



Universidad de Valladolid

E. T. S. de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Mención en Tecnologías de la Información

Aplicación web rutas y deportes: milrutas

Autor: Álvaro Estébanez Barrena

Tutor: Benjamín Sahelices Fernández

Este trabajo fin de grado significa la continuación de mi proyecto de vida y de las innumerables experiencias, que me han hecho crecer aceleradamente y por otro lado, es para mí el comienzo de una nueva etapa como profesional.

Durante su realización he aprendido a valorar mucho más a mis seres queridos, amigos y todas aquellas personas que han sido un gran apoyo para llegar al final de esta etapa y a quienes indiscutiblemente dedico este símbolo de esfuerzo tan importante para mí

Índice

Bloque I.	1
Introducción	1
Capítulo 1. Introducción y objetivos	3
1.1 Descripción del proyecto	3
1.2 Alcance del proyecto	3
1.3 Objetivos	4
1.4 Motivación	4
1.5 Limitaciones	4
1.6 Documentación	5
Capítulo 2. Deportes y rutas	7
2.1 El origen de las 3 ideas	7
2.2 ¿Qué es Deportes y rutas?	8
2.3 Viabilidad	8
2.4 Metas	9
Sociales:	9
Económicas	10
Web	10
Bloque II.	11
Fundamentos teóricos y tecnologías aplicadas	11
Capítulo 3. Fundamentos teóricos y tecnologías aplicadas	13
3.1 PHP	13
3.2 HTML 5	14
3.3 JavaScript	14
3.4 MySQL	15
3.5 AJAX	15
3.6 API Google Maps v3	16
3.7 JQuery	17
3.8 Apache	18

3.9	UML	18
3.10	CSS3	19
3.11	Twitter Bootstrap	19
3.12	API SDK for PHP	20
3.13	Twitter for Websites	20
3.14	API de JavaScript de Google +	20
3.15	XAMPP	20
3.16	Simple Weather	21
Bloque III.		23
Análisis		23
Capítulo 4. Introducción al análisis y plan de gestión del proceso		25
4.1	Prologo	25
4.2	Propósito del sistema	25
4.3	Sistema propuesto	26
4.4	Plan de trabajo	26
4.4.1	Actividades de trabajo	26
4.4.2	Asignación de recursos y calendario	27
4.5	Plan de proceso técnico	29
4.5.1	Modelo de proceso	29
4.5.2	Métodos, herramientas y técnicas	29
4.5.3	Plan de despliegue	30
4.6	Gestión de riesgos	30
4.6.1	Introducción	30
4.6.2	Identificación de los riesgos	31
4.6.3	Análisis de riesgos	31
4.6.4	Planificación de respuesta a los riesgos	33
4.7	Costes del proyecto	33
4.7.1	Recursos materiales	33
4.7.2	Recursos humanos	34

4.7.3	Costes totales _____	35
Capítulo 5.	Elicitación de requisitos _____	37
5.1	Requisitos de usuario (Objetivos) _____	37
5.2	Requisitos funcionales _____	38
5.3	Requisitos no funcionales _____	40
Capítulo 6.	_____	43
Casos de uso	_____	43
6.1	Descripción de los actores _____	43
	Usuario no registrado _____	43
	Usuario registrado _____	43
	Usuario administrador _____	44
6.2	Diagrama de casos de uso _____	44
6.3	Especificación de casos de uso _____	45
6.4	Modelo de dominio _____	55
Bloque IV.	_____	57
Diseño	_____	57
Capítulo 7. Diseño de persistencia	_____	59
7.1	Decisiones de diseño de la base de datos _____	59
7.2	Diagrama entidad-relación _____	59
7.2.1	usuario _____	60
7.2.2	ruta _____	61
7.2.3	rutacabecera _____	61
7.2.4	rutanodo _____	62
7.2.5	valoracion _____	63
Capítulo 8. Arquitectura del sistema	_____	65
8.1	Introducción _____	65
8.2	Descripción del sistema _____	65
8.2.1	Diseño arquitectónico _____	65
8.2.2	Organización del sistema _____	66

8.2.3	Descomposición modular	67
8.2.4	Flujo de control	68
8.2.5	Diseño físico	69
8.3	Modelos del sistema	70
8.3.1	Patrones utilizados	70
8.3.2.	Modelos estáticos	72
	Diagrama de clases global	73
	Diagrama de clases de los controladores	74
	ControladorUsuario:	75
	ControladorRutaPublica:	76
	ControladorRutaPrivada:	76
	ControladorCiudad:	77
	ControladorContacto:	77
	ControladorAdmin:	78
	Diagrama de clases de los Datamappers	78
	DataMapperRuta	80
	DataMapperUsuario:	81
	DataMapperAdmin:	81
8.3.3	Modelos dinámicos	82
8.3.4.1	Diagrama de secuencia, Crea ruta	83
8.3.4.2	Diagrama de secuencia, Empieza ruta	84
8.3.4.3	Diagrama de secuencia, Termina ruta	85
8.3.4.4	Diagrama de secuencia, Registro	86
8.3.4.5	Diagrama de secuencia (Login)	86
Bloque V.		89
Implementación		89
Capítulo 9. Estructura de directorios		91
9.1	Introducción	91
9.2	Estructura	91
Bloque VI.		97
Pruebas y resultados		97
Capítulo 10. Pruebas de funcionalidad y rendimiento		99
10.1	Pruebas de instalación	99

10.2	Pruebas de sistema	99
10.3	Pruebas de caja negra	100
10.4	Casos de prueba	102
Bloque VII.		105
Manual de instalación y usuario		105
Capítulo 11.		107
Manual de instalación		107
11.1	Introducción	107
11.2	Requisitos de instalación	107
11.3	Instalación	108
	Instalación previa	108
	Instalación del sitio web	108
Capítulo 12.		111
Manual de usuario		111
12.1	Usuario no registrado	111
	Introducción	111
	Ver rutas	113
	Cargar Rutas	115
	Realizar ruta	115
	Ver mis rutas (registro)	116
	Contacto	117
	Redes Sociales	117
12.2	Usuario registrado	118
	Introducción	118
	Realizar Ruta	119
	Crear ruta manual	120
	Ver mis rutas	121
12.3	Usuario Administrador	122
	Gestion Rutas	122
	Gestion Usuarios	122
12.4	Optimización al realizar rutas	123
Bloque VIII.		125
Conclusiones		125

Capítulo 13. Conclusiones	127
13.1 Objetivos logrados	127
13.2 Análisis global	127
13.3 Trabajo futuro	128
Bloque IX.	129
Apéndice	129
Capítulo 14. Apéndices	131
14.1 Contenido del CD	131
14.2 Referencias	131
14.2.1 Libros y manuales	131
14.2.2 Documentación web	132
14.3 Glosario de términos	132

Índice de ilustraciones

<i>Ilustración 1: Análisis DAFO</i>	9
<i>Ilustración 2: Diagrama de casos de uso</i>	44
<i>Ilustración 3: Clases del dominio</i>	56
<i>Ilustración 4: Diagrama entidad-relación</i>	59
<i>Ilustración 5: Tabla usuario</i>	60
<i>Ilustración 6: Tabla ruta</i>	61
<i>Ilustración 7: Tabla rutacabecera</i>	61
<i>Ilustración 8: Tabla rutanodos</i>	62
<i>Ilustración 9: Tabla de valoracion</i>	63
<i>Ilustración 10: Diagrama de capas del sistema</i>	67
<i>Ilustración 11: Diagrama de componentes</i>	68
<i>Ilustración 12: Diagrama de despliegue</i>	69
<i>Ilustración 13: Patrón DataMapper</i>	72
<i>Ilustración 14: Diagrama de clases global</i>	73
<i>Ilustración 15: Diagrama de clases de los controladores</i>	74
<i>Ilustración 16: ControladorUsuario</i>	75
<i>Ilustración 17: ControladorRutaPublica</i>	76
<i>Ilustración 18: ControladorRutaPrivada</i>	76
<i>Ilustración 19: ControladorCiudad</i>	77
<i>Ilustración 20: ControladorContacto</i>	77
<i>Ilustración 21: ControladorAdmin</i>	78
<i>Ilustración 22 : Diagrama de clases de los DataMappers</i>	79
<i>Ilustración 23: DataMapperRuta</i>	80
<i>Ilustración 24: DataMapperUsuario</i>	81
<i>Ilustración 25: DataMapperAdmin</i>	82
<i>Ilustración 26: Diagrama de secuencia crear ruta</i>	83
<i>Ilustración 27: Diagrama de secuencia empezar ruta</i>	84
<i>Ilustración 28: Diagrama de secuencia terminar ruta</i>	85
<i>Ilustración 29: Diagrama de secuencia registro</i>	86
<i>Ilustración 30: Diagrama de secuencia login</i>	86
<i>Ilustración 31: Estructura de directorios</i>	91
<i>Ilustración 32: Implementación: Configuraciones</i>	92
<i>Ilustración 33: Implementación: Controladores</i>	92
<i>Ilustración 34: Implementación: css</i>	92
<i>Ilustración 35: Implementación: fuentes</i>	93
<i>Ilustración 36: Implementación: imágenes</i>	93
<i>Ilustración 37 Implementación: JavaScript</i>	94
<i>Ilustración 38: Implementación: Libs</i>	94
<i>Ilustración 39 Implementación: DataMappers</i>	95
<i>Ilustración 40: Implementación: Modelos</i>	95

