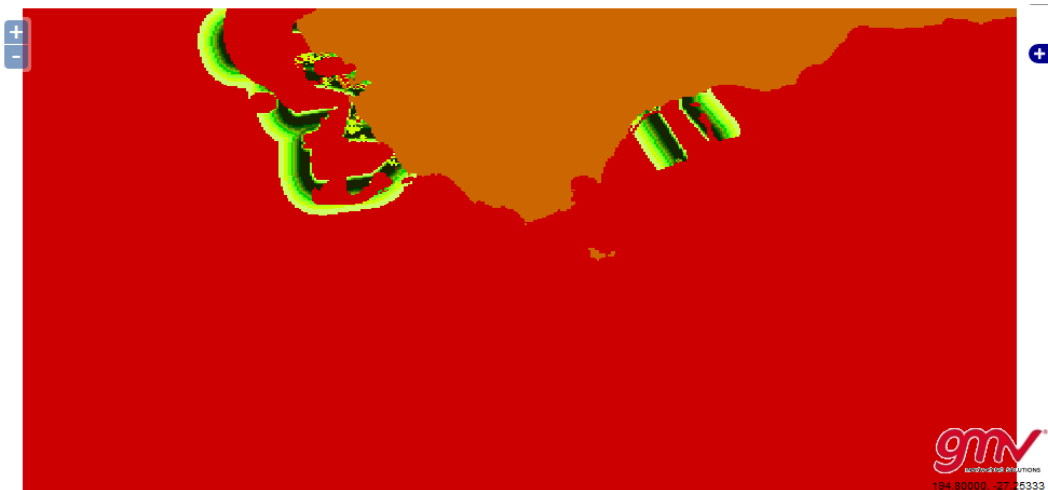
```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1330 layer=geofis
where="Agrup='Roca'"
v.to.rast input=R_1330 layer=1 type=point,line,area output=R_1331
use=val value=0.5 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1332 layer=geofis
where="Agrup='Vegetado'"
v.to.rast input=R_1332 layer=1 type=point,line,area output=R_1333
use=val value=1 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1334 layer=geofis
where="Agrup='Blando cohesivo'"
v.to.rast input=R_1334 layer=1 type=point,line,area output=R_1335
use=val value=0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1336 layer=geofis
where="Agrup='Blando no cohesivo'"
v.to.rast input=R_1336 layer=1 type=point,line,area output=R_1337
use=val value=1 rows=4096
r.patch input=R_1331,R_1333 output=R_1338
r.patch input=R_1335,R_1337 output=R_1339
r.patch input=R_1338,R_1339 output=R_1340
r.null map=R_1340 null=1
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1341 layer=formas
where="(TIPO='Megaripples') OR (TIPO='Abanico deltaico') OR
(TIPO='Abanico deltaico profundo') OR (TIPO='Caynon submarino') OR
(TIPO='Caynon submarino') OR (TIPO='Llanuras abisales') OR
(TIPO='Morfologia de crestas y valles') OR (TIPO='Megaripples') OR
(TIPO='Ondas de arena') OR (TIPO='Relieve volcanico')"
v.to.rast input=R_1341 layer=1 type=point,line,area output=R_1342
use=val value=0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1343 layer=formas
where="(TIPO='Ripples') OR (TIPO='Marcas de arrastre') OR (TIPO='Talud
continental') OR (TIPO='Afloramiento rocoso') OR (TIPO='Playas
sumergidas') OR (TIPO='Plataforma continental')"
```



```
v.to.rast input=R_1343 layer=1 type=point,line,area output=R_1344
use=val value=1 rows=4096
r.patch input=R_1342,R_1344 output=R_1345
r.null map=R_1345 null=1
r.mapcalc "R_1366 = R_1340 * R_1345 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=zona_cables output=R_1346
v.to.rast input=R_1346 layer=1 type=point,line,area output=R_1347
use=val value=0 rows=4096
r.grow.distance input=R_1347 distance=R_1348 metric=euclidean
r.mapcalc "R_1349 = R_1348 * 100"
r.mapcalc "R_1350 = if(isnull( R_1349 ), null(), if((R_1349 <= 1.5 ||
R_1349 >= 15), 0.0, if((R_1349 > 1.5 && R_1349 < 3), eval((R_1349 -
1.5)/(3 - 1.5)), if((R_1349 > 6 && R_1349 < 15), eval((15 - R_1349
)/(15 - 6)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=militar output=R_1361
v.to.rast input=R_1361 layer=1 type=point,line,area output=R_1362
use=val value=0 rows=4096
r.grow.distance input=R_1362 distance=R_1363 metric=euclidean
r.mapcalc "R_1364 = R_1363 * 100"
r.mapcalc "R_1365 = if(isnull( R_1364 ), null(), if((R_1364 <= 1 ||
R_1364 >= 99999), 0.0, if((R_1364 > 1 && R_1364 < 2), eval((R_1364 -
1)/(2 - 1)), if((R_1364 > 99999 && R_1364 < 99999), eval((99999 -
R_1364)/(99999 - 99999)), 1.0))))"
r.mapcalc "R_1367 = if(isnull( R_1340 ) || isnull( R_1345 ), null(),
if(R_1340 <= R_1345, R_1340, R_1345))"
r.mapcalc "R_1368 = if(isnull( R_1367 ) || isnull( R_1345 ), null(),
if(R_1367 <= R_1345, R_1367, R_1345))"
r.mapcalc "R_1369 = if(isnull( R_1368 ) || isnull( R_1350 ), null(),
if(R_1368 <= R_1350, R_1368, R_1350))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=aton output=R_1351
v.to.rast input=R_1351 layer=1 type=point,line,area output=R_1352
use=val value=0 rows=4096
r.grow.distance input=R_1352 distance=R_1353 metric=euclidean
r.mapcalc "R_1354 = R_1353 * 100"
r.mapcalc "R_1355 = if(isnull( R_1354 ), null(), if((R_1354 <= 0.5 ||
R_1354 >= 99999), 0.0, if((R_1354 > 0.5 && R_1354 < 1), eval((R_1354 -
0.5)/(1 - 0.5)), if((R_1354 > 99999 && R_1354 < 99999), eval((99999 -
R_1354)/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=dpp output=R_1356
v.to.rast input=R_1356 layer=1 type=point,line,area output=R_1357
use=val value=0 rows=4096
r.grow.distance input=R_1357 distance=R_1358 metric=euclidean
r.mapcalc "R_1359 = R_1358 * 100"
r.mapcalc "R_1360 = if(isnull( R_1359 ), null(), if((R_1359 <= 0 ||
R_1359 >= 99999), 0.0, if((R_1359 > 0 && R_1359 < 1), eval((R_1359 -
0)/(1 - 0)), if((R_1359 > 99999 && R_1359 < 99999), eval((99999 -
R_1359)/(99999 - 99999)), 1.0))))"
r.mapcalc "R_1370 = if(isnull( R_1340 ) || isnull( R_1345 ), null(),
if(R_1340 <= R_1345, R_1340, R_1345))"
r.mapcalc "R_1371 = if(isnull( R_1370 ) || isnull( R_1345 ), null(),
if(R_1370 <= R_1345, R_1370, R_1345))"
```

```
r.mapcalc "R_1372 = if(isnull( R_1371 ) || isnull( R_1350 ), null(),
if(R_1371 <= R_1350 , R_1371 , R_1350 ))"
r.mapcalc "R_1373 = if(isnull( R_1372 ) || isnull( R_1365 ), null(),
if(R_1372 <= R_1365 , R_1372 , R_1365 ))"
r.mapcalc "R_1374 = if(isnull( R_1373 ) || isnull( R_1355 ), null(),
if(R_1373 <= R_1355 , R_1373 , R_1355 ))"
r.mapcalc "R_1375 = R_1366 * R_1369 * R_1374 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1376 = if(isnull(ccaa),R_1375, -1.0)"
g.region n=36.781971227861 s=35.024158727861 e=-3.830121796875 w=-
7.433637421875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1340
output="C:\Users\GSI\OceanLider\Resultados\83\RI_R_1340.tif"
r.out.gdal input=R_1345
output="C:\Users\GSI\OceanLider\Resultados\83\RI_R_1345.tif"
r.out.gdal input=R_1350
output="C:\Users\GSI\OceanLider\Resultados\83\RI_R_1350.tif"
r.out.gdal input=R_1365
output="C:\Users\GSI\OceanLider\Resultados\83\RI_R_1365.tif"
r.out.gdal input=R_1355
output="C:\Users\GSI\OceanLider\Resultados\83\RI_R_1355.tif"
r.out.gdal input=R_1360
output="C:\Users\GSI\OceanLider\Resultados\83\RI_R_1360.tif"
r.out.gdal input=R_1376
output="C:\Users\GSI\OceanLider\Resultados\83\RF_R_1376.tif"
```





El método de razonamiento que se ha escogido para la combinación de los tres criterios: **pesimista (mínimo de los valores).**

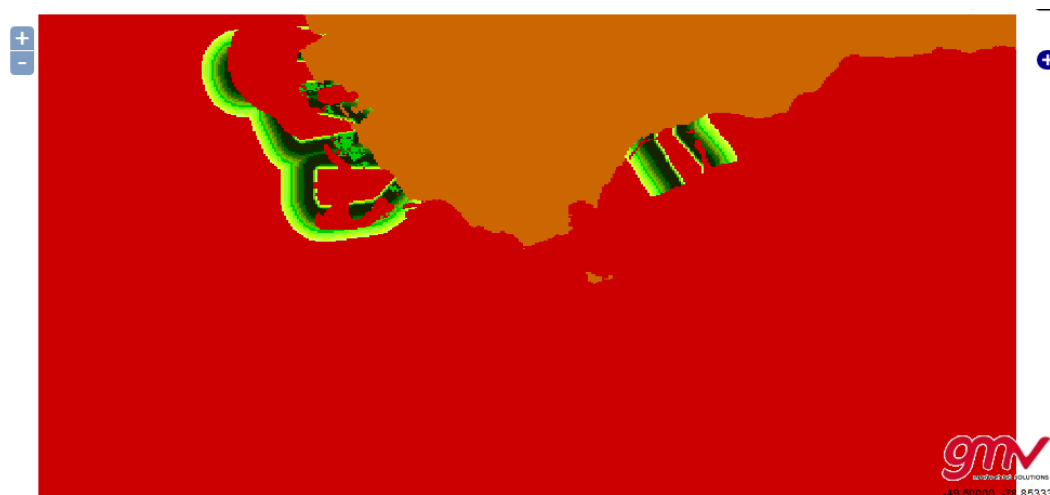
```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1377 layer=geofis
where="Agrup='Roca'"
v.to.rast input=R_1377 layer=1 type=point,line,area output=R_1378
use=val value=0.5 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1379 layer=geofis
where="Agrup='Vegetado'"
v.to.rast input=R_1379 layer=1 type=point,line,area output=R_1380
use=val value=1 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1381 layer=geofis
where="Agrup='Blando cohesivo'"
v.to.rast input=R_1381 layer=1 type=point,line,area output=R_1382
use=val value=0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1383 layer=geofis
where="Agrup='Blando no cohesivo'"
v.to.rast input=R_1383 layer=1 type=point,line,area output=R_1384
use=val value=1 rows=4096
r.patch input=R_1378,R_1380 output=R_1385
r.patch input=R_1382,R_1384 output=R_1386
r.patch input=R_1385,R_1386 output=R_1387
r.null map=R_1387 null=1
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1388 layer=formas
where="(TIPO='Megaripples') OR (TIPO='Abanico deltaico') OR
(TIPO='Abanico deltaico profundo') OR (TIPO='Caynon submarino') OR
(TIPO='Caynon submarino') OR (TIPO='Llanuras abisales') OR
```

```

(TIPO='Morfología de crestas y valles') OR (TIPO='Megaripples') OR
(TIPO='Ondas de arena') OR (TIPO='Relieve volcanico')"
v.to.rast input=R_1388 layer=1 type=point,line,area output=R_1389
use=val value=0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1390 layer=formas
where="(TIPO='Ripples') OR (TIPO='Marcas de arrastre') OR (TIPO='Talud
continental') OR (TIPO='Afloramiento rocoso') OR (TIPO='Playas
sumergidas') OR (TIPO='Plataforma continental')"
v.to.rast input=R_1390 layer=1 type=point,line,area output=R_1391
use=val value=1 rows=4096
r.patch input=R_1389,R_1391 output=R_1392
r.null map=R_1392 null=1
r.mapcalc "R_1413 = R_1387 * R_1392 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=zona_cables output=R_1393
v.to.rast input=R_1393 layer=1 type=point,line,area output=R_1394
use=val value=0 rows=4096
r.grow.distance input=R_1394 distance=R_1395 metric=euclidean
r.mapcalc "R_1396 = R_1395 * 100"
r.mapcalc "R_1397 = if(isnull( R_1396 ) , null(), if((R_1396 <= 1.5 ||
R_1396 >= 15), 0.0, if((R_1396 > 1.5 && R_1396 < 3), eval((R_1396 -
1.5)/(3 - 1.5)), if((R_1396 > 6 && R_1396 < 15), eval((15 - R_1396
)/(15 - 6)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=militar output=R_1408
v.to.rast input=R_1408 layer=1 type=point,line,area output=R_1409
use=val value=0 rows=4096
r.grow.distance input=R_1409 distance=R_1410 metric=euclidean
r.mapcalc "R_1411 = R_1410 * 100"
r.mapcalc "R_1412 = if(isnull( R_1411 ) , null(), if((R_1411 <= 1 ||
R_1411 >= 99999), 0.0, if((R_1411 > 1 && R_1411 < 2), eval((R_1411 -
1)/(2 - 1)), if((R_1411 > 99999 && R_1411 < 99999), eval((99999 -
R_1411)/(99999 - 99999)), 1.0))))"
r.mapcalc "R_1414 = if(isnull( R_1387 ) || isnull( R_1392 ), null(),
if(R_1387 <= R_1392 , R_1387 , R_1392 ))"
r.mapcalc "R_1415 = if(isnull( R_1414 ) || isnull( R_1392 ), null(),
if(R_1414 <= R_1392 , R_1414 , R_1392 ))"
r.mapcalc "R_1416 = if(isnull( R_1415 ) || isnull( R_1397 ), null(),
if(R_1415 <= R_1397 , R_1415 , R_1397 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=aton output=R_1398
v.to.rast input=R_1398 layer=1 type=point,line,area output=R_1399
use=val value=0 rows=4096
r.grow.distance input=R_1399 distance=R_1400 metric=euclidean
r.mapcalc "R_1401 = R_1400 * 100"
r.mapcalc "R_1402 = if(isnull( R_1401 ) , null(), if((R_1401 <= 0.5 ||
R_1401 >= 99999), 0.0, if((R_1401 > 0.5 && R_1401 < 1), eval((R_1401 -
0.5)/(1 - 0.5)), if((R_1401 > 99999 && R_1401 < 99999), eval((99999 -
R_1401)/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=dpp output=R_1403
v.to.rast input=R_1403 layer=1 type=point,line,area output=R_1404
use=val value=0 rows=4096
r.grow.distance input=R_1404 distance=R_1405 metric=euclidean

```

```
r.mapcalc "R_1406 = R_1405 * 100"
r.mapcalc "R_1407 = if(isnull( R_1406 ), null(), if((R_1406 <= 0 ||
R_1406 >= 99999), 0.0, if((R_1406 > 0 && R_1406 < 1), eval((R_1406 -
0)/(1 - 0 )), if((R_1406 > 99999 && R_1406 < 99999 ), eval((99999 -
R_1406)/(99999 - 99999)), 1.0)))"
r.mapcalc "R_1417 = if(isnull( R_1387 ) || isnull( R_1392 ), null(),
if(R_1387 <= R_1392 , R_1387 , R_1392 ))"
r.mapcalc "R_1418 = if(isnull( R_1417 ) || isnull( R_1392 ), null(),
if(R_1417 <= R_1392 , R_1417 , R_1392 ))"
r.mapcalc "R_1419 = if(isnull( R_1418 ) || isnull( R_1397 ), null(),
if(R_1418 <= R_1397 , R_1418 , R_1397 ))"
r.mapcalc "R_1420 = if(isnull( R_1419 ) || isnull( R_1412 ), null(),
if(R_1419 <= R_1412 , R_1419 , R_1412 ))"
r.mapcalc "R_1421 = if(isnull( R_1420 ) || isnull( R_1402 ), null(),
if(R_1420 <= R_1402 , R_1420 , R_1402 ))"
r.mapcalc "R_1422 = if(isnull( R_1413 ) || isnull( R_1416 ), null(),
if(R_1413 <= R_1416 , R_1413 , R_1416 ))"
r.mapcalc "R_1423 = if(isnull( R_1422 ) || isnull( R_1416 ), null(),
if(R_1422 <= R_1416 , R_1422 , R_1416 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1424 = if(isnull(ccaa),R_1423, -1.0)"
g.region n=36.868762817717 s=35.067005005217 e=-3.786176484375 w=-
7.411664765625 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1387
output="C:\Users\GSI\OceanLider\Resultados\84\RI_R_1387.tif"
r.out.gdal input=R_1392
output="C:\Users\GSI\OceanLider\Resultados\84\RI_R_1392.tif"
r.out.gdal input=R_1397
output="C:\Users\GSI\OceanLider\Resultados\84\RI_R_1397.tif"
r.out.gdal input=R_1412
output="C:\Users\GSI\OceanLider\Resultados\84\RI_R_1412.tif"
r.out.gdal input=R_1402
output="C:\Users\GSI\OceanLider\Resultados\84\RI_R_1402.tif"
r.out.gdal input=R_1407
output="C:\Users\GSI\OceanLider\Resultados\84\RI_R_1407.tif"
r.out.gdal input=R_1424
output="C:\Users\GSI\OceanLider\Resultados\84\RF_R_1424.tif"
```



El método de razonamiento que se ha escogido para la combinación de los tres criterios: **optimista (máximo de los valores)**.

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1425 layer=geofis
where="Agrup='Roca' "
v.to.rast input=R_1425 layer=1 type=point,line,area output=R_1426
use=val value=0.5 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1427 layer=geofis
where="Agrup='Vegetado' "
```

```
v.to.rast input=R_1427 layer=1 type=point,line,area output=R_1428
use=val value=1 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1429 layer=geofis
where="Agrup='Blando cohesivo'"
v.to.rast input=R_1429 layer=1 type=point,line,area output=R_1430
use=val value=0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1431 layer=geofis
where="Agrup='Blando no cohesivo'"
v.to.rast input=R_1431 layer=1 type=point,line,area output=R_1432
use=val value=1 rows=4096
r.patch input=R_1426,R_1428 output=R_1433
r.patch input=R_1430,R_1432 output=R_1434
r.patch input=R_1433,R_1434 output=R_1435
r.null map=R_1435 null=1
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1436 layer=formas
where="(TIPO='Megaripples') OR (TIPO='Abanico deltaico') OR
(TIPO='Abanico deltaico profundo') OR (TIPO='Caynon submarino') OR
(TIPO='Caynon submarino') OR (TIPO='Llanuras abisales') OR
(TIPO='Morfologia de crestas y valles') OR (TIPO='Megaripples') OR
(TIPO='Ondas de arena') OR (TIPO='Relieve volcanico')"
v.to.rast input=R_1436 layer=1 type=point,line,area output=R_1437
use=val value=0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1438 layer=formas
where="(TIPO='Ripples') OR (TIPO='Marcas de arrastre') OR (TIPO='Talud
continental') OR (TIPO='Afloramiento rocoso') OR (TIPO='Playas
sumergidas') OR (TIPO='Plataforma continental')"
v.to.rast input=R_1438 layer=1 type=point,line,area output=R_1439
use=val value=1 rows=4096
r.patch input=R_1437,R_1439 output=R_1440
r.null map=R_1440 null=1
r.mapcalc "R_1461 = R_1435 * R_1440 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=zona_cables output=R_1441
v.to.rast input=R_1441 layer=1 type=point,line,area output=R_1442
use=val value=0 rows=4096
r.grow.distance input=R_1442 distance=R_1443 metric=euclidean
r.mapcalc "R_1444 = R_1443 * 100"
r.mapcalc "R_1445 = if(isnull( R_1444 ), null(), if((R_1444 <= 1.5 ||
R_1444 >= 15), 0.0, if((R_1444 > 1.5 && R_1444 < 3), eval((R_1444 -
1.5)/(3 -1.5 )), if((R_1444 > 6 && R_1444 < 15 ), eval((15 - R_1444
)/(15 - 6)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=militar output=R_1456
v.to.rast input=R_1456 layer=1 type=point,line,area output=R_1457
use=val value=0 rows=4096
r.grow.distance input=R_1457 distance=R_1458 metric=euclidean
r.mapcalc "R_1459 = R_1458 * 100"
r.mapcalc "R_1460 = if(isnull( R_1459 ), null(), if((R_1459 <= 1 ||
R_1459 >= 99999), 0.0, if((R_1459 > 1 && R_1459 < 2), eval((R_1459 -
1)/(2 -1 )), if((R_1459 > 99999 && R_1459 < 99999 ), eval((99999 -
R_1459)/(99999 - 99999)), 1.0))))"
```

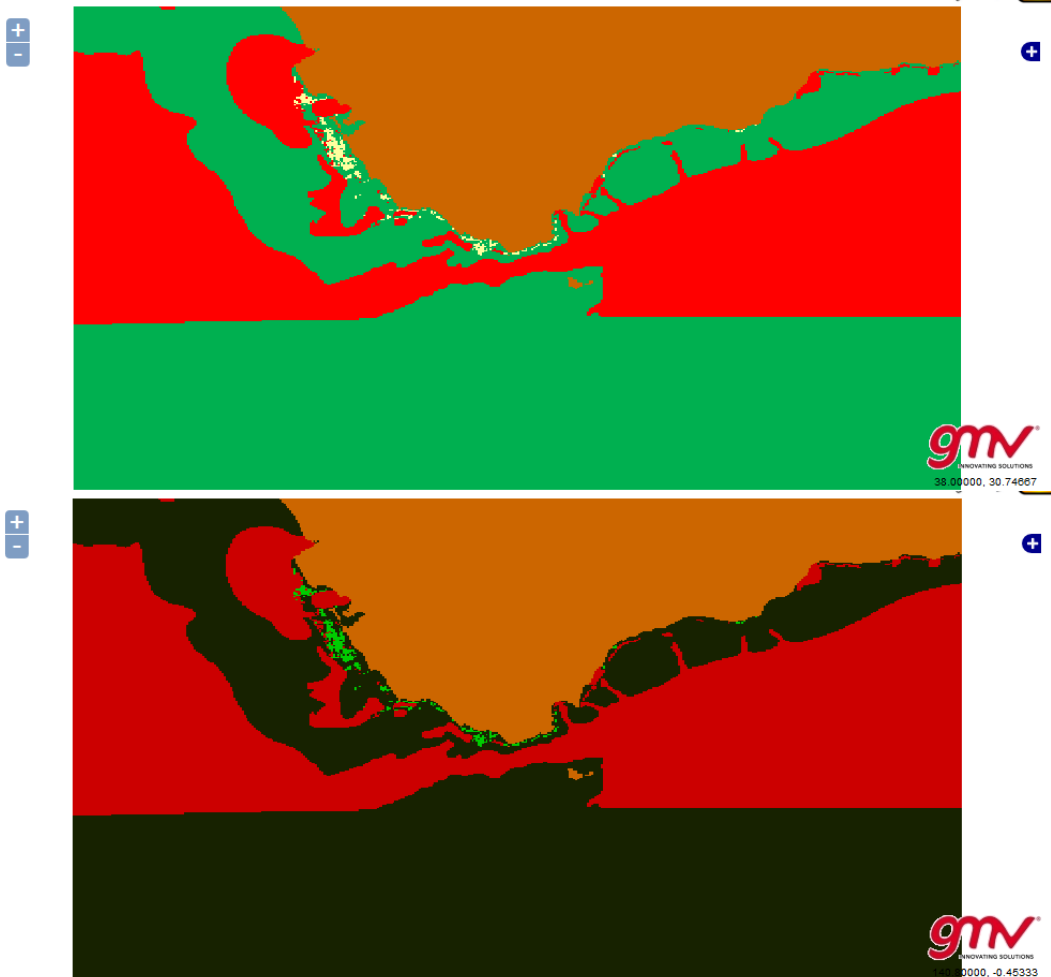
```

r.mapcalc "R_1462 = if(isnull( R_1435 ) || isnull( R_1440 ), null(),
if(R_1435 <= R_1440 , R_1435 , R_1440 ))"
r.mapcalc "R_1463 = if(isnull( R_1462 ) || isnull( R_1440 ), null(),
if(R_1462 <= R_1440 , R_1462 , R_1440 ))"
r.mapcalc "R_1464 = if(isnull( R_1463 ) || isnull( R_1445 ), null(),
if(R_1463 <= R_1445 , R_1463 , R_1445 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=aton output=R_1446
v.to.rast input=R_1446 layer=1 type=point,line,area output=R_1447
use=val value=0 rows=4096
r.grow.distance input=R_1447 distance=R_1448 metric=euclidean
r.mapcalc "R_1449 = R_1448 * 100"
r.mapcalc "R_1450 = if(isnull( R_1449 ) , null(), if((R_1449 <= 0.5 ||
R_1449 >= 99999), 0.0, if((R_1449 > 0.5 && R_1449 < 1), eval((R_1449 -
0.5)/(1 - 0.5 )), if((R_1449 > 99999 && R_1449 < 99999 ), eval((99999 -
R_1449 )/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=dpp output=R_1451
v.to.rast input=R_1451 layer=1 type=point,line,area output=R_1452
use=val value=0 rows=4096
r.grow.distance input=R_1452 distance=R_1453 metric=euclidean
r.mapcalc "R_1454 = R_1453 * 100"
r.mapcalc "R_1455 = if(isnull( R_1454 ) , null(), if((R_1454 <= 0 ||
R_1454 >= 99999), 0.0, if((R_1454 > 0 && R_1454 < 1), eval((R_1454 -
0)/(1 - 0 )), if((R_1454 > 99999 && R_1454 < 99999 ), eval((99999 -
R_1454 )/(99999 - 99999)), 1.0))))"
r.mapcalc "R_1465 = if(isnull( R_1435 ) || isnull( R_1440 ), null(),
if(R_1435 <= R_1440 , R_1435 , R_1440 ))"
r.mapcalc "R_1466 = if(isnull( R_1465 ) || isnull( R_1440 ), null(),
if(R_1465 <= R_1440 , R_1465 , R_1440 ))"
r.mapcalc "R_1467 = if(isnull( R_1466 ) || isnull( R_1445 ), null(),
if(R_1466 <= R_1445 , R_1466 , R_1445 ))"
r.mapcalc "R_1468 = if(isnull( R_1467 ) || isnull( R_1460 ), null(),
if(R_1467 <= R_1460 , R_1467 , R_1460 ))"
r.mapcalc "R_1469 = if(isnull( R_1468 ) || isnull( R_1450 ), null(),
if(R_1468 <= R_1450 , R_1468 , R_1450 ))"
r.mapcalc "R_1470 = if(isnull( R_1461 ) || isnull( R_1464 ), null(),
if(R_1461 >= R_1464 , R_1461 , R_1464 ))"
r.mapcalc "R_1471 = if(isnull( R_1470 ) || isnull( R_1464 ), null(),
if(R_1470 >= R_1464 , R_1470 , R_1464 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1472 = if(isnull(ccaa),R_1471, -1.0)"
g.region n=36.961048242187 s=35.093372460937 e=-3.852094453125 w=-
7.301801484375 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1435
output="C:\Users\GSI\OceanLider\Resultados\85\RI_R_1435.tif"
r.out.gdal input=R_1440
output="C:\Users\GSI\OceanLider\Resultados\85\RI_R_1440.tif"
r.out.gdal input=R_1445
output="C:\Users\GSI\OceanLider\Resultados\85\RI_R_1445.tif"
r.out.gdal input=R_1460
output="C:\Users\GSI\OceanLider\Resultados\85\RI_R_1460.tif"

```


Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

```
r.out.gdal input=R_1450  
output="C:\Users\GSI\OceanLider\Resultados\85\RI_R_1450.tif"  
r.out.gdal input=R_1455  
output="C:\Users\GSI\OceanLider\Resultados\85\RI_R_1455.tif"  
r.out.gdal input=R_1472  
output="C:\Users\GSI\OceanLider\Resultados\85\RF_R_1472.tif"
```



El método de razonamiento que se ha escogido para la combinación de los tres criterios: **MLP con los pesos 0.2** para el criterio mantenimiento de la posición, **0.5** para el criterio viabilidad de instalación en zonas ocupadas y **0.3** para el criterio viabilidad de instalación en zonas restringidas.

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84  
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224  
cols=3453
```

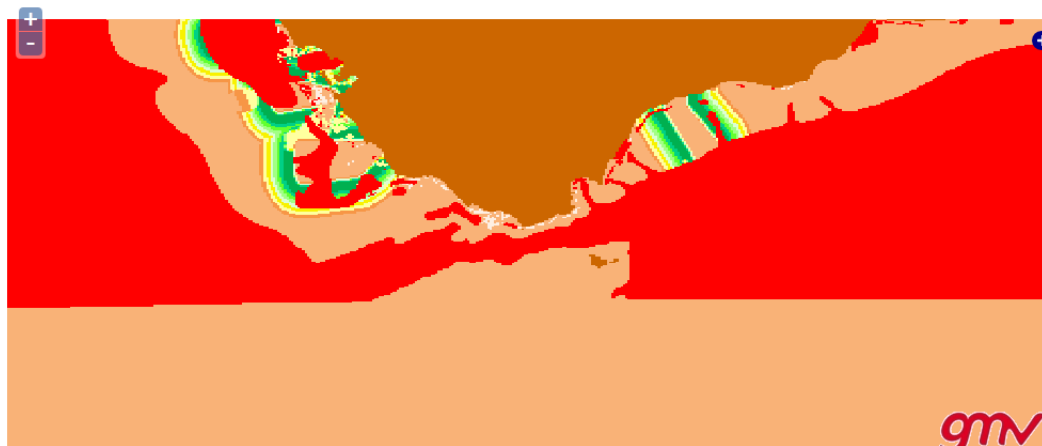
```

v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1473 layer=geofis
where="Agrup='Roca'"
v.to.rast input=R_1473 layer=1 type=point,line,area output=R_1474
use=val value=0.5 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1475 layer=geofis
where="Agrup='Vegetado'"
v.to.rast input=R_1475 layer=1 type=point,line,area output=R_1476
use=val value=1 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1477 layer=geofis
where="Agrup='Blando cohesivo'"
v.to.rast input=R_1477 layer=1 type=point,line,area output=R_1478
use=val value=0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1479 layer=geofis
where="Agrup='Blando no cohesivo'"
v.to.rast input=R_1479 layer=1 type=point,line,area output=R_1480
use=val value=1 rows=4096
r.patch input=R_1474,R_1476 output=R_1481
r.patch input=R_1478,R_1480 output=R_1482
r.patch input=R_1481,R_1482 output=R_1483
r.null map=R_1483 null=1
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1484 layer=formas
where="(TIPO='Megaripples') OR (TIPO='Abanico deltaico') OR
(TIPO='Abanico deltaico profundo') OR (TIPO='Caynon submarino') OR
(TIPO='Caynon submarino') OR (TIPO='Llanuras abisales') OR
(TIPO='Morfologia de crestas y valles') OR (TIPO='Megaripples') OR
(TIPO='Ondas de arena') OR (TIPO='Relieve volcanico')"
v.to.rast input=R_1484 layer=1 type=point,line,area output=R_1485
use=val value=0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_1486 layer=formas
where="(TIPO='Ripples') OR (TIPO='Marcas de arrastre') OR (TIPO='Talud
continental') OR (TIPO='Afloramiento rocoso') OR (TIPO='Playas
sumergidas') OR (TIPO='Plataforma continental')"
v.to.rast input=R_1486 layer=1 type=point,line,area output=R_1487
use=val value=1 rows=4096
r.patch input=R_1485,R_1487 output=R_1488
r.null map=R_1488 null=1
r.mapcalc "R_1509 = R_1483 * R_1488 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=zona_cables output=R_1489
v.to.rast input=R_1489 layer=1 type=point,line,area output=R_1490
use=val value=0 rows=4096
r.grow.distance input=R_1490 distance=R_1491 metric=euclidean
r.mapcalc "R_1492 = R_1491 * 100"
r.mapcalc "R_1493 = if(isnull( R_1492 ), null(), if((R_1492 <= 1.5 ||
R_1492 >= 15), 0.0, if((R_1492 > 1.5 && R_1492 < 3), eval((R_1492 -
1.5)/(3 -1.5 )), if((R_1492 > 6 && R_1492 < 15 ), eval((15 - R_1492
)/(15 - 6)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=militar output=R_1504

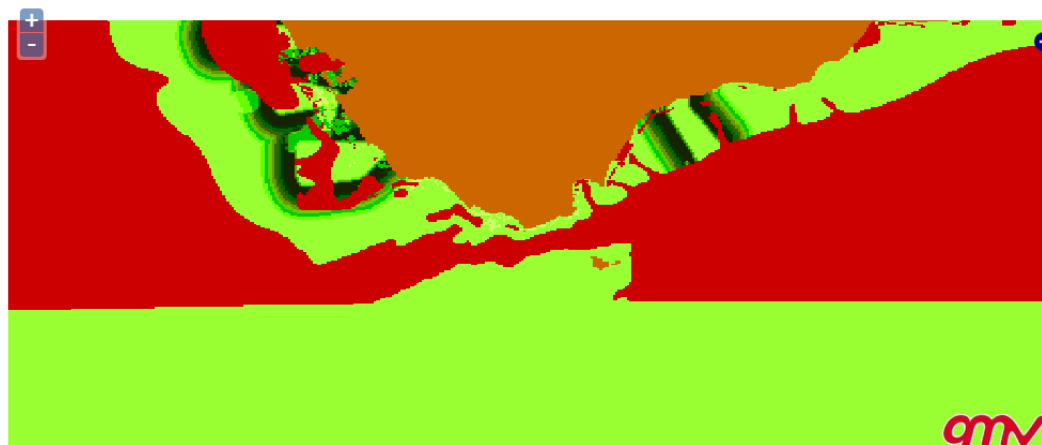
```

```
v.to.rast input=R_1504 layer=1 type=point,line,area output=R_1505
use=val value=0 rows=4096
r.grow.distance input=R_1505 distance=R_1506 metric=euclidean
r.mapcalc "R_1507 = R_1506 * 100"
r.mapcalc "R_1508 = if(isnull( R_1507 ) , null(), if((R_1507 <= 1 ||
R_1507 >= 99999), 0.0, if((R_1507 > 1 && R_1507 < 2), eval((R_1507 -
1)/(2 - 1)), if((R_1507 > 99999 && R_1507 < 99999), eval((99999 -
R_1507)/(99999 - 99999)), 1.0))))"
r.mapcalc "R_1510 = if(isnull( R_1483 ) || isnull( R_1488 ), null(),
if(R_1483 <= R_1488 , R_1483 , R_1488 ))"
r.mapcalc "R_1511 = if(isnull( R_1510 ) || isnull( R_1488 ), null(),
if(R_1510 <= R_1488 , R_1510 , R_1488 ))"
r.mapcalc "R_1512 = if(isnull( R_1511 ) || isnull( R_1493 ), null(),
if(R_1511 <= R_1493 , R_1511 , R_1493 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=aton output=R_1494
v.to.rast input=R_1494 layer=1 type=point,line,area output=R_1495
use=val value=0 rows=4096
r.grow.distance input=R_1495 distance=R_1496 metric=euclidean
r.mapcalc "R_1497 = R_1496 * 100"
r.mapcalc "R_1498 = if(isnull( R_1497 ) , null(), if((R_1497 <= 0.5 ||
R_1497 >= 99999), 0.0, if((R_1497 > 0.5 && R_1497 < 1), eval((R_1497 -
0.5)/(1 - 0.5)), if((R_1497 > 99999 && R_1497 < 99999), eval((99999 -
R_1497)/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=dpp output=R_1499
v.to.rast input=R_1499 layer=1 type=point,line,area output=R_1500
use=val value=0 rows=4096
r.grow.distance input=R_1500 distance=R_1501 metric=euclidean
r.mapcalc "R_1502 = R_1501 * 100"
r.mapcalc "R_1503 = if(isnull( R_1502 ) , null(), if((R_1502 <= 0 ||
R_1502 >= 99999), 0.0, if((R_1502 > 0 && R_1502 < 1), eval((R_1502 -
0)/(1 - 0)), if((R_1502 > 99999 && R_1502 < 99999), eval((99999 -
R_1502)/(99999 - 99999)), 1.0))))"
r.mapcalc "R_1513 = if(isnull( R_1483 ) || isnull( R_1488 ), null(),
if(R_1483 <= R_1488 , R_1483 , R_1488 ))"
r.mapcalc "R_1514 = if(isnull( R_1513 ) || isnull( R_1488 ), null(),
if(R_1513 <= R_1488 , R_1513 , R_1488 ))"
r.mapcalc "R_1515 = if(isnull( R_1514 ) || isnull( R_1493 ), null(),
if(R_1514 <= R_1493 , R_1514 , R_1493 ))"
r.mapcalc "R_1516 = if(isnull( R_1515 ) || isnull( R_1508 ), null(),
if(R_1515 <= R_1508 , R_1515 , R_1508 ))"
r.mapcalc "R_1517 = if(isnull( R_1516 ) || isnull( R_1498 ), null(),
if(R_1516 <= R_1498 , R_1516 , R_1498 ))"
r.mapcalc "R_1518 = 0.2 * R_1509 + 0.3 * R_1512 + 0.5 * R_1517 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_1519 = if(isnull(ccaa),R_1518, -1.0)"
g.region n=36.714954223967 s=35.242786255217 e=-3.808149140625 w=-
7.367719453125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_1483
output="C:\Users\GSI\OceanLider\Resultados\86\RI_R_1483.tif"
```

```
r.out.gdal input=R_1488  
output="C:\Users\GSI\OceanLider\Resultados\86\RI_R_1488.tif"  
r.out.gdal input=R_1493  
output="C:\Users\GSI\OceanLider\Resultados\86\RI_R_1493.tif"  
r.out.gdal input=R_1508  
output="C:\Users\GSI\OceanLider\Resultados\86\RI_R_1508.tif"  
r.out.gdal input=R_1498  
output="C:\Users\GSI\OceanLider\Resultados\86\RI_R_1498.tif"  
r.out.gdal input=R_1503  
output="C:\Users\GSI\OceanLider\Resultados\86\RI_R_1503.tif"  
r.out.gdal input=R_1519  
output="C:\Users\GSI\OceanLider\Resultados\86\RF_R_1519.tif"
```



gmv
INNOVATING SOLUTIONS
194.80000, 2.74887



gmv
INNOVATING SOLUTIONS
194.80000, 3.54887

2.3. Combinación de criterios medioambientales

2.3.1. Combinación 9


Combinación de los subcriterios del criterio **de espacios protegidos** (de exclusión parcial): *geoparques, reservas de la biosfera, patrimonio de la humanidad, red de áreas marinas protegidas del convenio OSPAR, zonas especialmente protegidas de importancia para el Mediterráneo, sitios reconocidos con el Diploma Europeo de Calidad de Espacios Protegidos, lugares de importancia comunitaria, zonas de especial protección de aves, parques nacionales, biotopos protegidos, microrreservas, monumentos naturales, parajes naturales, parques naturales, reservas con ley de protección propia, reservas naturales y zonas de especial protección de los valores naturales.*

Se le deja la libertad al usuario para configurar el valor de idoneidad para las áreas que intervienen en cada subcriterio y se le muestra uno por defecto dependiendo del tipo de dispositivo. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.