<i>Ilustración 41: Implementación: Vistas</i>	95
<i>Ilustración 42: Caso de prueba: Realizar una ruta previamente creada</i>	103
<i>Ilustración 43: Página principal escritorio 1/2, usuario no registrado</i>	111
<i>Ilustración 44: Página principal escritorio 2/2, usuario no registrado</i>	112
<i>Ilustración 45: Página principal móvil 2/2, usuario no registrado</i>	113
<i>Ilustración 46: Página principal móvil 1/2, usuario no registrado</i>	113
<i>Ilustración 47: Página rutas disponibles escritorio 1/2, usuario no registrado</i>	114
<i>Ilustración 48: Página rutas disponibles móvil, usuario no registrado</i>	114
<i>Ilustración 49: Página cargar rutas, usuario no registrado</i>	115
<i>Ilustración 50: Realizando ruta, usuario no registrado</i>	115
<i>Ilustración 51: Fin de ruta, usuario no registrado</i>	116
<i>Ilustración 52: Registro de usuario, usuario no registrado</i>	116
<i>Ilustración 53: Contacto, usuario no registrado</i>	117
<i>Ilustración 54: Redes sociales</i>	117
<i>Ilustración 55: Twittear</i>	118
<i>Ilustración 56: Compartir en Facebook</i>	118
<i>Ilustración 57: Página principal, usuario registrado</i>	119
<i>Ilustración 58: Datos ruta realizada, usuario registrado</i>	119
<i>Ilustración 59: Datos ruta manual, usuario registrado</i>	120
<i>Ilustración 60: Introducción de nodos manual, usuario registrado</i>	120
<i>Ilustración 61: Ver rutas privadas, usuario registrado</i>	121
<i>Ilustración 62: Ver rutas privadas móvil</i>	121
<i>Ilustración 63: Ver gestión de rutas, administrador</i>	122
<i>Ilustración 64: Ver gestión de usuarios, administrador</i>	122
<i>Ilustración 65: Configuración ubicación</i>	123

Índice de figuras

<i>Figura 1: Actividades de la Fase de Comienzo</i>	27
<i>Figura 2: Actividades de la Fase de Elaboración</i>	28
<i>Figura 3: Actividades de la Fase de Construcción</i>	28
<i>Figura 4: Actividades de la Fase de Transición</i>	28
<i>Figura 5: Actividades presentes en todas las fases</i>	28
<i>Figura 6: Modelo Vista Controlador</i>	71
<i>Figura 7: Controlador frontal</i>	71

Índice de tablas

<i>Tabla 1: Fases del proyecto</i>	27
<i>Tabla 2: Riesgos del proyecto</i>	31
<i>Tabla 3: Probabilidad de los riesgos</i>	31
<i>Tabla 4: Prioridad de los riesgos</i>	32
<i>Tabla 5: Matriz de probabilidad</i>	32
<i>Tabla 6: Prioridad de los riesgos y acción</i>	32
<i>Tabla 7: Impacto de los riesgos</i>	33
<i>Tabla 8: Respuesta específica de riesgos</i>	33
<i>Tabla 9: Costes de recursos materiales</i>	34
<i>Tabla 10: Costes por recursos humanos</i>	35
<i>Tabla 11: Costes totales del proyecto</i>	35
<i>Tabla 12: Objetivo control de usuario</i>	37
<i>Tabla 13: Objetivo control de usuarios registrados</i>	37
<i>Tabla 14: Objetivo administración de rutas</i>	37
<i>Tabla 15: Objetivo multiplataforma</i>	38
<i>Tabla 16: Requisito de gestión de usuario</i>	38
<i>Tabla 17: Requisito de gestión de ruta realizada</i>	38
<i>Tabla 18: Requisito de gestión de ruta creada</i>	38
<i>Tabla 19: Requisito de realizar ruta</i>	38
<i>Tabla 20: Requisito de mostrar rutas</i>	38
<i>Tabla 21: Requisito de crear ruta</i>	39
<i>Tabla 22: Requisito de buscar ruta</i>	39
<i>Tabla 23: Requisito de contacto</i>	39
<i>Tabla 24: Requisito de seleccionar ciudad</i>	39
<i>Tabla 25: Requisito de localización</i>	39
<i>Tabla 26: Requisito de mostrar ruta</i>	39
<i>Tabla 27: Requisito de distancia ruta</i>	40
<i>Tabla 28: Requisito de valoración</i>	40
<i>Tabla 29: Requisito administración</i>	40
<i>Tabla 30: Requisito tiempo meteorológico</i>	40
<i>Tabla 31: Requisito no funcional de tiempo de respuesta</i>	40
<i>Tabla 32: Requisito no funcional de servidor</i>	41
<i>Tabla 33: Requisito no funcional de lenguaje de programación</i>	41
<i>Tabla 34: Requisito no funcional de base de datos</i>	41
<i>Tabla 35: Requisito no funcional de seguridad</i>	41
<i>Tabla 36: Requisito no funcional de datos personales</i>	41
<i>Tabla 37: Requisito no funcional de copia de seguridad</i>	41
<i>Tabla 38: Requisito no funcional de asincronicidad</i>	42
<i>Tabla 39: Requisito no funcional de usabilidad</i>	42
<i>Tabla 40: Requisito no funcional de compatibilidad</i>	42

<i>Tabla 41: Actor usuario no registrado</i>	43
<i>Tabla 42: Actor usuario registrado</i>	43
<i>Tabla 43: Actor usuario administrador</i>	44
<i>Tabla 44: Especificación del caso de uso crear ruta</i>	45
<i>Tabla 45: Especificación del caso de uso ver rutas</i>	46
<i>Tabla 46: Especificación del caso de uso contacto</i>	47
<i>Tabla 47: Especificación del caso de uso realizar ruta como usuario registrado</i>	48
<i>Tabla 48: Especificación del caso de uso realizar ruta como usuario no registrado</i>	49
<i>Tabla 49: Especificación del caso de uso buscar ruta pública</i>	50
<i>Tabla 50: Especificación del caso de uso buscar ruta privada usuario registrado</i>	51
<i>Tabla 51: Especificación del caso de uso buscar ruta privada usuario no registrado</i>	52
<i>Tabla 52: Especificación del caso de uso registro de usuario</i>	53
<i>Tabla 53: Especificación del caso de uso identificar usuario</i>	53
<i>Tabla 54: Especificación del caso de uso gestión de usuarios</i>	54
<i>Tabla 55: Especificación del caso de uso gestión de rutas</i>	55
<i>Tabla 56: Especificación tabla usuario</i>	60
<i>Tabla 57: Especificación tabla ruta</i>	61
<i>Tabla 58: Especificación tabla rutacabecera</i>	62
<i>Tabla 59: Especificación rutanodos</i>	63
<i>Tabla 60: Especificación de la tabla valoracion</i>	63
<i>Tabla 61: Descripción de la clase ControladorUsuario</i>	75
<i>Tabla 62: Descripción de la clase ControladorRutaPublica</i>	76
<i>Tabla 63: Descripción de la clase ControladorRutaPrivada</i>	77
<i>Tabla 64: Descripción de la clase ControladorCiudad</i>	77
<i>Tabla 65: Descripción de la clase ControladorContacto</i>	78
<i>Tabla 66: Descripción de la clase ControladorAdmin</i>	78
<i>Tabla 67: Descripción DataMapperRuta</i>	81
<i>Tabla 68: Descripción DataMapperUsuario</i>	81
<i>Tabla 69: Descripción DataMapperAdmin</i>	82
<i>Tabla 70: Pruebas de caja negra: email de usuario</i>	100
<i>Tabla 71: Pruebas de caja negra: comprobación email de usuario</i>	100
<i>Tabla 72: Pruebas de caja negra: password</i>	100
<i>Tabla 73: Pruebas de caja negra: comprobación password</i>	101
<i>Tabla 74: Pruebas de caja negra: nombre</i>	101
<i>Tabla 75: Pruebas de caja negra: comprobación nombre</i>	101
<i>Tabla 76: Pruebas de caja negra: apellidos</i>	101
<i>Tabla 77: Pruebas de caja negra: comprobación apellidos</i>	101
<i>Tabla 78: Pruebas de caja negra: nombre de la ruta</i>	102
<i>Tabla 79: Pruebas de caja negra: comprobación nombre de la ruta</i>	102
<i>Tabla 80: Pruebas de caja negra: comentarios de una ruta</i>	102
<i>Tabla 81: Pruebas de caja negra: comprobación comentarios de una ruta</i>	102
<i>Tabla 82: Caso de prueba: Realizar ruta nueva</i>	103

Bloque I.

Introducción

BLOQUE I. INTRODUCCIÓN

Capítulo 1.

Introducción y objetivos

1.1 Descripción del proyecto

Dado el exponencial crecimiento del uso de dispositivos Smartphone, surgen por parte de los usuarios nuevas expectativas que van más allá del simple uso del teléfono, demandando a dichos dispositivos aplicaciones cada vez más sofisticadas y que nos ayuden y faciliten nuestro día a día. Que aporten un valor añadido a tareas diarias.

En los últimos años se ha popularizado la tendencia de realizar deporte junto con aplicaciones móviles (app's) para así tener mayor información del ejercicio físico realizado.

El mercado de las aplicaciones relacionadas con el deporte ha sufrido un crecimiento exponencial, la variabilidad de los dispositivos y el continuo cambio de los mismos, hacen que la web sea una solución apropiada para todos los aparatos.

El proyecto consta de una aplicación web para rutas de diferentes deportes, tanto para ver las rutas que un usuario ha realizado, como para unirse a otras creadas por diferentes usuarios.

1.2 Alcance del proyecto

La aplicación web está pensada para ser utilizado en eventos deportivos y/o de ocio tanto individuales como colectivos. Ya sea una persona que realiza una ruta o un grupo de amigos, realizando una ruta nueva u otra previamente creada o realizada por otros usuarios.

Respecto a los usuarios finales, se presume que tienen unos conocimientos mínimos para hacer un uso correcto de la aplicación, aun así, en este documento se detalla un manual de usuario con las funcionalidades que ofrece deportes y rutas

BLOQUE I. INTRODUCCIÓN

1.3 Objetivos

El objetivo de este proyecto es la implementación de una aplicación software que permita al usuario ver los eventos y rutas disponibles, así como sus propias rutas realizadas, todo ello ha de ser posible tanto en dispositivos móviles como en ordenadores de sobremesa.

Estoy convencido que el proponer a los deportistas o simples practicantes del deporte una aplicación de estas características será una motivación más para una práctica deportiva continuada y controlada.

La posibilidad de compartir rutas, valorarlas, etc, sin duda alguna fomentara una intercomunicación entre personas dentro de las distintas actividades ociosas que éstas desarrollen, andar, correr, pasear en moto o cualquier simple desplazamiento que se desee monitorizar...

Por último, es necesario tener un apartado de administración donde el responsable del sitio web pueda modificar los atributos de los usuarios y/o borrar rutas.

1.4 Motivación

Uno de los principales motivos para el desarrollo del proyecto ha sido la posibilidad de flexibilizar la aplicación hacia otras actividades, si bien algo alejadas del deporte y su práctica, siempre relacionadas con la geolocalización y control de los desplazamientos.

Queda abierto, a mi juicio, en este sentido un enorme potencial de evolución, circunstancia obviamente fuera de las pretensiones de este trabajo.

1.5 Limitaciones

Dado el carácter NO comercial del trabajo, éste ha renunciado deliberadamente al uso de herramientas y servicios comerciales o empresariales (mayormente de pago). En este contexto se ha tenido en cuenta sobre todo motivos pedagógicos y el poder poner en valor la mayoría de los conocimientos adquiridos y, aplicarlos en la medida de lo posible a circunstancias y necesidades del mundo real.

De tal manera, se ha optado por un desarrollo web en lugar de una aplicación móvil, lo cual invita a ulteriores mejoras y ampliaciones.

Por consiguiente, se ha buscado un punto de convergencia en lo que significa un trabajo fin de grado acercar estas nuevas (algunas incipientes) tecnologías al deporte y su entorno

BLOQUE I. INTRODUCCIÓN

1.6 Documentación

En este apartado se resumirá brevemente como está organizada o esquematizada esta memoria así como la información que se puede encontrar en cada uno de los bloques en los que está dividida.

Bloque I: Introducción. En este bloque, que es en el que nos encontramos, se da una visión general del proyecto, su alcance, motivación y estructura de la memoria del mismo. El objetivo es presentar de forma breve en que consiste el software que se ha desarrollado.

Bloque II: Fundamentos teóricos y tecnologías aplicadas. En este segundo bloque se detallarán los contenidos teóricos sobre los que está apoyado el proyecto, así como Apis, frameworks, plugins... utilizados para programar la aplicación.

Bloque III: Análisis. En el bloque de análisis, se pretende presentar que es lo que se quiere hacer, definir los actores que interactúan con el sistema y presentar los primeros diagramas UML de análisis, en definitiva, el objetivo de este bloque es contextualizar el trabajo a realizar.

Bloque IV: Diseño. El bloque de diseño pretende explicar cómo vamos a hacer el sistema, presentar los diagramas UML de diseño, dar una visión global de la arquitectura del sistema, así como, explicar cómo se estructurarán y funcionaran las capas de presentación, lógica y persistencia de datos de la aplicación.

Bloque V: Implementación. En este bloque se explicará cómo se ha llevado a cabo la implementación del sistema estudiado en los bloques III y IV.

Bloque VI: Pruebas y resultados. En el bloque V, se presentarán las diferentes pruebas que se han ejecutado sobre la aplicación y los resultados que se han obtenido.

Bloque VII: Manual de instalación y usuario. En el bloque de manual de usuario, se explicará que se necesita para poder utilizar la aplicación, como instalarla y por supuesto como utilizarla.

Bloque VIII: Conclusiones. Una vez se haya terminado de implementar la aplicación y de realizar los manuales de instalación y usuario, pasaremos al apartado de conclusiones, en el que explicaremos que cosas hemos aprendido o de que cosas nos hemos dado cuenta, y también, que cosas han ido bien y mal sin olvidar que cosas se pueden mejorar.

Bloque IX: Apéndices. En este último bloque, se presentará el contenido del CD adjunto con el proyecto y las referencias o recursos en los que nos hemos apoyado para analizar, diseñar e implementar.

BLOQUE I. INTRODUCCIÓN

Capítulo 2.

Deportes y rutas

2.1 El origen de las 3 ideas

El uso de las nuevas tecnologías en nuestra sociedad es un hecho cada vez más cotidiano del que para muchos se ha convertido hasta en dependencia. Y es que las facilidades que nos permite hoy en día Internet hace que sea realmente complicado pensar en un mundo diferente.

Los expertos denominan a nuestros días la era tecnológica y no es difícil de utilizar ese apellido si en nuestro país, uno de los más envejecidos, más de la mitad de la población de más de 15 años ya tiene un Smartphone y casi el 70% usa internet.

Internet hace que tengamos a nuestra disposición cualquier tipo de información a tiempo real desde cualquier parte del mundo. Y desde esta perspectiva globalizadora, informativa y digital nace la primera idea de este proyecto: usar las nuevas tecnologías para facilitar la vida a las personas.

También internet tiene su “parte negativa”. Nuestros jóvenes cada día empiezan antes con el uso de dispositivos móviles e incluso utilizan las redes sociales e internet con un dudoso control paternal. Internet permite alcanzar información de cualquier parte del mundo pero lo cierto es que también aísla de la realidad más cotidiana sobre todo a los jóvenes. También crea sedentarismo por culpa de los trabajos online y las aficiones relacionadas con el mundo virtual. De estos problemas nace nuestra segunda idea: fomentar la práctica deportiva.