El tipo de dispositivo al que esté ligado el escenario: **undimotriz- fuera costa (flotante).**

Los parámetros establecidos en cada subcriterio:

✓ *Subcriterio Geoparques*


<input checked="" type="checkbox"/> Geoparques 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Reservas de la biosfera*

Reservas de la biosfera 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Patrimonio de la humanidad*

Patrimonio de la humanidad 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Red de Áreas Marinas Protegidas del Convenio OSPAR*

Red de Áreas Marinas Protegidas del Convenio OSPAR 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Zonas Especialmente Protegidas de Importancia para el Mediterráneo*

Zonas Especialmente Protegidas de Importancia para el Mediterráneo 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Sitios reconocidos con el Diploma Europeo de Calidad de Espacios Protegidos*

Sitios reconocidos con el Diploma Europeo de Calidad de Espacios Protegidos 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Lugares de Importancia Comunitaria*

Lugares de Importancia Comunitaria 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Zonas de Especial Protección para las Aves*

Zonas de Especial Protección para las Aves 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Parques Nacionales*

Parques Nacionales 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Biotopos protegidos*

Biotopos protegidos 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Microrreservas*

Microrreservas 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Monumentos Naturales*

Monumentos Naturales 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Parajes Naturales*

Parajes Naturales 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Parques Naturales*

Parques Naturales 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Reservas con ley de protección propia*

Reservas con ley de protección propia 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Reservas Naturales*

Reservas Naturales 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Zonas de Especial Protección de los Valores Naturales*

Zonas de Especial Protección de los Valores Naturales 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El método de razonamiento que se ha escogido para la combinación de los criterios: **más que pesimista (multiplicación de los valores).**

```

SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ppnat output=R_5422
v.to.rast input=R_5422 layer=1 type=point,line,area output=R_5423
use=val value=0.0 rows=4096
r.grow.distance input=R_5423 distance=R_5424 metric=euclidean
r.mapcalc "R_5425 = R_5424 * 100"
r.mapcalc "R_5426 = if(isnull( R_5425 ), null(), if((R_5425 <= 1 ||
R_5425 >= 99999), 0.0, if((R_5425 > 1 && R_5425 < 4), eval((R_5425 -
1)/(4 -1 )), if((R_5425 > 99999 && R_5425 < 99999 ), eval((99999 -
R_5425)/(99999 - 99999)), 1.0))))"
r.patch input=R_5423,R_5426 output=R_5427
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepims output=R_5428
v.to.rast input=R_5428 layer=1 type=point,line,area output=R_5429
use=val value=0.0 rows=4096
r.grow.distance input=R_5429 distance=R_5430 metric=euclidean
r.mapcalc "R_5431 = R_5430 * 100"
r.mapcalc "R_5432 = if(isnull( R_5431 ), null(), if((R_5431 <= 1 ||
R_5431 >= 99999), 0.0, if((R_5431 > 1 && R_5431 < 4), eval((R_5431 -
1)/(4 -1 )), if((R_5431 > 99999 && R_5431 < 99999 ), eval((99999 -
R_5431)/(99999 - 99999)), 1.0))))"
r.patch input=R_5429,R_5432 output=R_5433
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=lics output=R_5434
v.to.rast input=R_5434 layer=1 type=point,line,area output=R_5435
use=val value=0.0 rows=4096
r.grow.distance input=R_5435 distance=R_5436 metric=euclidean
r.mapcalc "R_5437 = R_5436 * 100"
r.mapcalc "R_5438 = if(isnull( R_5437 ), null(), if((R_5437 <= 1 ||
R_5437 >= 99999), 0.0, if((R_5437 > 1 && R_5437 < 4), eval((R_5437 -
1)/(4 -1 )), if((R_5437 > 99999 && R_5437 < 99999 ), eval((99999 -
R_5437)/(99999 - 99999)), 1.0))))"
r.patch input=R_5435,R_5438 output=R_5439
    
```

```
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mmnat output=R_5440
v.to.rast input=R_5440 layer=1 type=point,line,area output=R_5441
use=val value=0.0 rows=4096
r.grow.distance input=R_5441 distance=R_5442 metric=euclidean
r.mapcalc "R_5443 = R_5442 * 100"
r.mapcalc "R_5444 = if(isnull( R_5443 ) , null(), if((R_5443 <= 1 ||
R_5443 >= 99999), 0.0, if((R_5443 > 1 && R_5443 < 4), eval((R_5443 -
1)/(4 -1 )), if((R_5443 > 99999 && R_5443 < 99999 ), eval((99999 -
R_5443)/(99999 - 99999)), 1.0))))"
r.patch input=R_5441,R_5444 output=R_5445
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepas output=R_5446
v.to.rast input=R_5446 layer=1 type=point,line,area output=R_5447
use=val value=0.0 rows=4096
r.grow.distance input=R_5447 distance=R_5448 metric=euclidean
r.mapcalc "R_5449 = R_5448 * 100"
r.mapcalc "R_5450 = if(isnull( R_5449 ) , null(), if((R_5449 <= 1 ||
R_5449 >= 99999), 0.0, if((R_5449 > 1 && R_5449 < 4), eval((R_5449 -
1)/(4 -1 )), if((R_5449 > 99999 && R_5449 < 99999 ), eval((99999 -
R_5449)/(99999 - 99999)), 1.0))))"
r.patch input=R_5447,R_5450 output=R_5451
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ppnn output=R_5452
v.to.rast input=R_5452 layer=1 type=point,line,area output=R_5453
use=val value=0.0 rows=4096
r.grow.distance input=R_5453 distance=R_5454 metric=euclidean
r.mapcalc "R_5455 = R_5454 * 100"
r.mapcalc "R_5456 = if(isnull( R_5455 ) , null(), if((R_5455 <= 1 ||
R_5455 >= 99999), 0.0, if((R_5455 > 1 && R_5455 < 4), eval((R_5455 -
1)/(4 -1 )), if((R_5455 > 99999 && R_5455 < 99999 ), eval((99999 -
R_5455)/(99999 - 99999)), 1.0))))"
r.patch input=R_5453,R_5456 output=R_5457
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=rrmn output=R_5458
v.to.rast input=R_5458 layer=1 type=point,line,area output=R_5459
use=val value=0.0 rows=4096
r.grow.distance input=R_5459 distance=R_5460 metric=euclidean
r.mapcalc "R_5461 = R_5460 * 100"
r.mapcalc "R_5462 = if(isnull( R_5461 ) , null(), if((R_5461 <= 1 ||
R_5461 >= 99999), 0.0, if((R_5461 > 1 && R_5461 < 4), eval((R_5461 -
1)/(4 -1 )), if((R_5461 > 99999 && R_5461 < 99999 ), eval((99999 -
R_5461)/(99999 - 99999)), 1.0))))"
r.patch input=R_5459,R_5462 output=R_5463
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=geoparques
output=R_5464
v.to.rast input=R_5464 layer=1 type=point,line,area output=R_5465
use=val value=0.0 rows=4096
r.grow.distance input=R_5465 distance=R_5466 metric=euclidean
r.mapcalc "R_5467 = R_5466 * 100"
r.mapcalc "R_5468 = if(isnull( R_5467 ) , null(), if((R_5467 <= 1 ||
R_5467 >= 99999), 0.0, if((R_5467 > 1 && R_5467 < 4), eval((R_5467 -
1)/(4 -1 )), if((R_5467 > 99999 && R_5467 < 99999 ), eval((99999 -
R_5467)/(99999 - 99999)), 1.0))))"
```

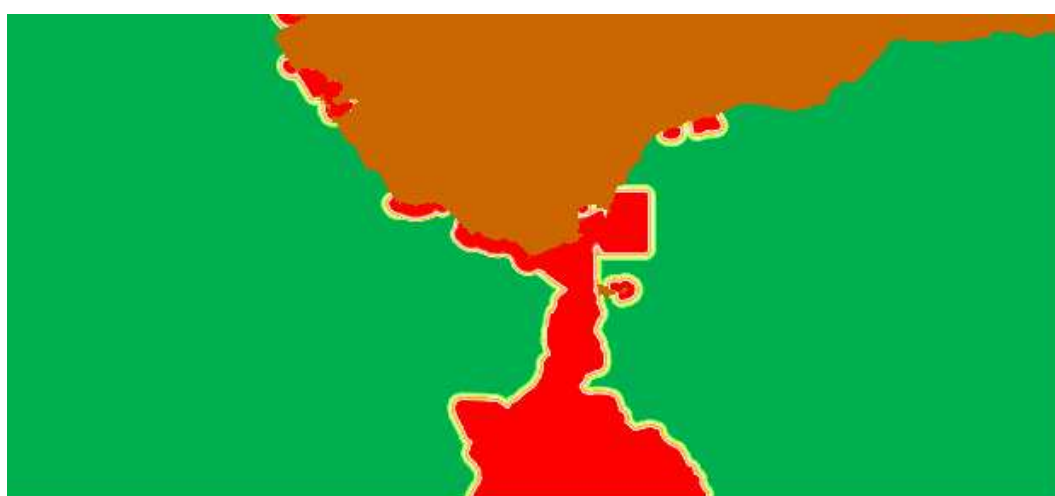
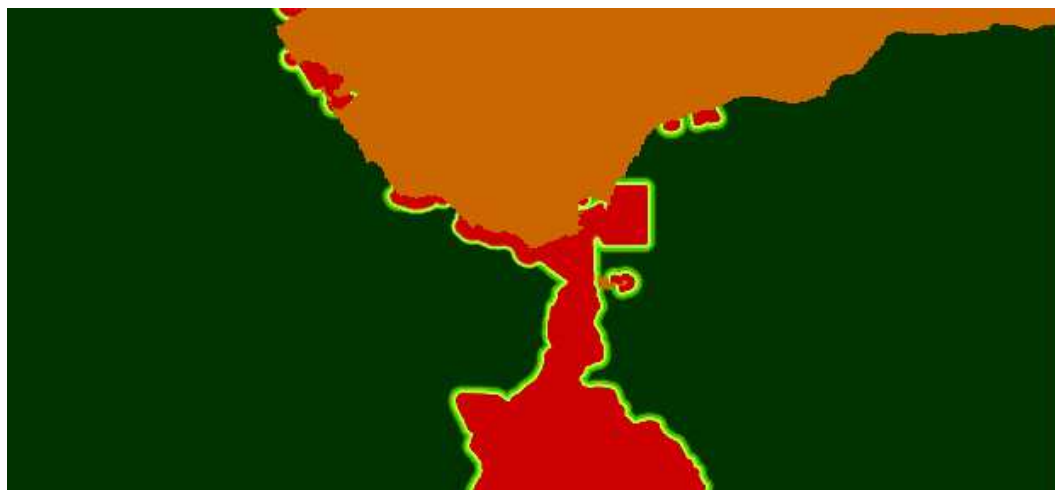
```
r.patch input=R_5465,R_5468 output=R_5469
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=p_humanidad
output=R_5470
v.to.rast input=R_5470 layer=1 type=point,line,area output=R_5471
use=val value=0.0 rows=4096
r.grow.distance input=R_5471 distance=R_5472 metric=euclidean
r.mapcalc "R_5473 = R_5472 * 100"
r.mapcalc "R_5474 = if(isnull( R_5473 ) , null(), if((R_5473 <= 1 ||
R_5473 >= 99999), 0.0, if((R_5473 > 1 && R_5473 < 4), eval((R_5473 -
1)/(4 -1 )), if((R_5473 > 99999 && R_5473 < 99999 ), eval((99999 -
R_5473)/(99999 - 99999)), 1.0)))"
r.patch input=R_5471,R_5474 output=R_5475
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=dipeuro output=R_5476
v.to.rast input=R_5476 layer=1 type=point,line,area output=R_5477
use=val value=0.0 rows=4096
r.grow.distance input=R_5477 distance=R_5478 metric=euclidean
r.mapcalc "R_5479 = R_5478 * 100"
r.mapcalc "R_5480 = if(isnull( R_5479 ) , null(), if((R_5479 <= 1 ||
R_5479 >= 99999), 0.0, if((R_5479 > 1 && R_5479 < 4), eval((R_5479 -
1)/(4 -1 )), if((R_5479 > 99999 && R_5479 < 99999 ), eval((99999 -
R_5479)/(99999 - 99999)), 1.0)))"
r.patch input=R_5477,R_5480 output=R_5481
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=micror output=R_5482
v.to.rast input=R_5482 layer=1 type=point,line,area output=R_5483
use=val value=0.0 rows=4096
r.grow.distance input=R_5483 distance=R_5484 metric=euclidean
r.mapcalc "R_5485 = R_5484 * 100"
r.mapcalc "R_5486 = if(isnull( R_5485 ) , null(), if((R_5485 <= 1 ||
R_5485 >= 99999), 0.0, if((R_5485 > 1 && R_5485 < 4), eval((R_5485 -
1)/(4 -1 )), if((R_5485 > 99999 && R_5485 < 99999 ), eval((99999 -
R_5485)/(99999 - 99999)), 1.0)))"
r.patch input=R_5483,R_5486 output=R_5487
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=rr_ley_prop
output=R_5488
v.to.rast input=R_5488 layer=1 type=point,line,area output=R_5489
use=val value=0.0 rows=4096
r.grow.distance input=R_5489 distance=R_5490 metric=euclidean
r.mapcalc "R_5491 = R_5490 * 100"
r.mapcalc "R_5492 = if(isnull( R_5491 ) , null(), if((R_5491 <= 1 ||
R_5491 >= 99999), 0.0, if((R_5491 > 1 && R_5491 < 4), eval((R_5491 -
1)/(4 -1 )), if((R_5491 > 99999 && R_5491 < 99999 ), eval((99999 -
R_5491)/(99999 - 99999)), 1.0)))"
r.patch input=R_5489,R_5492 output=R_5493
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=bbppro output=R_5494
v.to.rast input=R_5494 layer=1 type=point,line,area output=R_5495
use=val value=0.0 rows=4096
r.grow.distance input=R_5495 distance=R_5496 metric=euclidean
r.mapcalc "R_5497 = R_5496 * 100"
r.mapcalc "R_5498 = if(isnull( R_5497 ) , null(), if((R_5497 <= 1 ||
R_5497 >= 99999), 0.0, if((R_5497 > 1 && R_5497 < 4), eval((R_5497 -
```

```

1)/(4 -1 )), if((R_5497 > 99999 && R_5497 < 99999 ), eval((99999 -
R_5497)/(99999 - 99999)), 1.0))))"
r.patch input=R_5495,R_5498 output=R_5499
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mabs output=R_5500
v.to.rast input=R_5500 layer=1 type=point,line,area output=R_5501
use=val value=0.0 rows=4096
r.grow.distance input=R_5501 distance=R_5502 metric=euclidean
r.mapcalc "R_5503 = R_5502 * 100"
r.mapcalc "R_5504 = if(isnull( R_5503 ), null(), if((R_5503 <= 1 ||
R_5503 >= 99999), 0.0, if((R_5503 > 1 && R_5503 < 4), eval((R_5503 -
1)/(4 -1 )), if((R_5503 > 99999 && R_5503 < 99999 ), eval((99999 -
R_5503)/(99999 - 99999)), 1.0))))"
r.patch input=R_5501,R_5504 output=R_5505
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ospar_amps
output=R_5506
v.to.rast input=R_5506 layer=1 type=point,line,area output=R_5507
use=val value=0.0 rows=4096
r.grow.distance input=R_5507 distance=R_5508 metric=euclidean
r.mapcalc "R_5509 = R_5508 * 100"
r.mapcalc "R_5510 = if(isnull( R_5509 ), null(), if((R_5509 <= 1 ||
R_5509 >= 99999), 0.0, if((R_5509 > 1 && R_5509 < 4), eval((R_5509 -
1)/(4 -1 )), if((R_5509 > 99999 && R_5509 < 99999 ), eval((99999 -
R_5509)/(99999 - 99999)), 1.0))))"
r.patch input=R_5507,R_5510 output=R_5511
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=pparnnat output=R_5512
v.to.rast input=R_5512 layer=1 type=point,line,area output=R_5513
use=val value=0.0 rows=4096
r.grow.distance input=R_5513 distance=R_5514 metric=euclidean
r.mapcalc "R_5515 = R_5514 * 100"
r.mapcalc "R_5516 = if(isnull( R_5515 ), null(), if((R_5515 <= 1 ||
R_5515 >= 99999), 0.0, if((R_5515 > 1 && R_5515 < 4), eval((R_5515 -
1)/(4 -1 )), if((R_5515 > 99999 && R_5515 < 99999 ), eval((99999 -
R_5515)/(99999 - 99999)), 1.0))))"
r.patch input=R_5513,R_5516 output=R_5517
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepvn output=R_5518
v.to.rast input=R_5518 layer=1 type=point,line,area output=R_5519
use=val value=0.0 rows=4096
r.grow.distance input=R_5519 distance=R_5520 metric=euclidean
r.mapcalc "R_5521 = R_5520 * 100"
r.mapcalc "R_5522 = if(isnull( R_5521 ), null(), if((R_5521 <= 1 ||
R_5521 >= 99999), 0.0, if((R_5521 > 1 && R_5521 < 4), eval((R_5521 -
1)/(4 -1 )), if((R_5521 > 99999 && R_5521 < 99999 ), eval((99999 -
R_5521)/(99999 - 99999)), 1.0))))"
r.patch input=R_5519,R_5522 output=R_5523
r.mapcalc "R_5524 = R_5427 * R_5433 * R_5439 * R_5445 * R_5451 * R_5457
* R_5463 * R_5469 * R_5475 * R_5481 * R_5487 * R_5493 * R_5499 * R_5505
* R_5511 * R_5517 * R_5523 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096

```

```
r.mapcalc "R_5525 = if(isnull(ccaa),R_5524, -1.0)"
g.region n=36.934680786467 s=35.067005005217 e=-3.852094453125 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_5427
output="C:\Users\GSI\OceanLider\Resultados\87\RI_R_5427.tif"
r.out.gdal input=R_5433
output="C:\Users\GSI\OceanLider\Resultados\87\RI_R_5433.tif"
r.out.gdal input=R_5439
output="C:\Users\GSI\OceanLider\Resultados\87\RI_R_5439.tif"
r.out.gdal input=R_5445
output="C:\Users\GSI\OceanLider\Resultados\87\RI_R_5445.tif"
r.out.gdal input=R_5451
output="C:\Users\GSI\OceanLider\Resultados\87\RI_R_5451.tif"
r.out.gdal input=R_5457
output="C:\Users\GSI\OceanLider\Resultados\87\RI_R_5457.tif"
r.out.gdal input=R_5463
output="C:\Users\GSI\OceanLider\Resultados\87\RI_R_5463.tif"
r.out.gdal input=R_5469
output="C:\Users\GSI\OceanLider\Resultados\87\RI_R_5469.tif"
r.out.gdal input=R_5475
output="C:\Users\GSI\OceanLider\Resultados\87\RI_R_5475.tif"
r.out.gdal input=R_5481
output="C:\Users\GSI\OceanLider\Resultados\87\RI_R_5481.tif"
r.out.gdal input=R_5487
output="C:\Users\GSI\OceanLider\Resultados\87\RI_R_5487.tif"
r.out.gdal input=R_5493
output="C:\Users\GSI\OceanLider\Resultados\87\RI_R_5493.tif"
r.out.gdal input=R_5499
output="C:\Users\GSI\OceanLider\Resultados\87\RI_R_5499.tif"
r.out.gdal input=R_5505
output="C:\Users\GSI\OceanLider\Resultados\87\RI_R_5505.tif"
r.out.gdal input=R_5511
output="C:\Users\GSI\OceanLider\Resultados\87\RI_R_5511.tif"
r.out.gdal input=R_5517
output="C:\Users\GSI\OceanLider\Resultados\87\RI_R_5517.tif"
r.out.gdal input=R_5523
output="C:\Users\GSI\OceanLider\Resultados\87\RI_R_5523.tif"
r.out.gdal input=R_5525
output="C:\Users\GSI\OceanLider\Resultados\87\RF_R_5525.tif"
```



El método de razonamiento que se ha escogido para la combinación de los subcriterios: **pesimista (mínimo de los valores).**

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ppnat output=R_5565
v.to.rast input=R_5565 layer=1 type=point,line,area output=R_5566
use=val value=0.0 rows=4096
r.grow.distance input=R_5566 distance=R_5567 metric=euclidean
r.mapcalc "R_5568 = R_5567 * 100"
r.mapcalc "R_5569 = if(isnull( R_5568 ), null(), if((R_5568 <= 1 ||
R_5568 >= 99999), 0.0, if((R_5568 > 1 && R_5568 < 4), eval((R_5568 -
1)/(4 -1 )), if((R_5568 > 99999 && R_5568 < 99999 ), eval((99999 -
R_5568 )/(99999 - 99999)), 1.0))))"
```



```
r.patch input=R_5566,R_5569 output=R_5570
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepims output=R_5571
v.to.rast input=R_5571 layer=1 type=point,line,area output=R_5572
use=val value=0.0 rows=4096
r.grow.distance input=R_5572 distance=R_5573 metric=euclidean
r.mapcalc "R_5574 = R_5573 * 100"
r.mapcalc "R_5575 = if(isnull( R_5574 ) , null(), if((R_5574 <= 1 ||
R_5574 >= 99999), 0.0, if((R_5574 > 1 && R_5574 < 4), eval((R_5574 -
1)/(4 -1 )), if((R_5574 > 99999 && R_5574 < 99999 ), eval((99999 -
R_5574)/(99999 - 99999)), 1.0))))"
r.patch input=R_5572,R_5575 output=R_5576
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=lics output=R_5577
v.to.rast input=R_5577 layer=1 type=point,line,area output=R_5578
use=val value=0.0 rows=4096
r.grow.distance input=R_5578 distance=R_5579 metric=euclidean
r.mapcalc "R_5580 = R_5579 * 100"
r.mapcalc "R_5581 = if(isnull( R_5580 ) , null(), if((R_5580 <= 1 ||
R_5580 >= 99999), 0.0, if((R_5580 > 1 && R_5580 < 4), eval((R_5580 -
1)/(4 -1 )), if((R_5580 > 99999 && R_5580 < 99999 ), eval((99999 -
R_5580)/(99999 - 99999)), 1.0))))"
r.patch input=R_5578,R_5581 output=R_5582
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mmnat output=R_5583
v.to.rast input=R_5583 layer=1 type=point,line,area output=R_5584
use=val value=0.0 rows=4096
r.grow.distance input=R_5584 distance=R_5585 metric=euclidean
r.mapcalc "R_5586 = R_5585 * 100"
r.mapcalc "R_5587 = if(isnull( R_5586 ) , null(), if((R_5586 <= 1 ||
R_5586 >= 99999), 0.0, if((R_5586 > 1 && R_5586 < 4), eval((R_5586 -
1)/(4 -1 )), if((R_5586 > 99999 && R_5586 < 99999 ), eval((99999 -
R_5586)/(99999 - 99999)), 1.0))))"
r.patch input=R_5584,R_5587 output=R_5588
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepas output=R_5589
v.to.rast input=R_5589 layer=1 type=point,line,area output=R_5590
use=val value=0.0 rows=4096
r.grow.distance input=R_5590 distance=R_5591 metric=euclidean
r.mapcalc "R_5592 = R_5591 * 100"
r.mapcalc "R_5593 = if(isnull( R_5592 ) , null(), if((R_5592 <= 1 ||
R_5592 >= 99999), 0.0, if((R_5592 > 1 && R_5592 < 4), eval((R_5592 -
1)/(4 -1 )), if((R_5592 > 99999 && R_5592 < 99999 ), eval((99999 -
R_5592)/(99999 - 99999)), 1.0))))"
r.patch input=R_5590,R_5593 output=R_5594
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ppnn output=R_5595
v.to.rast input=R_5595 layer=1 type=point,line,area output=R_5596
use=val value=0.0 rows=4096
r.grow.distance input=R_5596 distance=R_5597 metric=euclidean
r.mapcalc "R_5598 = R_5597 * 100"
r.mapcalc "R_5599 = if(isnull( R_5598 ) , null(), if((R_5598 <= 1 ||
R_5598 >= 99999), 0.0, if((R_5598 > 1 && R_5598 < 4), eval((R_5598 -
1)/(4 -1 )), if((R_5598 > 99999 && R_5598 < 99999 ), eval((99999 -
R_5598)/(99999 - 99999)), 1.0))))"
```

```

r.patch input=R_5596,R_5599 output=R_5600
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=rrrn output=R_5601
v.to.rast input=R_5601 layer=1 type=point,line,area output=R_5602
use=val value=0.0 rows=4096
r.grow.distance input=R_5602 distance=R_5603 metric=euclidean
r.mapcalc "R_5604 = R_5603 * 100"
r.mapcalc "R_5605 = if(isnull( R_5604 ) , null(), if((R_5604 <= 1 ||
R_5604 >= 99999), 0.0, if((R_5604 > 1 && R_5604 < 4), eval((R_5604 -
1)/(4 -1 )), if((R_5604 > 99999 && R_5604 < 99999 ), eval((99999 -
R_5604)/(99999 - 99999)), 1.0))))"
r.patch input=R_5602,R_5605 output=R_5606
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=geoparques
output=R_5607
v.to.rast input=R_5607 layer=1 type=point,line,area output=R_5608
use=val value=0.0 rows=4096
r.grow.distance input=R_5608 distance=R_5609 metric=euclidean
r.mapcalc "R_5610 = R_5609 * 100"
r.mapcalc "R_5611 = if(isnull( R_5610 ) , null(), if((R_5610 <= 1 ||
R_5610 >= 99999), 0.0, if((R_5610 > 1 && R_5610 < 4), eval((R_5610 -
1)/(4 -1 )), if((R_5610 > 99999 && R_5610 < 99999 ), eval((99999 -
R_5610)/(99999 - 99999)), 1.0))))"
r.patch input=R_5608,R_5611 output=R_5612
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=p_humanidad
output=R_5613
v.to.rast input=R_5613 layer=1 type=point,line,area output=R_5614
use=val value=0.0 rows=4096
r.grow.distance input=R_5614 distance=R_5615 metric=euclidean
r.mapcalc "R_5616 = R_5615 * 100"
r.mapcalc "R_5617 = if(isnull( R_5616 ) , null(), if((R_5616 <= 1 ||
R_5616 >= 99999), 0.0, if((R_5616 > 1 && R_5616 < 4), eval((R_5616 -
1)/(4 -1 )), if((R_5616 > 99999 && R_5616 < 99999 ), eval((99999 -
R_5616)/(99999 - 99999)), 1.0))))"
r.patch input=R_5614,R_5617 output=R_5618
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=dipeuro output=R_5619
v.to.rast input=R_5619 layer=1 type=point,line,area output=R_5620
use=val value=0.0 rows=4096
r.grow.distance input=R_5620 distance=R_5621 metric=euclidean
r.mapcalc "R_5622 = R_5621 * 100"
r.mapcalc "R_5623 = if(isnull( R_5622 ) , null(), if((R_5622 <= 1 ||
R_5622 >= 99999), 0.0, if((R_5622 > 1 && R_5622 < 4), eval((R_5622 -
1)/(4 -1 )), if((R_5622 > 99999 && R_5622 < 99999 ), eval((99999 -
R_5622)/(99999 - 99999)), 1.0))))"
r.patch input=R_5620,R_5623 output=R_5624
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=micror output=R_5625
v.to.rast input=R_5625 layer=1 type=point,line,area output=R_5626
use=val value=0.0 rows=4096
r.grow.distance input=R_5626 distance=R_5627 metric=euclidean
r.mapcalc "R_5628 = R_5627 * 100"
r.mapcalc "R_5629 = if(isnull( R_5628 ) , null(), if((R_5628 <= 1 ||
R_5628 >= 99999), 0.0, if((R_5628 > 1 && R_5628 < 4), eval((R_5628 -

```

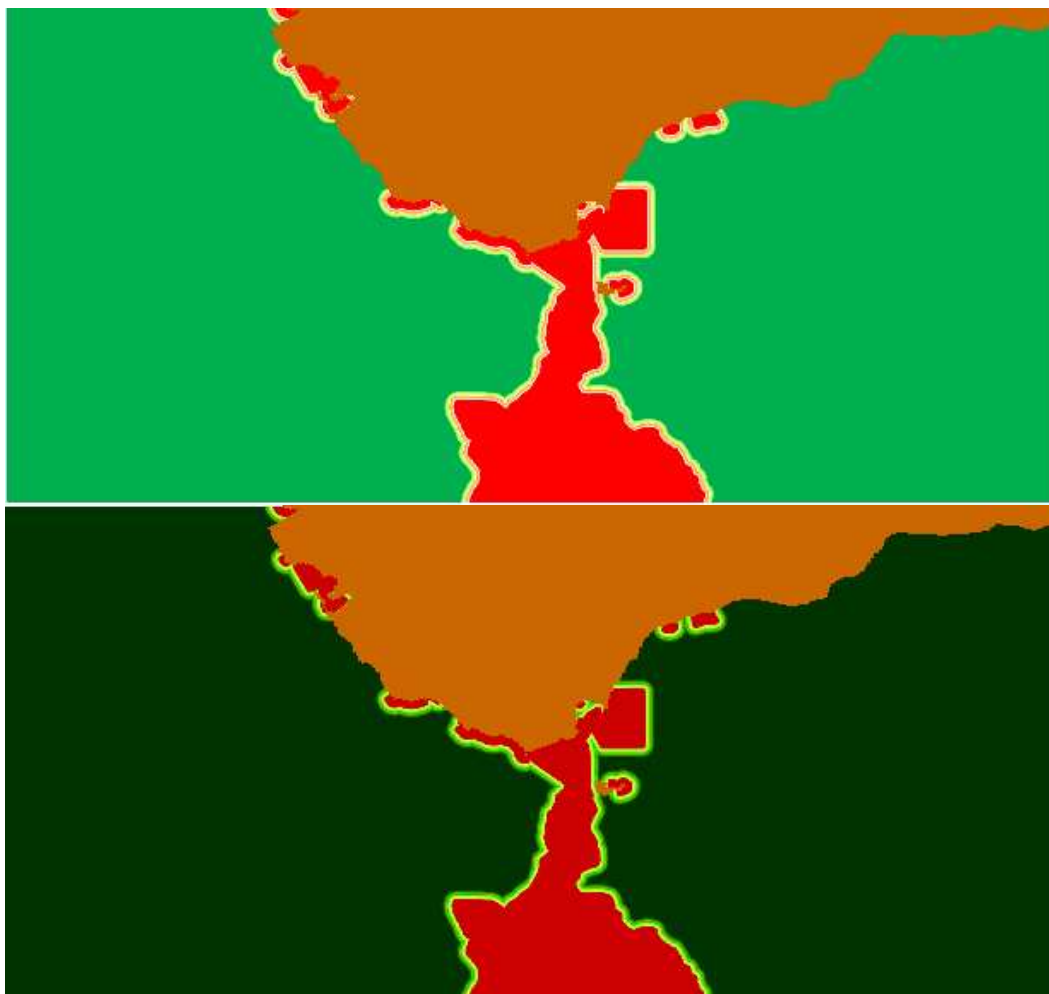
```
1)/(4 -1 )), if((R_5628 > 99999 && R_5628 < 99999 ), eval((99999 -
R_5628)/(99999 - 99999)), 1.0))))"
r.patch input=R_5626,R_5629 output=R_5630
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=rr_ley_prop
output=R_5631
v.to.rast input=R_5631 layer=1 type=point,line,area output=R_5632
use=val value=0.0 rows=4096
r.grow.distance input=R_5632 distance=R_5633 metric=euclidean
r.mapcalc "R_5634 = R_5633 * 100"
r.mapcalc "R_5635 = if(isnull( R_5634 ), null(), if((R_5634 <= 1 ||
R_5634 >= 99999), 0.0, if((R_5634 > 1 && R_5634 < 4), eval((R_5634 -
1)/(4 -1 )), if((R_5634 > 99999 && R_5634 < 99999 ), eval((99999 -
R_5634)/(99999 - 99999)), 1.0))))"
r.patch input=R_5632,R_5635 output=R_5636
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=bbppro output=R_5637
v.to.rast input=R_5637 layer=1 type=point,line,area output=R_5638
use=val value=0.0 rows=4096
r.grow.distance input=R_5638 distance=R_5639 metric=euclidean
r.mapcalc "R_5640 = R_5639 * 100"
r.mapcalc "R_5641 = if(isnull( R_5640 ), null(), if((R_5640 <= 1 ||
R_5640 >= 99999), 0.0, if((R_5640 > 1 && R_5640 < 4), eval((R_5640 -
1)/(4 -1 )), if((R_5640 > 99999 && R_5640 < 99999 ), eval((99999 -
R_5640)/(99999 - 99999)), 1.0))))"
r.patch input=R_5638,R_5641 output=R_5642
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mabs output=R_5643
v.to.rast input=R_5643 layer=1 type=point,line,area output=R_5644
use=val value=0.0 rows=4096
r.grow.distance input=R_5644 distance=R_5645 metric=euclidean
r.mapcalc "R_5646 = R_5645 * 100"
r.mapcalc "R_5647 = if(isnull( R_5646 ), null(), if((R_5646 <= 1 ||
R_5646 >= 99999), 0.0, if((R_5646 > 1 && R_5646 < 4), eval((R_5646 -
1)/(4 -1 )), if((R_5646 > 99999 && R_5646 < 99999 ), eval((99999 -
R_5646)/(99999 - 99999)), 1.0))))"
r.patch input=R_5644,R_5647 output=R_5648
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ospar_amps
output=R_5649
v.to.rast input=R_5649 layer=1 type=point,line,area output=R_5650
use=val value=0.0 rows=4096
r.grow.distance input=R_5650 distance=R_5651 metric=euclidean
r.mapcalc "R_5652 = R_5651 * 100"
r.mapcalc "R_5653 = if(isnull( R_5652 ), null(), if((R_5652 <= 1 ||
R_5652 >= 99999), 0.0, if((R_5652 > 1 && R_5652 < 4), eval((R_5652 -
1)/(4 -1 )), if((R_5652 > 99999 && R_5652 < 99999 ), eval((99999 -
R_5652)/(99999 - 99999)), 1.0))))"
r.patch input=R_5650,R_5653 output=R_5654
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=pparnnat output=R_5655
v.to.rast input=R_5655 layer=1 type=point,line,area output=R_5656
use=val value=0.0 rows=4096
r.grow.distance input=R_5656 distance=R_5657 metric=euclidean
r.mapcalc "R_5658 = R_5657 * 100"
```

```

r.mapcalc "R_5659 = if(isnull( R_5658 ), null(), if((R_5658 <= 1 ||
R_5658 >= 99999), 0.0, if((R_5658 > 1 && R_5658 < 4), eval((R_5658 -
1)/(4 -1 )), if((R_5658 > 99999 && R_5658 < 99999 ), eval((99999 -
R_5658)/(99999 - 99999)), 1.0))))"
r.patch input=R_5656,R_5659 output=R_5660
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepvn output=R_5661
v.to.rast input=R_5661 layer=1 type=point,line,area output=R_5662
use=val value=0.0 rows=4096
r.grow.distance input=R_5662 distance=R_5663 metric=euclidean
r.mapcalc "R_5664 = R_5663 * 100"
r.mapcalc "R_5665 = if(isnull( R_5664 ), null(), if((R_5664 <= 1 ||
R_5664 >= 99999), 0.0, if((R_5664 > 1 && R_5664 < 4), eval((R_5664 -
1)/(4 -1 )), if((R_5664 > 99999 && R_5664 < 99999 ), eval((99999 -
R_5664)/(99999 - 99999)), 1.0))))"
r.patch input=R_5662,R_5665 output=R_5666
r.mapcalc "R_5667 = if(isnull( R_5570 ) || isnull( R_5576 ), null(),
if(R_5570 <= R_5576 , R_5570 , R_5576 ))"
r.mapcalc "R_5668 = if(isnull( R_5667 ) || isnull( R_5576 ), null(),
if(R_5667 <= R_5576 , R_5667 , R_5576 ))"
r.mapcalc "R_5669 = if(isnull( R_5668 ) || isnull( R_5582 ), null(),
if(R_5668 <= R_5582 , R_5668 , R_5582 ))"
r.mapcalc "R_5670 = if(isnull( R_5669 ) || isnull( R_5588 ), null(),
if(R_5669 <= R_5588 , R_5669 , R_5588 ))"
r.mapcalc "R_5671 = if(isnull( R_5670 ) || isnull( R_5594 ), null(),
if(R_5670 <= R_5594 , R_5670 , R_5594 ))"
r.mapcalc "R_5672 = if(isnull( R_5671 ) || isnull( R_5600 ), null(),
if(R_5671 <= R_5600 , R_5671 , R_5600 ))"
r.mapcalc "R_5673 = if(isnull( R_5672 ) || isnull( R_5606 ), null(),
if(R_5672 <= R_5606 , R_5672 , R_5606 ))"
r.mapcalc "R_5674 = if(isnull( R_5673 ) || isnull( R_5612 ), null(),
if(R_5673 <= R_5612 , R_5673 , R_5612 ))"
r.mapcalc "R_5675 = if(isnull( R_5674 ) || isnull( R_5618 ), null(),
if(R_5674 <= R_5618 , R_5674 , R_5618 ))"
r.mapcalc "R_5676 = if(isnull( R_5675 ) || isnull( R_5624 ), null(),
if(R_5675 <= R_5624 , R_5675 , R_5624 ))"
r.mapcalc "R_5677 = if(isnull( R_5676 ) || isnull( R_5630 ), null(),
if(R_5676 <= R_5630 , R_5676 , R_5630 ))"
r.mapcalc "R_5678 = if(isnull( R_5677 ) || isnull( R_5636 ), null(),
if(R_5677 <= R_5636 , R_5677 , R_5636 ))"
r.mapcalc "R_5679 = if(isnull( R_5678 ) || isnull( R_5642 ), null(),
if(R_5678 <= R_5642 , R_5678 , R_5642 ))"
r.mapcalc "R_5680 = if(isnull( R_5679 ) || isnull( R_5648 ), null(),
if(R_5679 <= R_5648 , R_5679 , R_5648 ))"
r.mapcalc "R_5681 = if(isnull( R_5680 ) || isnull( R_5654 ), null(),
if(R_5680 <= R_5654 , R_5680 , R_5654 ))"
r.mapcalc "R_5682 = if(isnull( R_5681 ) || isnull( R_5660 ), null(),
if(R_5681 <= R_5660 , R_5681 , R_5660 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_5683 = if(isnull(ccaa),R_5682, -1.0)"
g.region n=37.070911523437 s=35.159290429687 e=-3.874067109375 w=-
7.301801484375 nsres=300 ewres=300 rows=3224 cols=3453

```

```
r.out.gdal input=R_5570
output="C:\Users\GSI\OceanLider\Resultados\88\RI_R_5570.tif"
r.out.gdal input=R_5576
output="C:\Users\GSI\OceanLider\Resultados\88\RI_R_5576.tif"
r.out.gdal input=R_5582
output="C:\Users\GSI\OceanLider\Resultados\88\RI_R_5582.tif"
r.out.gdal input=R_5588
output="C:\Users\GSI\OceanLider\Resultados\88\RI_R_5588.tif"
r.out.gdal input=R_5594
output="C:\Users\GSI\OceanLider\Resultados\88\RI_R_5594.tif"
r.out.gdal input=R_5600
output="C:\Users\GSI\OceanLider\Resultados\88\RI_R_5600.tif"
r.out.gdal input=R_5606
output="C:\Users\GSI\OceanLider\Resultados\88\RI_R_5606.tif"
r.out.gdal input=R_5612
output="C:\Users\GSI\OceanLider\Resultados\88\RI_R_5612.tif"
r.out.gdal input=R_5618
output="C:\Users\GSI\OceanLider\Resultados\88\RI_R_5618.tif"
r.out.gdal input=R_5624
output="C:\Users\GSI\OceanLider\Resultados\88\RI_R_5624.tif"
r.out.gdal input=R_5630
output="C:\Users\GSI\OceanLider\Resultados\88\RI_R_5630.tif"
r.out.gdal input=R_5636
output="C:\Users\GSI\OceanLider\Resultados\88\RI_R_5636.tif"
r.out.gdal input=R_5642
output="C:\Users\GSI\OceanLider\Resultados\88\RI_R_5642.tif"
r.out.gdal input=R_5648
output="C:\Users\GSI\OceanLider\Resultados\88\RI_R_5648.tif"
r.out.gdal input=R_5654
output="C:\Users\GSI\OceanLider\Resultados\88\RI_R_5654.tif"
r.out.gdal input=R_5660
output="C:\Users\GSI\OceanLider\Resultados\88\RI_R_5660.tif"
r.out.gdal input=R_5666
output="C:\Users\GSI\OceanLider\Resultados\88\RI_R_5666.tif"
r.out.gdal input=R_5683
output="C:\Users\GSI\OceanLider\Resultados\88\RF_R_5683.tif"
```



El método de razonamiento que se ha escogido para la combinación de los subcriterios: **optimista (máximo de los valores)**.

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ppnat output=R_3488
v.to.rast input=R_3488 layer=1 type=point,line,area output=R_3489
use=val value=0.0 rows=4096
r.grow.distance input=R_3489 distance=R_3490 metric=euclidean
r.mapcalc "R_3491 = R_3490 * 100"
r.mapcalc "R_3492 = if(isnull( R_3491 ), null(), if((R_3491 <= 1 ||
R_3491 >= 99999), 0.0, if((R_3491 > 1 && R_3491 < 4), eval((R_3491 -
1)/(4 -1 )), if((R_3491 > 99999 && R_3491 < 99999 ), eval((99999 -
R_3491 )/(99999 - 99999)), 1.0))))"
r.patch input=R_3489,R_3492 output=R_3493
```

```
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepims output=R_3494
v.to.rast input=R_3494 layer=1 type=point,line,area output=R_3495
use=val value=0.0 rows=4096
r.grow.distance input=R_3495 distance=R_3496 metric=euclidean
r.mapcalc "R_3497 = R_3496 * 100"
r.mapcalc "R_3498 = if(isnull( R_3497 ) , null(), if((R_3497 <= 1 ||
R_3497 >= 99999), 0.0, if((R_3497 > 1 && R_3497 < 4), eval((R_3497 -
1)/(4 -1 )), if((R_3497 > 99999 && R_3497 < 99999 ), eval((99999 -
R_3497)/(99999 - 99999)), 1.0))))"
r.patch input=R_3495,R_3498 output=R_3499
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=lics output=R_3500
v.to.rast input=R_3500 layer=1 type=point,line,area output=R_3501
use=val value=0.0 rows=4096
r.grow.distance input=R_3501 distance=R_3502 metric=euclidean
r.mapcalc "R_3503 = R_3502 * 100"
r.mapcalc "R_3504 = if(isnull( R_3503 ) , null(), if((R_3503 <= 1 ||
R_3503 >= 99999), 0.0, if((R_3503 > 1 && R_3503 < 4), eval((R_3503 -
1)/(4 -1 )), if((R_3503 > 99999 && R_3503 < 99999 ), eval((99999 -
R_3503)/(99999 - 99999)), 1.0))))"
r.patch input=R_3501,R_3504 output=R_3505
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mmnat output=R_3506
v.to.rast input=R_3506 layer=1 type=point,line,area output=R_3507
use=val value=0.0 rows=4096
r.grow.distance input=R_3507 distance=R_3508 metric=euclidean
r.mapcalc "R_3509 = R_3508 * 100"
r.mapcalc "R_3510 = if(isnull( R_3509 ) , null(), if((R_3509 <= 1 ||
R_3509 >= 99999), 0.0, if((R_3509 > 1 && R_3509 < 4), eval((R_3509 -
1)/(4 -1 )), if((R_3509 > 99999 && R_3509 < 99999 ), eval((99999 -
R_3509)/(99999 - 99999)), 1.0))))"
r.patch input=R_3507,R_3510 output=R_3511
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepas output=R_3512
v.to.rast input=R_3512 layer=1 type=point,line,area output=R_3513
use=val value=0.0 rows=4096
r.grow.distance input=R_3513 distance=R_3514 metric=euclidean
r.mapcalc "R_3515 = R_3514 * 100"
r.mapcalc "R_3516 = if(isnull( R_3515 ) , null(), if((R_3515 <= 1 ||
R_3515 >= 99999), 0.0, if((R_3515 > 1 && R_3515 < 4), eval((R_3515 -
1)/(4 -1 )), if((R_3515 > 99999 && R_3515 < 99999 ), eval((99999 -
R_3515)/(99999 - 99999)), 1.0))))"
r.patch input=R_3513,R_3516 output=R_3517
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ppnn output=R_3518
v.to.rast input=R_3518 layer=1 type=point,line,area output=R_3519
use=val value=0.0 rows=4096
r.grow.distance input=R_3519 distance=R_3520 metric=euclidean
r.mapcalc "R_3521 = R_3520 * 100"
r.mapcalc "R_3522 = if(isnull( R_3521 ) , null(), if((R_3521 <= 1 ||
R_3521 >= 99999), 0.0, if((R_3521 > 1 && R_3521 < 4), eval((R_3521 -
1)/(4 -1 )), if((R_3521 > 99999 && R_3521 < 99999 ), eval((99999 -
R_3521)/(99999 - 99999)), 1.0))))"
r.patch input=R_3519,R_3522 output=R_3523
```

```

v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=rrnn output=R_3524
v.to.rast input=R_3524 layer=1 type=point,line,area output=R_3525
use=val value=0.0 rows=4096
r.grow.distance input=R_3525 distance=R_3526 metric=euclidean
r.mapcalc "R_3527 = R_3526 * 100"
r.mapcalc "R_3528 = if(isnull( R_3527 ) , null(), if((R_3527 <= 1 ||
R_3527 >= 99999), 0.0, if((R_3527 > 1 && R_3527 < 4), eval((R_3527 -
1)/(4 -1 )), if((R_3527 > 99999 && R_3527 < 99999 ), eval((99999 -
R_3527)/(99999 - 99999)), 1.0))))"
r.patch input=R_3525,R_3528 output=R_3529
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=geoparques
output=R_3530
v.to.rast input=R_3530 layer=1 type=point,line,area output=R_3531
use=val value=0.0 rows=4096
r.grow.distance input=R_3531 distance=R_3532 metric=euclidean
r.mapcalc "R_3533 = R_3532 * 100"
r.mapcalc "R_3534 = if(isnull( R_3533 ) , null(), if((R_3533 <= 1 ||
R_3533 >= 99999), 0.0, if((R_3533 > 1 && R_3533 < 4), eval((R_3533 -
1)/(4 -1 )), if((R_3533 > 99999 && R_3533 < 99999 ), eval((99999 -
R_3533)/(99999 - 99999)), 1.0))))"
r.patch input=R_3531,R_3534 output=R_3535
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=p_humanidad
output=R_3536
v.to.rast input=R_3536 layer=1 type=point,line,area output=R_3537
use=val value=0.0 rows=4096
r.grow.distance input=R_3537 distance=R_3538 metric=euclidean
r.mapcalc "R_3539 = R_3538 * 100"
r.mapcalc "R_3540 = if(isnull( R_3539 ) , null(), if((R_3539 <= 1 ||
R_3539 >= 99999), 0.0, if((R_3539 > 1 && R_3539 < 4), eval((R_3539 -
1)/(4 -1 )), if((R_3539 > 99999 && R_3539 < 99999 ), eval((99999 -
R_3539)/(99999 - 99999)), 1.0))))"
r.patch input=R_3537,R_3540 output=R_3541
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=dipeuro output=R_3542
v.to.rast input=R_3542 layer=1 type=point,line,area output=R_3543
use=val value=0.0 rows=4096
r.grow.distance input=R_3543 distance=R_3544 metric=euclidean
r.mapcalc "R_3545 = R_3544 * 100"
r.mapcalc "R_3546 = if(isnull( R_3545 ) , null(), if((R_3545 <= 1 ||
R_3545 >= 99999), 0.0, if((R_3545 > 1 && R_3545 < 4), eval((R_3545 -
1)/(4 -1 )), if((R_3545 > 99999 && R_3545 < 99999 ), eval((99999 -
R_3545)/(99999 - 99999)), 1.0))))"
r.patch input=R_3543,R_3546 output=R_3547
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=micror output=R_3548
v.to.rast input=R_3548 layer=1 type=point,line,area output=R_3549
use=val value=0.0 rows=4096
r.grow.distance input=R_3549 distance=R_3550 metric=euclidean
r.mapcalc "R_3551 = R_3550 * 100"
r.mapcalc "R_3552 = if(isnull( R_3551 ) , null(), if((R_3551 <= 1 ||
R_3551 >= 99999), 0.0, if((R_3551 > 1 && R_3551 < 4), eval((R_3551 -

```



```
1)/(4 -1 )), if((R_3551 > 99999 && R_3551 < 99999 ), eval((99999 -
R_3551)/(99999 - 99999)), 1.0))))"
r.patch input=R_3549,R_3552 output=R_3553
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=rr_ley_prop
output=R_3554
v.to.rast input=R_3554 layer=1 type=point,line,area output=R_3555
use=val value=0.0 rows=4096
r.grow.distance input=R_3555 distance=R_3556 metric=euclidean
r.mapcalc "R_3557 = R_3556 * 100"
r.mapcalc "R_3558 = if(isnull( R_3557 ), null(), if((R_3557 <= 1 ||
R_3557 >= 99999), 0.0, if((R_3557 > 1 && R_3557 < 4), eval((R_3557 -
1)/(4 -1 )), if((R_3557 > 99999 && R_3557 < 99999 ), eval((99999 -
R_3557)/(99999 - 99999)), 1.0))))"
r.patch input=R_3555,R_3558 output=R_3559
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=bbppro output=R_3560
v.to.rast input=R_3560 layer=1 type=point,line,area output=R_3561
use=val value=0.0 rows=4096
r.grow.distance input=R_3561 distance=R_3562 metric=euclidean
r.mapcalc "R_3563 = R_3562 * 100"
r.mapcalc "R_3564 = if(isnull( R_3563 ), null(), if((R_3563 <= 1 ||
R_3563 >= 99999), 0.0, if((R_3563 > 1 && R_3563 < 4), eval((R_3563 -
1)/(4 -1 )), if((R_3563 > 99999 && R_3563 < 99999 ), eval((99999 -
R_3563)/(99999 - 99999)), 1.0))))"
r.patch input=R_3561,R_3564 output=R_3565
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mabs output=R_3566
v.to.rast input=R_3566 layer=1 type=point,line,area output=R_3567
use=val value=0.0 rows=4096
r.grow.distance input=R_3567 distance=R_3568 metric=euclidean
r.mapcalc "R_3569 = R_3568 * 100"
r.mapcalc "R_3570 = if(isnull( R_3569 ), null(), if((R_3569 <= 1 ||
R_3569 >= 99999), 0.0, if((R_3569 > 1 && R_3569 < 4), eval((R_3569 -
1)/(4 -1 )), if((R_3569 > 99999 && R_3569 < 99999 ), eval((99999 -
R_3569)/(99999 - 99999)), 1.0))))"
r.patch input=R_3567,R_3570 output=R_3571
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ospar_amps
output=R_3572
v.to.rast input=R_3572 layer=1 type=point,line,area output=R_3573
use=val value=0.0 rows=4096
r.grow.distance input=R_3573 distance=R_3574 metric=euclidean
r.mapcalc "R_3575 = R_3574 * 100"
r.mapcalc "R_3576 = if(isnull( R_3575 ), null(), if((R_3575 <= 1 ||
R_3575 >= 99999), 0.0, if((R_3575 > 1 && R_3575 < 4), eval((R_3575 -
1)/(4 -1 )), if((R_3575 > 99999 && R_3575 < 99999 ), eval((99999 -
R_3575)/(99999 - 99999)), 1.0))))"
r.patch input=R_3573,R_3576 output=R_3577
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=pparnnat output=R_3578
v.to.rast input=R_3578 layer=1 type=point,line,area output=R_3579
use=val value=0.0 rows=4096
r.grow.distance input=R_3579 distance=R_3580 metric=euclidean
r.mapcalc "R_3581 = R_3580 * 100"
```

```

r.mapcalc "R_3582 = if(isnull( R_3581 ) , null(), if((R_3581 <= 1 ||
R_3581 >= 99999), 0.0, if((R_3581 > 1 && R_3581 < 4), eval((R_3581 -
1)/(4 -1 )), if((R_3581 > 99999 && R_3581 < 99999 ), eval((99999 -
R_3581)/(99999 - 99999)), 1.0))))"
r.patch input=R_3579,R_3582 output=R_3583
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepvn output=R_3584
v.to.rast input=R_3584 layer=1 type=point,line,area output=R_3585
use=val value=0.0 rows=4096
r.grow.distance input=R_3585 distance=R_3586 metric=euclidean
r.mapcalc "R_3587 = R_3586 * 100"
r.mapcalc "R_3588 = if(isnull( R_3587 ) , null(), if((R_3587 <= 1 ||
R_3587 >= 99999), 0.0, if((R_3587 > 1 && R_3587 < 4), eval((R_3587 -
1)/(4 -1 )), if((R_3587 > 99999 && R_3587 < 99999 ), eval((99999 -
R_3587)/(99999 - 99999)), 1.0))))"
r.patch input=R_3585,R_3588 output=R_3589
r.mapcalc "R_3590 = if(isnull( R_3493 ) || isnull( R_3499 ) , null(),
if(R_3493 >= R_3499 , R_3493 , R_3499 ))"
r.mapcalc "R_3591 = if(isnull( R_3590 ) || isnull( R_3499 ) , null(),
if(R_3590 >= R_3499 , R_3590 , R_3499 ))"
r.mapcalc "R_3592 = if(isnull( R_3591 ) || isnull( R_3505 ) , null(),
if(R_3591 >= R_3505 , R_3591 , R_3505 ))"
r.mapcalc "R_3593 = if(isnull( R_3592 ) || isnull( R_3511 ) , null(),
if(R_3592 >= R_3511 , R_3592 , R_3511 ))"
r.mapcalc "R_3594 = if(isnull( R_3593 ) || isnull( R_3517 ) , null(),
if(R_3593 >= R_3517 , R_3593 , R_3517 ))"
r.mapcalc "R_3595 = if(isnull( R_3594 ) || isnull( R_3523 ) , null(),
if(R_3594 >= R_3523 , R_3594 , R_3523 ))"
r.mapcalc "R_3596 = if(isnull( R_3595 ) || isnull( R_3529 ) , null(),
if(R_3595 >= R_3529 , R_3595 , R_3529 ))"
r.mapcalc "R_3597 = if(isnull( R_3596 ) || isnull( R_3535 ) , null(),
if(R_3596 >= R_3535 , R_3596 , R_3535 ))"
r.mapcalc "R_3598 = if(isnull( R_3597 ) || isnull( R_3541 ) , null(),
if(R_3597 >= R_3541 , R_3597 , R_3541 ))"
r.mapcalc "R_3599 = if(isnull( R_3598 ) || isnull( R_3547 ) , null(),
if(R_3598 >= R_3547 , R_3598 , R_3547 ))"
r.mapcalc "R_3600 = if(isnull( R_3599 ) || isnull( R_3553 ) , null(),
if(R_3599 >= R_3553 , R_3599 , R_3553 ))"
r.mapcalc "R_3601 = if(isnull( R_3600 ) || isnull( R_3559 ) , null(),
if(R_3600 >= R_3559 , R_3600 , R_3559 ))"
r.mapcalc "R_3602 = if(isnull( R_3601 ) || isnull( R_3565 ) , null(),
if(R_3601 >= R_3565 , R_3601 , R_3565 ))"
r.mapcalc "R_3603 = if(isnull( R_3602 ) || isnull( R_3571 ) , null(),
if(R_3602 >= R_3571 , R_3602 , R_3571 ))"
r.mapcalc "R_3604 = if(isnull( R_3603 ) || isnull( R_3577 ) , null(),
if(R_3603 >= R_3577 , R_3603 , R_3577 ))"
r.mapcalc "R_3605 = if(isnull( R_3604 ) || isnull( R_3583 ) , null(),
if(R_3604 >= R_3583 , R_3604 , R_3583 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_3606 = if(isnull(ccaa),R_3605, -1.0)"
g.region n=36.803943884111 s=35.134022009111 e=-3.830121796875 w=-
7.301801484375 nsres=300 ewres=300 rows=3224 cols=3453

```

```
r.out.gdal input=R_3493
output="C:\Users\GSI\OceanLider\Resultados\89\RI_R_3493.tif"
r.out.gdal input=R_3499
output="C:\Users\GSI\OceanLider\Resultados\89\RI_R_3499.tif"
r.out.gdal input=R_3505
output="C:\Users\GSI\OceanLider\Resultados\89\RI_R_3505.tif"
r.out.gdal input=R_3511
output="C:\Users\GSI\OceanLider\Resultados\89\RI_R_3511.tif"
r.out.gdal input=R_3517
output="C:\Users\GSI\OceanLider\Resultados\89\RI_R_3517.tif"
r.out.gdal input=R_3523
output="C:\Users\GSI\OceanLider\Resultados\89\RI_R_3523.tif"
r.out.gdal input=R_3529
output="C:\Users\GSI\OceanLider\Resultados\89\RI_R_3529.tif"
r.out.gdal input=R_3535
output="C:\Users\GSI\OceanLider\Resultados\89\RI_R_3535.tif"
r.out.gdal input=R_3541
output="C:\Users\GSI\OceanLider\Resultados\89\RI_R_3541.tif"
r.out.gdal input=R_3547
output="C:\Users\GSI\OceanLider\Resultados\89\RI_R_3547.tif"
r.out.gdal input=R_3553
output="C:\Users\GSI\OceanLider\Resultados\89\RI_R_3553.tif"
r.out.gdal input=R_3559
output="C:\Users\GSI\OceanLider\Resultados\89\RI_R_3559.tif"
r.out.gdal input=R_3565
output="C:\Users\GSI\OceanLider\Resultados\89\RI_R_3565.tif"
r.out.gdal input=R_3571
output="C:\Users\GSI\OceanLider\Resultados\89\RI_R_3571.tif"
r.out.gdal input=R_3577
output="C:\Users\GSI\OceanLider\Resultados\89\RI_R_3577.tif"
r.out.gdal input=R_3583
output="C:\Users\GSI\OceanLider\Resultados\89\RI_R_3583.tif"
r.out.gdal input=R_3589
output="C:\Users\GSI\OceanLider\Resultados\89\RI_R_3589.tif"
r.out.gdal input=R_3606
output="C:\Users\GSI\OceanLider\Resultados\89\RF_R_3606.tif"
```



2.3.2. Combinación 10


Combinación de tres subcriterios utilizando el método de razonamiento MLP para su combinación **del criterio conservación de la biodiversidad** (de exclusión parcial): *reservas de la biosfera, lugares de importancia comunitaria y zonas de especial protección para las aves.*

Se le deja la libertad al usuario para configurar el valor de idoneidad para las áreas que intervienen en cada subcriterio y se le muestra uno por defecto dependiendo del tipo de dispositivo. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.


El tipo de dispositivo al que esté ligado el escenario: **undimotriz- fuera costa (flotante).**

Los parámetros establecidos en cada subcriterio:


- ✓ *Subcriterio Reservas de la biosfera*

<input checked="" type="checkbox"/> Reservas de la biosfera 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.7

- ✓ *Subcriterio Lugares de Importancia Comunitaria*

<input checked="" type="checkbox"/> Lugares de Importancia Comunitaria 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.2

- ✓ *Subcriterio Zonas de Especial Protección para las Aves*

<input checked="" type="checkbox"/> Zonas de Especial Protección para las Aves 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.1

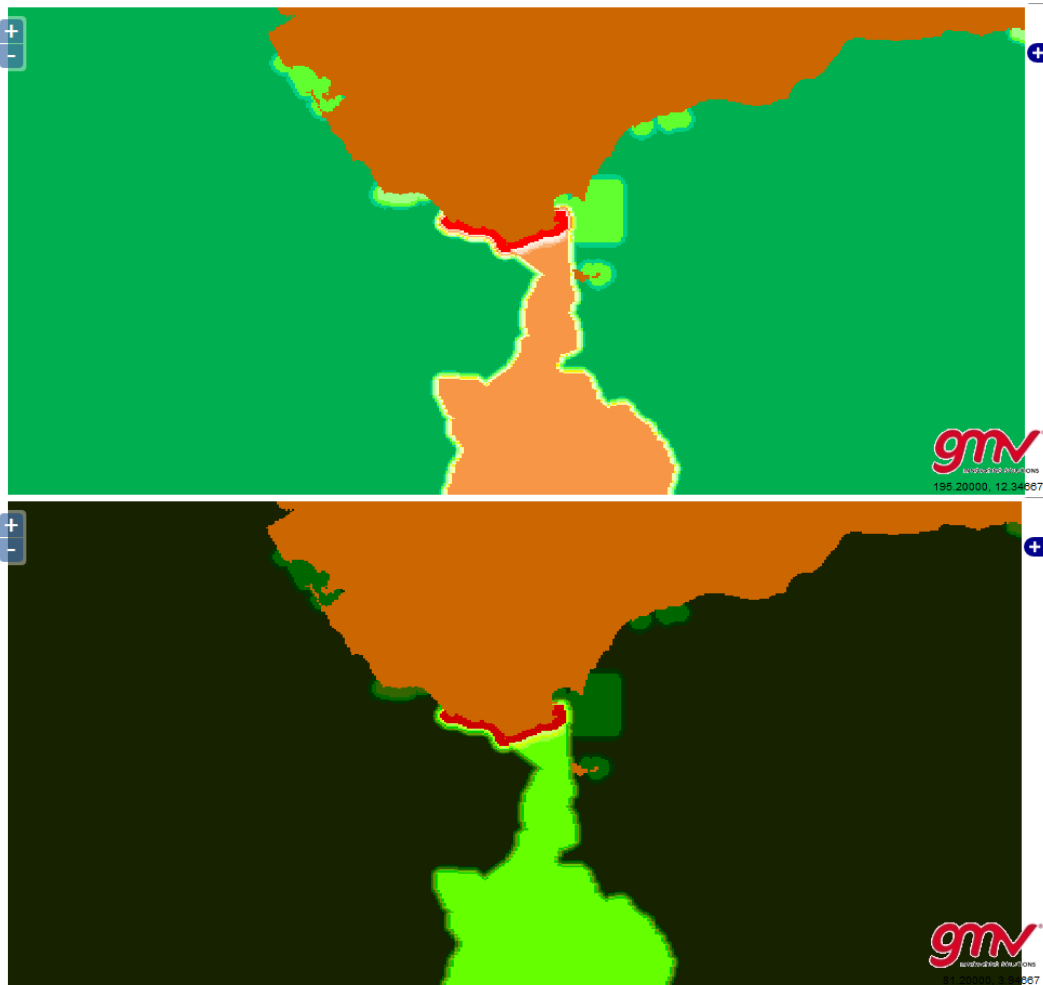
Los pesos de cada subcriterio que interviene: **0.7** a reservas de la biosfera, **0.2** a lugares de importancia comunitaria y **0.1** a zonas de especial protección para las aves.

```

SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepas output=R_4443
v.to.rast input=R_4443 layer=1 type=point,line,area output=R_4444
use=val value=0.0 rows=4096
r.grow.distance input=R_4444 distance=R_4445 metric=euclidean
r.mapcalc "R_4446 = R_4445 * 100"
r.mapcalc "R_4447 = if(isnull( R_4446 ) , null(), if((R_4446 <= 1 ||
R_4446 >= 99999), 0.0, if((R_4446 > 1 && R_4446 < 4), eval((R_4446 -
1)/(4 - 1)), if((R_4446 > 99999 && R_4446 < 99999), eval((99999 -
R_4446)/(99999 - 99999)), 1.0))))"
r.patch input=R_4444,R_4447 output=R_4448
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mabs output=R_4449
v.to.rast input=R_4449 layer=1 type=point,line,area output=R_4450
use=val value=0.0 rows=4096
r.grow.distance input=R_4450 distance=R_4451 metric=euclidean
r.mapcalc "R_4452 = R_4451 * 100"
r.mapcalc "R_4453 = if(isnull( R_4452 ) , null(), if((R_4452 <= 1 ||
R_4452 >= 99999), 0.0, if((R_4452 > 1 && R_4452 < 4), eval((R_4452 -
1)/(4 - 1)), if((R_4452 > 99999 && R_4452 < 99999), eval((99999 -
R_4452)/(99999 - 99999)), 1.0))))"
r.patch input=R_4450,R_4453 output=R_4454
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=lics output=R_4455
v.to.rast input=R_4455 layer=1 type=point,line,area output=R_4456
use=val value=0.0 rows=4096
r.grow.distance input=R_4456 distance=R_4457 metric=euclidean
r.mapcalc "R_4458 = R_4457 * 100"
r.mapcalc "R_4459 = if(isnull( R_4458 ) , null(), if((R_4458 <= 1 ||
R_4458 >= 99999), 0.0, if((R_4458 > 1 && R_4458 < 4), eval((R_4458 -
1)/(4 - 1)), if((R_4458 > 99999 && R_4458 < 99999), eval((99999 -
R_4458)/(99999 - 99999)), 1.0))))"
r.patch input=R_4456,R_4459 output=R_4460
r.mapcalc "R_4461 = 0.1 * R_4448 + 0.7 * R_4454 + 0.2 * R_4460 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_4462 = if(isnull(ccaa),R_4461, -1.0)"
g.region n=36.824817505217 s=35.132922973967 e=-3.808149140625 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_4448
output="C:\Users\GSI\OceanLider\Resultados\90\RI_R_4448.tif"
r.out.gdal input=R_4454
output="C:\Users\GSI\OceanLider\Resultados\90\RI_R_4454.tif"
r.out.gdal input=R_4460
output="C:\Users\GSI\OceanLider\Resultados\90\RI_R_4460.tif"

```

```
r.out.gdal input=R_4462  
output="C:\Users\GSI\OceanLider\Resultados\90\RF_R_4462.tif"
```



2.3.3. Combinación 11

Combinación de los subcriterios del **criterio de biodiversidad** (de exclusión parcial): *áreas de interés para la conservación de los cetáceos, áreas compatibles con la figura de parque nacional, áreas importantes para las aves, zonas propuestas para realizar el inventario de la biodiversidad marina, áreas importantes para las aves marinas, zonas marina especialmente sensible, y rutas migratorias de cetáceos.*


Se le deja la libertad al usuario para configurar el valor de idoneidad para las áreas que intervienen en cada subcriterio y se le muestra uno por defecto dependiendo del tipo de dispositivo. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el

valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.


El tipo de dispositivo al que esté ligado el escenario: **undimotriz- fuera costa (flotante)**.

Los parámetros establecidos en cada subcriterio:

- ✓ *Subcriterio Áreas de interés para la conservación de los cetáceos*

<input checked="" type="checkbox"/> Áreas de interés para la conservación de los cetáceos 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

- ✓ *Subcriterio Áreas compatibles con la figura de Parque Nacional*

<input checked="" type="checkbox"/> Áreas compatibles con la figura de Parque Nacional 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

- ✓ *Subcriterio Áreas importantes para las aves*

Áreas importantes para las aves 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Zonas propuestas para realizar el Inventario de la biodiversidad marina*

Zonas propuestas para realizar el Inventario de la biodiversidad marina 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Áreas importantes para las aves marinas*

Áreas importantes para las aves marinas 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Zona marina especialmente sensible*

Zona marina especialmente sensible 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio Rutas migratorias de cetáceos*

Rutas migratorias de cetáceos 

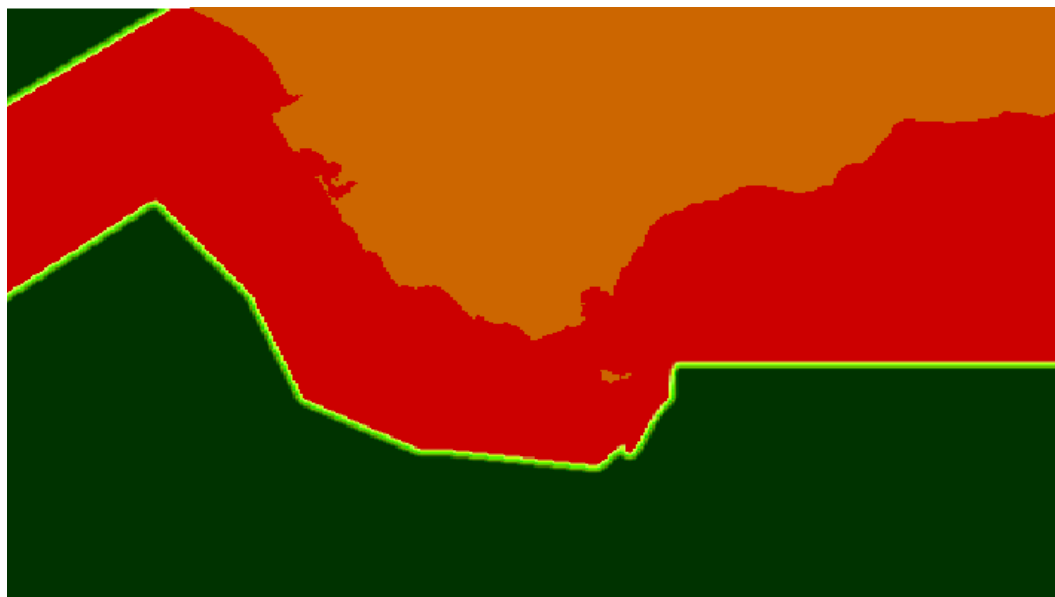
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

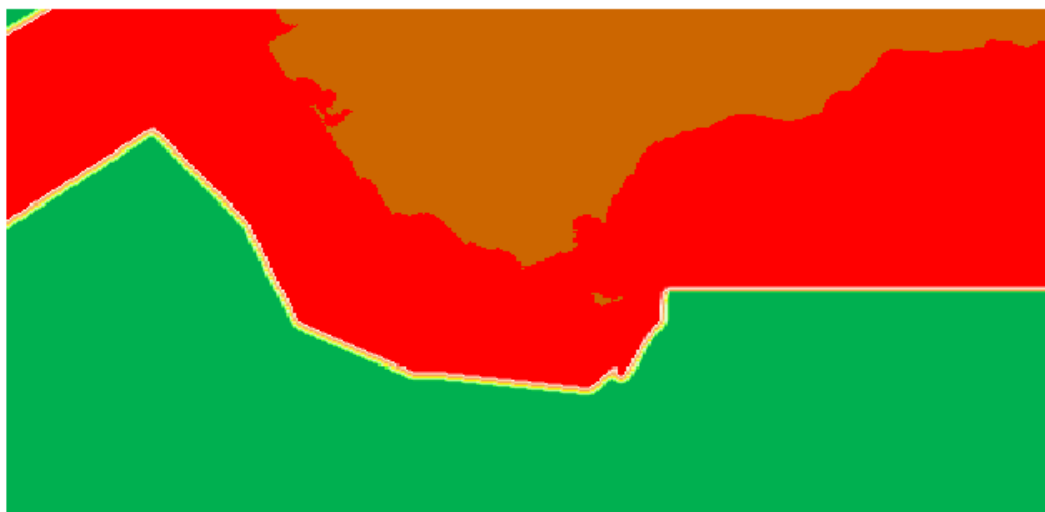
El método de razonamiento que se ha escogido para la combinación de los subcriterios: **más que pesimista (multiplicación de valores)**.

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ibas output=R_3607
v.to.rast input=R_3607 layer=1 type=point,line,area output=R_3608
use=val value=0.0 rows=4096
r.grow.distance input=R_3608 distance=R_3609 metric=euclidean
r.mapcalc "R_3610 = R_3609 * 100"
r.mapcalc "R_3611 = if(isnull( R_3610 ), null(), if((R_3610 <= 1 ||
R_3610 >= 99999), 0.0, if((R_3610 > 1 && R_3610 < 4), eval((R_3610 -
1)/(4 -1 )), if((R_3610 > 99999 && R_3610 < 99999 ), eval((99999 -
R_3610 )/(99999 - 99999)), 1.0))))"
r.patch input=R_3608,R_3611 output=R_3612
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ibm output=R_3613
v.to.rast input=R_3613 layer=1 type=point,line,area output=R_3614
use=val value=0.0 rows=4096
r.grow.distance input=R_3614 distance=R_3615 metric=euclidean
r.mapcalc "R_3616 = R_3615 * 100"
```

```
r.mapcalc "R_3617 = if(isnull( R_3616 ) , null(), if((R_3616 <= 1 ||
R_3616 >= 99999), 0.0, if((R_3616 > 1 && R_3616 < 4), eval((R_3616 -
1)/(4 -1 )), if((R_3616 > 99999 && R_3616 < 99999 ), eval((99999 -
R_3616)/(99999 - 99999)), 1.0))))"
r.patch input=R_3614,R_3617 output=R_3618
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=iba_recom output=R_3619
v.to.rast input=R_3619 layer=1 type=point,line,area output=R_3620
use=val value=0.0 rows=4096
r.grow.distance input=R_3620 distance=R_3621 metric=euclidean
r.mapcalc "R_3622 = R_3621 * 100"
r.mapcalc "R_3623 = if(isnull( R_3622 ) , null(), if((R_3622 <= 1 ||
R_3622 >= 99999), 0.0, if((R_3622 > 1 && R_3622 < 4), eval((R_3622 -
1)/(4 -1 )), if((R_3622 > 99999 && R_3622 < 99999 ), eval((99999 -
R_3622)/(99999 - 99999)), 1.0))))"
r.patch input=R_3620,R_3623 output=R_3624
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=aicp output=R_3625
v.to.rast input=R_3625 layer=1 type=point,line,area output=R_3626
use=val value=0.0 rows=4096
r.grow.distance input=R_3626 distance=R_3627 metric=euclidean
r.mapcalc "R_3628 = R_3627 * 100"
r.mapcalc "R_3629 = if(isnull( R_3628 ) , null(), if((R_3628 <= 1 ||
R_3628 >= 99999), 0.0, if((R_3628 > 1 && R_3628 < 4), eval((R_3628 -
1)/(4 -1 )), if((R_3628 > 99999 && R_3628 < 99999 ), eval((99999 -
R_3628)/(99999 - 99999)), 1.0))))"
r.patch input=R_3626,R_3629 output=R_3630
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zmes output=R_3631
v.to.rast input=R_3631 layer=1 type=point,line,area output=R_3632
use=val value=0.0 rows=4096
r.grow.distance input=R_3632 distance=R_3633 metric=euclidean
r.mapcalc "R_3634 = R_3633 * 100"
r.mapcalc "R_3635 = if(isnull( R_3634 ) , null(), if((R_3634 <= 1 ||
R_3634 >= 99999), 0.0, if((R_3634 > 1 && R_3634 < 4), eval((R_3634 -
1)/(4 -1 )), if((R_3634 > 99999 && R_3634 < 99999 ), eval((99999 -
R_3634)/(99999 - 99999)), 1.0))))"
r.patch input=R_3632,R_3635 output=R_3636
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=cetaceos output=R_3637
v.to.rast input=R_3637 layer=1 type=point,line,area output=R_3638
use=val value=0.0 rows=4096
r.grow.distance input=R_3638 distance=R_3639 metric=euclidean
r.mapcalc "R_3640 = R_3639 * 100"
r.mapcalc "R_3641 = if(isnull( R_3640 ) , null(), if((R_3640 <= 1 ||
R_3640 >= 99999), 0.0, if((R_3640 > 1 && R_3640 < 4), eval((R_3640 -
1)/(4 -1 )), if((R_3640 > 99999 && R_3640 < 99999 ), eval((99999 -
R_3640)/(99999 - 99999)), 1.0))))"
r.patch input=R_3638,R_3641 output=R_3642
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=aopn output=R_3643
v.to.rast input=R_3643 layer=1 type=point,line,area output=R_3644
use=val value=0.0 rows=4096
r.grow.distance input=R_3644 distance=R_3645 metric=euclidean
r.mapcalc "R_3646 = R_3645 * 100"
```

```
r.mapcalc "R_3647 = if(isnull( R_3646 ) , null(), if((R_3646 <= 1 ||
R_3646 >= 99999), 0.0, if((R_3646 > 1 && R_3646 < 4), eval((R_3646 -
1)/(4 -1 )), if((R_3646 > 99999 && R_3646 < 99999 ), eval((99999 -
R_3646)/(99999 - 99999)), 1.0))))"
r.patch input=R_3644,R_3647 output=R_3648
r.mapcalc "R_3649 = R_3612 * R_3618 * R_3624 * R_3630 * R_3636 * R_3642
* R_3648 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_3650 = if(isnull(ccaa),R_3649, -1.0)"
g.region n=36.781971227861 s=35.134022009111 e=-3.830121796875 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_3612
output="C:\Users\GSI\OceanLider\Resultados\91\RI_R_3612.tif"
r.out.gdal input=R_3618
output="C:\Users\GSI\OceanLider\Resultados\91\RI_R_3618.tif"
r.out.gdal input=R_3624
output="C:\Users\GSI\OceanLider\Resultados\91\RI_R_3624.tif"
r.out.gdal input=R_3630
output="C:\Users\GSI\OceanLider\Resultados\91\RI_R_3630.tif"
r.out.gdal input=R_3636
output="C:\Users\GSI\OceanLider\Resultados\91\RI_R_3636.tif"
r.out.gdal input=R_3642
output="C:\Users\GSI\OceanLider\Resultados\91\RI_R_3642.tif"
r.out.gdal input=R_3648
output="C:\Users\GSI\OceanLider\Resultados\91\RI_R_3648.tif"
r.out.gdal input=R_3650
output="C:\Users\GSI\OceanLider\Resultados\91\RF_R_3650.tif"
```





El método de razonamiento que se ha escogido para la combinación de los subcriterios: **pesimista (mínimo de valores).**

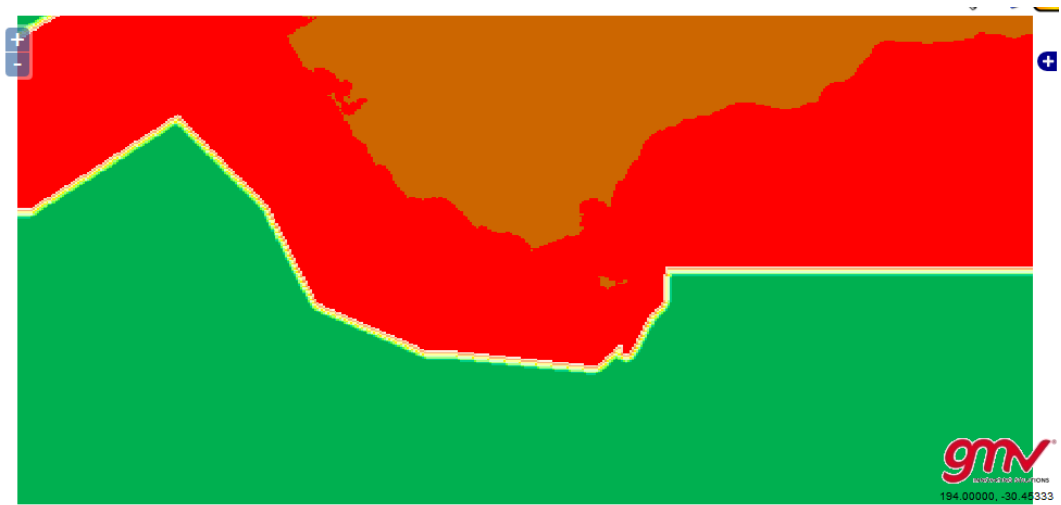
```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ibas output=R_4463
v.to.rast input=R_4463 layer=1 type=point,line,area output=R_4464
use=val value=0.0 rows=4096
r.grow.distance input=R_4464 distance=R_4465 metric=euclidean
r.mapcalc "R_4466 = R_4465 * 100"
r.mapcalc "R_4467 = if(isnull( R_4466 ), null(), if((R_4466 <= 1 ||
R_4466 >= 99999), 0.0, if((R_4466 > 1 && R_4466 < 4), eval((R_4466 -
1)/(4 -1 )), if((R_4466 > 99999 && R_4466 < 99999 ), eval((99999 -
R_4466)/(99999 - 99999)), 1.0))))"
r.patch input=R_4464,R_4467 output=R_4468
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ibm output=R_4469
v.to.rast input=R_4469 layer=1 type=point,line,area output=R_4470
use=val value=0.0 rows=4096
r.grow.distance input=R_4470 distance=R_4471 metric=euclidean
r.mapcalc "R_4472 = R_4471 * 100"
r.mapcalc "R_4473 = if(isnull( R_4472 ), null(), if((R_4472 <= 1 ||
R_4472 >= 99999), 0.0, if((R_4472 > 1 && R_4472 < 4), eval((R_4472 -
1)/(4 -1 )), if((R_4472 > 99999 && R_4472 < 99999 ), eval((99999 -
R_4472)/(99999 - 99999)), 1.0))))"
r.patch input=R_4470,R_4473 output=R_4474
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=iba_recom output=R_4475
v.to.rast input=R_4475 layer=1 type=point,line,area output=R_4476
use=val value=0.0 rows=4096
r.grow.distance input=R_4476 distance=R_4477 metric=euclidean
r.mapcalc "R_4478 = R_4477 * 100"
```

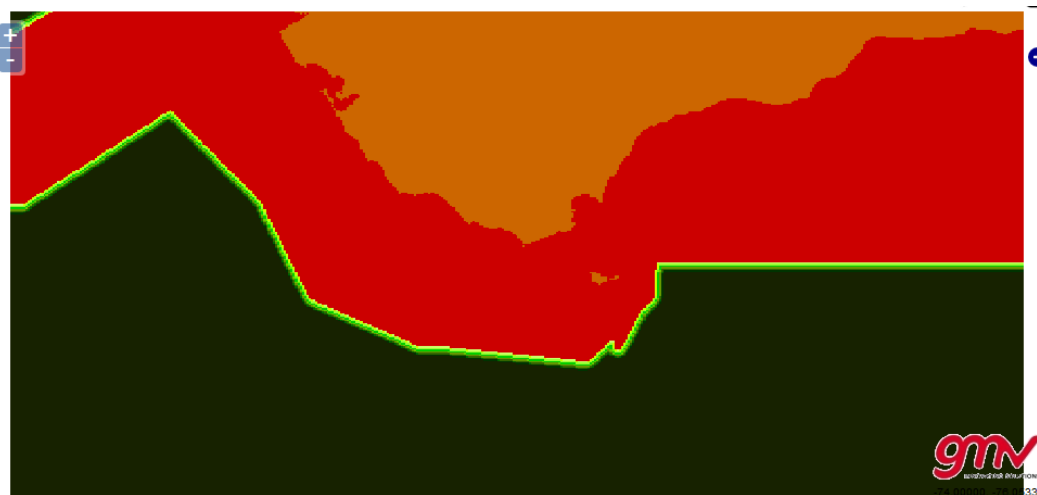
```

r.mapcalc "R_4479 = if(isnull( R_4478 ) , null(), if((R_4478 <= 1 ||
R_4478 >= 99999), 0.0, if((R_4478 > 1 && R_4478 < 4), eval((R_4478 -
1)/(4 -1 )), if((R_4478 > 99999 && R_4478 < 99999 ), eval((99999 -
R_4478 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4476,R_4479 output=R_4480
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=aicp output=R_4481
v.to.rast input=R_4481 layer=1 type=point,line,area output=R_4482
use=val value=0.0 rows=4096
r.grow.distance input=R_4482 distance=R_4483 metric=euclidean
r.mapcalc "R_4484 = R_4483 * 100"
r.mapcalc "R_4485 = if(isnull( R_4484 ) , null(), if((R_4484 <= 1 ||
R_4484 >= 99999), 0.0, if((R_4484 > 1 && R_4484 < 4), eval((R_4484 -
1)/(4 -1 )), if((R_4484 > 99999 && R_4484 < 99999 ), eval((99999 -
R_4484 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4482,R_4485 output=R_4486
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zmes output=R_4487
v.to.rast input=R_4487 layer=1 type=point,line,area output=R_4488
use=val value=0.0 rows=4096
r.grow.distance input=R_4488 distance=R_4489 metric=euclidean
r.mapcalc "R_4490 = R_4489 * 100"
r.mapcalc "R_4491 = if(isnull( R_4490 ) , null(), if((R_4490 <= 1 ||
R_4490 >= 99999), 0.0, if((R_4490 > 1 && R_4490 < 4), eval((R_4490 -
1)/(4 -1 )), if((R_4490 > 99999 && R_4490 < 99999 ), eval((99999 -
R_4490 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4488,R_4491 output=R_4492
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=cetaceos output=R_4493
v.to.rast input=R_4493 layer=1 type=point,line,area output=R_4494
use=val value=0.0 rows=4096
r.grow.distance input=R_4494 distance=R_4495 metric=euclidean
r.mapcalc "R_4496 = R_4495 * 100"
r.mapcalc "R_4497 = if(isnull( R_4496 ) , null(), if((R_4496 <= 1 ||
R_4496 >= 99999), 0.0, if((R_4496 > 1 && R_4496 < 4), eval((R_4496 -
1)/(4 -1 )), if((R_4496 > 99999 && R_4496 < 99999 ), eval((99999 -
R_4496 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4494,R_4497 output=R_4498
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=aopn output=R_4499
v.to.rast input=R_4499 layer=1 type=point,line,area output=R_4500
use=val value=0.0 rows=4096
r.grow.distance input=R_4500 distance=R_4501 metric=euclidean
r.mapcalc "R_4502 = R_4501 * 100"
r.mapcalc "R_4503 = if(isnull( R_4502 ) , null(), if((R_4502 <= 1 ||
R_4502 >= 99999), 0.0, if((R_4502 > 1 && R_4502 < 4), eval((R_4502 -
1)/(4 -1 )), if((R_4502 > 99999 && R_4502 < 99999 ), eval((99999 -
R_4502 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4500,R_4503 output=R_4504
r.mapcalc "R_4505 = if(isnull( R_4468 ) || isnull( R_4474 ) , null(),
if(R_4468 <= R_4474 , R_4468 , R_4474 )"
r.mapcalc "R_4506 = if(isnull( R_4505 ) || isnull( R_4474 ) , null(),
if(R_4505 <= R_4474 , R_4505 , R_4474 )"
r.mapcalc "R_4507 = if(isnull( R_4506 ) || isnull( R_4480 ) , null(),
if(R_4506 <= R_4480 , R_4506 , R_4480 )"

```