Por último, nuestra tercera idea trata sobre la movilidad, cómo nos desplazamos. En el ámbito deportivo en los últimos años han proliferado innumerables runners, rollers y ciclistas que día a día practican deportes exteriores abandonando gimnasios y sus cuotas. Pero además de los deportistas, existen cada vez más asociaciones de moteros, concentraciones de coches míticos, quedadas de coches de competición... que buscan aunar su pasión con la necesidad de descubrir nuevos lugares paisajes. De esta idea sobre la movilidad nace el tercer pilar y más importante: Las rutas, es decir, dónde vamos.

BLOQUE I. INTRODUCCIÓN

A través de estas 3 ideas que conforman los 3 pilares de este proyecto, nace milrutas. Una web para facilitar, promocionar el deporte y encontrar nuevos sitios que descubrir ya sea corriendo, andando o en moto.

2.2 ¿Qué es Deportes y rutas?

milrutas es una web adaptada a todo tipo de terminales desde la que se pueden realizar rutas de desplazamiento en la que se mide la distancia y la velocidad con una doble finalidad:

Por un lado, el resto de usuarios pueden descubrir nuevos lugares para practicar su deporte favorito.

Por otro lado, el propio usuario puede utilizar sus datos guardados para utilizarlos en forma de comparativa con su historial.

Se trata de una doble herramienta totalmente gratuita que proporciona al usuario una respuesta ante la pregunta: dónde voy.

El elemento gratuito proporciona una base de datos en la que se le puede informar de acontecimientos y hechos relacionado con las rutas dentro en la web.

El usuario que entra en milrutas tiene tres opciones cuando entra a la web:

- Participar de una ruta ya creada por otro usuario.
- Crear una ruta planificada previamente.
- Crear una ruta sin planificación previa.

En cualquiera de los casos, la web recoge el tiempo, la distancia y la velocidad del participante y lo registra en un historial que el usuario puede consultar en cualquier momento.

Además, solo es necesario un Smartphone con conexión a internet para participar de la experiencia de milrutas ya que esta web está optimizada para cualquier dispositivo de cualquier sistema operativo.

Por último, milrutas.com ofrece la posibilidad de interactuar y captar usuarios a través de las principales redes sociales: Facebook, twitter y google plus. La web permite el registro a través de las redes sociales así como poder compartir rutas realizadas.

2.3 Viabilidad

milrutas tiene un sentido social ya que es una herramienta web gratuita pero no renuncia a posibles beneficios que hagan viable este proyecto. Entre estas posibilidades se encuentran:

- Empresas que quieran publicitarse en la web mediante banners.
- Publicidad de Google Adwords.

BLOQUE I. INTRODUCCIÓN

- Venta de base de datos.
- Anuncios a través de redes sociales.
- Rutas patrocinadas.
- Alianzas con otras webs complementarias.

Análisis DAFO:

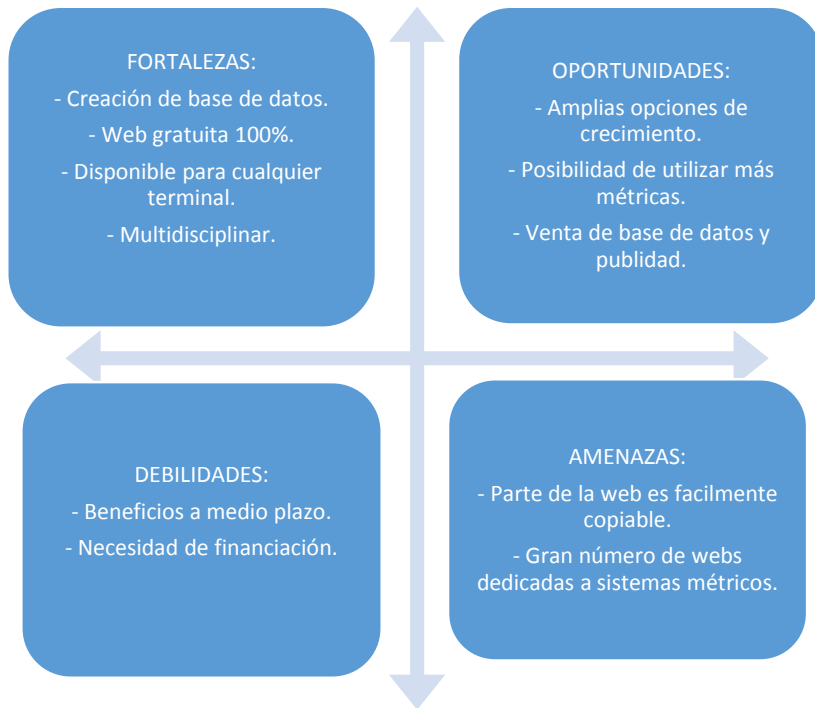


Ilustración 1: Análisis DAFO

2.4 Metas

Sociales:

- Llegar al público objetivo ofreciendo una herramienta web gratuita a través de foros y blogs.
- Ofrecer una experiencia simple al usuario pero satisfactoria.
- Extender al mayor número de disciplinas posibles esta web.
- Buscar alianzas con webs complementarias.
- Diferenciarse de la competencia mediante la gratuidad de todo el servicio.

BLOQUE I. INTRODUCCIÓN

Económicas

- Crecer la base de datos al 5% mensual los dos primeros años.
- Encontrar 5 empresas publicitarias en la web en el primer año.

Web

- Mantenimiento de los usuarios ya registrados con retornos del 30%
- Tener un índice de rebote inferior al 15%.

Bloque II.

Fundamentos teóricos y tecnologías aplicadas

BLOQUE II. FUNDAMENTOS TEÓRICOS Y TECNOLOGÍAS APLICADAS

Capítulo 3.

Fundamentos teóricos y tecnologías aplicadas

3.1 PHP

PHP es el lenguaje en el que está desarrollado gran parte de la lógica de la aplicación, es por ello, que en este apartado daremos una breve introducción en la que explicaremos que es PHP y cuáles son sus principales características.

PHP (acrónimo recursivo de PHP: Hypertext Pre-processor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en código HTML. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente.

BLOQUE II. FUNDAMENTOS TEÓRICOS Y TECNOLOGÍAS APLICADAS

Este lenguaje forma parte del software libre publicado bajo la licencia PHP, que es incompatible con la Licencia Pública General de GNU debido a las restricciones del uso del término PHP.

Este proyecto ha sido desarrollado con la versión 5.5.15, siendo la versión 5.6 la última ya que la 6 ha sido abandonado por la 7 previsto su lanzamiento a finales del 2015. Las principales motivaciones que me han llevado a utilizar este lenguaje han sido: en primer lugar que se trata de un lenguaje libre, ya que también existía la opción de realizar el desarrollo en ASP. La segunda es la facilidad que proporciona para aplicar programación orientada a objetos.

3.2 HTML 5

La estructura del proyecto en la parte del cliente está desarrollada en HTML5. La elección de este lenguaje viene dado por su potencial para el desarrollo de aplicaciones móviles así como la integración de la localización mediante APIs.

HTML5 (HyperText Markup Language, versión 5) es la quinta revisión del lenguaje de programación “básico” de la World Wide Web, el HTML. Los documentos HTML se encuentran estrictamente organizados. Cada parte del documento está diferenciada, declarada y determinada por etiquetas específicas, ya que se trata de un lenguaje de marcas.

HTML5 provee básicamente tres características: estructura, estilo y funcionalidad. Nunca fue declarado oficialmente pero, incluso cuando algunas APIs y la especificación de CSS3 por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, CSS y Javascript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y Javascript hace el resto.

3.3 JavaScript

Javascript se ha utilizado en el desarrollo de este software para realizar la función de localización mediante la API de geolocalización que nos permite obtener la posición del usuario.

Javascript es un lenguaje interpretado usado para múltiples propósitos pero solo considerado como un complemento hasta ahora. Es un lenguaje pequeño y ligero, no es útil como un lenguaje independiente, más bien está diseñado para una fácil incrustación en otros productos y aplicaciones. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS).

Contrariamente a la falsa idea popular, JavaScript no es "Java interpretativo". En pocas palabras, JavaScript es un lenguaje de programación dinámico que soporta construcción de objetos

BLOQUE II. FUNDAMENTOS TEÓRICOS Y TECNOLOGÍAS APLICADAS

basado en prototipos. La sintaxis básica es similar a Java y C++ con la intención de reducir el número de nuevos conceptos necesarios para aprender el lenguaje.

3.4 MySQL

Para la persistencia de los datos se ha optado por usar el SGBD MySQL. MySQL es un sistema gestor de bases de datos, se trata de un sistema de libre distribución y de código abierto.

MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger y hacer volcados de datos. Puede desarrollar sus propias aplicaciones de base de datos en la mayor parte de los lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos. MySQL utiliza el lenguaje de consulta estructurado (SQL). Se trata del lenguaje utilizado por todas las bases de datos relacionales.

MySQL es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python). Su popularidad como aplicación web está muy ligada a PHP. Es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación.

3.5 AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

BLOQUE II. FUNDAMENTOS TEÓRICOS Y TECNOLOGÍAS APLICADAS

3.6 API Google Maps v3

El API consiste de archivos JavaScript que contienen las clases, métodos y propiedades que se usan para el comportamiento de los mapas. Es sólo HTML, CSS y JavaScript trabajando junto. Los mapas son solo imágenes que se cargan en el fondo a través de peticiones ejecutadas por la tecnología de AJAX, y se insertan en un `<div>` en la página HTML. Mientras navegas en el mapa, el API envía información acerca de las nuevas coordenadas, los niveles de “zoom” del mapa a través de AJAX, etc. y esto retorna las imágenes. Las coordenadas están expresadas usando números decimales separados por coma. La latitud siempre precede la longitud.

La API de Google Maps se encuentra alojada en los servidores de Google. Para poder cargar la API de Google Maps debemos hacer una referencia desde nuestro archivo HTML hacia el lugar en el que se encuentra ésta.

API Key

Para poder utilizar los servicios de Google Maps es necesario disponer de una API Key. La política de Google se está volviendo rigurosa respecto a esto, debiéramos pararnos a pensar de qué forma nos arrastran gratuitamente hacia sus productos de pago. De manera “gratuita” nos permite lanzar 2500 peticiones diarias, y si se requieren más peticiones es necesario crear una cuenta de pago Google Business con más servicios y hasta 2.000.000 de peticiones diarias. Para poder disponer de dicha Key es necesario disponer de una cuenta Gmail y aceptar los términos de uso.

Sensor

La API de Google Maps requiere que se indique si la aplicación que estamos creando utiliza algún tipo de sensor, para determinar la ubicación del usuario a través de un localizador de GPS, 4G, 3G o GSM. Al indicar “sensor = true”, la API intenta utilizar el mayor grado de precisión que disponga el dispositivo, siendo transparente para el desarrollador y usuario final.

Localización (idioma)

La API de Google Maps intentará determinar automáticamente qué idioma utiliza la interfaz de usuario. No obstante, también podemos indicarlo en el `<script>` que hemos utilizado para referenciar la API de Google Maps. Para hacerlo, debemos añadir el parámetro (opcional) `&language=”codigo_del_idioma”` al final de la cadena:

MapOptions

Se deben definir primero cuáles serán las opciones y características de nuestro mapa a través de la declaración de un objeto `mapOptions`. A continuación indicamos las más representativas:

BLOQUE II. FUNDAMENTOS TEÓRICOS Y TECNOLOGÍAS APLICADAS

- center: El punto central de nuestro mapa, es decir, lo primero que se verá cuando se cargue. Se debe crear un objeto LatLng para especificar las coordenadas del punto.
- zoom: La distancia de la cámara a la superficie del mapa.
- mapTypeId: Tipo de mapa que se debe cargar. Dentro de esta categoría tenemos varias opciones:
 - ROADMAP: Carga el típico mapa con las calles.
 - SATELLITE: Muestra el mapa con imágenes tomadas desde satélite (se vería como el Google Earth).
 - TERRAIN: Muestra características físicas como elevación del terreno y vegetación.
 - HYBRID: Carga las calles principales sobre imágenes de satélite

3.7 JQuery

JQuery es una biblioteca de JavaScript, 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. JQuery es la biblioteca de JavaScript más utilizada.

JQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU, permitiendo su uso en proyectos libres y privados. JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

JQuery consiste en un único fichero JavaScript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX.

La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX. Para ello utiliza las funciones `$()` o `jQuery()`.

JQuery TimeCircles

TimeCircles es un plugin de JQuery que proporciona una interfaz en forma de reloj agradable. Utilizado para indicar al usuario el tiempo que lleva empleado en la ruta que está haciendo. Dependiendo de las opciones, puede utilizarse tanto como cronometro como de cuenta regresiva.

Uso básico:

- Cargar el archivo CSS correspondiente a TimeCircles.
- Crear un contenedor y ajustar el formato.

BLOQUE II. FUNDAMENTOS TEÓRICOS Y TECNOLOGÍAS APLICADAS

- Cargar la librería JQuery TimeCircles.
- Llamar al plugin y activar las opciones deseadas

JQuery UI Dialog

Dialog es una utilidad de JQuery UI para la creación de cuadros de diálogos personalizados, es una manera de dar estilo al funcionamiento de la página web, en lugar de utilizar las alertas incómodas de Javascript. Se utiliza para el contacto e introducir los datos de una ruta.

JQuery DataTable

DataTables es un plugin para jQuery que nos permite darle dinamismo a nuestras tablas de una forma sencilla y aumentando la vistosidad de nuestras tablas. Con solo un par de líneas, se cambiará completamente la apariencia de nuestra tabla.

Con este plugin se conseguirán las siguientes acciones con las tablas:

- Ordenar por campos.
- Paginar resultados
- Filtrar
- Sistema de búsqueda
- Obtención de datos de forma dinámica.

3.8 Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual.

3.9 UML

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group).

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el

BLOQUE II. FUNDAMENTOS TEÓRICOS Y TECNOLOGÍAS APLICADAS

sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

3.10 CSS3

Hoja de estilo en cascada o CSS (siglas en inglés de cascading style sheets) es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El World Wide Web Consortium (W3C) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. La versión 3 de CSS sigue el mismo camino, pero esta vez con un mayor compromiso. La especificación de HTML5 fue desarrollada considerando CSS a cargo del diseño. Debido a esta consideración, la integración entre HTML y CSS es ahora vital para el desarrollo web y esta es la razón por la que cada vez que mencionamos HTML5 también estamos haciendo referencia a CSS3, aunque oficialmente se trate de dos tecnologías completamente separadas.

3.11 Twitter Bootstrap

Twitter Bootstrap es un framework o conjunto de herramientas de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales.

Bootstrap tiene un soporte relativamente incompleto para HTML5 y CSS 3, pero es compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores. Existe un concepto de compatibilidad parcial que hace disponible la información básica de un sitio web para todos los dispositivos y navegadores.

Desde la versión 2.0 también soporta diseños sensibles. Esto significa que el diseño gráfico de la página se ajusta dinámicamente, tomando en cuenta las características del dispositivo usado (ordenador de sobremesa / portátil, tabletas, teléfonos móviles).

Bootstrap es de código abierto y está disponible en GitHub.

Para usar Bootstrap en una página HTML, se debe descargar la hoja de estilo Bootstrap CSS y enlazarla en el archivo HTML. Si se desea usar los componentes de JavaScript, éstos deben estar referenciados junto con la librería jQuery en el documento HTML.

BLOQUE II. FUNDAMENTOS TEÓRICOS Y TECNOLOGÍAS APLICADAS

3.12 API SDK for PHP

El SDK de Facebook para PHP es una biblioteca con potentes funciones que permiten a los desarrolladores de PHP integrar fácilmente Facebook login y hacer peticiones a la API de gráficos. También, con el SDK de Facebook JavaScript es posible dar al usuario (front-end) la mejor experiencia de usuario posible. Pero la cosa no termina ahí, el SDK de Facebook para PHP hace que sea fácil subir fotos y videos y enviar solicitudes de lotes a la Graph API entre otras cosas.

Esta Api es utilizada para publicar en el muro del usuario las páginas de la aplicación que quiera compartir, siendo posible la autenticación por medio de la web a Facebook y el contenido de la publicación.

3.13 Twitter for Websites

Twitter for Websites es utilizada para los sitios web, compuesta por un conjunto de widgets que traen el contenido de Twitter en nuestra web y botones que animar al público de Twitter para compartir su contenido y suscribirse a las actualizaciones de la cuenta de Twitter.

Se empleara para que los usuarios puedan publicar tweets desde el sistema, ya sea para una ruta o para una parte de la aplicación, ya que estará disponible durante todas las partes del sistema

3.14 API de JavaScript de Google +

La API de Google+ es la interfaz de programación de Google+. Se puede utilizar la API para integrar la aplicación web con Google +. Esto permite a los usuarios conectarse entre sí para el compromiso máximo uso de las funciones de Google+ desde su aplicación.