```
r.mapcalc "R_4508 = if(isnull( R_4507 ) || isnull( R_4486 ), null(),  
if(R_4507 <= R_4486 , R_4507 , R_4486 ))"  
r.mapcalc "R_4509 = if(isnull( R_4508 ) || isnull( R_4492 ), null(),  
if(R_4508 <= R_4492 , R_4508 , R_4492 ))"  
r.mapcalc "R_4510 = if(isnull( R_4509 ) || isnull( R_4498 ), null(),  
if(R_4509 <= R_4498 , R_4509 , R_4498 ))"  
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432  
user=postgres password=968XCdPh" layer=ccaa output=ccaa  
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val  
value=-1.0 rows=4096  
r.mapcalc "R_4511 = if(isnull(ccaa),R_4510, -1.0)"  
g.region n=36.802844848967 s=35.132922973967 e=-3.896039765625 w=-  
7.367719453125 nsres=300 ewres=300 rows=3224 cols=3453  
r.out.gdal input=R_4468  
output="C:\Users\GSI\OceanLider\Resultados\92\RI_R_4468.tif"  
r.out.gdal input=R_4474  
output="C:\Users\GSI\OceanLider\Resultados\92\RI_R_4474.tif"  
r.out.gdal input=R_4480  
output="C:\Users\GSI\OceanLider\Resultados\92\RI_R_4480.tif"  
r.out.gdal input=R_4486  
output="C:\Users\GSI\OceanLider\Resultados\92\RI_R_4486.tif"  
r.out.gdal input=R_4492  
output="C:\Users\GSI\OceanLider\Resultados\92\RI_R_4492.tif"  
r.out.gdal input=R_4498  
output="C:\Users\GSI\OceanLider\Resultados\92\RI_R_4498.tif"  
r.out.gdal input=R_4504  
output="C:\Users\GSI\OceanLider\Resultados\92\RI_R_4504.tif"  
r.out.gdal input=R_4511  
output="C:\Users\GSI\OceanLider\Resultados\92\RF_R_4511.tif"
```





El método de razonamiento que se ha escogido para la combinación de los subcriterios: **optimista (máximo de los valores).**

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ibas output=R_4886
v.to.rast input=R_4886 layer=1 type=point,line,area output=R_4887
use=val value=0.0 rows=4096
r.grow.distance input=R_4887 distance=R_4888 metric=euclidean
r.mapcalc "R_4889 = R_4888 * 100"
r.mapcalc "R_4890 = if(isnull( R_4889 ), null(), if((R_4889 <= 1 ||
R_4889 >= 99999), 0.0, if((R_4889 > 1 && R_4889 < 4), eval((R_4889 -
1)/(4 -1 )), if((R_4889 > 99999 && R_4889 < 99999 ), eval((99999 -
R_4889 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4887,R_4890 output=R_4891
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=iba output=R_4892
v.to.rast input=R_4892 layer=1 type=point,line,area output=R_4893
use=val value=0.0 rows=4096
r.grow.distance input=R_4893 distance=R_4894 metric=euclidean
r.mapcalc "R_4895 = R_4894 * 100"
r.mapcalc "R_4896 = if(isnull( R_4895 ), null(), if((R_4895 <= 1 ||
R_4895 >= 99999), 0.0, if((R_4895 > 1 && R_4895 < 4), eval((R_4895 -
1)/(4 -1 )), if((R_4895 > 99999 && R_4895 < 99999 ), eval((99999 -
R_4895 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4893,R_4896 output=R_4897
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=iba_recom output=R_4898
v.to.rast input=R_4898 layer=1 type=point,line,area output=R_4899
use=val value=0.0 rows=4096
r.grow.distance input=R_4899 distance=R_4900 metric=euclidean
r.mapcalc "R_4901 = R_4900 * 100"
```



```
r.mapcalc "R_4902 = if(isnull( R_4901 ) , null(), if((R_4901 <= 1 || R_4901 >= 99999), 0.0, if((R_4901 > 1 && R_4901 < 4), eval((R_4901 - 1)/(4 - 1)), if((R_4901 > 99999 && R_4901 < 99999 ), eval((99999 - R_4901)/(99999 - 99999)), 1.0))))"
r.patch input=R_4899,R_4902 output=R_4903
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost port=5432 user=postgres password=968XCdPh" layer=aicp output=R_4904
v.to.rast input=R_4904 layer=1 type=point,line,area output=R_4905
use=val value=0.0 rows=4096
r.grow.distance input=R_4905 distance=R_4906 metric=euclidean
r.mapcalc "R_4907 = R_4906 * 100"
r.mapcalc "R_4908 = if(isnull( R_4907 ) , null(), if((R_4907 <= 1 || R_4907 >= 99999), 0.0, if((R_4907 > 1 && R_4907 < 4), eval((R_4907 - 1)/(4 - 1)), if((R_4907 > 99999 && R_4907 < 99999 ), eval((99999 - R_4907)/(99999 - 99999)), 1.0))))"
r.patch input=R_4905,R_4908 output=R_4909
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost port=5432 user=postgres password=968XCdPh" layer=zmes output=R_4910
v.to.rast input=R_4910 layer=1 type=point,line,area output=R_4911
use=val value=0.0 rows=4096
r.grow.distance input=R_4911 distance=R_4912 metric=euclidean
r.mapcalc "R_4913 = R_4912 * 100"
r.mapcalc "R_4914 = if(isnull( R_4913 ) , null(), if((R_4913 <= 1 || R_4913 >= 99999), 0.0, if((R_4913 > 1 && R_4913 < 4), eval((R_4913 - 1)/(4 - 1)), if((R_4913 > 99999 && R_4913 < 99999 ), eval((99999 - R_4913)/(99999 - 99999)), 1.0))))"
r.patch input=R_4911,R_4914 output=R_4915
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost port=5432 user=postgres password=968XCdPh" layer=cetaceos output=R_4916
v.to.rast input=R_4916 layer=1 type=point,line,area output=R_4917
use=val value=0.0 rows=4096
r.grow.distance input=R_4917 distance=R_4918 metric=euclidean
r.mapcalc "R_4919 = R_4918 * 100"
r.mapcalc "R_4920 = if(isnull( R_4919 ) , null(), if((R_4919 <= 1 || R_4919 >= 99999), 0.0, if((R_4919 > 1 && R_4919 < 4), eval((R_4919 - 1)/(4 - 1)), if((R_4919 > 99999 && R_4919 < 99999 ), eval((99999 - R_4919)/(99999 - 99999)), 1.0))))"
r.patch input=R_4917,R_4920 output=R_4921
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost port=5432 user=postgres password=968XCdPh" layer=aopn output=R_4922
v.to.rast input=R_4922 layer=1 type=point,line,area output=R_4923
use=val value=0.0 rows=4096
r.grow.distance input=R_4923 distance=R_4924 metric=euclidean
r.mapcalc "R_4925 = R_4924 * 100"
r.mapcalc "R_4926 = if(isnull( R_4925 ) , null(), if((R_4925 <= 1 || R_4925 >= 99999), 0.0, if((R_4925 > 1 && R_4925 < 4), eval((R_4925 - 1)/(4 - 1)), if((R_4925 > 99999 && R_4925 < 99999 ), eval((99999 - R_4925)/(99999 - 99999)), 1.0))))"
r.patch input=R_4923,R_4926 output=R_4927
r.mapcalc "R_4928 = if(isnull( R_4891 ) || isnull( R_4897 ), null(), if(R_4891 >= R_4897 , R_4891 , R_4897 ))"
r.mapcalc "R_4929 = if(isnull( R_4928 ) || isnull( R_4897 ), null(), if(R_4928 >= R_4897 , R_4928 , R_4897 ))"
r.mapcalc "R_4930 = if(isnull( R_4929 ) || isnull( R_4903 ), null(), if(R_4929 >= R_4903 , R_4929 , R_4903 ))"
```

```
r.mapcalc "R_4931 = if(isnull( R_4930 ) || isnull( R_4909 ), null(),
if(R_4930 >= R_4909 , R_4930 , R_4909 ))"
r.mapcalc "R_4932 = if(isnull( R_4931 ) || isnull( R_4915 ), null(),
if(R_4931 >= R_4915 , R_4931 , R_4915 ))"
r.mapcalc "R_4933 = if(isnull( R_4932 ) || isnull( R_4921 ), null(),
if(R_4932 >= R_4921 , R_4932 , R_4921 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_4934 = if(isnull(ccaa),R_4933, -1.0)"
g.region n=36.758899536467 s=35.176868286467 e=-3.808149140625 w=-
7.367719453125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_4891
output="C:\Users\GSI\OceanLider\Resultados\93\RI_R_4891.tif"
r.out.gdal input=R_4897
output="C:\Users\GSI\OceanLider\Resultados\93\RI_R_4897.tif"
r.out.gdal input=R_4903
output="C:\Users\GSI\OceanLider\Resultados\93\RI_R_4903.tif"
r.out.gdal input=R_4909
output="C:\Users\GSI\OceanLider\Resultados\93\RI_R_4909.tif"
r.out.gdal input=R_4915
output="C:\Users\GSI\OceanLider\Resultados\93\RI_R_4915.tif"
r.out.gdal input=R_4921
output="C:\Users\GSI\OceanLider\Resultados\93\RI_R_4921.tif"
r.out.gdal input=R_4927
output="C:\Users\GSI\OceanLider\Resultados\93\RI_R_4927.tif"
r.out.gdal input=R_4934
output="C:\Users\GSI\OceanLider\Resultados\93\RF_R_4934.tif"
```






2.3.4. Combinación 12

Combinación de tres subcriterios del criterio **de biodiversidad** (de exclusión parcial) mediante el método de razonamiento **MLP**. Los subcriterios que intervienen con sus respectivos pesos son: *áreas importantes para las aves* con un peso de **0.6**, *zonas propuestas para realizar el inventario de la biodiversidad marina* con un peso de **0.3**, y *áreas importantes para las aves marinas* con un peso de **0.1**.

Se le deja la libertad al usuario para configurar el valor de idoneidad para las áreas que intervienen en cada subcriterio y se le muestra uno por defecto dependiendo del tipo de dispositivo. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.


El tipo de dispositivo al que esté ligado el escenario: **undimotriz- fuera costa (flotante)**.
Los parámetros establecidos en cada subcriterio:

- ✓ *Subcriterio Áreas Importantes para las Aves*

Áreas importantes para las aves 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.6

✓ *Subcriterio Zonas propuestas para realizar el Inventario de la biodiversidad marina*

Zonas propuestas para realizar el Inventario de la biodiversidad marina 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.3

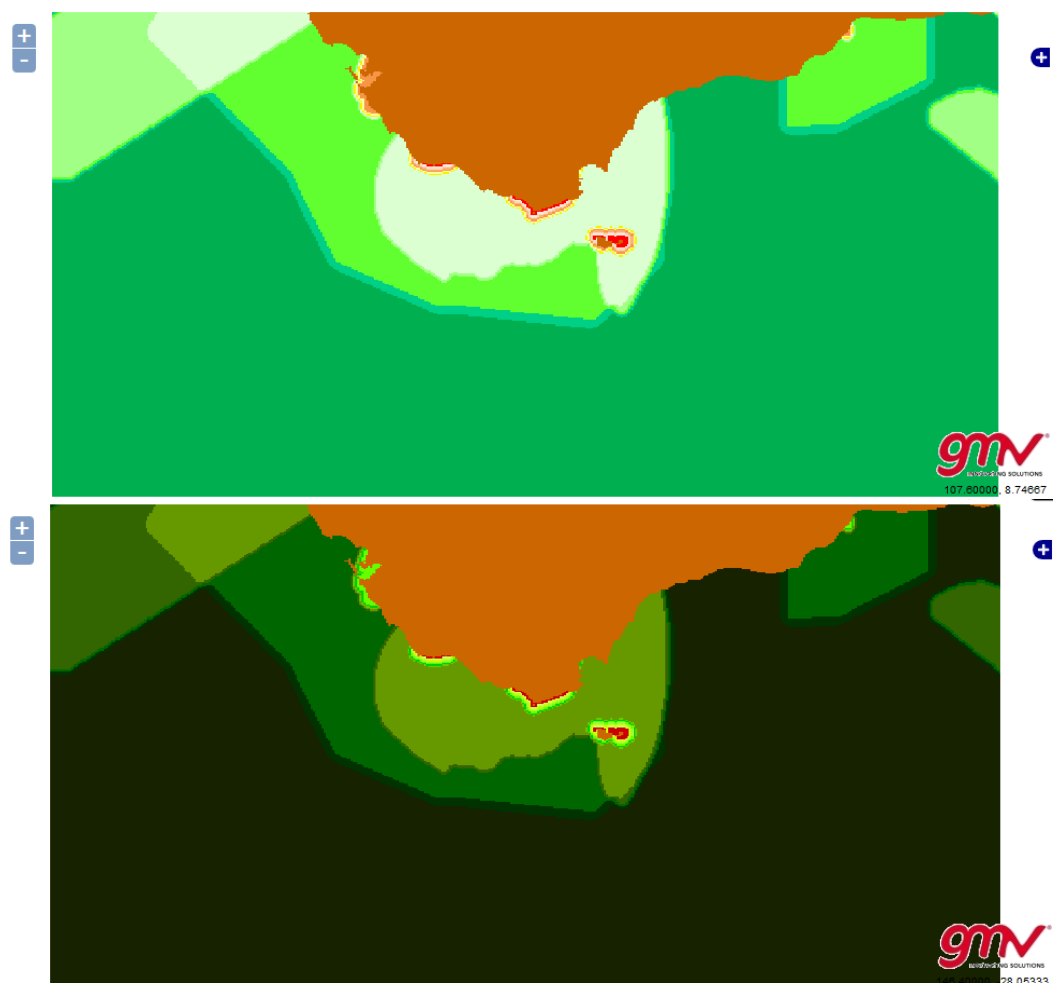
✓ *Subcriterio Áreas importantes para las aves marinas*

Áreas importantes para las aves marinas 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.1

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ibas output=R_2128
v.to.rast input=R_2128 layer=1 type=point,line,area output=R_2129
use=val value=0.0 rows=4096
r.grow.distance input=R_2129 distance=R_2130 metric=euclidean
r.mapcalc "R_2131 = R_2130 * 100"
```

```
r.mapcalc "R_2132 = if(isnull( R_2131 ) , null(), if((R_2131 <= 1 ||
R_2131 >= 99999), 0.0, if((R_2131 > 1 && R_2131 < 4), eval((R_2131 -
1)/(4 -1 )), if((R_2131 > 99999 && R_2131 < 99999 ), eval((99999 -
R_2131)/(99999 - 99999)), 1.0))))"
r.patch input=R_2129,R_2132 output=R_2133
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ibm output=R_2134
v.to.rast input=R_2134 layer=1 type=point,line,area output=R_2135
use=val value=0.0 rows=4096
r.grow.distance input=R_2135 distance=R_2136 metric=euclidean
r.mapcalc "R_2137 = R_2136 * 100"
r.mapcalc "R_2138 = if(isnull( R_2137 ) , null(), if((R_2137 <= 1 ||
R_2137 >= 99999), 0.0, if((R_2137 > 1 && R_2137 < 4), eval((R_2137 -
1)/(4 -1 )), if((R_2137 > 99999 && R_2137 < 99999 ), eval((99999 -
R_2137)/(99999 - 99999)), 1.0))))"
r.patch input=R_2135,R_2138 output=R_2139
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=iba_recom output=R_2140
v.to.rast input=R_2140 layer=1 type=point,line,area output=R_2141
use=val value=0.0 rows=4096
r.grow.distance input=R_2141 distance=R_2142 metric=euclidean
r.mapcalc "R_2143 = R_2142 * 100"
r.mapcalc "R_2144 = if(isnull( R_2143 ) , null(), if((R_2143 <= 1 ||
R_2143 >= 99999), 0.0, if((R_2143 > 1 && R_2143 < 4), eval((R_2143 -
1)/(4 -1 )), if((R_2143 > 99999 && R_2143 < 99999 ), eval((99999 -
R_2143)/(99999 - 99999)), 1.0))))"
r.patch input=R_2141,R_2144 output=R_2145
r.mapcalc "R_2146 = 0.6 * R_2133 + 0.3 * R_2139 + 0.1 * R_2145 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_2147 = if(isnull(ccaa),R_2146, -1.0)"
g.region n=36.738025915361 s=34.958240759111 e=-3.896039765625 w=-
7.389692109375 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_2133
output="C:\Users\GSI\OceanLider\Resultados\94\RI_R_2133.tif"
r.out.gdal input=R_2139
output="C:\Users\GSI\OceanLider\Resultados\94\RI_R_2139.tif"
r.out.gdal input=R_2145
output="C:\Users\GSI\OceanLider\Resultados\94\RI_R_2145.tif"
r.out.gdal input=R_2147
output="C:\Users\GSI\OceanLider\Resultados\94\RF_R_2147.tif"
```



2.3.5. Combinación 13

Combinación de los subcriterios del **criterio de usos pesqueros** (de exclusión parcial): *subcriterio instalaciones de acuicultura, subcriterio almadrabas, subcriterio arrecifes, subcriterio caladeros, subcriterio viveros, subcriterio zonas preotegidas de interés pesquero, subcriterio piscifactorías, subcriterio reservas marinas y subcriterio zonas de producción de moluscos y de otros invertebrados marinos.*

Se le deja la libertad al usuario para configurar el valor de idoneidad para las áreas que intervienen en cada subcriterio y se le muestra uno por defecto dependiendo del tipo de dispositivo. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

El tipo de dispositivo al que esté ligado el escenario: **undimotriz- fuera costa (flotante)**.

Los parámetros establecidos en cada subcriterio:

- ✓ *Subcriterio instalaciones de acuicultura*

Instalaciones de acuicultura 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

- ✓ *Subcriterio almadrabas*

Almadrabas 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

- ✓ *Subcriterio arrecifes*

Arrecifes 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio caladeros*

Caladeros 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio viveros*

Viveros 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio zonas preotegidas de interés pesquero*

Zonas protegidas de interés pesquero 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio piscifactoría*

Piscifactorías 


Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio reservas marinas*

Reservas marinas 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio zonas de producción de moluscos y de otros invertebrados marinos*

Zonas de producción de moluscos y de otros invertebrados marinos 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

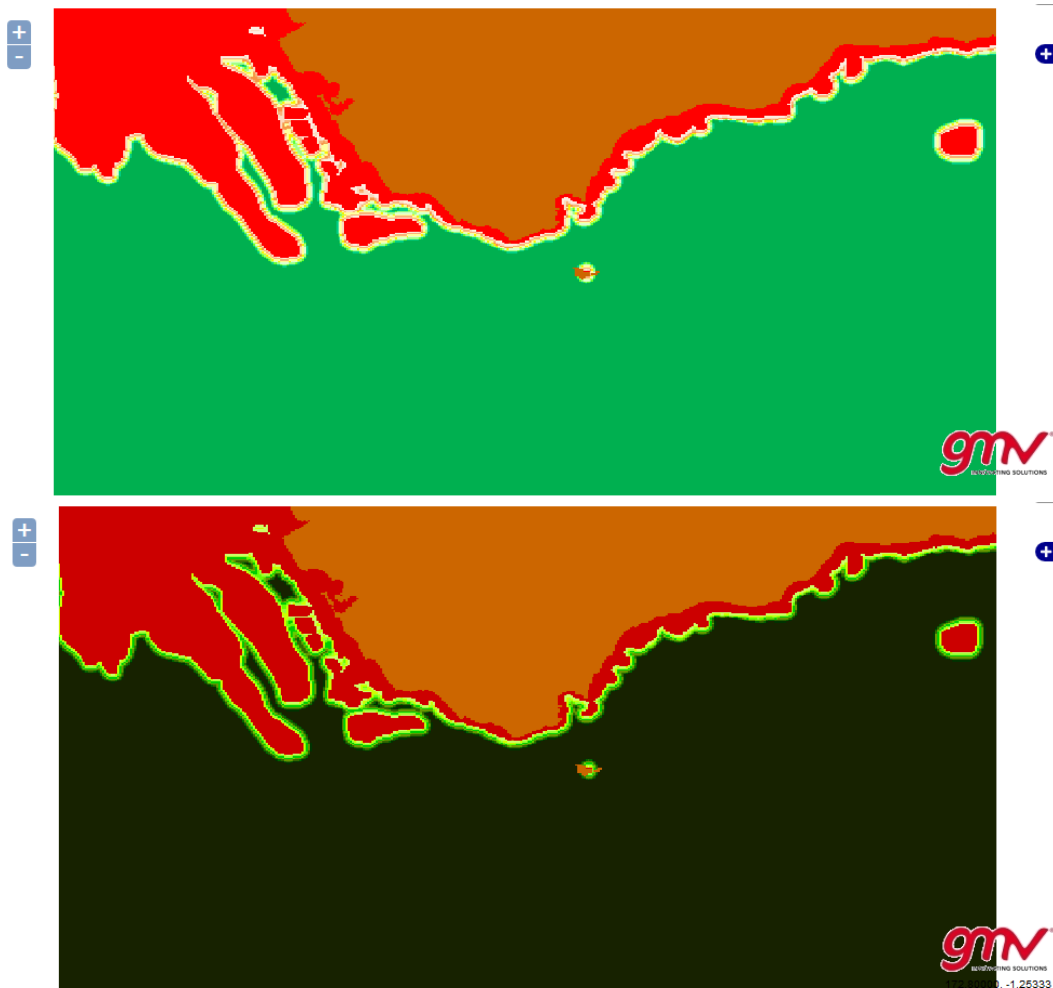
El método de razonamiento que se ha escogido para la combinación de los subcriterios: **más que pesimista (multiplicación de valores).**

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
```

```
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=acuicultura
output=R_5126
v.to.rast input=R_5126 layer=1 type=point,line,area output=R_5127
use=val value=0.0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=acuicultura_buf
output=R_5128
v.to.rast input=R_5128 layer=1 type=point,line,area output=R_5129
use=val value=0.0 rows=4096
r.patch input=R_5127,R_5129 output=R_5130
r.grow.distance input=R_5130 distance=R_5131 metric=euclidean
r.mapcalc "R_5132 = R_5131 * 100"
r.mapcalc "R_5133 = if(isnull( R_5132 ), null(), if((R_5132 <= 1 ||
R_5132 >= 99999), 0.0, if((R_5132 > 1 && R_5132 < 4), eval((R_5132 -
1)/(4 -1 )), if((R_5132 > 99999 && R_5132 < 99999 ), eval((99999 -
R_5132)/(99999 - 99999)), 1.0))))"
r.patch input=R_5130,R_5133 output=R_5134
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=cria_moluscos
output=R_5135
v.to.rast input=R_5135 layer=1 type=point,line,area output=R_5136
use=val value=0.0 rows=4096
r.grow.distance input=R_5136 distance=R_5137 metric=euclidean
r.mapcalc "R_5138 = R_5137 * 100"
r.mapcalc "R_5139 = if(isnull( R_5138 ), null(), if((R_5138 <= 1 ||
R_5138 >= 99999), 0.0, if((R_5138 > 1 && R_5138 < 4), eval((R_5138 -
1)/(4 -1 )), if((R_5138 > 99999 && R_5138 < 99999 ), eval((99999 -
R_5138)/(99999 - 99999)), 1.0))))"
r.patch input=R_5136,R_5139 output=R_5140
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=almadrabas
output=R_5141
v.to.rast input=R_5141 layer=1 type=point,line,area output=R_5142
use=val value=0.0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=almadrabas_buf
output=R_5143
v.to.rast input=R_5143 layer=1 type=point,line,area output=R_5144
use=val value=0.0 rows=4096
r.patch input=R_5142,R_5144 output=R_5145
r.grow.distance input=R_5145 distance=R_5146 metric=euclidean
r.mapcalc "R_5147 = R_5146 * 100"
r.mapcalc "R_5148 = if(isnull( R_5147 ), null(), if((R_5147 <= 1 ||
R_5147 >= 99999), 0.0, if((R_5147 > 1 && R_5147 < 4), eval((R_5147 -
1)/(4 -1 )), if((R_5147 > 99999 && R_5147 < 99999 ), eval((99999 -
R_5147)/(99999 - 99999)), 1.0))))"
r.patch input=R_5145,R_5148 output=R_5149
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=rrmm output=R_5150
v.to.rast input=R_5150 layer=1 type=point,line,area output=R_5151
use=val value=0.0 rows=4096
r.grow.distance input=R_5151 distance=R_5152 metric=euclidean
r.mapcalc "R_5153 = R_5152 * 100"
```

```
r.mapcalc "R_5154 = if(isnull( R_5153 ) , null(), if((R_5153 <= 1 || R_5153 >= 99999), 0.0, if((R_5153 > 1 && R_5153 < 4), eval((R_5153 - 1)/(4 - 1)), if((R_5153 > 99999 && R_5153 < 99999), eval((99999 - R_5153)/(99999 - 99999)), 1.0))))"
r.patch input=R_5151,R_5154 output=R_5155
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost port=5432 user=postgres password=968XCdPh" layer=caladeros output=R_5156
v.to.rast input=R_5156 layer=1 type=point,line,area output=R_5157
use=val value=0.0 rows=4096
r.grow.distance input=R_5157 distance=R_5158 metric=euclidean
r.mapcalc "R_5159 = R_5158 * 100"
r.mapcalc "R_5160 = if(isnull( R_5159 ) , null(), if((R_5159 <= 1 || R_5159 >= 99999), 0.0, if((R_5159 > 1 && R_5159 < 4), eval((R_5159 - 1)/(4 - 1)), if((R_5159 > 99999 && R_5159 < 99999), eval((99999 - R_5159)/(99999 - 99999)), 1.0))))"
r.patch input=R_5157,R_5160 output=R_5161
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost port=5432 user=postgres password=968XCdPh" layer=viveros output=R_5162
v.to.rast input=R_5162 layer=1 type=point,line,area output=R_5163
use=val value=0.0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost port=5432 user=postgres password=968XCdPh" layer=viveros_buf output=R_5164
v.to.rast input=R_5164 layer=1 type=point,line,area output=R_5165
use=val value=0.0 rows=4096
r.patch input=R_5163,R_5165 output=R_5166
r.grow.distance input=R_5166 distance=R_5167 metric=euclidean
r.mapcalc "R_5168 = R_5167 * 100"
r.mapcalc "R_5169 = if(isnull( R_5168 ) , null(), if((R_5168 <= 1 || R_5168 >= 99999), 0.0, if((R_5168 > 1 && R_5168 < 4), eval((R_5168 - 1)/(4 - 1)), if((R_5168 > 99999 && R_5168 < 99999), eval((99999 - R_5168)/(99999 - 99999)), 1.0))))"
r.patch input=R_5166,R_5169 output=R_5170
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost port=5432 user=postgres password=968XCdPh" layer=zpips output=R_5171
v.to.rast input=R_5171 layer=1 type=point,line,area output=R_5172
use=val value=0.0 rows=4096
r.grow.distance input=R_5172 distance=R_5173 metric=euclidean
r.mapcalc "R_5174 = R_5173 * 100"
r.mapcalc "R_5175 = if(isnull( R_5174 ) , null(), if((R_5174 <= 1 || R_5174 >= 99999), 0.0, if((R_5174 > 1 && R_5174 < 4), eval((R_5174 - 1)/(4 - 1)), if((R_5174 > 99999 && R_5174 < 99999), eval((99999 - R_5174)/(99999 - 99999)), 1.0))))"
r.patch input=R_5172,R_5175 output=R_5176
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost port=5432 user=postgres password=968XCdPh" layer=arrecifes output=R_5177
v.to.rast input=R_5177 layer=1 type=point,line,area output=R_5178
use=val value=0.0 rows=4096
r.grow.distance input=R_5178 distance=R_5179 metric=euclidean
r.mapcalc "R_5180 = R_5179 * 100"
r.mapcalc "R_5181 = if(isnull( R_5180 ) , null(), if((R_5180 <= 1 || R_5180 >= 99999), 0.0, if((R_5180 > 1 && R_5180 < 4), eval((R_5180 - 1)/(4 - 1)), if((R_5180 > 99999 && R_5180 < 99999), eval((99999 - R_5180)/(99999 - 99999)), 1.0))))"
r.patch input=R_5178,R_5181 output=R_5182
```

```
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=piscifactoria
output=R_5183
v.to.rast input=R_5183 layer=1 type=point,line,area output=R_5184
use=val value=0.0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=piscifactoria_buf
output=R_5185
v.to.rast input=R_5185 layer=1 type=point,line,area output=R_5186
use=val value=0.0 rows=4096
r.patch input=R_5184,R_5186 output=R_5187
r.grow.distance input=R_5187 distance=R_5188 metric=euclidean
r.mapcalc "R_5189 = R_5188 * 100"
r.mapcalc "R_5190 = if(isnull( R_5189 ), null(), if((R_5189 <= 1 ||
R_5189 >= 99999), 0.0, if((R_5189 > 1 && R_5189 < 4), eval((R_5189 -
1)/(4 -1 )), if((R_5189 > 99999 && R_5189 < 99999 ), eval((99999 -
R_5189)/(99999 - 99999)), 1.0))))"
r.patch input=R_5187,R_5190 output=R_5191
r.mapcalc "R_5192 = R_5134 * R_5140 * R_5149 * R_5155 * R_5161 * R_5170
* R_5176 * R_5182 * R_5191 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_5193 = if(isnull(ccaa),R_5192, -1.0)"
g.region n=36.847889196611 s=35.090076696611 e=-3.852094453125 w=-
7.257856171875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_5134
output="C:\Users\GSI\OceanLider\Resultados\95\RI_R_5134.tif"
r.out.gdal input=R_5140
output="C:\Users\GSI\OceanLider\Resultados\95\RI_R_5140.tif"
r.out.gdal input=R_5149
output="C:\Users\GSI\OceanLider\Resultados\95\RI_R_5149.tif"
r.out.gdal input=R_5155
output="C:\Users\GSI\OceanLider\Resultados\95\RI_R_5155.tif"
r.out.gdal input=R_5161
output="C:\Users\GSI\OceanLider\Resultados\95\RI_R_5161.tif"
r.out.gdal input=R_5170
output="C:\Users\GSI\OceanLider\Resultados\95\RI_R_5170.tif"
r.out.gdal input=R_5176
output="C:\Users\GSI\OceanLider\Resultados\95\RI_R_5176.tif"
r.out.gdal input=R_5182
output="C:\Users\GSI\OceanLider\Resultados\95\RI_R_5182.tif"
r.out.gdal input=R_5191
output="C:\Users\GSI\OceanLider\Resultados\95\RI_R_5191.tif"
r.out.gdal input=R_5193
output="C:\Users\GSI\OceanLider\Resultados\95\RF_R_5193.tif"
```



El método de razonamiento que se ha escogido para la combinación de los subcriterios: **pesimista (mínimo de valores).**

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=acuicultura
output=R_3977
v.to.rast input=R_3977 layer=1 type=point,line,area output=R_3978
use=val value=0.0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=acuicultura_buf
output=R_3979
v.to.rast input=R_3979 layer=1 type=point,line,area output=R_3980
use=val value=0.0 rows=4096
```

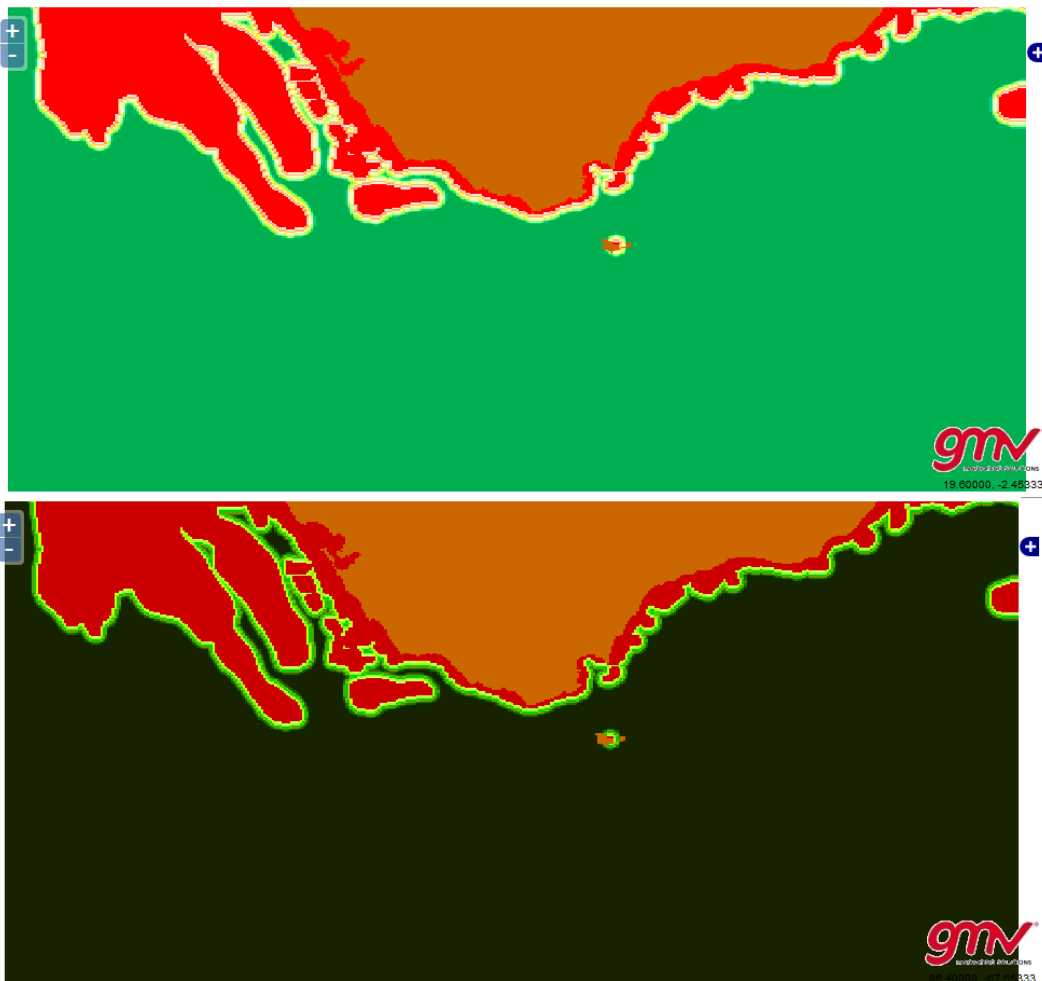
```
r.patch input=R_3978,R_3980 output=R_3981
r.grow.distance input=R_3981 distance=R_3982 metric=euclidean
r.mapcalc "R_3983 = R_3982 * 100"
r.mapcalc "R_3984 = if(isnull( R_3983 ) , null(), if((R_3983 <= 1 ||
R_3983 >= 99999), 0.0, if((R_3983 > 1 && R_3983 < 4), eval((R_3983 -
1)/(4 -1 )), if((R_3983 > 99999 && R_3983 < 99999 ), eval((99999 -
R_3983 )/(99999 - 99999)), 1.0))))"
r.patch input=R_3981,R_3984 output=R_3985
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=cria_moluscos
output=R_3986
v.to.rast input=R_3986 layer=1 type=point,line,area output=R_3987
use=val value=0.0 rows=4096
r.grow.distance input=R_3987 distance=R_3988 metric=euclidean
r.mapcalc "R_3989 = R_3988 * 100"
r.mapcalc "R_3990 = if(isnull( R_3989 ) , null(), if((R_3989 <= 1 ||
R_3989 >= 99999), 0.0, if((R_3989 > 1 && R_3989 < 4), eval((R_3989 -
1)/(4 -1 )), if((R_3989 > 99999 && R_3989 < 99999 ), eval((99999 -
R_3989 )/(99999 - 99999)), 1.0))))"
r.patch input=R_3987,R_3990 output=R_3991
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=almadrabas
output=R_3992
v.to.rast input=R_3992 layer=1 type=point,line,area output=R_3993
use=val value=0.0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=almadrabas_buf
output=R_3994
v.to.rast input=R_3994 layer=1 type=point,line,area output=R_3995
use=val value=0.0 rows=4096
r.patch input=R_3993,R_3995 output=R_3996
r.grow.distance input=R_3996 distance=R_3997 metric=euclidean
r.mapcalc "R_3998 = R_3997 * 100"
r.mapcalc "R_3999 = if(isnull( R_3998 ) , null(), if((R_3998 <= 1 ||
R_3998 >= 99999), 0.0, if((R_3998 > 1 && R_3998 < 4), eval((R_3998 -
1)/(4 -1 )), if((R_3998 > 99999 && R_3998 < 99999 ), eval((99999 -
R_3998 )/(99999 - 99999)), 1.0))))"
r.patch input=R_3996,R_3999 output=R_4000
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=rrmm output=R_4001
v.to.rast input=R_4001 layer=1 type=point,line,area output=R_4002
use=val value=0.0 rows=4096
r.grow.distance input=R_4002 distance=R_4003 metric=euclidean
r.mapcalc "R_4004 = R_4003 * 100"
r.mapcalc "R_4005 = if(isnull( R_4004 ) , null(), if((R_4004 <= 1 ||
R_4004 >= 99999), 0.0, if((R_4004 > 1 && R_4004 < 4), eval((R_4004 -
1)/(4 -1 )), if((R_4004 > 99999 && R_4004 < 99999 ), eval((99999 -
R_4004 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4002,R_4005 output=R_4006
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=caladeros output=R_4007
v.to.rast input=R_4007 layer=1 type=point,line,area output=R_4008
use=val value=0.0 rows=4096
r.grow.distance input=R_4008 distance=R_4009 metric=euclidean
r.mapcalc "R_4010 = R_4009 * 100"
```

```
r.mapcalc "R_4011 = if(isnull( R_4010 ) , null(), if((R_4010 <= 1 ||
R_4010 >= 99999), 0.0, if((R_4010 > 1 && R_4010 < 4), eval((R_4010 -
1)/(4 -1 )), if((R_4010 > 99999 && R_4010 < 99999 ), eval((99999 -
R_4010)/(99999 - 99999)), 1.0))))"
r.patch input=R_4008,R_4011 output=R_4012
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=viveros output=R_4013
v.to.rast input=R_4013 layer=1 type=point,line,area output=R_4014
use=val value=0.0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=viveros_buf
output=R_4015
v.to.rast input=R_4015 layer=1 type=point,line,area output=R_4016
use=val value=0.0 rows=4096
r.patch input=R_4014,R_4016 output=R_4017
r.grow.distance input=R_4017 distance=R_4018 metric=euclidean
r.mapcalc "R_4019 = R_4018 * 100"
r.mapcalc "R_4020 = if(isnull( R_4019 ) , null(), if((R_4019 <= 1 ||
R_4019 >= 99999), 0.0, if((R_4019 > 1 && R_4019 < 4), eval((R_4019 -
1)/(4 -1 )), if((R_4019 > 99999 && R_4019 < 99999 ), eval((99999 -
R_4019)/(99999 - 99999)), 1.0))))"
r.patch input=R_4017,R_4020 output=R_4021
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zpips output=R_4022
v.to.rast input=R_4022 layer=1 type=point,line,area output=R_4023
use=val value=0.0 rows=4096
r.grow.distance input=R_4023 distance=R_4024 metric=euclidean
r.mapcalc "R_4025 = R_4024 * 100"
r.mapcalc "R_4026 = if(isnull( R_4025 ) , null(), if((R_4025 <= 1 ||
R_4025 >= 99999), 0.0, if((R_4025 > 1 && R_4025 < 4), eval((R_4025 -
1)/(4 -1 )), if((R_4025 > 99999 && R_4025 < 99999 ), eval((99999 -
R_4025)/(99999 - 99999)), 1.0))))"
r.patch input=R_4023,R_4026 output=R_4027
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=arrecifes output=R_4028
v.to.rast input=R_4028 layer=1 type=point,line,area output=R_4029
use=val value=0.0 rows=4096
r.grow.distance input=R_4029 distance=R_4030 metric=euclidean
r.mapcalc "R_4031 = R_4030 * 100"
r.mapcalc "R_4032 = if(isnull( R_4031 ) , null(), if((R_4031 <= 1 ||
R_4031 >= 99999), 0.0, if((R_4031 > 1 && R_4031 < 4), eval((R_4031 -
1)/(4 -1 )), if((R_4031 > 99999 && R_4031 < 99999 ), eval((99999 -
R_4031)/(99999 - 99999)), 1.0))))"
r.patch input=R_4029,R_4032 output=R_4033
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=piscifactoria
output=R_4034
v.to.rast input=R_4034 layer=1 type=point,line,area output=R_4035
use=val value=0.0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=piscifactoria_buf
output=R_4036
v.to.rast input=R_4036 layer=1 type=point,line,area output=R_4037
use=val value=0.0 rows=4096
r.patch input=R_4035,R_4037 output=R_4038
```

```

r.grow.distance input=R_4038 distance=R_4039 metric=euclidean
r.mapcalc "R_4040 = R_4039 * 100"
r.mapcalc "R_4041 = if(isnull( R_4040 ) , null(), if((R_4040 <= 1 ||
R_4040 >= 99999), 0.0, if((R_4040 > 1 && R_4040 < 4), eval((R_4040 -
1)/(4 - 1)), if((R_4040 > 99999 && R_4040 < 99999), eval((99999 -
R_4040)/(99999 - 99999)), 1.0))))"
r.patch input=R_4038,R_4041 output=R_4042
r.mapcalc "R_4043 = if(isnull( R_3985 ) || isnull( R_3991 ), null(),
if(R_3985 <= R_3991 , R_3985 , R_3991 ))"
r.mapcalc "R_4044 = if(isnull( R_4043 ) || isnull( R_3991 ), null(),
if(R_4043 <= R_3991 , R_4043 , R_3991 ))"
r.mapcalc "R_4045 = if(isnull( R_4044 ) || isnull( R_4000 ), null(),
if(R_4044 <= R_4000 , R_4044 , R_4000 ))"
r.mapcalc "R_4046 = if(isnull( R_4045 ) || isnull( R_4006 ), null(),
if(R_4045 <= R_4006 , R_4045 , R_4006 ))"
r.mapcalc "R_4047 = if(isnull( R_4046 ) || isnull( R_4012 ), null(),
if(R_4046 <= R_4012 , R_4046 , R_4012 ))"
r.mapcalc "R_4048 = if(isnull( R_4047 ) || isnull( R_4021 ), null(),
if(R_4047 <= R_4021 , R_4047 , R_4021 ))"
r.mapcalc "R_4049 = if(isnull( R_4048 ) || isnull( R_4027 ), null(),
if(R_4048 <= R_4027 , R_4048 , R_4027 ))"
r.mapcalc "R_4050 = if(isnull( R_4049 ) || isnull( R_4033 ), null(),
if(R_4049 <= R_4033 , R_4049 , R_4033 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_4051 = if(isnull(ccaa),R_4050, -1.0)"
g.region n=36.692981567717 s=35.067005005217 e=-3.961957734375 w=-
7.367719453125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_3985
output="C:\Users\GSI\OceanLider\Resultados\96\RI_R_3985.tif"
r.out.gdal input=R_3991
output="C:\Users\GSI\OceanLider\Resultados\96\RI_R_3991.tif"
r.out.gdal input=R_4000
output="C:\Users\GSI\OceanLider\Resultados\96\RI_R_4000.tif"
r.out.gdal input=R_4006
output="C:\Users\GSI\OceanLider\Resultados\96\RI_R_4006.tif"
r.out.gdal input=R_4012
output="C:\Users\GSI\OceanLider\Resultados\96\RI_R_4012.tif"
r.out.gdal input=R_4021
output="C:\Users\GSI\OceanLider\Resultados\96\RI_R_4021.tif"
r.out.gdal input=R_4027
output="C:\Users\GSI\OceanLider\Resultados\96\RI_R_4027.tif"
r.out.gdal input=R_4033
output="C:\Users\GSI\OceanLider\Resultados\96\RI_R_4033.tif"
r.out.gdal input=R_4042
output="C:\Users\GSI\OceanLider\Resultados\96\RI_R_4042.tif"
r.out.gdal input=R_4051
output="C:\Users\GSI\OceanLider\Resultados\96\RF_R_4051.tif"

```

El método de razonamiento que se ha escogido para la combinación de los subcriterios: **optimista (máximo de los valores).**

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=acuicultura
output=R_5308
v.to.rast input=R_5308 layer=1 type=point,line,area output=R_5309
use=val value=0.0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=acuicultura_buf
output=R_5310
v.to.rast input=R_5310 layer=1 type=point,line,area output=R_5311
use=val value=0.0 rows=4096
r.patch input=R_5309,R_5311 output=R_5312
```

```
r.grow.distance input=R_5312 distance=R_5313 metric=euclidean
r.mapcalc "R_5314 = R_5313 * 100"
r.mapcalc "R_5315 = if(isnull( R_5314 ) , null(), if((R_5314 <= 1 ||
R_5314 >= 99999), 0.0, if((R_5314 > 1 && R_5314 < 4), eval((R_5314 -
1)/(4 -1 )), if((R_5314 > 99999 && R_5314 < 99999 ), eval((99999 -
R_5314 )/(99999 - 99999)), 1.0))))"
r.patch input=R_5312,R_5315 output=R_5316
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=cria_moluscos
output=R_5317
v.to.rast input=R_5317 layer=1 type=point,line,area output=R_5318
use=val value=0.0 rows=4096
r.grow.distance input=R_5318 distance=R_5319 metric=euclidean
r.mapcalc "R_5320 = R_5319 * 100"
r.mapcalc "R_5321 = if(isnull( R_5320 ) , null(), if((R_5320 <= 1 ||
R_5320 >= 99999), 0.0, if((R_5320 > 1 && R_5320 < 4), eval((R_5320 -
1)/(4 -1 )), if((R_5320 > 99999 && R_5320 < 99999 ), eval((99999 -
R_5320 )/(99999 - 99999)), 1.0))))"
r.patch input=R_5318,R_5321 output=R_5322
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=almadrabas
output=R_5323
v.to.rast input=R_5323 layer=1 type=point,line,area output=R_5324
use=val value=0.0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=almadrabas_buf
output=R_5325
v.to.rast input=R_5325 layer=1 type=point,line,area output=R_5326
use=val value=0.0 rows=4096
r.patch input=R_5324,R_5326 output=R_5327
r.grow.distance input=R_5327 distance=R_5328 metric=euclidean
r.mapcalc "R_5329 = R_5328 * 100"
r.mapcalc "R_5330 = if(isnull( R_5329 ) , null(), if((R_5329 <= 1 ||
R_5329 >= 99999), 0.0, if((R_5329 > 1 && R_5329 < 4), eval((R_5329 -
1)/(4 -1 )), if((R_5329 > 99999 && R_5329 < 99999 ), eval((99999 -
R_5329 )/(99999 - 99999)), 1.0))))"
r.patch input=R_5327,R_5330 output=R_5331
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=rrmm output=R_5332
v.to.rast input=R_5332 layer=1 type=point,line,area output=R_5333
use=val value=0.0 rows=4096
r.grow.distance input=R_5333 distance=R_5334 metric=euclidean
r.mapcalc "R_5335 = R_5334 * 100"
r.mapcalc "R_5336 = if(isnull( R_5335 ) , null(), if((R_5335 <= 1 ||
R_5335 >= 99999), 0.0, if((R_5335 > 1 && R_5335 < 4), eval((R_5335 -
1)/(4 -1 )), if((R_5335 > 99999 && R_5335 < 99999 ), eval((99999 -
R_5335 )/(99999 - 99999)), 1.0))))"
r.patch input=R_5333,R_5336 output=R_5337
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=caladeros output=R_5338
v.to.rast input=R_5338 layer=1 type=point,line,area output=R_5339
use=val value=0.0 rows=4096
r.grow.distance input=R_5339 distance=R_5340 metric=euclidean
r.mapcalc "R_5341 = R_5340 * 100"
```

```
r.mapcalc "R_5342 = if(isnull( R_5341 ) , null(), if((R_5341 <= 1 || R_5341 >= 99999), 0.0, if((R_5341 > 1 && R_5341 < 4), eval((R_5341 - 1)/(4 -1 )), if((R_5341 > 99999 && R_5341 < 99999 ), eval((99999 - R_5341)/(99999 - 99999)), 1.0))))"
r.patch input=R_5339,R_5342 output=R_5343
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost port=5432 user=postgres password=968XCdPh" layer=viveros output=R_5344
v.to.rast input=R_5344 layer=1 type=point,line,area output=R_5345
use=val value=0.0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost port=5432 user=postgres password=968XCdPh" layer=viveros_buf output=R_5346
v.to.rast input=R_5346 layer=1 type=point,line,area output=R_5347
use=val value=0.0 rows=4096
r.patch input=R_5345,R_5347 output=R_5348
r.grow.distance input=R_5348 distance=R_5349 metric=euclidean
r.mapcalc "R_5350 = R_5349 * 100"
r.mapcalc "R_5351 = if(isnull( R_5350 ) , null(), if((R_5350 <= 1 || R_5350 >= 99999), 0.0, if((R_5350 > 1 && R_5350 < 4), eval((R_5350 - 1)/(4 -1 )), if((R_5350 > 99999 && R_5350 < 99999 ), eval((99999 - R_5350)/(99999 - 99999)), 1.0))))"
r.patch input=R_5348,R_5351 output=R_5352
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost port=5432 user=postgres password=968XCdPh" layer=zpips output=R_5353
v.to.rast input=R_5353 layer=1 type=point,line,area output=R_5354
use=val value=0.0 rows=4096
r.grow.distance input=R_5354 distance=R_5355 metric=euclidean
r.mapcalc "R_5356 = R_5355 * 100"
r.mapcalc "R_5357 = if(isnull( R_5356 ) , null(), if((R_5356 <= 1 || R_5356 >= 99999), 0.0, if((R_5356 > 1 && R_5356 < 4), eval((R_5356 - 1)/(4 -1 )), if((R_5356 > 99999 && R_5356 < 99999 ), eval((99999 - R_5356)/(99999 - 99999)), 1.0))))"
r.patch input=R_5354,R_5357 output=R_5358
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost port=5432 user=postgres password=968XCdPh" layer=arrecifes output=R_5359
v.to.rast input=R_5359 layer=1 type=point,line,area output=R_5360
use=val value=0.0 rows=4096
r.grow.distance input=R_5360 distance=R_5361 metric=euclidean
r.mapcalc "R_5362 = R_5361 * 100"
r.mapcalc "R_5363 = if(isnull( R_5362 ) , null(), if((R_5362 <= 1 || R_5362 >= 99999), 0.0, if((R_5362 > 1 && R_5362 < 4), eval((R_5362 - 1)/(4 -1 )), if((R_5362 > 99999 && R_5362 < 99999 ), eval((99999 - R_5362)/(99999 - 99999)), 1.0))))"
r.patch input=R_5360,R_5363 output=R_5364
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost port=5432 user=postgres password=968XCdPh" layer=piscifactoria output=R_5365
v.to.rast input=R_5365 layer=1 type=point,line,area output=R_5366
use=val value=0.0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost port=5432 user=postgres password=968XCdPh" layer=piscifactoria_buf output=R_5367
v.to.rast input=R_5367 layer=1 type=point,line,area output=R_5368
use=val value=0.0 rows=4096
r.patch input=R_5366,R_5368 output=R_5369
```

```

r.grow.distance input=R_5369 distance=R_5370 metric=euclidean
r.mapcalc "R_5371 = R_5370 * 100"
r.mapcalc "R_5372 = if(isnull( R_5371 ) , null(), if((R_5371 <= 1 ||
R_5371 >= 99999), 0.0, if((R_5371 > 1 && R_5371 < 4), eval((R_5371 -
1)/(4 -1 )), if((R_5371 > 99999 && R_5371 < 99999 ), eval((99999 -
R_5371 )/(99999 - 99999)), 1.0))))"
r.patch input=R_5369,R_5372 output=R_5373
r.mapcalc "R_5374 = if(isnull( R_5316 ) || isnull( R_5322 ), null(),
if(R_5316 >= R_5322 , R_5316 , R_5322 ))"
r.mapcalc "R_5375 = if(isnull( R_5374 ) || isnull( R_5322 ), null(),
if(R_5374 >= R_5322 , R_5374 , R_5322 ))"
r.mapcalc "R_5376 = if(isnull( R_5375 ) || isnull( R_5331 ), null(),
if(R_5375 >= R_5331 , R_5375 , R_5331 ))"
r.mapcalc "R_5377 = if(isnull( R_5376 ) || isnull( R_5337 ), null(),
if(R_5376 >= R_5337 , R_5376 , R_5337 ))"
r.mapcalc "R_5378 = if(isnull( R_5377 ) || isnull( R_5343 ), null(),
if(R_5377 >= R_5343 , R_5377 , R_5343 ))"
r.mapcalc "R_5379 = if(isnull( R_5378 ) || isnull( R_5352 ), null(),
if(R_5378 >= R_5352 , R_5378 , R_5352 ))"
r.mapcalc "R_5380 = if(isnull( R_5379 ) || isnull( R_5358 ), null(),
if(R_5379 >= R_5358 , R_5379 , R_5358 ))"
r.mapcalc "R_5381 = if(isnull( R_5380 ) || isnull( R_5364 ), null(),
if(R_5380 >= R_5364 , R_5380 , R_5364 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_5382 = if(isnull(ccaa),R_5381, -1.0)"
g.region n=37.026966210937 s=35.137317773437 e=-3.786176484375 w=-
7.323774140625 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_5316
output="C:\Users\GSI\OceanLider\Resultados\97\RI_R_5316.tif"
r.out.gdal input=R_5322
output="C:\Users\GSI\OceanLider\Resultados\97\RI_R_5322.tif"
r.out.gdal input=R_5331
output="C:\Users\GSI\OceanLider\Resultados\97\RI_R_5331.tif"
r.out.gdal input=R_5337
output="C:\Users\GSI\OceanLider\Resultados\97\RI_R_5337.tif"
r.out.gdal input=R_5343
output="C:\Users\GSI\OceanLider\Resultados\97\RI_R_5343.tif"
r.out.gdal input=R_5352
output="C:\Users\GSI\OceanLider\Resultados\97\RI_R_5352.tif"
r.out.gdal input=R_5358
output="C:\Users\GSI\OceanLider\Resultados\97\RI_R_5358.tif"
r.out.gdal input=R_5364
output="C:\Users\GSI\OceanLider\Resultados\97\RI_R_5364.tif"
r.out.gdal input=R_5373
output="C:\Users\GSI\OceanLider\Resultados\97\RI_R_5373.tif"
r.out.gdal input=R_5382
output="C:\Users\GSI\OceanLider\Resultados\97\RF_R_5382.tif"

```



2.3.6. Combinación 14


Combinación de tres subcriterios del **criterio de usos pesqueros** (de exclusión parcial) por el método de razonamiento con los consiguientes pesos: *subcriterio arrecifes* con un peso de **0.6**, *subcriterio caladeros* con un peso de **0.3** y *subcriterio zonas de producción de moluscos y de otros invertebrados marinos* con un peso de **0.1**.

Se le deja la libertad al usuario para configurar el valor de idoneidad para las áreas que intervienen en cada subcriterio y se le muestra uno por defecto dependiendo del tipo de dispositivo. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.


El tipo de dispositivo al que esté ligado el escenario: **undimotriz- fuera costa (flotante)**.

Los parámetros establecidos en cada subcriterio:


✓ *Subcriterio Arrecifes*

<input checked="" type="checkbox"/> Arrecifes 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.6

✓ *Subcriterio Caladeros*

<input checked="" type="checkbox"/> Caladeros 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.3

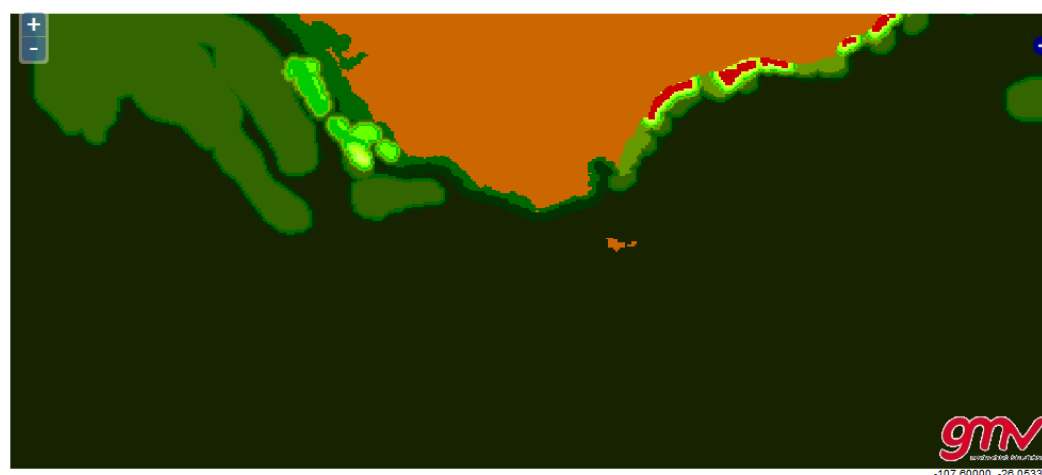
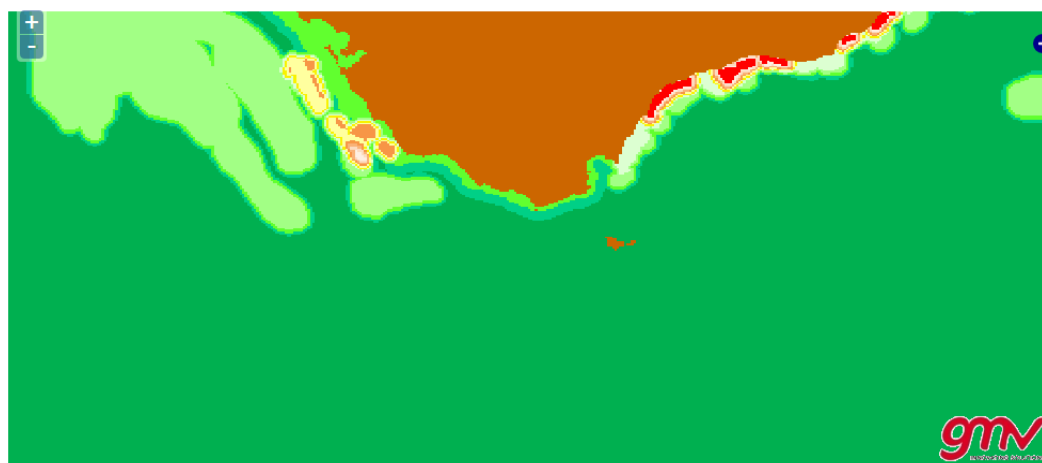
✓ *Subcriterio Zonas de producción de moluscos y de otros invertebrados marinos*

<input checked="" type="checkbox"/> Zonas de producción de moluscos y de otros invertebrados marinos 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.1

SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84

Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

```
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=cria_moluscos
output=R_2366
v.to.rast input=R_2366 layer=1 type=point,line,area output=R_2367
use=val value=0.0 rows=4096
r.grow.distance input=R_2367 distance=R_2368 metric=euclidean
r.mapcalc "R_2369 = R_2368 * 100"
r.mapcalc "R_2370 = if(isnull( R_2369 ) , null(), if((R_2369 <= 1 ||
R_2369 >= 99999), 0.0, if((R_2369 > 1 && R_2369 < 4), eval((R_2369 -
1)/(4 -1)), if((R_2369 > 99999 && R_2369 < 99999), eval((99999 -
R_2369)/(99999 - 99999)), 1.0))))"
r.patch input=R_2367,R_2370 output=R_2371
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=arrecifes output=R_2372
v.to.rast input=R_2372 layer=1 type=point,line,area output=R_2373
use=val value=0.0 rows=4096
r.grow.distance input=R_2373 distance=R_2374 metric=euclidean
r.mapcalc "R_2375 = R_2374 * 100"
r.mapcalc "R_2376 = if(isnull( R_2375 ) , null(), if((R_2375 <= 1 ||
R_2375 >= 99999), 0.0, if((R_2375 > 1 && R_2375 < 4), eval((R_2375 -
1)/(4 -1)), if((R_2375 > 99999 && R_2375 < 99999), eval((99999 -
R_2375)/(99999 - 99999)), 1.0))))"
r.patch input=R_2373,R_2376 output=R_2377
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=caladeros output=R_2378
v.to.rast input=R_2378 layer=1 type=point,line,area output=R_2379
use=val value=0.0 rows=4096
r.grow.distance input=R_2379 distance=R_2380 metric=euclidean
r.mapcalc "R_2381 = R_2380 * 100"
r.mapcalc "R_2382 = if(isnull( R_2381 ) , null(), if((R_2381 <= 1 ||
R_2381 >= 99999), 0.0, if((R_2381 > 1 && R_2381 < 4), eval((R_2381 -
1)/(4 -1)), if((R_2381 > 99999 && R_2381 < 99999), eval((99999 -
R_2381)/(99999 - 99999)), 1.0))))"
r.patch input=R_2379,R_2382 output=R_2383
r.mapcalc "R_2384 = 0.1 * R_2371 + 0.6 * R_2377 + 0.3 * R_2383 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_2385 = if(isnull(ccaa),R_2384, -1.0)"
g.region n=36.649036255217 s=35.154895630217 e=-3.918012421875 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_2371
output="C:\Users\GSI\OceanLider\Resultados\98\RI_R_2371.tif"
r.out.gdal input=R_2377
output="C:\Users\GSI\OceanLider\Resultados\98\RI_R_2377.tif"
r.out.gdal input=R_2383
output="C:\Users\GSI\OceanLider\Resultados\98\RI_R_2383.tif"
r.out.gdal input=R_2385
output="C:\Users\GSI\OceanLider\Resultados\98\RF_R_2385.tif"
```



2.3.7. Combinación 15

Combinación de tres criterios medioambientales: **criterio de espacios protegidos** (de exclusión parcial), **criterio de biodiversidad** (de exclusión parcial) y **criterio de usos pesqueros** (de exclusión parcial).

En el **criterio de espacios protegidos** intervienen dos subcriterios combinados con el método **pesimista**: *subcriterio reservas de la biosfera* y *subcriterio lugares de importancia comunitaria*.

En el **criterio de biodiversidad** intervienen dos subcriterios combinados con el método **pesimista**: *subcriterio áreas importantes para las aves* y *subcriterio zonas propuestas para realizar el inventario de la biodiversidad marina*.


En el **criterio de usos pesqueros** intervienen dos subcriterios combinados con el método **pesimista**: *subcriterio arrecifes* y *subcriterio zonas de producción de moluscos y de otros invertebrados marinos*.

Se le deja la libertad al usuario para configurar el valor de idoneidad para las áreas que intervienen en cada subcriterio y se le muestra uno por defecto dependiendo del tipo de dispositivo. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.


El tipo de dispositivo al que esté ligado el escenario: **undimotriz- fuera costa (flotante)**.

Los parámetros establecidos en cada subcriterio:

- ✓ *Subcriterio reservas de la biosfera*

<input checked="" type="checkbox"/> Reservas de la biosfera 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

- ✓ *Subcriterio lugares de importancia comunitaria*

<input checked="" type="checkbox"/> Lugares de Importancia Comunitaria 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.2

- ✓ *Subcriterio áreas importantes para las aves*

Áreas importantes para las aves

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.6

✓ *Subcriterio zonas propuestas para realizar el inventario de la biodiversidad marina*

 Zonas propuestas para realizar el Inventario de la biodiversidad marina

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.3

✓ *Subcriterio arrecifes*

 Arrecifes

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *Subcriterio zonas de producción de moluscos y de otros invertebrados marinos*

Zonas de producción de moluscos y de otros invertebrados marinos 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

El método de razonamiento que se ha escogido para la combinación de los tres criterios: **más que pesimista (multiplicación de valores).**

```

SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ibas output=R_4704
v.to.rast input=R_4704 layer=1 type=point,line,area output=R_4705
use=val value=0.0 rows=4096
r.grow.distance input=R_4705 distance=R_4706 metric=euclidean
r.mapcalc "R_4707 = R_4706 * 100"
r.mapcalc "R_4708 = if(isnull( R_4707 ), null(), if((R_4707 <= 1 ||
R_4707 >= 99999), 0.0, if((R_4707 > 1 && R_4707 < 4), eval((R_4707 -
1)/(4 -1 )), if((R_4707 > 99999 && R_4707 < 99999 ), eval((99999 -
R_4707)/(99999 - 99999)), 1.0)))"
r.patch input=R_4705,R_4708 output=R_4709
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ibm output=R_4710
v.to.rast input=R_4710 layer=1 type=point,line,area output=R_4711
use=val value=0.0 rows=4096
r.grow.distance input=R_4711 distance=R_4712 metric=euclidean
r.mapcalc "R_4713 = R_4712 * 100"
r.mapcalc "R_4714 = if(isnull( R_4713 ), null(), if((R_4713 <= 1 ||
R_4713 >= 99999), 0.0, if((R_4713 > 1 && R_4713 < 4), eval((R_4713 -
1)/(4 -1 )), if((R_4713 > 99999 && R_4713 < 99999 ), eval((99999 -
R_4713)/(99999 - 99999)), 1.0)))"
r.patch input=R_4711,R_4714 output=R_4715
r.mapcalc "R_4740 = if(isnull( R_4709 ) || isnull( R_4715 ), null(),
if(R_4709 <= R_4715 , R_4709 , R_4715 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=cria_moluscos
output=R_4716
v.to.rast input=R_4716 layer=1 type=point,line,area output=R_4717
use=val value=0.0 rows=4096
r.grow.distance input=R_4717 distance=R_4718 metric=euclidean
r.mapcalc "R_4719 = R_4718 * 100"
r.mapcalc "R_4720 = if(isnull( R_4719 ), null(), if((R_4719 <= 1 ||
R_4719 >= 99999), 0.0, if((R_4719 > 1 && R_4719 < 4), eval((R_4719 -
1)/(4 -1 )), if((R_4719 > 99999 && R_4719 < 99999 ), eval((99999 -
R_4719)/(99999 - 99999)), 1.0)))"
    
```

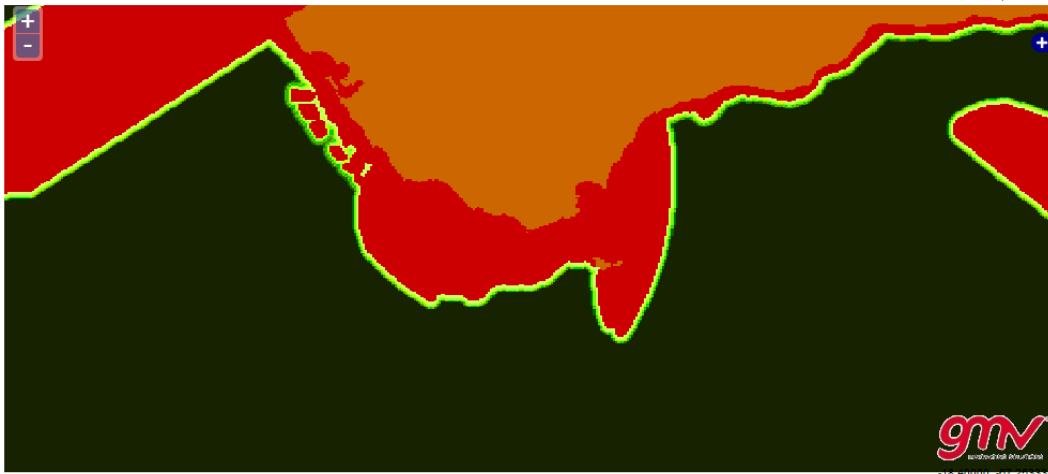
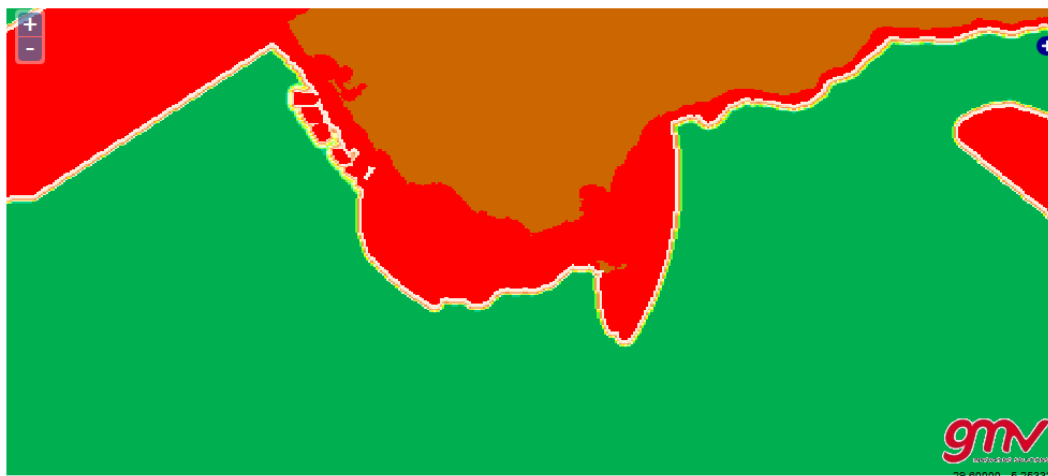
```

r.patch input=R_4717,R_4720 output=R_4721
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=arrecifes output=R_4728
v.to.rast input=R_4728 layer=1 type=point,line,area output=R_4729
use=val value=0.0 rows=4096
r.grow.distance input=R_4729 distance=R_4730 metric=euclidean
r.mapcalc "R_4731 = R_4730 * 100"
r.mapcalc "R_4732 = if(isnull( R_4731 ) , null(), if((R_4731 <= 1 ||
R_4731 >= 99999), 0.0, if((R_4731 > 1 && R_4731 < 4), eval((R_4731 -
1)/(4 -1 )), if((R_4731 > 99999 && R_4731 < 99999 ), eval((99999 -
R_4731)/(99999 - 99999)), 1.0))))"
r.patch input=R_4729,R_4732 output=R_4733
r.mapcalc "R_4741 = if(isnull( R_4709 ) || isnull( R_4715 ), null(),
if(R_4709 <= R_4715 , R_4709 , R_4715 ))"
r.mapcalc "R_4742 = if(isnull( R_4741 ) || isnull( R_4715 ), null(),
if(R_4741 <= R_4715 , R_4741 , R_4715 ))"
r.mapcalc "R_4743 = if(isnull( R_4742 ) || isnull( R_4721 ), null(),
if(R_4742 <= R_4721 , R_4742 , R_4721 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=lics output=R_4722
v.to.rast input=R_4722 layer=1 type=point,line,area output=R_4723
use=val value=0.0 rows=4096
r.grow.distance input=R_4723 distance=R_4724 metric=euclidean
r.mapcalc "R_4725 = R_4724 * 100"
r.mapcalc "R_4726 = if(isnull( R_4725 ) , null(), if((R_4725 <= 1 ||
R_4725 >= 99999), 0.0, if((R_4725 > 1 && R_4725 < 4), eval((R_4725 -
1)/(4 -1 )), if((R_4725 > 99999 && R_4725 < 99999 ), eval((99999 -
R_4725)/(99999 - 99999)), 1.0))))"
r.patch input=R_4723,R_4726 output=R_4727
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mabs output=R_4734
v.to.rast input=R_4734 layer=1 type=point,line,area output=R_4735
use=val value=0.0 rows=4096
r.grow.distance input=R_4735 distance=R_4736 metric=euclidean
r.mapcalc "R_4737 = R_4736 * 100"
r.mapcalc "R_4738 = if(isnull( R_4737 ) , null(), if((R_4737 <= 1 ||
R_4737 >= 99999), 0.0, if((R_4737 > 1 && R_4737 < 4), eval((R_4737 -
1)/(4 -1 )), if((R_4737 > 99999 && R_4737 < 99999 ), eval((99999 -
R_4737)/(99999 - 99999)), 1.0))))"
r.patch input=R_4735,R_4738 output=R_4739
r.mapcalc "R_4744 = if(isnull( R_4709 ) || isnull( R_4715 ), null(),
if(R_4709 <= R_4715 , R_4709 , R_4715 ))"
r.mapcalc "R_4745 = if(isnull( R_4744 ) || isnull( R_4715 ), null(),
if(R_4744 <= R_4715 , R_4744 , R_4715 ))"
r.mapcalc "R_4746 = if(isnull( R_4745 ) || isnull( R_4721 ), null(),
if(R_4745 <= R_4721 , R_4745 , R_4721 ))"
r.mapcalc "R_4747 = if(isnull( R_4746 ) || isnull( R_4733 ), null(),
if(R_4746 <= R_4733 , R_4746 , R_4733 ))"
r.mapcalc "R_4748 = if(isnull( R_4747 ) || isnull( R_4727 ), null(),
if(R_4747 <= R_4727 , R_4747 , R_4727 ))"
r.mapcalc "R_4749 = R_4740 * R_4743 * R_4748 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096

```

Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

```
r.mapcalc "R_4750 = if(isnull(ccaa),R_4749, -1.0)"  
g.region n=36.780872192717 s=35.176868286467 e=-3.808149140625 w=-  
7.411664765625 nsres=300 ewres=300 rows=3224 cols=3453  
r.out.gdal input=R_4709  
output="C:\Users\GSI\OceanLider\Resultados\99\RI_R_4709.tif"  
r.out.gdal input=R_4715  
output="C:\Users\GSI\OceanLider\Resultados\99\RI_R_4715.tif"  
r.out.gdal input=R_4721  
output="C:\Users\GSI\OceanLider\Resultados\99\RI_R_4721.tif"  
r.out.gdal input=R_4733  
output="C:\Users\GSI\OceanLider\Resultados\99\RI_R_4733.tif"  
r.out.gdal input=R_4727  
output="C:\Users\GSI\OceanLider\Resultados\99\RI_R_4727.tif"  
r.out.gdal input=R_4739  
output="C:\Users\GSI\OceanLider\Resultados\99\RI_R_4739.tif"  
r.out.gdal input=R_4750  
output="C:\Users\GSI\OceanLider\Resultados\99\RF_R_4750.tif"
```



El método de razonamiento que se ha escogido para la combinación de los tres criterios: **pesimista (mínimo de valores)**.