3.15 XAMPP

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl.

El programa está liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas.

XAMPP solamente requiere descargar y ejecutar un archivo ZIP, tar, exe o fkl, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará. XAMPP se actualiza regularmente para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL y phpMyAdmin. Para instalar XAMPP se requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar los programas

BLOQUE II. FUNDAMENTOS TEÓRICOS Y TECNOLOGÍAS APLICADAS

por separado. Puede encontrarse tanto en versión completa, así como en una versión más ligera que es portátil.

3.16 Simple Weather

SimpleWeather es un Plugin de jQuery para visualizar fácilmente un widget del tiempo de forma limpia y personalizable. Se utiliza para mostrar la información meteorológica de la posición del usuario.

Bloque III.

Análisis

BLOQUE III. ANÁLISIS

Capítulo 4.

Introducción al análisis y plan de gestión del proceso

4.1 Prologo

En este capítulo se presentará una primera aproximación al análisis del diseño, incluyendo el plan de trabajo que se llevara a cabo. En él se describirá la gestión de los recursos disponibles junto con las actividades fundamentales para el éxito del proyecto. También se tratara el proceso técnico empleado y las herramientas necesarias.

En las últimas partes del tema, se abordará brevemente la gestión de riesgos desglosando su identificación, el análisis y las respuestas asociadas. Y para finalizar se verá, de manera aproximada, los costes parciales y totales que engloban este proyecto.

4.2 Propósito del sistema

La aspiración del sistema consiste en disponer de un sistema vía web para la creación, realización y visionado de rutas por parte de los usuarios sea cual sea la modalidad de deporte y/o aventura que se vaya a realizar.

Es importante recalcar, la necesidad de que la aplicación esté disponible en cualquier dispositivo con una conexión a internet y un navegador web. Aunque se tiene que tener en cuenta, la posibilidad, en un futuro, de la migración a sistemas nativos para una mayor funcionalidad de la

BLOQUE III. ANÁLISIS

aplicación. Por lo cual, se procurara, que el sistema sea lo más escalable y modular, que sea posible.

4.3 Sistema propuesto

El sistema propuesto estará basado en componentes software de código libre, se necesitara un sistema cliente – servidor, un conjunto de servicios externos para poder hacer uso de la geolocalización y visualización atractiva adaptativa a cualquier tipo de dispositivo.

Una primera solución general, consta de clientes semi-delgados, se encargarán de la visualización en el dispositivo y de la recopilación de los datos de localización para su envío al servidor. La interacción con el usuario estará basada en los estándares web convencionales y en la filosofía de los 3 click (realizar cualquier funcionalidad de la aplicación en, como máximo 3 click).

La parte del cliente será amigable y fácil de utilizar por cualquier usuario con conocimientos básicos de informática.

La parte del servidor será la encargada de gestionar la lógica de la aplicación y la persistencia de los datos. Se realizarán conexiones síncronas (cuando el usuario este interactuando manualmente con el sistema) y asíncronas (para que sea transparente para el usuario las tareas de geolocalización).

La parte restante del sistema, son los servicios que se harán uso para poder gestionar la geolocalización de una manera rápida y eficiente de los datos, incluyendo las transferencias de los mismos.

4.4 Plan de trabajo

El plan de trabajo está apoyado por medio de la herramienta MS Project, obteniéndose los diagramas y datos.

4.4.1 Actividades de trabajo

De la aplicación del modelo de Proceso Unificado, se obtienen 4 fases determinantes en el proyecto:

Fases	Producto resultante
Fase de comienzo	<ul style="list-style-type: none">- Visión general de los requisitos principales, características y restricciones principales.- Primera versión del modelo de casos de uso, completado al 10% aproximadamente.- Primera aproximación al diseño arquitectónico.

BLOQUE III. ANÁLISIS

	<ul style="list-style-type: none"> - Elaboración preliminar del plan de gestión de riesgos. - Modelo completo de negocio - Requisitos funcionales y no funcionales revisados. - Descripción de la arquitectura software. - Revisión del plan de gestión de riesgos.
Fase de elaboración	<ul style="list-style-type: none"> - Sistema software ejecutable en versión beta. - Modelo de sistema completo al 100% - Primeras versiones del manual de usuario e instalación
Fase de construcción	<ul style="list-style-type: none"> - Sistema software al completo, junto con la instalación. - Última versión de los manuales de usuario e instalación. - Valoración de la satisfacción por parte de los usuarios.
Fase de transición	

Tabla 1: Fases del proyecto

4.4.2 Asignación de recursos y calendario

Se adjuntan las actividades identificadas en MS Project para cada una de las fases.

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
▣ Fase de Comienzo	4,75 días	lun 22/06/15	vie 26/06/15		
▣ I1	4,75 días	lun 22/06/15	vie 26/06/15		
▣ Plan de desarrollo del proyecto	4,75 días	lun 22/06/15	vie 26/06/15		
Determinar el alcance del proyecto	2 días	lun 22/06/15	mar 23/06/15		Responsable del proyecto
Elaboración del plan de proyecto	3 días	mié 24/06/15	vie 26/06/15	4	Responsable del proyecto
Fin fase de comienzo	0 días	vie 26/06/15	vie 26/06/15	2	

Figura 1: Actividades de la Fase de Comienzo

BLOQUE III. ANÁLISIS

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recurso
▸ Fase de Comienzo	4,75 días	lun 22/06/15	vie 26/06/15		
▸ Fase de Elaboración	15,13 días	lun 29/06/15	lun 20/07/15	1	
▸ 12	3,75 días	lun 29/06/15	jue 02/07/15	2	
▸ Análisis y requisitos del software	2,75 días	lun 29/06/15	mié 01/07/15		
Acotación de las especificaciones del proyecto	1 día	lun 29/06/15	lun 29/06/15		Analista
Elaboración de requisitos funcionales y no funcionales	4 horas	mar 30/06/15	mar 30/06/15	10	Analista
Modelo de casos de uso (hasta 10%)	4 horas	mar 30/06/15	mar 30/06/15	11	Analista
Versión inicial del modelo de dominio	1 día	mié 01/07/15	mié 01/07/15	12	Analista
▸ Diseño	0,75 días	jue 02/07/15	jue 02/07/15	9	
Planteamiento del diseño arquitectónico	1 día	jue 02/07/15	jue 02/07/15	13	Diseñador
▸ 13	7,38 días	vie 03/07/15	mar 14/07/15	8	
Identificación de requisitos	6 horas	vie 03/07/15	vie 03/07/15	6	Analista
Revisión de requisitos funcionales y no funcionales	1 día	vie 03/07/15	lun 06/07/15	17	Analista
Identificación de casos de uso	6 horas	lun 06/07/15	mar 07/07/15	18	Analista
Modelo de casos de uso (hasta 80%)	2 días	mar 07/07/15	jue 09/07/15	19	Analista
Especificación de casos de uso	2 días	jue 09/07/15	lun 13/07/15	20	Analista
Modelo de dominio	1 día	lun 13/07/15	mar 14/07/15	21	Analista
▸ 14	3,63 días	mar 14/07/15	lun 20/07/15	16	
Revisión de análisis	6 horas	mar 14/07/15	mié 15/07/15		Diseñador
Diagrama de clases de diseño parcial	1 día	mié 15/07/15	jue 16/07/15	24	Diseñador
Diagrama arquitectónico	2 días	jue 16/07/15	lun 20/07/15	25	Diseñador
Diagrama de secuencia	2 días	jue 16/07/15	lun 20/07/15	25	Diseñador
Diagrama de clases de diseño completo	2 días	jue 16/07/15	lun 20/07/15	25	Diseñador
Fin fase de elaboración	0 días	lun 20/07/15	lun 20/07/15	28	

Figura 2: Actividades de la Fase de Elaboración

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
▸ Fase de Comienzo	4,75 días	lun 22/06/15	vie 26/06/15		
▸ Fase de Elaboración	15,13 días	lun 29/06/15	lun 20/07/15	1	
▸ Fase de construcción	24,88 días	lun 20/07/15	lun 24/08/15	7	
▸ Fase de transición	2,63 días	lun 24/08/15	mié 26/08/15	30	
▸ 19	2,63 días	lun 24/08/15	mié 26/08/15		
Despliegue	6 horas	lun 24/08/15	lun 24/08/15		Desarrollador
Pruebas de despliegue	2 días	lun 24/08/15	mié 26/08/15	51	Evaluador
Fin de fase de transición	0 días	mié 26/08/15	mié 26/08/15	52	

Figura 4: Actividades de la Fase de Transición

Por último se han definido dos actividades que estarán presentes a lo largo de todo el ciclo de desarrollo del proyecto: el seguimiento y control del mismo y la documentación asociada a dicho control y seguimiento.