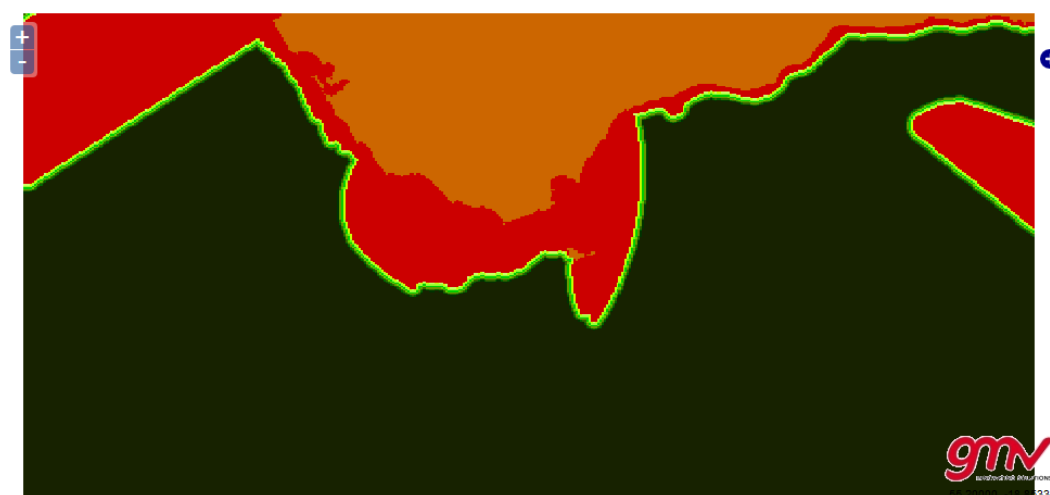
```

SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ibas output=R_5078
v.to.rast input=R_5078 layer=1 type=point,line,area output=R_5079
use=val value=0.0 rows=4096
r.grow.distance input=R_5079 distance=R_5080 metric=euclidean
r.mapcalc "R_5081 = R_5080 * 100"
r.mapcalc "R_5082 = if(isnull( R_5081 ) , null(), if((R_5081 <= 1 ||
R_5081 >= 99999), 0.0, if((R_5081 > 1 && R_5081 < 4), eval((R_5081 -
1)/(4 -1 )), if((R_5081 > 99999 && R_5081 < 99999 ), eval((99999 -
R_5081 )/(99999 - 99999)), 1.0))))"
r.patch input=R_5079,R_5082 output=R_5083
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ibm output=R_5084
v.to.rast input=R_5084 layer=1 type=point,line,area output=R_5085
use=val value=0.0 rows=4096
r.grow.distance input=R_5085 distance=R_5086 metric=euclidean
r.mapcalc "R_5087 = R_5086 * 100"
r.mapcalc "R_5088 = if(isnull( R_5087 ) , null(), if((R_5087 <= 1 ||
R_5087 >= 99999), 0.0, if((R_5087 > 1 && R_5087 < 4), eval((R_5087 -
1)/(4 -1 )), if((R_5087 > 99999 && R_5087 < 99999 ), eval((99999 -
R_5087 )/(99999 - 99999)), 1.0))))"
r.patch input=R_5085,R_5088 output=R_5089
r.mapcalc "R_5114 = if(isnull( R_5083 ) || isnull( R_5089 ), null(),
if(R_5083 <= R_5089 , R_5083 , R_5089 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=cria_moluscos
output=R_5090
v.to.rast input=R_5090 layer=1 type=point,line,area output=R_5091
use=val value=0.0 rows=4096
r.grow.distance input=R_5091 distance=R_5092 metric=euclidean
r.mapcalc "R_5093 = R_5092 * 100"
r.mapcalc "R_5094 = if(isnull( R_5093 ) , null(), if((R_5093 <= 1 ||
R_5093 >= 99999), 0.0, if((R_5093 > 1 && R_5093 < 4), eval((R_5093 -
1)/(4 -1 )), if((R_5093 > 99999 && R_5093 < 99999 ), eval((99999 -
R_5093 )/(99999 - 99999)), 1.0))))"
r.patch input=R_5091,R_5094 output=R_5095
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=arrecifes output=R_5102
v.to.rast input=R_5102 layer=1 type=point,line,area output=R_5103
use=val value=0.0 rows=4096
r.grow.distance input=R_5103 distance=R_5104 metric=euclidean
r.mapcalc "R_5105 = R_5104 * 100"
r.mapcalc "R_5106 = if(isnull( R_5105 ) , null(), if((R_5105 <= 1 ||
R_5105 >= 99999), 0.0, if((R_5105 > 1 && R_5105 < 4), eval((R_5105 -
1)/(4 -1 )), if((R_5105 > 99999 && R_5105 < 99999 ), eval((99999 -
R_5105 )/(99999 - 99999)), 1.0))))"
r.patch input=R_5103,R_5106 output=R_5107

```

```
r.mapcalc "R_5115 = if(isnull( R_5083 ) || isnull( R_5089 ), null(),
if(R_5083 <= R_5089 , R_5083 , R_5089 ))"
r.mapcalc "R_5116 = if(isnull( R_5115 ) || isnull( R_5089 ), null(),
if(R_5115 <= R_5089 , R_5115 , R_5089 ))"
r.mapcalc "R_5117 = if(isnull( R_5116 ) || isnull( R_5095 ), null(),
if(R_5116 <= R_5095 , R_5116 , R_5095 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=lics output=R_5096
v.to.rast input=R_5096 layer=1 type=point,line,area output=R_5097
use=val value=0.0 rows=4096
r.grow.distance input=R_5097 distance=R_5098 metric=euclidean
r.mapcalc "R_5099 = R_5098 * 100"
r.mapcalc "R_5100 = if(isnull( R_5099 ) , null(), if((R_5099 <= 1 ||
R_5099 >= 99999), 0.0, if((R_5099 > 1 && R_5099 < 4), eval((R_5099 -
1)/(4 -1 )), if((R_5099 > 99999 && R_5099 < 99999 ), eval((99999 -
R_5099)/(99999 - 99999)), 1.0)))"
r.patch input=R_5097,R_5100 output=R_5101
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mabs output=R_5108
v.to.rast input=R_5108 layer=1 type=point,line,area output=R_5109
use=val value=0.0 rows=4096
r.grow.distance input=R_5109 distance=R_5110 metric=euclidean
r.mapcalc "R_5111 = R_5110 * 100"
r.mapcalc "R_5112 = if(isnull( R_5111 ) , null(), if((R_5111 <= 1 ||
R_5111 >= 99999), 0.0, if((R_5111 > 1 && R_5111 < 4), eval((R_5111 -
1)/(4 -1 )), if((R_5111 > 99999 && R_5111 < 99999 ), eval((99999 -
R_5111)/(99999 - 99999)), 1.0)))"
r.patch input=R_5109,R_5112 output=R_5113
r.mapcalc "R_5118 = if(isnull( R_5083 ) || isnull( R_5089 ), null(),
if(R_5083 <= R_5089 , R_5083 , R_5089 ))"
r.mapcalc "R_5119 = if(isnull( R_5118 ) || isnull( R_5089 ), null(),
if(R_5118 <= R_5089 , R_5118 , R_5089 ))"
r.mapcalc "R_5120 = if(isnull( R_5119 ) || isnull( R_5095 ), null(),
if(R_5119 <= R_5095 , R_5119 , R_5095 ))"
r.mapcalc "R_5121 = if(isnull( R_5120 ) || isnull( R_5107 ), null(),
if(R_5120 <= R_5107 , R_5120 , R_5107 ))"
r.mapcalc "R_5122 = if(isnull( R_5121 ) || isnull( R_5101 ), null(),
if(R_5121 <= R_5101 , R_5121 , R_5101 ))"
r.mapcalc "R_5123 = if(isnull( R_5114 ) || isnull( R_5117 ), null(),
if(R_5114 <= R_5117 , R_5114 , R_5117 ))"
r.mapcalc "R_5124 = if(isnull( R_5123 ) || isnull( R_5117 ), null(),
if(R_5123 <= R_5117 , R_5123 , R_5117 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_5125 = if(isnull(ccaa),R_5124, -1.0)"
g.region n=36.759998571611 s=35.002186071611 e=-3.698285859375 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_5083
output="C:\Users\GSI\OceanLider\Resultados\100\RI_R_5083.tif"
r.out.gdal input=R_5089
output="C:\Users\GSI\OceanLider\Resultados\100\RI_R_5089.tif"
r.out.gdal input=R_5095
output="C:\Users\GSI\OceanLider\Resultados\100\RI_R_5095.tif"
```

```
r.out.gdal input=R_5107
output="C:\Users\GSI\OceanLider\Resultados\100\RI_R_5107.tif"
r.out.gdal input=R_5101
output="C:\Users\GSI\OceanLider\Resultados\100\RI_R_5101.tif"
r.out.gdal input=R_5113
output="C:\Users\GSI\OceanLider\Resultados\100\RI_R_5113.tif"
r.out.gdal input=R_5125
output="C:\Users\GSI\OceanLider\Resultados\100\RF_R_5125.tif"
```



El método de razonamiento que se ha escogido para la combinación de los tres criterios: **optimista (máximo de los valores)**.

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
```

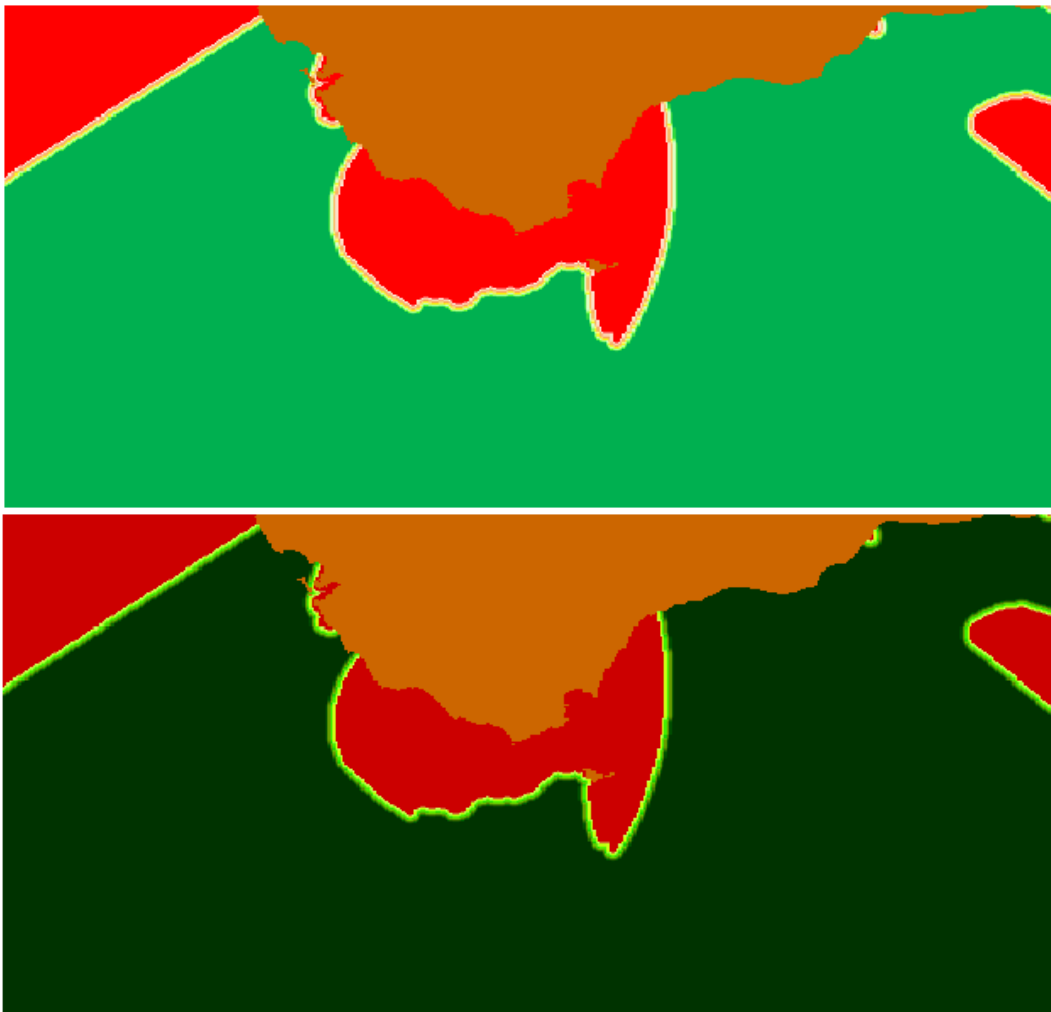


```
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ibas output=R_4799
v.to.rast input=R_4799 layer=1 type=point,line,area output=R_4800
use=val value=0.0 rows=4096
r.grow.distance input=R_4800 distance=R_4801 metric=euclidean
r.mapcalc "R_4802 = R_4801 * 100"
r.mapcalc "R_4803 = if(isnull( R_4802 ) , null(), if((R_4802 <= 1 ||
R_4802 >= 99999), 0.0, if((R_4802 > 1 && R_4802 < 4), eval((R_4802 -
1)/(4 -1 )), if((R_4802 > 99999 && R_4802 < 99999 ), eval((99999 -
R_4802)/(99999 - 99999)), 1.0))))"
r.patch input=R_4800,R_4803 output=R_4804
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=ibm output=R_4805
v.to.rast input=R_4805 layer=1 type=point,line,area output=R_4806
use=val value=0.0 rows=4096
r.grow.distance input=R_4806 distance=R_4807 metric=euclidean
r.mapcalc "R_4808 = R_4807 * 100"
r.mapcalc "R_4809 = if(isnull( R_4808 ) , null(), if((R_4808 <= 1 ||
R_4808 >= 99999), 0.0, if((R_4808 > 1 && R_4808 < 4), eval((R_4808 -
1)/(4 -1 )), if((R_4808 > 99999 && R_4808 < 99999 ), eval((99999 -
R_4808)/(99999 - 99999)), 1.0))))"
r.patch input=R_4806,R_4809 output=R_4810
r.mapcalc "R_4835 = if(isnull( R_4804 ) || isnull( R_4810 ), null(),
if(R_4804 <= R_4810 , R_4804 , R_4810 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=cria_moluscos
output=R_4811
v.to.rast input=R_4811 layer=1 type=point,line,area output=R_4812
use=val value=0.0 rows=4096
r.grow.distance input=R_4812 distance=R_4813 metric=euclidean
r.mapcalc "R_4814 = R_4813 * 100"
r.mapcalc "R_4815 = if(isnull( R_4814 ) , null(), if((R_4814 <= 1 ||
R_4814 >= 99999), 0.0, if((R_4814 > 1 && R_4814 < 4), eval((R_4814 -
1)/(4 -1 )), if((R_4814 > 99999 && R_4814 < 99999 ), eval((99999 -
R_4814)/(99999 - 99999)), 1.0))))"
r.patch input=R_4812,R_4815 output=R_4816
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=arrecifes output=R_4823
v.to.rast input=R_4823 layer=1 type=point,line,area output=R_4824
use=val value=0.0 rows=4096
r.grow.distance input=R_4824 distance=R_4825 metric=euclidean
r.mapcalc "R_4826 = R_4825 * 100"
r.mapcalc "R_4827 = if(isnull( R_4826 ) , null(), if((R_4826 <= 1 ||
R_4826 >= 99999), 0.0, if((R_4826 > 1 && R_4826 < 4), eval((R_4826 -
1)/(4 -1 )), if((R_4826 > 99999 && R_4826 < 99999 ), eval((99999 -
R_4826)/(99999 - 99999)), 1.0))))"
r.patch input=R_4824,R_4827 output=R_4828
r.mapcalc "R_4836 = if(isnull( R_4804 ) || isnull( R_4810 ), null(),
if(R_4804 <= R_4810 , R_4804 , R_4810 ))"
r.mapcalc "R_4837 = if(isnull( R_4836 ) || isnull( R_4810 ), null(),
if(R_4836 <= R_4810 , R_4836 , R_4810 ))"
r.mapcalc "R_4838 = if(isnull( R_4837 ) || isnull( R_4816 ), null(),
if(R_4837 <= R_4816 , R_4837 , R_4816 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=lics output=R_4817
```

```

v.to.rast input=R_4817 layer=1 type=point,line,area output=R_4818
use=val value=0.0 rows=4096
r.grow.distance input=R_4818 distance=R_4819 metric=euclidean
r.mapcalc "R_4820 = R_4819 * 100"
r.mapcalc "R_4821 = if(isnull( R_4820 ) , null(), if((R_4820 <= 1 ||
R_4820 >= 99999), 0.0, if((R_4820 > 1 && R_4820 < 4), eval((R_4820 -
1)/(4 -1 )), if((R_4820 > 99999 && R_4820 < 99999 ), eval((99999 -
R_4820 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4818,R_4821 output=R_4822
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mabs output=R_4829
v.to.rast input=R_4829 layer=1 type=point,line,area output=R_4830
use=val value=0.0 rows=4096
r.grow.distance input=R_4830 distance=R_4831 metric=euclidean
r.mapcalc "R_4832 = R_4831 * 100"
r.mapcalc "R_4833 = if(isnull( R_4832 ) , null(), if((R_4832 <= 1 ||
R_4832 >= 99999), 0.0, if((R_4832 > 1 && R_4832 < 4), eval((R_4832 -
1)/(4 -1 )), if((R_4832 > 99999 && R_4832 < 99999 ), eval((99999 -
R_4832 )/(99999 - 99999)), 1.0))))"
r.patch input=R_4830,R_4833 output=R_4834
r.mapcalc "R_4839 = if(isnull( R_4804 ) || isnull( R_4810 ) , null(),
if(R_4804 <= R_4810 , R_4804 , R_4810 ))"
r.mapcalc "R_4840 = if(isnull( R_4839 ) || isnull( R_4810 ) , null(),
if(R_4839 <= R_4810 , R_4839 , R_4810 ))"
r.mapcalc "R_4841 = if(isnull( R_4840 ) || isnull( R_4816 ) , null(),
if(R_4840 <= R_4816 , R_4840 , R_4816 ))"
r.mapcalc "R_4842 = if(isnull( R_4841 ) || isnull( R_4828 ) , null(),
if(R_4841 <= R_4828 , R_4841 , R_4828 ))"
r.mapcalc "R_4843 = if(isnull( R_4842 ) || isnull( R_4822 ) , null(),
if(R_4842 <= R_4822 , R_4842 , R_4822 ))"
r.mapcalc "R_4844 = if(isnull( R_4835 ) || isnull( R_4838 ) , null(),
if(R_4835 >= R_4838 , R_4835 , R_4838 ))"
r.mapcalc "R_4845 = if(isnull( R_4844 ) || isnull( R_4838 ) , null(),
if(R_4844 >= R_4838 , R_4844 , R_4838 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_4846 = if(isnull(ccaa),R_4845, -1.0)"
g.region n=36.738025915361 s=35.112049352861 e=-3.720258515625 w=-
7.367719453125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_4804
output="C:\Users\GSI\OceanLider\Resultados\101\RI_R_4804.tif"
r.out.gdal input=R_4810
output="C:\Users\GSI\OceanLider\Resultados\101\RI_R_4810.tif"
r.out.gdal input=R_4816
output="C:\Users\GSI\OceanLider\Resultados\101\RI_R_4816.tif"
r.out.gdal input=R_4828
output="C:\Users\GSI\OceanLider\Resultados\101\RI_R_4828.tif"
r.out.gdal input=R_4822
output="C:\Users\GSI\OceanLider\Resultados\101\RI_R_4822.tif"
r.out.gdal input=R_4834
output="C:\Users\GSI\OceanLider\Resultados\101\RI_R_4834.tif"
r.out.gdal input=R_4846
output="C:\Users\GSI\OceanLider\Resultados\101\RF_R_4846.tif"

```



2.4. Combinación de criterios físicos y criterios operacionales


2.4.1. Combinación 16

Combinación del criterio **distancia a la costa** (criterio físico de exclusión parcial) y el criterio **viabilidad de instalación en zonas ocupadas** (criterio operacional de exclusión parcial).

En el criterio operacional **viabilidad de instalación en zonas ocupadas** se ha seleccionado el subcriterio viabilidad de instalación en *zonas ocupadas por las ayudas a la navegación del sistema portuario español*.


Los parámetros que intervienen:

✓ *En el criterio distancia a la costa.*

Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km. 

Valor A	<input type="text" value="8"/>
Valor B	<input type="text" value="9"/>
Valor C	<input type="text" value="20"/>
Valor D	<input type="text" value="25"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

✓ *En el subcriterio viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español.*

Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español. Se aplica una operación difusa sobre la distancia a las zonas ocupadas por las ayudas a la navegación del sistema portuario español. Las medidas se aplican en Km. 

De subida ▾

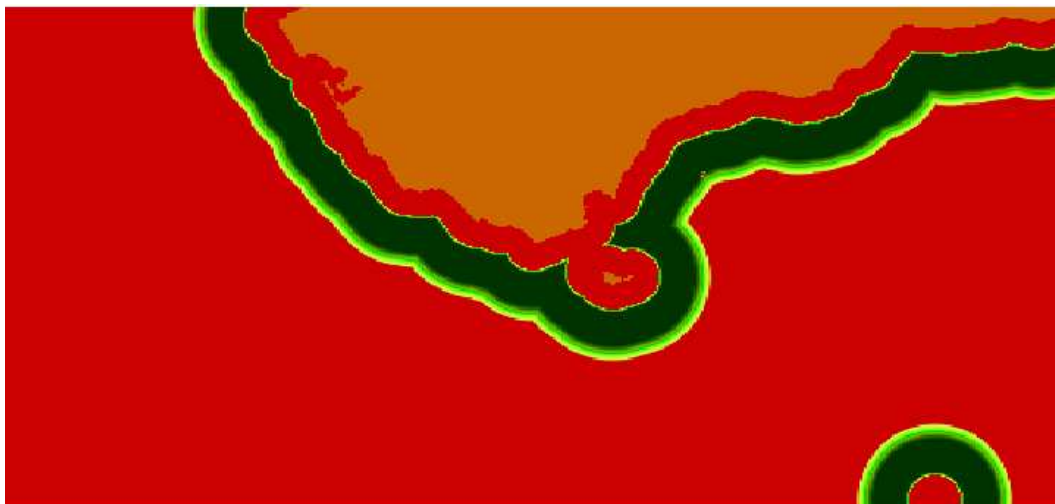
Valor A	<input type="text" value="0.5"/>
Valor B	<input type="text" value="1"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>

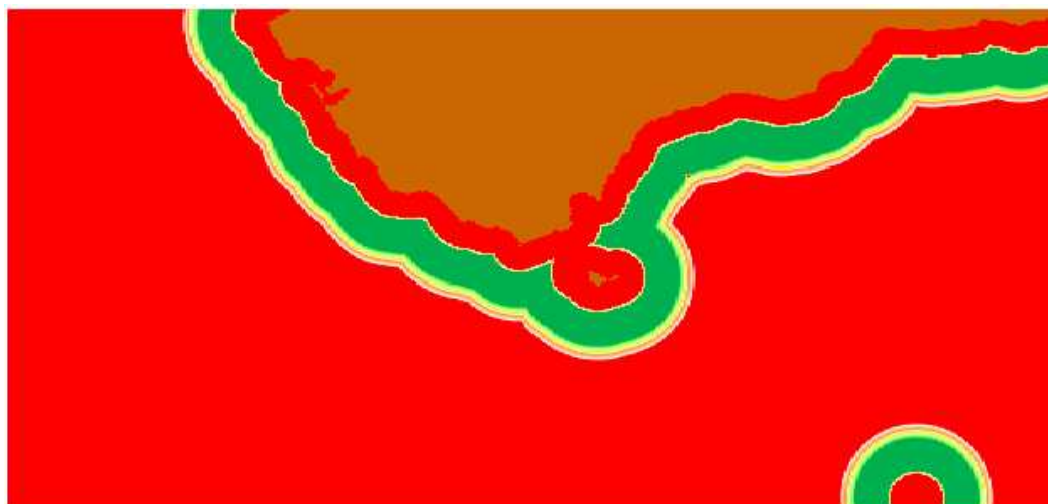
No hace falta configurar los parámetros de la toma de decisión final. Se han escogido criterios físicos y criterios operacionales, sólo interviene una rama en la toma de decisión.

Internamente el motor hace la combinación mediante el mínimo de los valores del resultado de los criterios físicos y el resultado de los criterios operacionales.

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_3799
v.to.rast input=R_3799 layer=1 type=point,line,area output=R_3800
use=val value=1 rows=4096
r.grow.distance input=R_3800 distance=R_3801 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_3802
v.to.rast input=R_3802 layer=1 type=point,line,area output=R_3803
use=val value=0 rows=4096
r.null map=R_3803 null=1
r.mapcalc "R_3804 = R_3801 * R_3803 "
r.mapcalc "R_3805 = R_3804 * 100"
r.mapcalc "R_3806 = if(isnull( R_3805 ), null(), if((R_3805 <= 8 ||
R_3805 >= 25), 0.0, if((R_3805 > 8 && R_3805 < 9), eval((R_3805 - 8)/(9
-8 )), if((R_3805 > 20 && R_3805 < 25 ), eval((25 - R_3805)/(25 - 20)),
1.0)))"
r.null map=R_3806 value=0
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=aton output=R_3794
v.to.rast input=R_3794 layer=1 type=point,line,area output=R_3795
use=val value=0 rows=4096
```

```
r.grow.distance input=R_3795 distance=R_3796 metric=euclidean
r.mapcalc "R_3797 = R_3796 * 100"
r.mapcalc "R_3798 = if(isnull( R_3797 ) , null(), if((R_3797 <= 0.5 ||
R_3797 >= 99999), 0.0, if((R_3797 > 0.5 && R_3797 < 1), eval((R_3797 -
0.5)/(1 -0.5 )), if((R_3797 > 99999 && R_3797 < 99999 ), eval((99999 -
R_3797 )/(99999 - 99999)), 1.0))))"
r.mapcalc "R_3807 = if(isnull( R_3798 ) || isnull( R_3806 ), null(),
if(R_3798 <= R_3806 , R_3798 , R_3806 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_3808 = if(isnull(ccaa),R_3807, -1.0)"
g.region n=36.759998571611 s=35.134022009111 e=-3.852094453125 w=-
7.367719453125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_3806
output="C:\Users\GSI\OceanLider\Resultados\102\RI_R_3806.tif"
r.out.gdal input=R_3798
output="C:\Users\GSI\OceanLider\Resultados\102\RI_R_3798.tif"
r.out.gdal input=R_3808
output="C:\Users\GSI\OceanLider\Resultados\102\RF_R_3808.tif"
```





2.4.2. Combinación 17

Combinación de **criterio distancia a la costa** (criterio físico de exclusión parcial), **criterio profundidad** (criterio físico de exclusión parcial), **criterio supervivencia del dispositivo** (criterio físico de exclusión total) y **criterio viabilidad de instalación en zonas restringidas** (criterio operacional de exclusión parcial).

En el **criterio supervivencia del dispositivo** intervienen los dos subcriterios: *altura de ola significativa* y *velocidad de corriente*. Se ha elegido el método de razonamiento **optimista** para su combinación.

En el **criterio viabilidad de instalación en zonas restringidas** intervienen los subcriterios: *viabilidad de instalación en zonas restringidas para uso militar*, *viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías*, *viabilidad de instalación en zonas restringidas por la presencia de vertidos* y *viabilidad de instalación en zonas restringidas por la existencia de vertidos de material militar*. Por defecto se combinan con el método de razonamiento **pesimista**.

Y el método de razonamiento elegido para la combinación de criterios físicos parciales ha sido el **pesimista**.

Los parámetros que intervienen:

- ✓ *En el subcriterio altura de ola significativa*. Se extrae el valor mayor de ola significativa admitido configurado en el dispositivo con un valor de 2.7 metros
- ✓ *En el subcriterio velocidad de corriente*. Se extrae el valor de velocidad máxima de corriente configurado en el dispositivo con un valor de 1.7 m/s.
- ✓ *En el criterio profundidad*. Se extrae el valor de profundidad máxima y mínima configurado en el dispositivo con un valor de -100 y -20 metros respectivamente.
- ✓ *En el criterio distancia a la costa*.

Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km. 

Valor A	<input type="text" value="8"/>
Valor B	<input type="text" value="9"/>
Valor C	<input type="text" value="20"/>
Valor D	<input type="text" value="25"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

✓ *En el subcriterio viabilidad de instalación en zonas restringidas para uso militar.*

Viabilidad de instalación en zonas restringidas para uso militar. Se aplica una operación difusa sobre la distancia a las zonas restringidas para uso militar. Las medidas se aplican en Km. 

De subida ▾

Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="2"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>

✓ *En el subcriterio viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías.*

Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Las medidas se aplican en Km. 

Normal ▾

Valor A	<input type="text" value="1.5"/>
Valor B	<input type="text" value="3"/>
Valor C	<input type="text" value="6"/>
Valor D	<input type="text" value="15"/>

✓ *En el subcriterio viabilidad de instalación en zonas restringidas por la presencia de vertidos.*

Viabilidad de instalación en zonas restringidas por la presencia de vertidos. Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de vertidos. Las medidas se aplican en Km. 

De subida ▾

Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="2"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>

✓ *En el subcriterio viabilidad de instalación en zonas restringidas por la existencia de vertidos de material militar.*

Viabilidad de instalación en zonas restringidas por la existencia de vertidos de material militar. Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de vertidos de material militar. Las medidas se aplican en Km.

De subida ▾	
Valor A	1
Valor B	2
Valor C	99999
Valor D	99999

No hace falta configurar los parámetros de la toma de decisión final. Se han escogido criterios físicos y criterios operacionales, sólo interviene una rama en la toma de decisión.

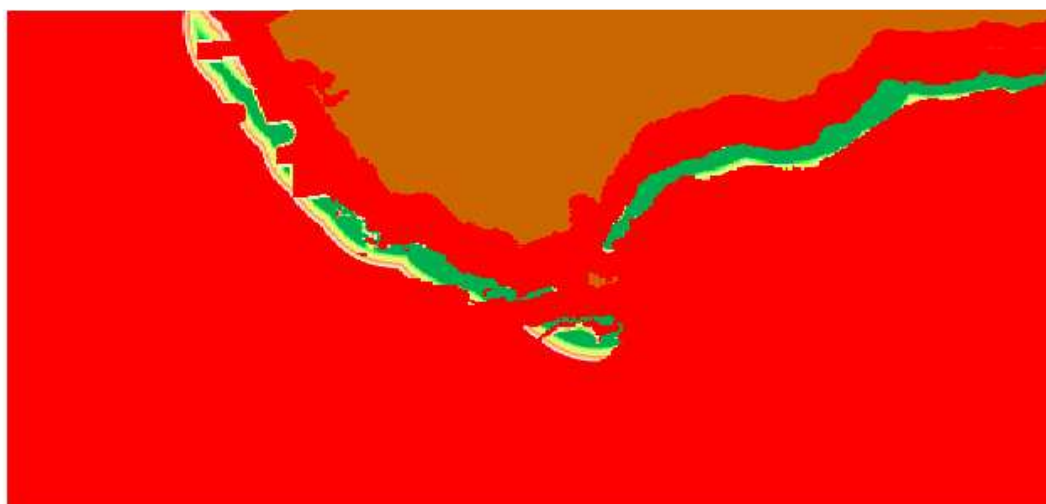
Internamente el motor hace la combinación mediante el mínimo de los valores del resultado de los criterios físicos y el resultado de los criterios operacionales.

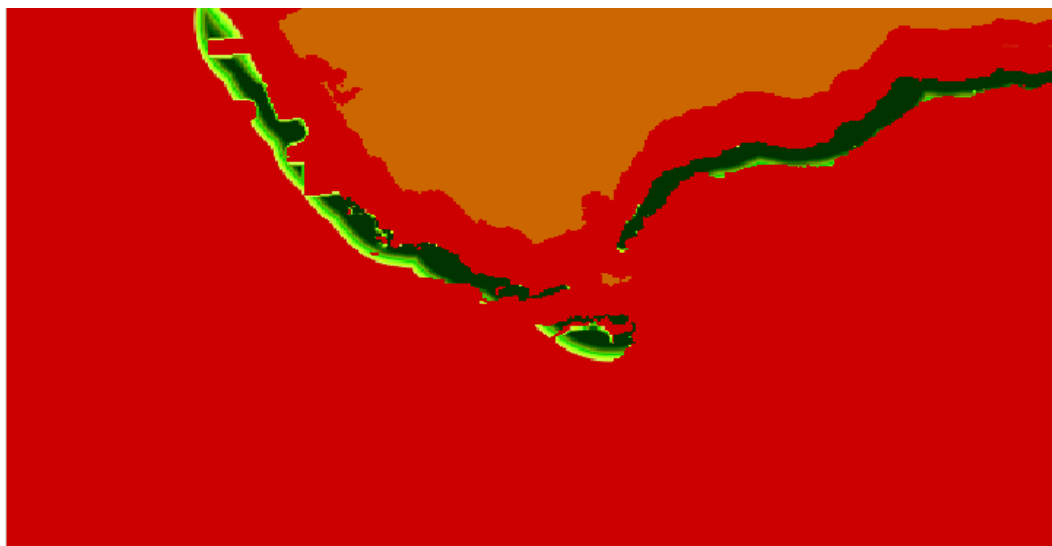
```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\unionhsmedia.tif"
output=R_3809
r.reclass input=R_3809 output=R_3810
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_3810.txt"
r.mapcalc "R_3811 = R_3810"
r.null map=R_3811 null=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\medviv.tif"
output=R_3817
r.reclass input=R_3817 output=R_3818
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_3818.txt"
r.mapcalc "R_3819 = R_3818"
r.null map=R_3819 null=0
r.mapcalc "R_3845 = if(isnull( R_3811 ) || isnull( R_3819 ), null(),
if(R_3811 >= R_3819 , R_3811 , R_3819 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_3825
v.to.rast input=R_3825 layer=1 type=point,line,area output=R_3826
use=val value=1 rows=4096
r.grow.distance input=R_3826 distance=R_3827 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_3828
v.to.rast input=R_3828 layer=1 type=point,line,area output=R_3829
use=val value=0 rows=4096
r.null map=R_3829 null=1
r.mapcalc "R_3830 = R_3827 * R_3829 "
r.mapcalc "R_3831 = R_3830 * 100"
r.mapcalc "R_3832 = if(isnull( R_3831 ) , null(), if((R_3831 <= 8 ||
R_3831 >= 25), 0.0, if((R_3831 > 8 && R_3831 < 9), eval((R_3831 - 8)/(9
-8)), if((R_3831 > 20 && R_3831 < 25), eval((25 - R_3831)/(25 - 20)),
1.0)))"
r.null map=R_3832 value=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_3838
r.mapcalc "R_3839 = if(isnull( R_3838 ) , null(), if((R_3838 <= -105.0
|| R_3838 >= -15.0), 0.0, if((R_3838 > -105.0 && R_3838 < -100.0),
```



```
eval((R_3838 - -105.0)/(-100.0 --105.0 )), if((R_3838 > -20.0 && R_3838 < -15.0 ), eval((-15.0 - R_3838 )/(-15.0 - -20.0)), 1.0)))"
r.mapcalc "R_3846 = if(isnull( R_3832 ) || isnull( R_3839 ), null(), if(R_3832 <= R_3839 , R_3832 , R_3839 ))"
r.mapcalc "R_3847 = if(isnull( R_3846 ) || isnull( R_3839 ), null(), if(R_3846 <= R_3839 , R_3846 , R_3839 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432 user=postgres password=968XCdPh" layer=vertido_material_militar output=R_3812
v.to.rast input=R_3812 layer=1 type=point,line,area output=R_3813 use=val value=0 rows=4096
r.grow.distance input=R_3813 distance=R_3814 metric=euclidean
r.mapcalc "R_3815 = R_3814 * 100"
r.mapcalc "R_3816 = if(isnull( R_3815 ) , null(), if((R_3815 <= 1 || R_3815 >= 99999), 0.0, if((R_3815 > 1 && R_3815 < 2), eval((R_3815 - 1)/(2 -1 )), if((R_3815 > 99999 && R_3815 < 99999 ), eval((99999 - R_3815)/(99999 - 99999)), 1.0)))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432 user=postgres password=968XCdPh" layer=vertidos output=R_3820
v.to.rast input=R_3820 layer=1 type=point,line,area output=R_3821 use=val value=0 rows=4096
r.grow.distance input=R_3821 distance=R_3822 metric=euclidean
r.mapcalc "R_3823 = R_3822 * 100"
r.mapcalc "R_3824 = if(isnull( R_3823 ) , null(), if((R_3823 <= 1 || R_3823 >= 99999), 0.0, if((R_3823 > 1 && R_3823 < 2), eval((R_3823 - 1)/(2 -1 )), if((R_3823 > 99999 && R_3823 < 99999 ), eval((99999 - R_3823)/(99999 - 99999)), 1.0)))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432 user=postgres password=968XCdPh" layer=militar output=R_3833
v.to.rast input=R_3833 layer=1 type=point,line,area output=R_3834 use=val value=0 rows=4096
r.grow.distance input=R_3834 distance=R_3835 metric=euclidean
r.mapcalc "R_3836 = R_3835 * 100"
r.mapcalc "R_3837 = if(isnull( R_3836 ) , null(), if((R_3836 <= 1 || R_3836 >= 99999), 0.0, if((R_3836 > 1 && R_3836 < 2), eval((R_3836 - 1)/(2 -1 )), if((R_3836 > 99999 && R_3836 < 99999 ), eval((99999 - R_3836)/(99999 - 99999)), 1.0)))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432 user=postgres password=968XCdPh" layer=zona_cables output=R_3840
v.to.rast input=R_3840 layer=1 type=point,line,area output=R_3841 use=val value=0 rows=4096
r.grow.distance input=R_3841 distance=R_3842 metric=euclidean
r.mapcalc "R_3843 = R_3842 * 100"
r.mapcalc "R_3844 = if(isnull( R_3843 ) , null(), if((R_3843 <= 1.5 || R_3843 >= 15), 0.0, if((R_3843 > 1.5 && R_3843 < 3), eval((R_3843 - 1.5)/(3 -1.5 )), if((R_3843 > 6 && R_3843 < 15 ), eval((15 - R_3843)/(15 - 6)), 1.0)))"
r.mapcalc "R_3848 = if(isnull( R_3816 ) || isnull( R_3824 ), null(), if(R_3816 <= R_3824 , R_3816 , R_3824 ))"
r.mapcalc "R_3849 = if(isnull( R_3848 ) || isnull( R_3824 ), null(), if(R_3848 <= R_3824 , R_3848 , R_3824 ))"
r.mapcalc "R_3850 = if(isnull( R_3849 ) || isnull( R_3837 ), null(), if(R_3849 <= R_3837 , R_3849 , R_3837 ))"
r.mapcalc "R_3851 = if(isnull( R_3850 ) || isnull( R_3847 ), null(), if(R_3850 <= R_3847 , R_3850 , R_3847 ))"
```

```
v.in.ogr -o "dsn=PG:dbname=BBDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_3852 = if(isnull(ccaa),R_3851, -1.0)"
g.region n=36.803943884111 s=34.980213415361 e=-3.874067109375 w=-
7.257856171875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_3811
output="C:\Users\GSI\OceanLider\Resultados\103\RI_R_3811.tif"
r.out.gdal input=R_3819
output="C:\Users\GSI\OceanLider\Resultados\103\RI_R_3819.tif"
r.out.gdal input=R_3832
output="C:\Users\GSI\OceanLider\Resultados\103\RI_R_3832.tif"
r.out.gdal input=R_3839
output="C:\Users\GSI\OceanLider\Resultados\103\RI_R_3839.tif"
r.out.gdal input=R_3816
output="C:\Users\GSI\OceanLider\Resultados\103\RI_R_3816.tif"
r.out.gdal input=R_3824
output="C:\Users\GSI\OceanLider\Resultados\103\RI_R_3824.tif"
r.out.gdal input=R_3837
output="C:\Users\GSI\OceanLider\Resultados\103\RI_R_3837.tif"
r.out.gdal input=R_3844
output="C:\Users\GSI\OceanLider\Resultados\103\RI_R_3844.tif"
r.out.gdal input=R_3852
output="C:\Users\GSI\OceanLider\Resultados\103\RF_R_3852.tif"
```






2.5. Combinación de criterios físicos y criterios de recurso energético

2.5.1. Combinación 18

Combinación del **criterio distancia a la costa** (criterio físico de exclusión parcial) y **criterio de recurso energético- densidad** (criterio de recurso energético de exclusión parcial).

Los parámetros que intervienen:

- ✓ *En el criterio distancia a la costa.*

Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km. 

Valor A	<input type="text" value="8"/>
Valor B	<input type="text" value="9"/>
Valor C	<input type="text" value="20"/>
Valor D	<input type="text" value="25"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

- ✓ *En el criterio de recurso energético- densidad.* Se aplica por tener el tipo de dispositivo undimotriz ligado al escenario. Hay que tener en cuenta que al introducir valores máximos mayores que 0 el resultado tenderá a 0.

Recurso energético

Peso criterio:

Densidad. Las unidades del valor máximo es en KWh/m.

Valor Máximo

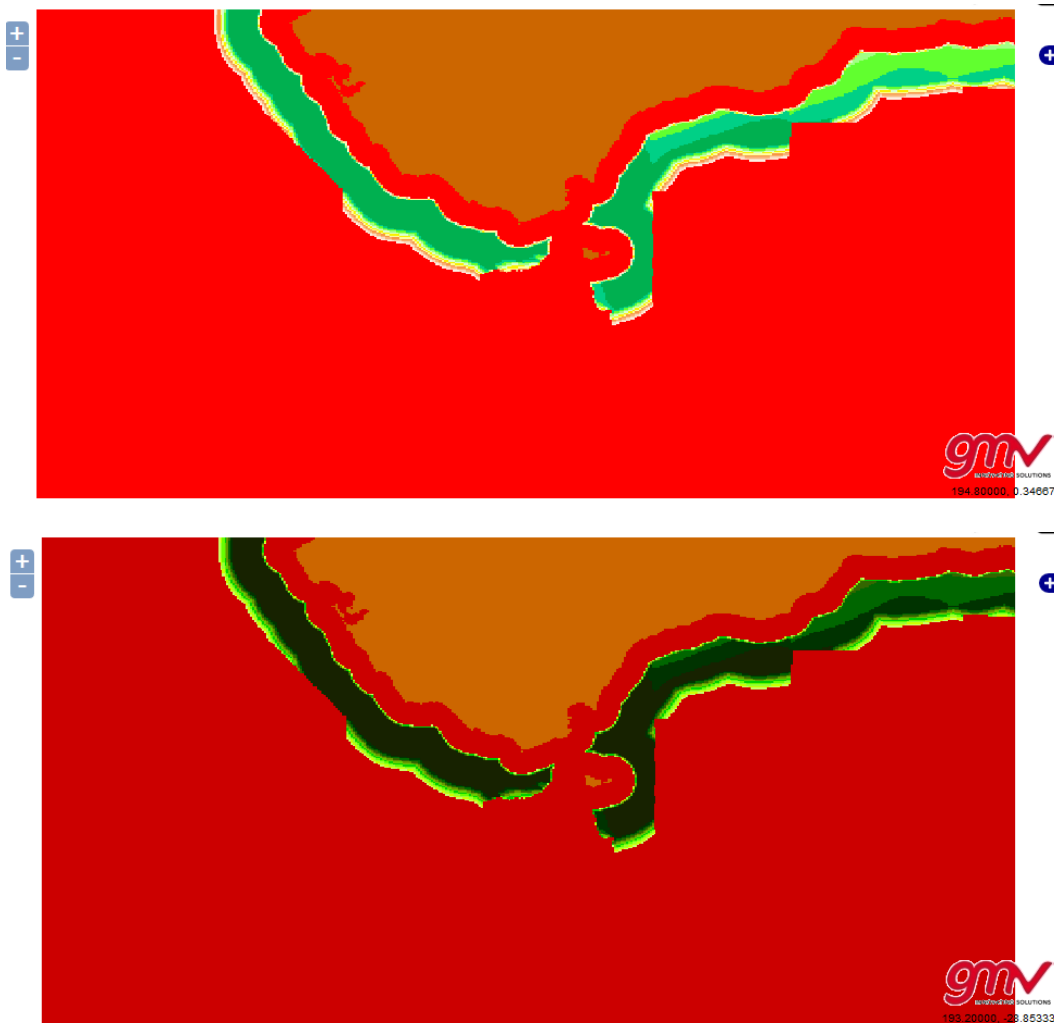
No hace falta configurar los parámetros de la toma de decisión final. Se han escogido criterios físicos y criterios de recurso energético, sólo interviene una rama en la toma de decisión.

Internamente el motor hace la combinación mediante el mínimo de los valores del resultado de los criterios físicos y el resultado de los criterios de recurso energético.

```

SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_3853
v.to.rast input=R_3853 layer=1 type=point,line,area output=R_3854
use=val value=1 rows=4096
r.grow.distance input=R_3854 distance=R_3855 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_3856
v.to.rast input=R_3856 layer=1 type=point,line,area output=R_3857
use=val value=0 rows=4096
r.null map=R_3857 null=1
r.mapcalc "R_3858 = R_3855 * R_3857 "
r.mapcalc "R_3859 = R_3858 * 100"
r.mapcalc "R_3860 = if(isnull( R_3859 ), null(), if((R_3859 <= 8 ||
R_3859 >= 25), 0.0, if((R_3859 > 8 && R_3859 < 9), eval((R_3859 - 8)/(9
-8)), if((R_3859 > 20 && R_3859 < 25 ), eval((25 - R_3859)/(25 - 20)),
1.0)))"
r.null map=R_3860 value=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\pw.tif" output=R_3861
r.mapcalc "R_3862= if(R_3861 >= 0 && R_3861 < 0.3, eval(R_3861/0.3), 1)"
r.null map=R_3862 null=0
r.mapcalc "R_3863 = if(isnull( R_3862 ) || isnull( R_3860 ), null(),
if(R_3862 <= R_3860 , R_3862 , R_3860 )"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_3864 = if(isnull(ccaa),R_3863, -1.0)"
g.region n=36.781971227861 s=35.002186071611 e=-3.808149140625 w=-
7.367719453125 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_3860
output="C:\Users\GSI\OceanLider\Resultados\104\RI_R_3860.tif"
r.out.gdal input=R_3862
output="C:\Users\GSI\OceanLider\Resultados\104\RI_R_3862.tif"
r.out.gdal input=R_3864
output="C:\Users\GSI\OceanLider\Resultados\104\RF_R_3864.tif"

```



2.5.2. Combinación 19


Combinación de los criterios: **supervivencia del dispositivo** (criterio físico de exclusión total), **profundidad** (criterio físico de exclusión parcial), distancia a la costa (criterio físico de exclusión parcial) y **densidad** (criterio de recurso energético de exclusión parcial).

En el **criterio supervivencia del dispositivo** intervienen los dos subcriterios: *altura de ola significativa* y *velocidad de corriente*. Se ha elegido el método de razonamiento **optimista** para su combinación.

El método de razonamiento elegido para la combinación de criterios físicos parciales ha sido el **pesimista**.

Los parámetros que intervienen:

- ✓ *En el subcriterio altura de ola significativa.* Se extrae el valor mayor de ola significativa admitido configurado en el dispositivo con un valor de 2.7 metros
- ✓ *En el subcriterio velocidad de corriente.* Se extrae el valor de velocidad máxima de corriente configurado en el dispositivo con un valor de 1.7 m/s.
- ✓ *En el criterio profundidad.* Se extrae el valor de profundidad máxima y mínima configurado en el dispositivo con un valor de -100 y -20 metros respectivamente.
- ✓ *En el criterio distancia a la costa.*


Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km. 

Valor A	<input type="text" value="8"/>
Valor B	<input type="text" value="9"/>
Valor C	<input type="text" value="20"/>
Valor D	<input type="text" value="25"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

- ✓ *En el criterio recurso energético-densidad.* Se aplica por tener el tipo de dispositivo undimotriz ligado al escenario. Hay que tener en cuenta que al introducir valores máximos mayores que 0 el resultado tenderá a 0.

Recurso energético

Peso criterio:

Densidad. Las unidades del valor máximo es en KWh/m. 

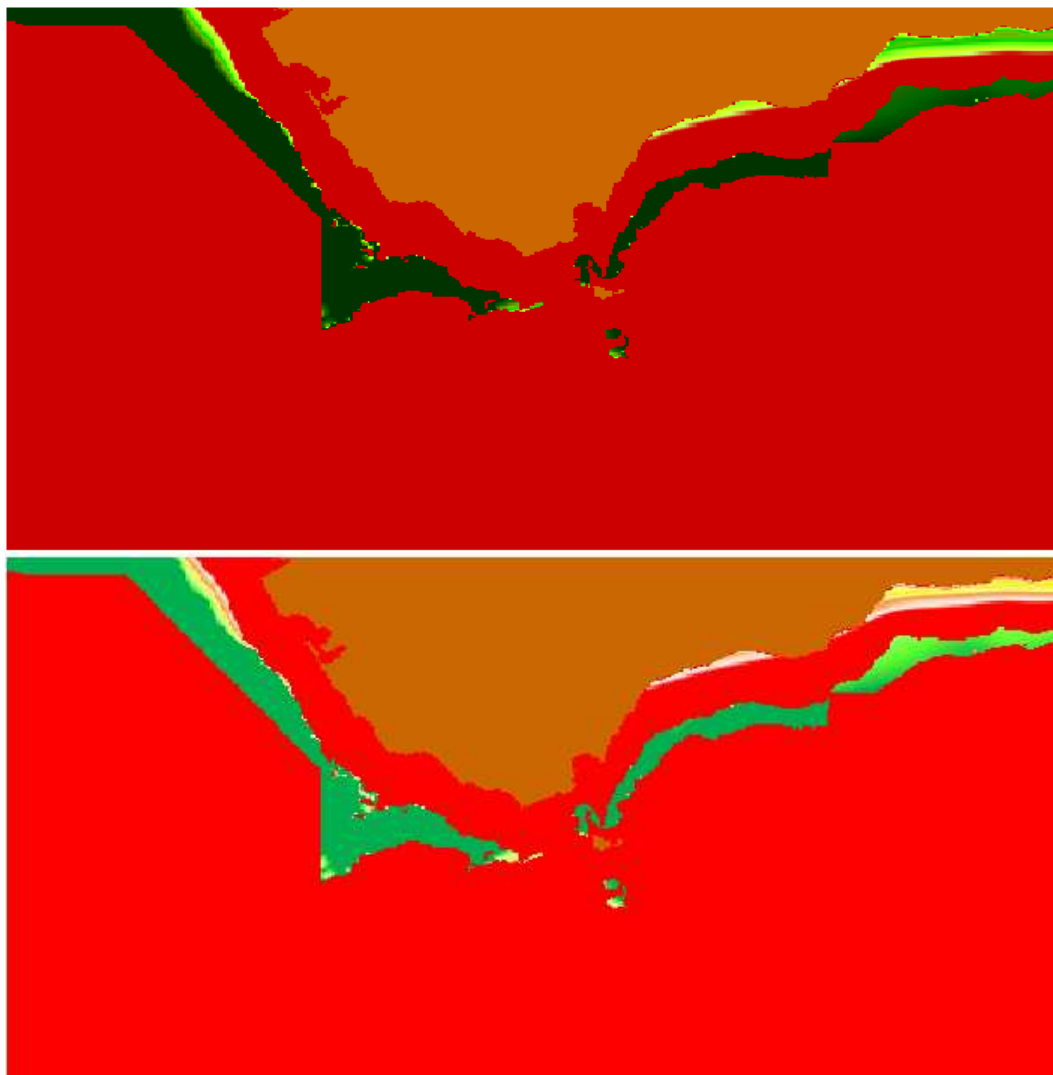
Valor Máximo

No hace falta configurar los parámetros de la toma de decisión final. Se han escogido criterios físicos y criterios de recurso energético, sólo interviene una rama en la toma de decisión.

Internamente el motor hace la combinación mediante el mínimo de los valores del resultado de los criterios físicos y el resultado de los criterios de recurso energético.

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\medviv.tif"
output=R_3865
r.reclass input=R_3865 output=R_3866
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_3866.txt"
r.mapcalc "R_3867 = R_3866"
r.null map=R_3867 null=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\unionhsmedia.tif"
output=R_3868
```

```
r.reclass input=R_3868 output=R_3869
rules="C:\Users\GSI\OceanLider\Resultados\reclass_R_3869.txt"
r.mapcalc "R_3870 = R_3869"
r.null map=R_3870 null=0
r.mapcalc "R_3875 = if(isnull( R_3867 ) || isnull( R_3870 ), null(),
if(R_3867 >= R_3870 , R_3867 , R_3870 ))"
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_3871
r.mapcalc "R_3872 = if(isnull( R_3871 ) , null(), if((R_3871 <= -105.0
|| R_3871 >= -15.0), 0.0, if((R_3871 > -105.0 && R_3871 < -100.0),
eval((R_3871 - -105.0)/(-100.0 --105.0 )), if((R_3871 > -20.0 && R_3871
< -15.0 ) , eval((-15.0 - R_3871 )/(-15.0 - -20.0)), 1.0)))"
r.mapcalc "R_3876 = if(isnull( R_3872 ) || isnull( R_3875 ), null(),
if(R_3872 <= R_3875 , R_3872 , R_3875 ))"
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\pw.tif" output=R_3873
r.mapcalc "R_3874= if(R_3873 >= 0 && R_3873 < 0.3, eval(R_3873/0.3), 1)"
r.null map=R_3874 null=0
r.mapcalc "R_3877 = if(isnull( R_3874 ) || isnull( R_3876 ), null(),
if(R_3874 <= R_3876 , R_3874 , R_3876 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_3878 = if(isnull(ccaa),R_3877, -1.0)"
g.region n=36.824817505217 s=34.935169067717 e=-3.720258515625 w=-
7.433637421875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_3867
output="C:\Users\GSI\OceanLider\Resultados\105\RI_R_3867.tif"
r.out.gdal input=R_3870
output="C:\Users\GSI\OceanLider\Resultados\105\RI_R_3870.tif"
r.out.gdal input=R_3872
output="C:\Users\GSI\OceanLider\Resultados\105\RI_R_3872.tif"
r.out.gdal input=R_3874
output="C:\Users\GSI\OceanLider\Resultados\105\RI_R_3874.tif"
r.out.gdal input=R_3878
output="C:\Users\GSI\OceanLider\Resultados\105\RF_R_3878.tif"
```



2.6. Combinación de criterios físicos y criterios medioambientales


2.6.1. Combinación 20

Combinación del criterio **distancia a la costa** (criterio físico de exclusión parcial) y **espacios protegidos** (criterio medio ambiental de exclusión parcial).

En el criterio de **espacios protegidos** intervienen los subcriterios: *reservas de la biosfera*, *lugares de importancia comunitaria* y *zonas de especial protección para las aves*. El método de razonamiento elegido para su combinación es el **pesimista**.


Los parámetros que intervienen:

- ✓ En el criterio distancia a la costa.

Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km. 


Valor A	<input type="text" value="8"/>
Valor B	<input type="text" value="9"/>
Valor C	<input type="text" value="20"/>
Valor D	<input type="text" value="25"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

- ✓ En el subcriterio de reservas de la biosfera

Reservas de la biosfera 


Valor de idoneidad	<input type="text" value="0.0"/>
Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="4"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

- ✓ En el subcriterio de lugares de importancia comunitaria

Lugares de Importancia Comunitaria 

Valor de idoneidad	<input type="text" value="0.0"/>
Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="4"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

- ✓ En el subcriterio de zonas de especial protección para las aves

Zonas de Especial Protección para las Aves 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

Se le deja la libertad al usuario para configurar el valor de idoneidad para las áreas que intervienen en cada subcriterio medioambiental y se le muestra uno por defecto dependiendo del tipo de dispositivo. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

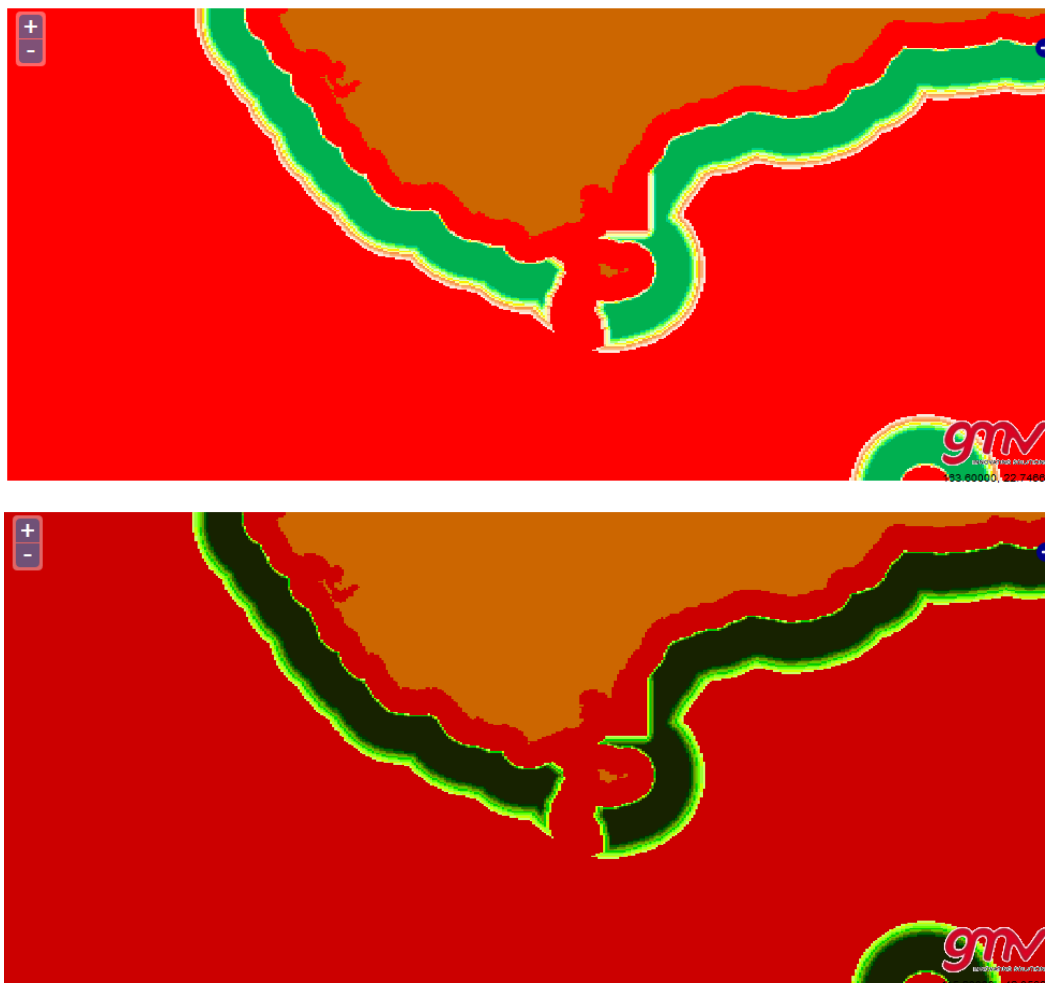
El tipo de dispositivo al que esté ligado el escenario: **undimotriz- fuera costa (flotante)**.

No hace falta configurar los parámetros de la toma de decisión final. Se han escogido criterios físicos y criterios medioambientales, sólo interviene una rama en la toma de decisión.

Internamente el motor hace la combinación mediante el mínimo de los valores del resultado de los criterios físicos y el resultado de los criterios medioambientales.

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_4058
v.to.rast input=R_4058 layer=1 type=point,line,area output=R_4059
use=val value=1 rows=4096
r.grow.distance input=R_4059 distance=R_4060 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_4061
v.to.rast input=R_4061 layer=1 type=point,line,area output=R_4062
use=val value=0 rows=4096
r.null map=R_4062 null=1
r.mapcalc "R_4063 = R_4060 * R_4062 "
r.mapcalc "R_4064 = R_4063 * 100"
r.mapcalc "R_4065 = if(isnull( R_4064 ), null(), if((R_4064 <= 8 ||
R_4064 >= 25), 0.0, if((R_4064 > 8 && R_4064 < 9), eval((R_4064 - 8)/(9
-8 )), if((R_4064 > 20 && R_4064 < 25 ), eval((25 - R_4064)/(25 - 20)),
1.0)))"
r.null map=R_4065 value=0
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mabs output=R_4052
```

```
v.to.rast input=R_4052 layer=1 type=point,line,area output=R_4053
use=val value=0.0 rows=4096
r.grow.distance input=R_4053 distance=R_4054 metric=euclidean
r.mapcalc "R_4055 = R_4054 * 100"
r.mapcalc "R_4056 = if(isnull( R_4055 ) , null(), if((R_4055 <= 1 ||
R_4055 >= 99999), 0.0, if((R_4055 > 1 && R_4055 < 4), eval((R_4055 -
1)/(4 -1 )), if((R_4055 > 99999 && R_4055 < 99999 ), eval((99999 -
R_4055)/(99999 - 99999)), 1.0))))"
r.patch input=R_4053,R_4056 output=R_4057
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=lics output=R_4066
v.to.rast input=R_4066 layer=1 type=point,line,area output=R_4067
use=val value=0.0 rows=4096
r.grow.distance input=R_4067 distance=R_4068 metric=euclidean
r.mapcalc "R_4069 = R_4068 * 100"
r.mapcalc "R_4070 = if(isnull( R_4069 ) , null(), if((R_4069 <= 1 ||
R_4069 >= 99999), 0.0, if((R_4069 > 1 && R_4069 < 4), eval((R_4069 -
1)/(4 -1 )), if((R_4069 > 99999 && R_4069 < 99999 ), eval((99999 -
R_4069)/(99999 - 99999)), 1.0))))"
r.patch input=R_4067,R_4070 output=R_4071
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepas output=R_4072
v.to.rast input=R_4072 layer=1 type=point,line,area output=R_4073
use=val value=0.0 rows=4096
r.grow.distance input=R_4073 distance=R_4074 metric=euclidean
r.mapcalc "R_4075 = R_4074 * 100"
r.mapcalc "R_4076 = if(isnull( R_4075 ) , null(), if((R_4075 <= 1 ||
R_4075 >= 99999), 0.0, if((R_4075 > 1 && R_4075 < 4), eval((R_4075 -
1)/(4 -1 )), if((R_4075 > 99999 && R_4075 < 99999 ), eval((99999 -
R_4075)/(99999 - 99999)), 1.0))))"
r.patch input=R_4073,R_4076 output=R_4077
r.mapcalc "R_4078 = if(isnull( R_4057 ) || isnull( R_4071 ) , null(),
if(R_4057 <= R_4071 , R_4057 , R_4071 ))"
r.mapcalc "R_4079 = if(isnull( R_4078 ) || isnull( R_4071 ) , null(),
if(R_4078 <= R_4071 , R_4078 , R_4071 ))"
r.mapcalc "R_4080 = if(isnull( R_4079 ) || isnull( R_4065 ) , null(),
if(R_4079 <= R_4065 , R_4079 , R_4065 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_4081 = if(isnull(ccaa),R_4080, -1.0)"
g.region n=36.758899536467 s=35.198840942717 e=-3.874067109375 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_4065
output="C:\Users\GSI\OceanLider\Resultados\106\RI_R_4065.tif"
r.out.gdal input=R_4057
output="C:\Users\GSI\OceanLider\Resultados\106\RI_R_4057.tif"
r.out.gdal input=R_4071
output="C:\Users\GSI\OceanLider\Resultados\106\RI_R_4071.tif"
r.out.gdal input=R_4077
output="C:\Users\GSI\OceanLider\Resultados\106\RI_R_4077.tif"
r.out.gdal input=R_4081
output="C:\Users\GSI\OceanLider\Resultados\106\RF_R_4081.tif"
```



2.7. Combinación de criterios físicos, criterios operacionales y criterios medioambientales

2.7.1. Combinación 21

Combinación del criterio **profundidad** (criterio físico de exclusión parcial), **distancia a la costa** (criterio físico de exclusión parcial), **mantenimiento de la posición** (criterio operacional de exclusión parcial), **viabilidad de instalación en zonas ocupadas** (criterio operacional de exclusión parcial), **viabilidad de instalación en zonas restringidas** (criterio operacional de exclusión parcial), y **usos pesqueros** (criterio medio ambiental de exclusión parcial).

En el criterio **mantenimiento de la posición** intervienen los subcriterios: *facilidad de anclaje al fondo* (*composición del sustrato marino*) y *facilidad de anclaje al fondo* (*morfología del fondo*).

marino). El método de razonamiento por defecto para su combinación es el **más que pesimista** (multiplicación de los valores).

En el criterio **viabilidad de instalación en zonas ocupadas** intervienen los subcriterios: *viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español* y *viabilidad de instalación en zonas designadas como dominio público portuario*. El método de razonamiento por defecto para su combinación es el **pesimista** (mínimo de los valores).

En el criterio **viabilidad de instalación en zonas restringidas** intervienen los subcriterios: *viabilidad de instalación en zonas restringidas para uso militar* y *viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías*. El método de razonamiento por defecto para su combinación es el **pesimista** (mínimo de los valores).

En el criterio de **usos pesqueros** intervienen los subcriterios: *arrecifes, caladeros y zonas de producción de moluscos y otros invertebrados marinos*. El método de razonamiento elegido es el **pesimista** (mínimo de los valores).

El método de razonamiento elegido para la combinación de los criterios físicos parciales es el **pesimista** (mínimo de los valores).

El método de razonamiento elegido para la combinación de los criterios operacionales parciales es el **pesimista** (mínimo de los valores).

Para la evaluación multicriterio, en la toma de decisión final se configuran los pesos:


- Rama de criterios operacionales: 0.25
- Rama de criterios medioambientales: 0.75

Y los pesos de orden:

- Peso de orden primero: 0.5
- Peso de orden segundo: 0.5


Los parámetros que intervienen:

- ✓ Criterio *profundidad*: los parámetros profundidad máxima y mínima: -100 metros y -20 metros configurados en los parámetros del dispositivo ligado al escenario.
- ✓ *En el criterio distancia a la costa*.

Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km. 

Valor A	<input type="text" value="8"/>
Valor B	<input type="text" value="9"/>
Valor C	<input type="text" value="20"/>
Valor D	<input type="text" value="25"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

- ✓ *En el subcriterio de facilidad de anclaje al fondo (composición del sustrato marino):* Se extraen los valores de idoneidad de las zonas configuradas para la capa Geofis dependiendo del tipo de dispositivo elegido (en este escenario se ha configurado un tipo de dispositivo undimotriz- fuera costa (flotante)) y se le deja la libertad al usuario para poder cambiar estos valores a través de la interfaz. Los valores de idoneidad configurados en la interfaz:

Facilidad de anclaje al fondo (composición del sustrato marino) 


Roca	0.5
Vegetado	1
Blando cohesivo	0
Blando no cohesivo	1

- ✓ *En el subcriterio de facilidad de anclaje al fondo (morfología al fondo marino):* Se extraen los valores de idoneidad de las zonas configuradas en la capa Formas dependiendo del tipo de dispositivo elegido (en este escenario se ha configurado un tipo de dispositivo undimotriz- fuera costa (flotante)) y se le deja la libertad al usuario de cambiar los valores según la agrupación para los diferentes tipos. Los valores de idoneidad configurados en la interfaz:

Facilidad de anclaje al fondo (morfología del fondo marino) 

Grupo0	0
Grupo1	1

- ✓ *En el subcriterio viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español.*

Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español. Se aplica una operación difusa sobre la distancia a las zonas ocupadas por las ayudas a la navegación del sistema portuario español. Las medidas se aplican en Km. 

De subida ▾

Valor A	0.5
Valor B	1
Valor C	99999
Valor D	99999

- ✓ *En el subcriterio viabilidad de instalación en zonas designadas como Dominio Público Portuario.*

Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

Viabilidad de instalación en zonas designadas como Dominio Público Portuario. Se aplica una operación difusa sobre la distancia a las zonas designadas como Dominio Público Portuario. Las medidas se aplican en Km.

De subida ▾

Valor A	0
Valor B	1
Valor C	99999
Valor D	99999

✓ En el subcriterio viabilidad de instalación en zonas restringidas para uso militar.

Viabilidad de instalación en zonas restringidas para uso militar. Se aplica una operación difusa sobre la distancia a las zonas restringidas para uso militar. Las medidas se aplican en Km.

De subida ▾

Valor A	1
Valor B	2
Valor C	99999
Valor D	99999

✓ En el subcriterio viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías.

Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Las medidas se aplican en Km.

De subida ▾

Valor A	1.5
Valor B	3
Valor C	99999
Valor D	99999

✓ En el subcriterio arrecifes

Arrecifes

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ En el subcriterio caladeros

<input checked="" type="checkbox"/> Caladeros	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ En el subcriterio zonas de producción de moluscos y otros invertebrados marinos

<input checked="" type="checkbox"/> Zonas de producción de moluscos y de otros invertebrados marinos	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

```

SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_4110
v.to.rast input=R_4110 layer=1 type=point,line,area output=R_4111
use=val value=1 rows=4096
r.grow.distance input=R_4111 distance=R_4112 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_4113
v.to.rast input=R_4113 layer=1 type=point,line,area output=R_4114
use=val value=0 rows=4096
r.null map=R_4114 null=1
r.mapcalc "R_4115 = R_4112 * R_4114 "
r.mapcalc "R_4116 = R_4115 * 100"
r.mapcalc "R_4117 = if(isnull( R_4116 ), null(), if((R_4116 <= 8 ||
R_4116 >= 25), 0.0, if((R_4116 > 8 && R_4116 < 9), eval((R_4116 - 8)/(9
-8 )), if((R_4116 > 20 && R_4116 < 25 ), eval((25 - R_4116)/(25 - 20)),
1.0)))"
r.null map=R_4117 value=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_4128
r.mapcalc "R_4129 = if(isnull( R_4128 ), null(), if((R_4128 <= -105.0
|| R_4128 >= -15.0), 0.0, if((R_4128 > -105.0 && R_4128 < -100.0),

```



```
eval((R_4128 - -105.0)/(-100.0 --105.0 )), if((R_4128 > -20.0 && R_4128
< -15.0 ), eval((-15.0 - R_4128 )/(-15.0 - -20.0)), 1.0)))"
r.mapcalc "R_4146 = if(isnull( R_4117 ) || isnull( R_4129 ), null(),
if(R_4117 <= R_4129 , R_4117 , R_4129 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_4082 layer=geofis
where="Agrup='Roca'"
v.to.rast input=R_4082 layer=1 type=point,line,area output=R_4083
use=val value=0.5 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_4084 layer=geofis
where="Agrup='Vegetado'"
v.to.rast input=R_4084 layer=1 type=point,line,area output=R_4085
use=val value=1 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_4086 layer=geofis
where="Agrup='Blando cohesivo'"
v.to.rast input=R_4086 layer=1 type=point,line,area output=R_4087
use=val value=0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_4088 layer=geofis
where="Agrup='Blando no cohesivo'"
v.to.rast input=R_4088 layer=1 type=point,line,area output=R_4089
use=val value=1 rows=4096
r.patch input=R_4083,R_4085 output=R_4090
r.patch input=R_4087,R_4089 output=R_4091
r.patch input=R_4090,R_4091 output=R_4092
r.null map=R_4092 null=1
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_4130 layer=formas
where="(TIPO='Megaripples') OR (TIPO='Abanico deltaico') OR
(TIPO='Abanico deltaico profundo') OR (TIPO='Caynon submarino') OR
(TIPO='Caynon submarino') OR (TIPO='Llanuras abisales') OR
(TIPO='Morfologia de crestas y valles') OR (TIPO='Megaripples') OR
(TIPO='Ondas de arena') OR (TIPO='Relieve volcanico')"
v.to.rast input=R_4130 layer=1 type=point,line,area output=R_4131
use=val value=0 rows=4096
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" output=R_4132 layer=formas
where="(TIPO='Ripples') OR (TIPO='Marcas de arrastre') OR (TIPO='Talud
continental') OR (TIPO='Afloramiento rocoso') OR (TIPO='Playas
sumergidas') OR (TIPO='Plataforma continental')"
v.to.rast input=R_4132 layer=1 type=point,line,area output=R_4133
use=val value=1 rows=4096
r.patch input=R_4131,R_4133 output=R_4134
r.null map=R_4134 null=1
r.mapcalc "R_4147 = R_4092 * R_4134 "
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=aton output=R_4105
v.to.rast input=R_4105 layer=1 type=point,line,area output=R_4106
use=val value=0 rows=4096
r.grow.distance input=R_4106 distance=R_4107 metric=euclidean
r.mapcalc "R_4108 = R_4107 * 100"
r.mapcalc "R_4109 = if(isnull( R_4108 ) , null(), if((R_4108 <= 0.5 ||
R_4108 >= 99999), 0.0, if((R_4108 > 0.5 && R_4108 < 1), eval((R_4108 -
```

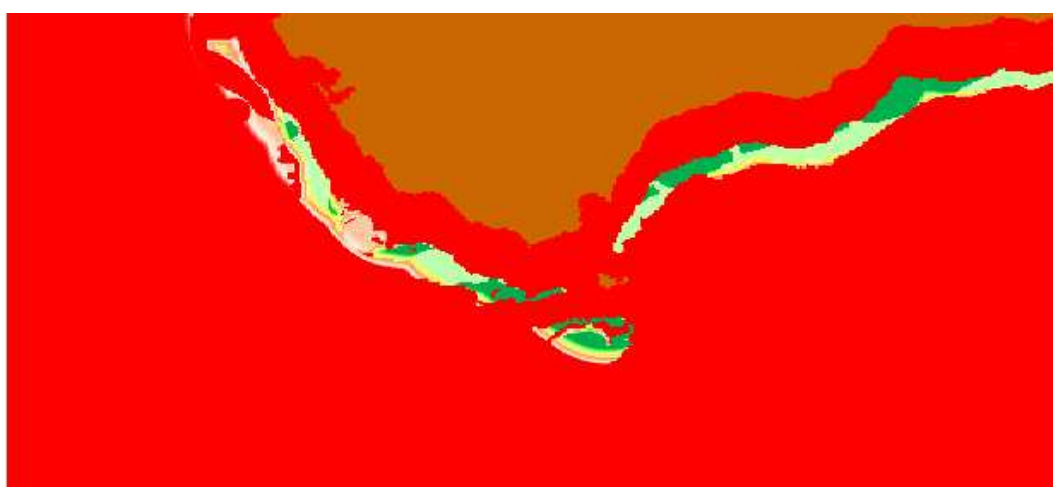
```

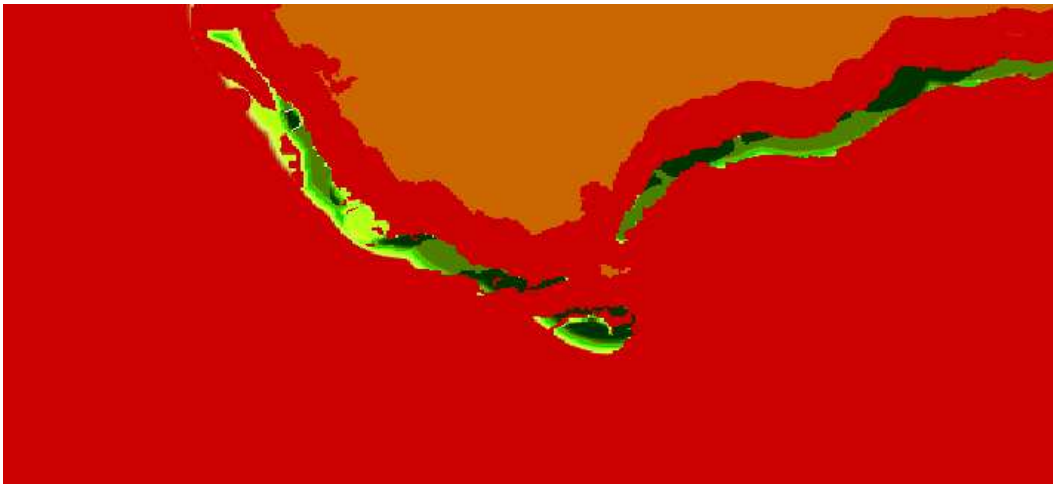
0.5)/(1 -0.5 )), if((R_4108 > 99999 && R_4108 < 99999 ), eval((99999 -
R_4108)/(99999 - 99999)), 1.0)))"
v.in.ogr -o "dsn=PG:dbname=BDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=dpp output=R_4118
v.to.rast input=R_4118 layer=1 type=point,line,area output=R_4119
use=val value=0 rows=4096
r.grow.distance input=R_4119 distance=R_4120 metric=euclidean
r.mapcalc "R_4121 = R_4120 * 100"
r.mapcalc "R_4122 = if(isnull( R_4121 ) , null(), if((R_4121 <= 0 ||
R_4121 >= 99999), 0.0, if((R_4121 > 0 && R_4121 < 1), eval((R_4121 -
0)/(1 -0 )), if((R_4121 > 99999 && R_4121 < 99999 ), eval((99999 -
R_4121)/(99999 - 99999)), 1.0)))"
r.mapcalc "R_4148 = if(isnull( R_4092 ) || isnull( R_4134 ), null(),
if(R_4092 <= R_4134 , R_4092 , R_4134 ))"
r.mapcalc "R_4149 = if(isnull( R_4148 ) || isnull( R_4134 ), null(),
if(R_4148 <= R_4134 , R_4148 , R_4134 ))"
r.mapcalc "R_4150 = if(isnull( R_4149 ) || isnull( R_4109 ), null(),
if(R_4149 <= R_4109 , R_4149 , R_4109 ))"
v.in.ogr -o "dsn=PG:dbname=BDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=militar output=R_4123
v.to.rast input=R_4123 layer=1 type=point,line,area output=R_4124
use=val value=0 rows=4096
r.grow.distance input=R_4124 distance=R_4125 metric=euclidean
r.mapcalc "R_4126 = R_4125 * 100"
r.mapcalc "R_4127 = if(isnull( R_4126 ) , null(), if((R_4126 <= 1 ||
R_4126 >= 99999), 0.0, if((R_4126 > 1 && R_4126 < 2), eval((R_4126 -
1)/(2 -1 )), if((R_4126 > 99999 && R_4126 < 99999 ), eval((99999 -
R_4126)/(99999 - 99999)), 1.0)))"
v.in.ogr -o "dsn=PG:dbname=BDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=zona_cables output=R_4135
v.to.rast input=R_4135 layer=1 type=point,line,area output=R_4136
use=val value=0 rows=4096
r.grow.distance input=R_4136 distance=R_4137 metric=euclidean
r.mapcalc "R_4138 = R_4137 * 100"
r.mapcalc "R_4139 = if(isnull( R_4138 ) , null(), if((R_4138 <= 1.5 ||
R_4138 >= 15), 0.0, if((R_4138 > 1.5 && R_4138 < 3), eval((R_4138 -
1.5)/(3 -1.5 )), if((R_4138 > 6 && R_4138 < 15 ), eval((15 - R_4138
)/(15 - 6)), 1.0)))"
r.mapcalc "R_4151 = if(isnull( R_4092 ) || isnull( R_4134 ), null(),
if(R_4092 <= R_4134 , R_4092 , R_4134 ))"
r.mapcalc "R_4152 = if(isnull( R_4151 ) || isnull( R_4134 ), null(),
if(R_4151 <= R_4134 , R_4151 , R_4134 ))"
r.mapcalc "R_4153 = if(isnull( R_4152 ) || isnull( R_4109 ), null(),
if(R_4152 <= R_4109 , R_4152 , R_4109 ))"
r.mapcalc "R_4154 = if(isnull( R_4153 ) || isnull( R_4122 ), null(),
if(R_4153 <= R_4122 , R_4153 , R_4122 ))"
r.mapcalc "R_4155 = if(isnull( R_4154 ) || isnull( R_4127 ), null(),
if(R_4154 <= R_4127 , R_4154 , R_4127 ))"
r.mapcalc "R_4156 = R_4147 * R_4150 * R_4155 "
r.mapcalc "R_4157 = if(isnull( R_4156 ) || isnull( R_4146 ), null(),
if(R_4156 <= R_4146 , R_4156 , R_4146 ))"
v.in.ogr -o "dsn=PG:dbname=BDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=cria_moluscos
output=R_4093

```

```
v.to.rast input=R_4093 layer=1 type=point,line,area output=R_4094
use=val value=0.0 rows=4096
r.grow.distance input=R_4094 distance=R_4095 metric=euclidean
r.mapcalc "R_4096 = R_4095 * 100"
r.mapcalc "R_4097 = if(isnull( R_4096 ) , null(), if((R_4096 <= 1 ||
R_4096 >= 99999), 0.0, if((R_4096 > 1 && R_4096 < 4), eval((R_4096 -
1)/(4 -1 )), if((R_4096 > 99999 && R_4096 < 99999 ), eval((99999 -
R_4096)/(99999 - 99999)), 1.0))))"
r.patch input=R_4094,R_4097 output=R_4098
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=caladeros output=R_4099
v.to.rast input=R_4099 layer=1 type=point,line,area output=R_4100
use=val value=0.0 rows=4096
r.grow.distance input=R_4100 distance=R_4101 metric=euclidean
r.mapcalc "R_4102 = R_4101 * 100"
r.mapcalc "R_4103 = if(isnull( R_4102 ) , null(), if((R_4102 <= 1 ||
R_4102 >= 99999), 0.0, if((R_4102 > 1 && R_4102 < 4), eval((R_4102 -
1)/(4 -1 )), if((R_4102 > 99999 && R_4102 < 99999 ), eval((99999 -
R_4102)/(99999 - 99999)), 1.0))))"
r.patch input=R_4100,R_4103 output=R_4104
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=arrecifes output=R_4140
v.to.rast input=R_4140 layer=1 type=point,line,area output=R_4141
use=val value=0.0 rows=4096
r.grow.distance input=R_4141 distance=R_4142 metric=euclidean
r.mapcalc "R_4143 = R_4142 * 100"
r.mapcalc "R_4144 = if(isnull( R_4143 ) , null(), if((R_4143 <= 1 ||
R_4143 >= 99999), 0.0, if((R_4143 > 1 && R_4143 < 4), eval((R_4143 -
1)/(4 -1 )), if((R_4143 > 99999 && R_4143 < 99999 ), eval((99999 -
R_4143)/(99999 - 99999)), 1.0))))"
r.patch input=R_4141,R_4144 output=R_4145
r.mapcalc "R_4158 = if(isnull( R_4098 ) || isnull( R_4104 ) , null(),
if(R_4098 <= R_4104 , R_4098 , R_4104 ))"
r.mapcalc "R_4159 = if(isnull( R_4158 ) || isnull( R_4104 ) , null(),
if(R_4158 <= R_4104 , R_4158 , R_4104 ))"
r.mapcalc "R_4160 = if(isnull( R_4159 ) || isnull( R_4146 ) , null(),
if(R_4159 <= R_4146 , R_4159 , R_4146 ))"
r.mapcalc "R_4161 = if(isnull( R_4157 ) || isnull( R_4160 ) ,
null(),if(R_4157 >= R_4160 , eval(((R_4157 * 0.25 * 0.5) + (R_4160 *
0.75 * 0.5 )) / (0.25 * 0.5 + 0.75 * 0.5 )), eval(((R_4157 * R_4157 *
0.5 ) + (R_4160 * 0.75 * 0.5 )) / (0.25 * 0.5 + 0.75 * 0.5))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_4162 = if(isnull(ccaa),R_4161, -1.0)"
g.region n=36.714954223967 s=35.198840942717 e=-3.852094453125 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_4117
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4117.tif"
r.out.gdal input=R_4129
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4129.tif"
r.out.gdal input=R_4092
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4092.tif"
```

```
r.out.gdal input=R_4134
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4134.tif"
r.out.gdal input=R_4109
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4109.tif"
r.out.gdal input=R_4122
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4122.tif"
r.out.gdal input=R_4127
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4127.tif"
r.out.gdal input=R_4139
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4139.tif"
r.out.gdal input=R_4098
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4098.tif"
r.out.gdal input=R_4104
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4104.tif"
r.out.gdal input=R_4145
output="C:\Users\GSI\OceanLider\Resultados\107\RI_R_4145.tif"
r.out.gdal input=R_4162
output="C:\Users\GSI\OceanLider\Resultados\107\RF_R_4162.tif"
```





2.8. Combinación de criterios físicos, criterios operacionales y criterios de recurso energético

2.8.1. Combinación 22

Combinación de criterio **distancia a la costa** (criterio físico de exclusión parcial), **viabilidad de instalación en zonas ocupadas** (criterio operacional de exclusión parcial), **viabilidad de instalación en zonas restringidas** (criterio operacional de exclusión parcial), y **densidad** (criterio de recurso energético de exclusión parcial).