Documentación	72 días	vie 22/05/15	lun 31/08/15		Responsable del proyecto
Seguimiento y control del proyecto	72 días	vie 22/05/15	lun 31/08/15		Responsable del proyecto
Entrega Fin del Proyecto	0 días	lun 31/08/15	lun 31/08/15	49:54:55	Responsable del proyecto

Figura 5: Actividades presentes en todas las fases

BLOQUE III. ANÁLISIS

4.5 Plan de proceso técnico

El Proceso Unificado (RUP) es un proceso de desarrollo de software. Según sus desarrolladores es más que un proceso, es un marco de trabajo genérico que puede especializarse para gran cantidad de sistemas de software, diferentes áreas de aplicación, diferentes tipos de organizaciones, etc.

Está basado en componentes, lo que indica que el sistema a construir estará formado por componentes software interconectados a través de interfaces.

Utiliza UML como lenguaje de modelado.

Los tres elementos claves que resumen el proceso unificado son:

- Dirigido por los casos de uso
- Centrado en la arquitectura
- Iterativo e incremental

4.5.1 Modelo de proceso

El modelo de proceso empleado será el Proceso Unificado, basado en componentes software conectados mediante interfaces bien definidas y siguiendo los elementos clave en este tipo de proceso.

De esta forma se diferencian 4 fases en el ciclo de vida del proyecto: fase de comienzo, fase de elaboración, fase de construcción y fase de transición. Cada una de ellas constará con una serie de iteraciones que permitan identificar y mitigar los riesgos más críticos en las etapas más tempranas del proyecto.

4.5.2 Métodos, herramientas y técnicas

Como se ha comentado anteriormente, se hará uso de herramientas software de código abierto para la elaboración del producto final.

En la siguiente lista se muestran las herramientas software necesarias en cada fase.

Fase de Comienzo:

- Herramientas ofimáticas: Se hará uso de estos productos para elaborar la documentación pertinente, así como los gráficos, tablas o el soporte para la planificación del proyecto. Se utilizará la familia de productos Office de Microsoft: concretamente Project y Word. También se hará uso de herramientas ofimáticas en la nube como Google Drive para tener copias en todo momento del proyecto y la documentación en medios remotos.

Fase de elaboración:

BLOQUE III. ANÁLISIS

- Herramientas CASE: Para la parte de ingeniería del software se buscará apoyo en herramientas informáticas que apoyen al proceso de ingeniería en lo que a productividad se refiere. Se utilizará Astah para la construcción de los diferentes diagramas de casos de uso, secuencia, actividad, flujo, etc...

Fase de construcción:

- Máquina Virtual con sistema operativo Linux Server Ubuntu 14.04.1.
- Cliente SSH para el acceso remoto a la máquina virtual asignada.
- Entornos de programación, IDEs:
 - Dreamweaver CS6
 - Netbeans 8.0.2
- Lenguajes de programación, Sistemas Gestores de Bases de Datos y Servidor Web: explicado en el capítulo 3.

4.5.3 Plan de despliegue

Una vez haya concluido la elaboración del producto final del proyecto, se requerirán una serie de elementos (hardware, software, personal...) para desplegar y hacer posible el uso de este producto por el cliente:

Hardware

- Una máquina que juegue el rol de servidor donde se instalara tanto los ficheros de la propia página, como la base de datos.
- Infraestructura de red capaz de dar salida a internet al servidor.
- Maquinas cliente para el acceso al servidor por medio de un navegador web.

Software

- Servidor web Apache.
- Interprete de lenguaje PHP.
- Base de datos relacional MySQL.
- SO compatible con las herramientas anteriormente mencionadas.

Personal

- Técnico encargado de llevar a cabo el despliegue del producto final para su correcto uso por parte del cliente.

4.6 Gestión de riesgos

4.6.1 Introducción

El plan de gestión de riesgos está concebido para guiar a identificar, analizar y gestionar los riesgos que se puedan producir durante la etapa de desarrollo del sistema gestión de incidencias. Se empleará una estrategia proactiva, integrada y sistémica, de tal forma que se identifiquen los riesgos potenciales teniendo en cuenta su probabilidad de ocurrencia, así como su posible impacto.

BLOQUE III. ANÁLISIS

Se establecerán políticas, sistemáticas y disciplinadas, de prioridad para los diferentes riesgos, que junto con los planes de acción correspondientes, controlará cualquier situación posible en caso de ocurrencia.

4.6.2 Identificación de los riesgos

Lista con los posibles riesgos identificados:

Identificador	Descripción	Tipo de riesgo
R01	Calidad inadecuada	De producto
R02	Planificación optimista	De proyecto
R03	Diseño inadecuado	De producto
R04	Inexperiencia	De proyecto
R05	Baja productividad	De proyecto
R06	Realización de pruebas no adecuada	De producto
R07	Falta de seguimiento	De proceso

Tabla 2: Riesgos del proyecto

4.6.3 Análisis de riesgos

Teniendo la lista de riesgos identificados, se definirá un nivel de probabilidad de ocurrencia. El impacto que supondrá en el alcance del objetivo y el coste.

Identificador	Descripción	Probabilidad	Impacto	Coste tiempo
R01	Calidad inadecuada	Alto	Muy alto	Elevado
R02	Planificación optimista	Bajo	Alto	Medio
R03	Diseño inadecuado	Medio	Alto	Corto
R04	Inexperiencia	Bajo	Bajo	Medio
R05	Baja productividad	Bajo	Bajo	Corto
R06	Realización de pruebas no adecuada	Alto	Alto	Corto
R07	Falta de seguimiento	Medio	Alto	Medio

Tabla 3: Probabilidad de los riesgos

BLOQUE III. ANÁLISIS

Se establecerá una clasificación basada en la prioridad de los riesgos. La tabla se ha elaborado basándose en una matriz de probabilidad-impacto, asignándose mayor prioridad a los riesgos que con más frecuencia ocurren y más impacto suponen.

Identificador	Descripción	Prioridad
R01	Calidad inadecuada	Muy alto
R02	Planificación optimista	Medio
R03	Diseño inadecuado	Alto
R04	Inexperiencia	Bajo
R05	Baja productividad	Bajo
R06	Realización de pruebas no adecuada	Medio
R07	Falta de seguimiento	Alto

Tabla 4: Prioridad de los riesgos

Matriz de probabilidad-impacto con su correspondiente agrupación de riesgos:

		Probabilidad			
		Muy Alto	Alto	Medio	Bajo
Impacto	Muy Alto		R01		
	Alto		R06	R03;R07	R02
	Medio				
	Bajo				R04;R05

Tabla 5: Matriz de probabilidad

Clasificación de los riesgos determinando la prioridad para cada uno de ellos:

Prioridad	Criterios	Acción
Muy alta	Riesgo crítico	Acción inmediata
Alta	Riesgo significativo y probable	Iniciar procedimientos
Media	Riesgo significativo y menos probable	Realizar una revisión
Baja	Riesgo poco probable	Volver a valorar

Tabla 6: Prioridad de los riesgos y acción

Estimamos los posibles impactos de cada riesgo en la siguiente clasificación:

Impacto	Descripción
Muy alto	Objetivos fundamentales del proyecto con alto grado de impacto o no

BLOQUE III. ANÁLISIS

	cumplimiento
Alto	Objetivos críticos del proyecto están amenazados
Medio	Algunos objetivos del proyecto pueden verse afectados
Bajo	Los objetivos del proyecto no serán afectados. Fácilmente remediable

Tabla 7: Impacto de los riesgos

4.6.4 Planificación de respuesta a los riesgos

Para los riesgos con prioridad alta y muy alta, se han hecho planes de respuesta específicos para cada uno de ellos en la siguiente tabla:

Identificador	Plan de mitigación	Plan de contingencia
R01	Establecer un plan de calidad, siguiendo el estándar ISO 9000 y 9003	Realizar un plan de calidad y llevarlo a cabo
R06	Revisión continua y periódica del desarrollo de pruebas y así detectar sus posibles anomalías	Realización de nuevos tipos de pruebas más concretas y documentadas

Tabla 8: Respuesta específica de riesgos

4.7 Costes del proyecto

En esta sección se detallarán los costes que el proyecto necesita para su desarrollo, agrupados por recursos.