En el criterio **viabilidad de instalación en zonas ocupadas** intervienen los subcriterios: *viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español* y *viabilidad de instalación en zonas designadas como dominio público portuario*. El método de razonamiento por defecto para su combinación es el **pesimista** (mínimo de los valores).

En el criterio **viabilidad de instalación en zonas restringidas** intervienen los subcriterios: *viabilidad de instalación en zonas restringidas para uso militar* y *viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías*. El método de razonamiento por defecto para su combinación es el **pesimista** (mínimo de los valores).

Para combinar los criterios operacionales parciales se ha elegido el método de razonamiento **MLP** con un peso de **0.6** para el *criterio viabilidad de instalación en zonas ocupadas* y **0.4** para el *criterio viabilidad de instalación en zonas restringidas*.

Para la evaluación multicriterio, en la toma de decisión final se configuran los pesos:


- Rama de criterios operacionales: 0.4
- Rama de criterios de recurso energético: 0.6

Y los pesos de orden:

- Peso de orden primero: 0.5
- Peso de orden segundo: 0.5


Los parámetros que intervienen:

- ✓ *En el criterio distancia a la costa.*

Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km. 

Valor A	8
Valor B	9
Valor C	20
Valor D	25
Peso del subcriterio:	0.0

- ✓ Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español.

Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español. Se aplica una operación difusa sobre la distancia a las zonas ocupadas por las ayudas a la navegación del sistema portuario español. Las medidas se aplican en Km. 

De subida ▾

Valor A	0.5
Valor B	1
Valor C	99999
Valor D	99999


- ✓ Viabilidad de instalación en zonas desinadas como Dominio Público Portuario.

Viabilidad de instalación en zonas designadas como Dominio Público Portuario. Se aplica una operación difusa sobre la distancia a las zonas designadas como Dominio Público Portuario. Las medidas se aplican en Km. 

De subida ▾

Valor A	0
Valor B	1
Valor C	99999
Valor D	99999

- ✓ Viabilidad de instalación en zonas restringidas para uso militar.

Viabilidad de instalación en zonas restringidas para uso militar. Se aplica una operación difusa sobre la distancia a las zonas restringidas para uso militar. Las medidas se aplican en Km. 

De subida ▾

Valor A	1
Valor B	2
Valor C	99999
Valor D	99999

- ✓ Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías.

Viabilidad de instalación en zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Se aplica una operación difusa sobre la distancia a las zonas restringidas por la presencia de cables eléctricos submarinos, emisarios o tuberías. Las medidas se aplican en Km. 

De subida ▼

Valor A	1.5
Valor B	3
Valor C	99999
Valor D	99999

- ✓ En el criterio recurso energético-densidad. Se aplica por tener el tipo de dispositivo undimotriz ligado al escenario. Hay que tener en cuenta que al introducir valores máximos mayores que 0 el resultado tenderá a 0.

Recurso energético

Peso criterio:

Densidad. Las unidades del valor máximo es en KWh/m. 

Valor Máximo

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_5533
v.to.rast input=R_5533 layer=1 type=point,line,area output=R_5534
use=val value=1 rows=4096
r.grow.distance input=R_5534 distance=R_5535 metric=euclidean
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_5536
v.to.rast input=R_5536 layer=1 type=point,line,area output=R_5537
use=val value=0 rows=4096
r.null map=R_5537 null=1
r.mapcalc "R_5538 = R_5535 * R_5537 "
r.mapcalc "R_5539 = R_5538 * 100"
r.mapcalc "R_5540 = if(isnull( R_5539 ), null(), if((R_5539 <= 8 ||
R_5539 >= 25), 0.0, if((R_5539 > 8 && R_5539 < 9), eval((R_5539 - 8)/(
-8 )), if((R_5539 > 20 && R_5539 < 25 ), eval((25 - R_5539 )/(25 - 20)),
1.0))))"
r.null map=R_5540 value=0
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=aton output=R_5528
v.to.rast input=R_5528 layer=1 type=point,line,area output=R_5529
use=val value=0 rows=4096
r.grow.distance input=R_5529 distance=R_5530 metric=euclidean
r.mapcalc "R_5531 = R_5530 * 100"
```

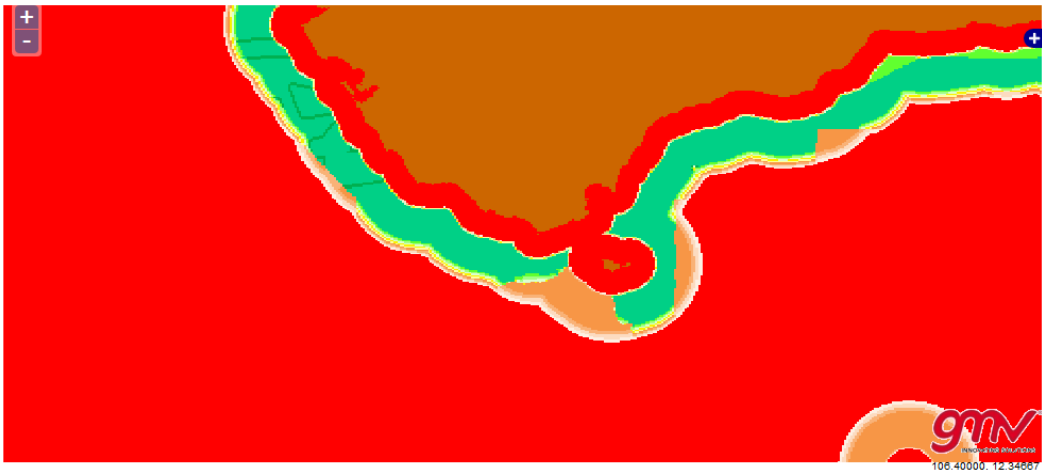
```

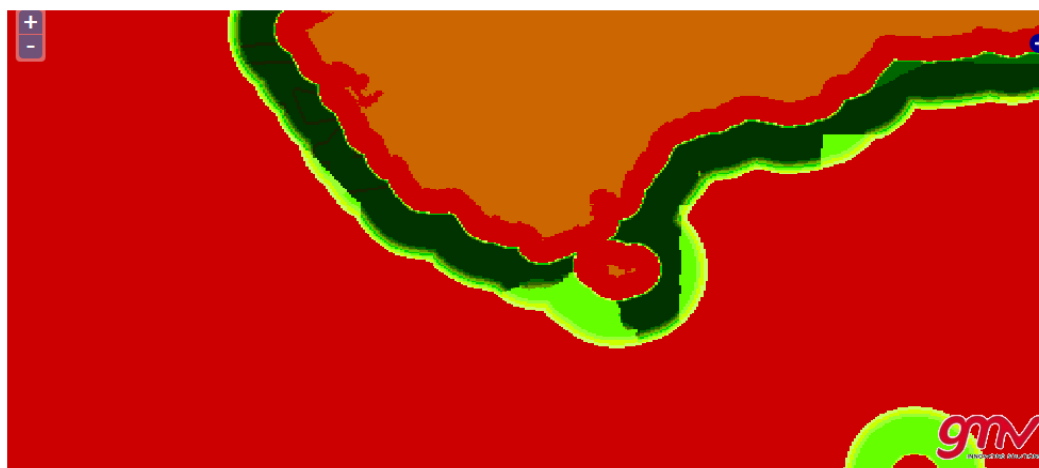
r.mapcalc "R_5532 = if(isnull( R_5531 ) , null(), if((R_5531 <= 0.5 ||
R_5531 >= 99999), 0.0, if((R_5531 > 0.5 && R_5531 < 1), eval((R_5531 -
0.5)/(1 - 0.5)), if((R_5531 > 99999 && R_5531 < 99999), eval((99999 -
R_5531)/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=dpp output=R_5541
v.to.rast input=R_5541 layer=1 type=point,line,area output=R_5542
use=val value=0 rows=4096
r.grow.distance input=R_5542 distance=R_5543 metric=euclidean
r.mapcalc "R_5544 = R_5543 * 100"
r.mapcalc "R_5545 = if(isnull( R_5544 ) , null(), if((R_5544 <= 0 ||
R_5544 >= 99999), 0.0, if((R_5544 > 0 && R_5544 < 1), eval((R_5544 -
0)/(1 - 0)), if((R_5544 > 99999 && R_5544 < 99999), eval((99999 -
R_5544)/(99999 - 99999)), 1.0))))"
r.mapcalc "R_5556 = if(isnull( R_5532 ) || isnull( R_5545 ), null(),
if(R_5532 <= R_5545 , R_5532 , R_5545 )"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=militar output=R_5546
v.to.rast input=R_5546 layer=1 type=point,line,area output=R_5547
use=val value=0 rows=4096
r.grow.distance input=R_5547 distance=R_5548 metric=euclidean
r.mapcalc "R_5549 = R_5548 * 100"
r.mapcalc "R_5550 = if(isnull( R_5549 ) , null(), if((R_5549 <= 1 ||
R_5549 >= 99999), 0.0, if((R_5549 > 1 && R_5549 < 2), eval((R_5549 -
1)/(2 - 1)), if((R_5549 > 99999 && R_5549 < 99999), eval((99999 -
R_5549)/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=zona_cables output=R_5551
v.to.rast input=R_5551 layer=1 type=point,line,area output=R_5552
use=val value=0 rows=4096
r.grow.distance input=R_5552 distance=R_5553 metric=euclidean
r.mapcalc "R_5554 = R_5553 * 100"
r.mapcalc "R_5555 = if(isnull( R_5554 ) , null(), if((R_5554 <= 1.5 ||
R_5554 >= 15), 0.0, if((R_5554 > 1.5 && R_5554 < 3), eval((R_5554 -
1.5)/(3 - 1.5)), if((R_5554 > 6 && R_5554 < 15), eval((15 - R_5554
)/(15 - 6)), 1.0))))"
r.mapcalc "R_5557 = if(isnull( R_5532 ) || isnull( R_5545 ), null(),
if(R_5532 <= R_5545 , R_5532 , R_5545 )"
r.mapcalc "R_5558 = if(isnull( R_5557 ) || isnull( R_5545 ), null(),
if(R_5557 <= R_5545 , R_5557 , R_5545 )"
r.mapcalc "R_5559 = if(isnull( R_5558 ) || isnull( R_5550 ), null(),
if(R_5558 <= R_5550 , R_5558 , R_5550 )"
r.mapcalc "R_5560 = 0.6 * R_5556 + 0.4 * R_5559"
r.mapcalc "R_5561 = if(isnull( R_5560 ) || isnull( R_5540 ), null(),
if(R_5560 <= R_5540 , R_5560 , R_5540 )"
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\pw.tif" output=R_5526
r.mapcalc "R_5527= if(R_5526 >= 0 && R_5526 < 0.3, eval(R_5526/0.3), 1)"
r.null map=R_5527 null=0
r.mapcalc "R_5562 = if(isnull( R_5527 ) || isnull( R_5540 ), null(),
if(R_5527 <= R_5540 , R_5527 , R_5540 )"
r.mapcalc "R_5563 = if(isnull( R_5561 ) || isnull( R_5562 ),
null(),if(R_5561 >= R_5562 , eval(((R_5561 * 0.4 * 0.5) + (R_5562 * 0.6
* 0.5)) / (0.4 * 0.5 + 0.6 * 0.5)), eval(((R_5561 * R_5561 * 0.5) +
(R_5562 * 0.6 * 0.5)) / (0.4 * 0.5 + 0.6 * 0.5))))"

```


Herramienta Web configurable para el apoyo a la toma de decisión usando GRASS y JSP

```
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
r.mapcalc "R_5564 = if(isnull(ccaa),R_5563, -1.0)"
g.region n=36.803943884111 s=35.199939977861 e=-3.830121796875 w=-
7.499555390625 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_5540
output="C:\Users\GSI\OceanLider\Resultados\108\RI_R_5540.tif"
r.out.gdal input=R_5532
output="C:\Users\GSI\OceanLider\Resultados\108\RI_R_5532.tif"
r.out.gdal input=R_5545
output="C:\Users\GSI\OceanLider\Resultados\108\RI_R_5545.tif"
r.out.gdal input=R_5550
output="C:\Users\GSI\OceanLider\Resultados\108\RI_R_5550.tif"
r.out.gdal input=R_5555
output="C:\Users\GSI\OceanLider\Resultados\108\RI_R_5555.tif"
r.out.gdal input=R_5527
output="C:\Users\GSI\OceanLider\Resultados\108\RI_R_5527.tif"
r.out.gdal input=R_5564
output="C:\Users\GSI\OceanLider\Resultados\108\RF_R_5564.tif"
```





2.9. Combinación de criterios físicos, criterios de recurso energético y criterios medioambientales

2.9.1. Combinación 23

Combinación de los criterios: **distancia a la costa** (criterio físico de exclusión parcial), **densidad** (criterio de recurso energético de exclusión parcial), **espacios protegidos** (criterio medio ambiental de exclusión parcial) y **usos pesqueros** (criterio medio ambiental de exclusión parcial).

En el **criterio de espacios protegidos** intervienen tres subcriterios combinados con el método **optimista**: *subcriterio reservas de la biosfera*, *subcriterio lugares de importancia comunitaria* y *zonas de especial protección para las aves*. El método de razonamiento elegido para su combinación es el **pesimista** (mínimo de los valores).

En el **criterio de usos pesqueros** intervienen tres subcriterios combinados con el método **optimista**: *subcriterio arrecifes*, *caladeros* y *subcriterio zonas de producción de moluscos y de otros invertebrados marinos*. El método de razonamiento elegido para su combinación es el **pesimista** (mínimo de los valores).

Se le deja la libertad al usuario para configurar el valor de idoneidad para las áreas que intervienen en cada subcriterio medioambiental y se le muestra uno por defecto dependiendo del tipo de dispositivo. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

El tipo de dispositivo al que esté ligado el escenario: **undimotriz- fuera costa (flotante)**.

Para la combinación de criterios medioambientales de exclusión parcial se ha elegido el método de razonamiento: **optimista** (máximo de los valores).

Para la evaluación multicriterio, en la toma de decisión final se configuran los pesos:


- Rama de criterios de recurso energético: 0.4
- Rama de criterios medioambientales: 0.6

Y los pesos de orden:


- Peso de orden primero: 0.5
- Peso de orden segundo: 0.5

Los parámetros que intervienen:


- ✓ *En el criterio distancia a la costa.*

<input checked="" type="checkbox"/> Distancia a la costa. Se aplica una operación difusa sobre distancia a la costa. Las medidas se aplican en Km. 	
Valor A	<input type="text" value="8"/>
Valor B	<input type="text" value="9"/>
Valor C	<input type="text" value="20"/>
Valor D	<input type="text" value="25"/>
Peso del subcriterio:	<input type="text" value="0.0"/>


- *En el criterio recurso energético-densidad.* Se aplica por tener el tipo de dispositivo undimotriz ligado al escenario. Hay que tener en cuenta que al introducir valores máximos mayores que 0 el resultado tenderá a 0.

<input checked="" type="checkbox"/> Recurso energético	
Peso criterio:	<input type="text" value="0.0"/>
<input checked="" type="checkbox"/> Densidad. Las unidades del valor máximo es en KWh/m. 	
Valor Máximo	<input type="text" value="0.3"/>


- ✓ *En el subcriterio reservas de la biosfera*

<input checked="" type="checkbox"/> Reservas de la biosfera 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.0

✓ *En el subcriterio lugares de importancia comunitaria*

<input checked="" type="checkbox"/> Lugares de Importancia Comunitaria 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.2

✓ *En el subcriterio zonas de especial protección para las aves*

<input checked="" type="checkbox"/> Zonas de Especial Protección para las Aves 	
Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.1

✓ *En el subcriterio arrecifes*

Arrecifes 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.6

✓ *En el subcriterio caladeros*

Caladeros 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.3

✓ *En el subcriterio zonas de producción de moluscos y de otros invertebrados marinos*

Zonas de producción de moluscos y de otros invertebrados marinos 

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.1

```
SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=lin_costa output=R_4234
v.to.rast input=R_4234 layer=1 type=point,line,area output=R_4235
use=val value=1 rows=4096
r.grow.distance input=R_4235 distance=R_4236 metric=euclidean
```

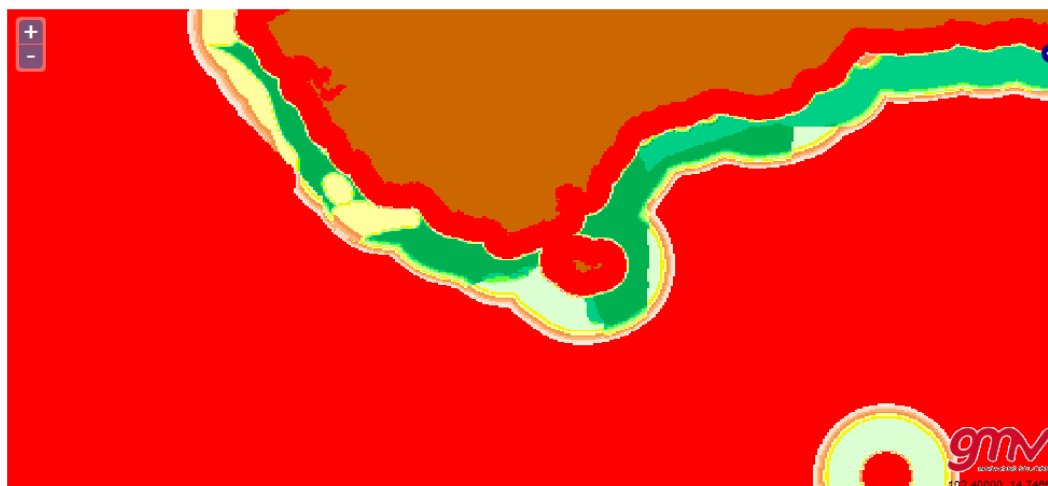
```

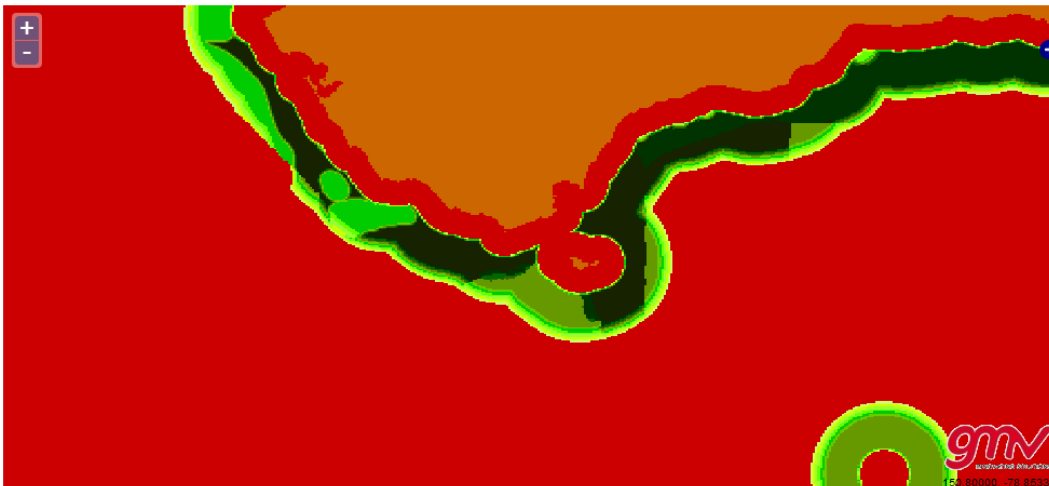
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=R_4237
v.to.rast input=R_4237 layer=1 type=point,line,area output=R_4238
use=val value=0 rows=4096
r.null map=R_4238 null=1
r.mapcalc "R_4239 = R_4236 * R_4238 "
r.mapcalc "R_4240 = R_4239 * 100"
r.mapcalc "R_4241 = if(isnull( R_4240 ) , null(), if((R_4240 <= 8 ||
R_4240 >= 25), 0.0, if((R_4240 > 8 && R_4240 < 9), eval((R_4240 - 8)/(9
-8 )), if((R_4240 > 20 && R_4240 < 25 ), eval((25 - R_4240)/(25 - 20)),
1.0))))"
r.null map=R_4241 value=0
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\pw.tif" output=R_4208
r.mapcalc "R_4209= if(R_4208 >= 0 && R_4208 < 0.3, eval(R_4208/0.3), 1)"
r.null map=R_4209 null=0
r.mapcalc "R_4248 = if(isnull( R_4209 ) || isnull( R_4241 ), null(),
if(R_4209 <= R_4241 , R_4209 , R_4241 ))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=cria_moluscos
output=R_4202
v.to.rast input=R_4202 layer=1 type=point,line,area output=R_4203
use=val value=0.0 rows=4096
r.grow.distance input=R_4203 distance=R_4204 metric=euclidean
r.mapcalc "R_4205 = R_4204 * 100"
r.mapcalc "R_4206 = if(isnull( R_4205 ) , null(), if((R_4205 <= 1 ||
R_4205 >= 99999), 0.0, if((R_4205 > 1 && R_4205 < 4), eval((R_4205 -
1)/(4 -1 )), if((R_4205 > 99999 && R_4205 < 99999 ), eval((99999 -
R_4205)/(99999 - 99999)), 1.0))))"
r.patch input=R_4203,R_4206 output=R_4207
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=caladeros output=R_4216
v.to.rast input=R_4216 layer=1 type=point,line,area output=R_4217
use=val value=0.0 rows=4096
r.grow.distance input=R_4217 distance=R_4218 metric=euclidean
r.mapcalc "R_4219 = R_4218 * 100"
r.mapcalc "R_4220 = if(isnull( R_4219 ) , null(), if((R_4219 <= 1 ||
R_4219 >= 99999), 0.0, if((R_4219 > 1 && R_4219 < 4), eval((R_4219 -
1)/(4 -1 )), if((R_4219 > 99999 && R_4219 < 99999 ), eval((99999 -
R_4219)/(99999 - 99999)), 1.0))))"
r.patch input=R_4217,R_4220 output=R_4221
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=arrecifes output=R_4242
v.to.rast input=R_4242 layer=1 type=point,line,area output=R_4243
use=val value=0.0 rows=4096
r.grow.distance input=R_4243 distance=R_4244 metric=euclidean
r.mapcalc "R_4245 = R_4244 * 100"
r.mapcalc "R_4246 = if(isnull( R_4245 ) , null(), if((R_4245 <= 1 ||
R_4245 >= 99999), 0.0, if((R_4245 > 1 && R_4245 < 4), eval((R_4245 -
1)/(4 -1 )), if((R_4245 > 99999 && R_4245 < 99999 ), eval((99999 -
R_4245)/(99999 - 99999)), 1.0))))"
r.patch input=R_4243,R_4246 output=R_4247
r.mapcalc "R_4249 = if(isnull( R_4207 ) || isnull( R_4221 ), null(),
if(R_4207 <= R_4221 , R_4207 , R_4221 ))"
r.mapcalc "R_4250 = if(isnull( R_4249 ) || isnull( R_4221 ), null(),
if(R_4249 <= R_4221 , R_4249 , R_4221 ))"

```

```
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=lics output=R_4210
v.to.rast input=R_4210 layer=1 type=point,line,area output=R_4211
use=val value=0.0 rows=4096
r.grow.distance input=R_4211 distance=R_4212 metric=euclidean
r.mapcalc "R_4213 = R_4212 * 100"
r.mapcalc "R_4214 = if(isnull( R_4213 ), null(), if((R_4213 <= 1 ||
R_4213 >= 99999), 0.0, if((R_4213 > 1 && R_4213 < 4), eval((R_4213 -
1)/(4 -1 )), if((R_4213 > 99999 && R_4213 < 99999 ), eval((99999 -
R_4213 ))/(99999 - 99999)), 1.0))))"
r.patch input=R_4211,R_4214 output=R_4215
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepas output=R_4222
v.to.rast input=R_4222 layer=1 type=point,line,area output=R_4223
use=val value=0.0 rows=4096
r.grow.distance input=R_4223 distance=R_4224 metric=euclidean
r.mapcalc "R_4225 = R_4224 * 100"
r.mapcalc "R_4226 = if(isnull( R_4225 ), null(), if((R_4225 <= 1 ||
R_4225 >= 99999), 0.0, if((R_4225 > 1 && R_4225 < 4), eval((R_4225 -
1)/(4 -1 )), if((R_4225 > 99999 && R_4225 < 99999 ), eval((99999 -
R_4225 ))/(99999 - 99999)), 1.0))))"
r.patch input=R_4223,R_4226 output=R_4227
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mabs output=R_4228
v.to.rast input=R_4228 layer=1 type=point,line,area output=R_4229
use=val value=0.0 rows=4096
r.grow.distance input=R_4229 distance=R_4230 metric=euclidean
r.mapcalc "R_4231 = R_4230 * 100"
r.mapcalc "R_4232 = if(isnull( R_4231 ), null(), if((R_4231 <= 1 ||
R_4231 >= 99999), 0.0, if((R_4231 > 1 && R_4231 < 4), eval((R_4231 -
1)/(4 -1 )), if((R_4231 > 99999 && R_4231 < 99999 ), eval((99999 -
R_4231 ))/(99999 - 99999)), 1.0))))"
r.patch input=R_4229,R_4232 output=R_4233
r.mapcalc "R_4251 = if(isnull( R_4207 ) || isnull( R_4221 ), null(),
if(R_4207 <= R_4221 , R_4207 , R_4221 ))"
r.mapcalc "R_4252 = if(isnull( R_4251 ) || isnull( R_4221 ), null(),
if(R_4251 <= R_4221 , R_4251 , R_4221 ))"
r.mapcalc "R_4253 = if(isnull( R_4252 ) || isnull( R_4247 ), null(),
if(R_4252 <= R_4247 , R_4252 , R_4247 ))"
r.mapcalc "R_4254 = if(isnull( R_4253 ) || isnull( R_4215 ), null(),
if(R_4253 <= R_4215 , R_4253 , R_4215 ))"
r.mapcalc "R_4255 = if(isnull( R_4254 ) || isnull( R_4227 ), null(),
if(R_4254 <= R_4227 , R_4254 , R_4227 ))"
r.mapcalc "R_4256 = if(isnull( R_4250 ) || isnull( R_4255 ), null(),
if(R_4250 >= R_4255 , R_4250 , R_4255 ))"
r.mapcalc "R_4257 = if(isnull( R_4256 ) || isnull( R_4241 ), null(),
if(R_4256 <= R_4241 , R_4256 , R_4241 ))"
r.mapcalc "R_4258 = if(isnull( R_4257 ) || isnull( R_4248 ),
null(),if(R_4257 >= R_4248 , eval(((R_4257 * 0.6 * 0.5) + (R_4248 * 0.4
* 0.5 )) / (0.6 * 0.5 + 0.4 * 0.5 )), eval(((R_4257 * R_4257 * 0.5 ) +
(R_4248 * 0.4 * 0.5 )) / (0.6 * 0.5 + 0.4 * 0.5))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
```

```
r.mapcalc "R_4259 = if(isnull(ccaa),R_4258, -1.0)"
g.region n=36.781971227861 s=35.112049352861 e=-3.698285859375 w=-
7.345746796875 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_4241
output="C:\Users\GSI\OceanLider\Resultados\109\RI_R_4241.tif"
r.out.gdal input=R_4207
output="C:\Users\GSI\OceanLider\Resultados\109\RI_R_4207.tif"
r.out.gdal input=R_4221
output="C:\Users\GSI\OceanLider\Resultados\109\RI_R_4221.tif"
r.out.gdal input=R_4247
output="C:\Users\GSI\OceanLider\Resultados\109\RI_R_4247.tif"
r.out.gdal input=R_4215
output="C:\Users\GSI\OceanLider\Resultados\109\RI_R_4215.tif"
r.out.gdal input=R_4227
output="C:\Users\GSI\OceanLider\Resultados\109\RI_R_4227.tif"
r.out.gdal input=R_4233
output="C:\Users\GSI\OceanLider\Resultados\109\RI_R_4233.tif"
r.out.gdal input=R_4209
output="C:\Users\GSI\OceanLider\Resultados\109\RI_R_4209.tif"
r.out.gdal input=R_4259
output="C:\Users\GSI\OceanLider\Resultados\109\RF_R_4259.tif"
```





2.10. Combinación de criterios físicos, criterios operacionales, criterios medioambientales y criterios de recurso energético

2.10.1. Combinación 24

Combinación de los criterios: **profundidad** (criterio físico de exclusión parcial), **viabilidad de instalación en zonas ocupadas** (criterio operacional de exclusión parcial), **espacios protegidos** (criterio medio ambiental de exclusión parcial), y **densidad** (criterio de recurso energético de exclusión parcial).

En el criterio **viabilidad de instalación en zonas ocupadas** intervienen los subcriterios: *viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español* y *viabilidad de instalación en zonas designadas como dominio público portuario*. El método de razonamiento por defecto para su combinación es el **pesimista** (mínimo de los valores).

En el **criterio de espacios protegidos** intervienen tres subcriterios combinados con el método **pesimista**: *subcriterio reservas de la biosfera*, *subcriterio lugares de importancia comunitaria* y *zonas de especial protección para las aves*. El método de razonamiento elegido para su combinación es el **pesimista** (mínimo de los valores).

Se le deja la libertad al usuario para configurar el valor de idoneidad para las áreas que intervienen en cada subcriterio medioambiental y se le muestra uno por defecto dependiendo del tipo de dispositivo. Posteriormente se aplica una operación difusa sobre la distancia a las áreas implicadas. Las medidas para la operación difusa se aplican en Km. El usuario debe introducir el valor de idoneidad que quiere aplicar a las áreas y los cuatro valores necesarios para realizar la operación difusa: valor A, valor B, valor C y valor D.

El tipo de dispositivo al que esté ligado el escenario: **undimotriz- fuera costa (flotante)**.

Para la evaluación multicriterio, en la toma de decisión final se configuran los pesos:


- Rama de criterios operacionales: 0.2
- Rama de criterios medioambientales: 0.5
- Rama de criterios de recurso energético: 0.3

Y los pesos de orden:

- Peso de orden primero: 0.33
- Peso de orden segundo: 0.33
- Peso de orden tercero: 0.33

Los parámetros que intervienen:

- ✓ El criterio *profundidad*: los parámetros profundidad máxima y mínima del dispositivo al que está ligado y tiene los valores -100 y -20, respectivamente.
- ✓ Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español.

Viabilidad de instalación en zonas ocupadas por las ayudas a la navegación del sistema portuario español. Se aplica una operación difusa sobre la distancia a las zonas ocupadas por las ayudas a la navegación del sistema portuario español. Las medidas se aplican en Km. 

De subida	
Valor A	0.5
Valor B	1
Valor C	99999
Valor D	99999

- ✓ Viabilidad de instalación en zonas desinadas como Dominio Público Portuario.

Viabilidad de instalación en zonas designadas como Dominio Público Portuario. Se aplica una operación difusa sobre la distancia a las zonas designadas como Dominio Público Portuario. Las medidas se aplican en Km. 

De subida	
Valor A	0
Valor B	1
Valor C	99999
Valor D	99999

- ✓ En el criterio *recurso energético-densidad*. Se aplica por tener el tipo de dispositivo undimotriz ligado al escenario. Hay que tener en cuenta que al introducir valores máximos mayores que 0 el resultado tenderá a 0.

Recurso energético

Peso criterio:

Densidad. Las unidades del valor máximo es en KWh/m. 

Valor Máximo

✓ *En el subcriterio reservas de la biosfera*

Reservas de la biosfera 

Valor de idoneidad	<input type="text" value="0.0"/>
Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="4"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>
Peso del subcriterio:	<input type="text" value="0.0"/>

✓ *En el subcriterio lugares de importancia comunitaria*

Lugares de Importancia Comunitaria 

Valor de idoneidad	<input type="text" value="0.0"/>
Valor A	<input type="text" value="1"/>
Valor B	<input type="text" value="4"/>
Valor C	<input type="text" value="99999"/>
Valor D	<input type="text" value="99999"/>
Peso del subcriterio:	<input type="text" value="0.2"/>

✓ *En el subcriterio zonas de especial protección para las aves*

Zonas de Especial Protección para las Aves

Valor de idoneidad	0.0
Valor A	1
Valor B	4
Valor C	99999
Valor D	99999
Peso del subcriterio:	0.1

```

SET GISRC=C:\Users\GSI\OceanLider\.grassrc6_wgs84
g.region n=44.83 s=26.67 e=5.57 w=-21.01 nsres=300 ewres=300 rows=3224
cols=3453
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\profundidad.tif"
output=R_4290
r.mapcalc "R_4291 = if(isnull( R_4290 ), null(), if((R_4290 <= -105.0
|| R_4290 >= -15.0), 0.0, if((R_4290 > -105.0 && R_4290 < -100.0),
eval((R_4290 - -105.0)/(-100.0 --105.0 )), if((R_4290 > -20.0 && R_4290
< -15.0 ), eval((-15.0 - R_4290 )/(-15.0 - -20.0)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=aton output=R_4268
v.to.rast input=R_4268 layer=1 type=point,line,area output=R_4269
use=val value=0 rows=4096
r.grow.distance input=R_4269 distance=R_4270 metric=euclidean
r.mapcalc "R_4271 = R_4270 * 100"
r.mapcalc "R_4272 = if(isnull( R_4271 ), null(), if((R_4271 <= 0.5 ||
R_4271 >= 99999), 0.0, if((R_4271 > 0.5 && R_4271 < 1), eval((R_4271 -
0.5)/(1 -0.5 )), if((R_4271 > 99999 && R_4271 < 99999 ), eval((99999 -
R_4271 )/(99999 - 99999)), 1.0))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=dpp output=R_4285
v.to.rast input=R_4285 layer=1 type=point,line,area output=R_4286
use=val value=0 rows=4096
r.grow.distance input=R_4286 distance=R_4287 metric=euclidean
r.mapcalc "R_4288 = R_4287 * 100"
r.mapcalc "R_4289 = if(isnull( R_4288 ), null(), if((R_4288 <= 0 ||
R_4288 >= 99999), 0.0, if((R_4288 > 0 && R_4288 < 1), eval((R_4288 -
0)/(1 -0 )), if((R_4288 > 99999 && R_4288 < 99999 ), eval((99999 -
R_4288 )/(99999 - 99999)), 1.0))))"
r.mapcalc "R_4292 = if(isnull( R_4272 ) || isnull( R_4289 ), null(),
if(R_4272 <= R_4289 , R_4272 , R_4289 )"
r.mapcalc "R_4293 = if(isnull( R_4292 ) || isnull( R_4291 ), null(),
if(R_4292 <= R_4291 , R_4292 , R_4291 )"
r.in.gdal -o input="%USERPROFILE%\OceanLider\Union\pw.tif" output=R_4266
r.mapcalc "R_4267= if(R_4266 >= 0 && R_4266 < 0.3, eval(R_4266/0.3), 1)"
r.null map=R_4267 null=0
r.mapcalc "R_4294 = if(isnull( R_4267 ) || isnull( R_4291 ), null(),
if(R_4267 <= R_4291 , R_4267 , R_4291 )"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=lics output=R_4260

```

```
v.to.rast input=R_4260 layer=1 type=point,line,area output=R_4261
use=val value=0.0 rows=4096
r.grow.distance input=R_4261 distance=R_4262 metric=euclidean
r.mapcalc "R_4263 = R_4262 * 100"
r.mapcalc "R_4264 = if(isnull( R_4263 ), null(), if((R_4263 <= 1 ||
R_4263 >= 99999), 0.0, if((R_4263 > 1 && R_4263 < 4), eval((R_4263 -
1)/(4 -1 )), if((R_4263 > 99999 && R_4263 < 99999 ), eval((99999 -
R_4263)/(99999 - 99999)), 1.0))))"
r.patch input=R_4261,R_4264 output=R_4265
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=zepas output=R_4273
v.to.rast input=R_4273 layer=1 type=point,line,area output=R_4274
use=val value=0.0 rows=4096
r.grow.distance input=R_4274 distance=R_4275 metric=euclidean
r.mapcalc "R_4276 = R_4275 * 100"
r.mapcalc "R_4277 = if(isnull( R_4276 ), null(), if((R_4276 <= 1 ||
R_4276 >= 99999), 0.0, if((R_4276 > 1 && R_4276 < 4), eval((R_4276 -
1)/(4 -1 )), if((R_4276 > 99999 && R_4276 < 99999 ), eval((99999 -
R_4276)/(99999 - 99999)), 1.0))))"
r.patch input=R_4274,R_4277 output=R_4278
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider_restricciones host=localhost
port=5432 user=postgres password=968XCdPh" layer=mabs output=R_4279
v.to.rast input=R_4279 layer=1 type=point,line,area output=R_4280
use=val value=0.0 rows=4096
r.grow.distance input=R_4280 distance=R_4281 metric=euclidean
r.mapcalc "R_4282 = R_4281 * 100"
r.mapcalc "R_4283 = if(isnull( R_4282 ), null(), if((R_4282 <= 1 ||
R_4282 >= 99999), 0.0, if((R_4282 > 1 && R_4282 < 4), eval((R_4282 -
1)/(4 -1 )), if((R_4282 > 99999 && R_4282 < 99999 ), eval((99999 -
R_4282)/(99999 - 99999)), 1.0))))"
r.patch input=R_4280,R_4283 output=R_4284
r.mapcalc "R_4295 = if(isnull( R_4265 ) || isnull( R_4278 ), null(),
if(R_4265 <= R_4278 , R_4265 , R_4278 ))"
r.mapcalc "R_4296 = if(isnull( R_4295 ) || isnull( R_4278 ), null(),
if(R_4295 <= R_4278 , R_4295 , R_4278 ))"
r.mapcalc "R_4297 = if(isnull( R_4296 ) || isnull( R_4291 ), null(),
if(R_4296 <= R_4291 , R_4296 , R_4291 ))"
r.mapcalc "R_4298 = if(isnull(R_4293) || isnull(R_4297) ||
isnull(R_4294), null(),if(R_4293 >= R_4297 && R_4297 >= R_4294,
eval(((R_4293*0.2*0.33) + (R_4297*0.5*0.33) + (R_4294*0.3*0.33)) /
(0.2*0.33+0.5*0.33+0.3*0.33)),if(R_4293 >= R_4294 && R_4294 > R_4297,
eval(((R_4293*0.2*0.33) + (R_4294*0.3*0.33) + (R_4297*0.5*0.33)) /
(0.2*0.33+0.3*0.33+0.5*0.33)),if(R_4297 > R_4293 && R_4293 >= R_4294,
eval(((R_4297*0.5*0.33) + (R_4293*0.2*0.33) + (R_4294*0.3*0.33))
/(0.5*0.33+0.2*0.33+0.3*0.33)),if(R_4297 >= R_4294 && R_4294 > R_4293,
eval(((R_4297*0.5*0.33) + (R_4294*0.3*0.33) + (R_4293*0.2*0.33)) /
(0.5*0.33+0.3*0.33+ 0.2*0.33)),if(R_4294 > R_4293 && R_4293 >= R_4297,
eval(((R_4294*0.3*0.33) + (R_4293*0.2*0.33) + (R_4297*0.5*0.33)) /
(0.3*0.33+0.2*0.33+0.5*0.33)),if(R_4294 > R_4297 && R_4297 > R_4293,
eval(((R_4294*0.3*0.33) + (R_4297*0.5*0.33) + (R_4293*0.2*0.33)) /
(0.3*0.33+0.5*0.33+0.2*0.33)), null()))))"
v.in.ogr -o "dsn=PG:dbname=BBDDOceanLider host=localhost port=5432
user=postgres password=968XCdPh" layer=ccaa output=ccaa
v.to.rast input=ccaa layer=1 type=point,line,area output=ccaa use=val
value=-1.0 rows=4096
```

```
r.mapcalc "R_4299 = if(isnull(ccaa),R_4298, -1.0)"
g.region n=36.692981567717 s=35.045032348967 e=-3.786176484375 w=-
7.301801484375 nsres=300 ewres=300 rows=3224 cols=3453
r.out.gdal input=R_4291
output="C:\Users\GSI\OceanLider\Resultados\111\RI_R_4291.tif"
r.out.gdal input=R_4272
output="C:\Users\GSI\OceanLider\Resultados\111\RI_R_4272.tif"
r.out.gdal input=R_4289
output="C:\Users\GSI\OceanLider\Resultados\111\RI_R_4289.tif"
r.out.gdal input=R_4265
output="C:\Users\GSI\OceanLider\Resultados\111\RI_R_4265.tif"
r.out.gdal input=R_4278
output="C:\Users\GSI\OceanLider\Resultados\111\RI_R_4278.tif"
r.out.gdal input=R_4284
output="C:\Users\GSI\OceanLider\Resultados\111\RI_R_4284.tif"
r.out.gdal input=R_4267
output="C:\Users\GSI\OceanLider\Resultados\111\RI_R_4267.tif"
r.out.gdal input=R_4299
output="C:\Users\GSI\OceanLider\Resultados\111\RF_R_4299.tif"
```