4.7.1 Recursos materiales

Se necesitarán una serie de recursos materiales básicos para poder llevar a cabo el desarrollo. Se ha priorizado en todo momento el uso de herramientas y lenguajes de programación Open Source, tanto por su coste de adquisición como por el de mantenimiento a corto y medio plazo.

También se han utilizado licencias de prueba y/o gratuitas, algunas de ellas facilitadas por la Escuela Técnica Superior de Ingeniería Informática de la Universidad de Valladolid.

A continuación se especifica el coste que supone cada una de ellas:

Recurso	Tipo	Detalles	Coste
Ordenador	Hardware	Ordenador portátil con SO Windows 7, (adquirido hace 3 años)	Valor actual: 350€
IDE	Software	Adobe Dreamweaver CC	0 € (licencia de prueba)
CASE	Software	Astah Community 6	0 € (licencia)

BLOQUE III. ANÁLISIS

			universitaria)
Servidor Web	Software/Hardware	Apache Localizado físicamente en máquina virtual	0 € (licencia Open Source) 30 € (IP estática incluida)
Servidor BBDD	Software/Hardware	MySQL (misma máquina virtual)	0 € (licencia Open Source)
Editor de texto	Software	MS Word	0 € (licencia universitaria)
Planificación y gestión de recursos	Software	MS Project	0 € (licencia universitaria)
Otros programas	Software	Navegadores web (Google Chrome, Mozilla FireFox, Opera)	0 € (licencia gratuita)
TOTALES			380 €

Tabla 9: Costes de recursos materiales

4.7.2 Recursos humanos

El desarrollo del proyecto ha sido realizado por el autor del mismo, si bien, se han ido variando los roles dependiendo de la etapa del proyecto.

A continuación se muestran los diferentes perfiles que han sido necesarios durante el desarrollo, incluyendo el coste asociado a cada uno de ellos:

Rol	Responsabilidades	Coste
Responsable del proyecto	<ul style="list-style-type: none"> - Controlar contenidos, estructura y formato del documento final. - Encargado de que las fases del proyecto se ajusten a las fechas establecidas y que la calidad del producto final del proyecto sea adecuada. - Reuniones con el tutor. 	42 €/hora persona 4 días * 8 horas Total 1344 €
Analista	<ul style="list-style-type: none"> - Extraer los requisitos funcionales y no funcionales. - Concretar el dominio del problema, mediante el uso de los diagramas correspondientes. - Elaborar los casos de uso. - Documentar todo el trabajo realizado. 	40 €/hora persona 7 días * 8 horas Total 2240 €
Diseñador	<ul style="list-style-type: none"> - Elaborar la arquitectura lógica 	40 €/hora persona

BLOQUE III. ANÁLISIS

	<ul style="list-style-type: none"> y física. - Reinterpretar el análisis teniendo en cuenta la tecnología disponible. - Construir los diagramas de diseño. - Construir los diagramas de secuencia para los casos de uso. - Definir esquemas de la capa de persistencia. - Documentación de todo el trabajo realizado. 	<p>9 días * 8 horas Total 2880 €</p>
Desarrollador	<ul style="list-style-type: none"> - Codificar las partes definidas en la fase de diseño. - Realización de la depuración. - Documentar todo el trabajo realizado. 	<p>35 €/hora persona 19 días * 8 horas Total 5320 €</p>
Evaluador	<ul style="list-style-type: none"> - Determinar los planes de prueba. - Documentar todo el trabajo realizado. - Encargado de probar la aplicación para asegurar que el resultado final se ajusta a las especificaciones capturadas en análisis. 	<p>40 €/hora persona 3 días * 8 horas Total 960 €</p>
TOTALES	-	12744 €

Tabla 10: Costes por recursos humanos

4.7.3 Costes totales

Recursos	Costes
Recursos materiales	380 €
Recursos humanos	12744 €
Total	13124 €

Tabla 11: Costes totales del proyecto

BLOQUE III. ANÁLISIS

Capítulo 5.

Elicitación de requisitos

5.1 Requisitos de usuario (Objetivos)

A continuación se muestran los principales objetivos:

OBJ-01	Control de usuarios.
Descripción	El sistema deberá distinguir entre los distintos tipos de usuarios.

Tabla 12: Objetivo control de usuario

OBJ-02	Control de usuarios registrados.
Descripción	El sistema deberá controlar el acceso a diferentes funcionalidades para usuarios registrados.

Tabla 13: Objetivo control de usuarios registrados

OBJ-03	Administración de rutas.
Descripción	El sistema deberá administrar la información relativa a los datos de las rutas.

Tabla 14: Objetivo administración de rutas

BLOQUE III. ANÁLISIS

OBJ-04	Sistema multiplataforma
Descripción	El sistema deberá poder usarse en cualquier dispositivo con internet y un navegador.

Tabla 15: Objetivo multiplataforma

5.2 Requisitos funcionales

Los requisitos funcionales que debe tener el sistema se enumeran y explican a continuación.

FRQ-01	Gestión de usuario
Descripción	El sistema deberá almacenar la información correspondiente de un usuario registrado.

Tabla 16: Requisito de gestión de usuario

FRQ-02	Gestión ruta realizada
Descripción	El sistema deberá guardar la información correspondiente a una ruta realizada.

Tabla 17: Requisito de gestión de ruta realizada

FRQ-03	Gestión ruta creada
Descripción	El sistema deberá guardar la información correspondiente a una ruta creada.

Tabla 18: Requisito de gestión de ruta creada

FRQ-04	Realizar ruta
Descripción	El sistema permitirá a los actores "usuario registrado" y "usuario no registrado" realizar una ruta libre.

Tabla 19: Requisito de realizar ruta

FRQ-05	Mostrar rutas
Descripción	El sistema permitirá visualizar las rutas realizadas por cualquier actor del sistema.

Tabla 20: Requisito de mostrar rutas

BLOQUE III. ANÁLISIS

FRQ-06	Crear ruta
Descripción	El sistema permitirá al actor “usuario registrado” crear una ruta manualmente.

Tabla 21: Requisito de crear ruta

FRQ-07	Buscar ruta
Descripción	El sistema permitirá a cualquier actor realizar una búsqueda de las rutas realizadas.

Tabla 22: Requisito de buscar ruta

FRQ-08	Contacto
Descripción	El sistema permitirá poner en contacto a cualquier usuario con el administrador del sistema.

Tabla 23: Requisito de contacto

FRQ-09	Seleccionar ciudad
Descripción	El sistema permitirá a cualquier actor seleccionar la ciudad donde desee realizar o buscar una ruta.

Tabla 24: Requisito de seleccionar ciudad

FRQ-10	Localización
Descripción	El sistema permitirá obtener la localización del usuario.

Tabla 25: Requisito de localización

FRQ-11	Mostrar ruta
Descripción	El sistema permitirá mostrar la ruta realizada hasta el momento en tiempo real.

Tabla 26: Requisito de mostrar ruta

BLOQUE III. ANÁLISIS

FRQ-12	Distancia ruta
Descripción	El sistema permitirá calcular la distancia recorrida en una ruta.

Tabla 27: Requisito de distancia ruta

FRQ-13	Valoración
Descripción	El sistema gestionara las valoraciones de las rutas dependiendo de la calificación de los usuarios.

Tabla 28: Requisito de valoración

FRQ-14	Administración
Descripción	El sistema gestionara un apartado específico de administración para el gestor del sitio web.

Tabla 29: Requisito administración

FRQ-15	Tiempo meteorológico
Descripción	El sistema gestionara las peticiones necesarias para poder mostrar el tiempo actual en la posición del usuario.

Tabla 30: Requisito tiempo meteorológico

5.3 Requisitos no funcionales

A continuación se describen los requisitos no funcionales que se han identificado:

NFR-01	Tiempo de respuesta
Descripción	El sistema deberá responder a las distintas peticiones en menos de 4 segundos.

Tabla 31: Requisito no funcional de tiempo de respuesta

BLOQUE III. ANÁLISIS

NFR-02	Servidor
Descripción	El sistema estará ubicado en un servidor web "Apache".

Tabla 32: Requisito no funcional de servidor

NFR-03	Lenguaje de programación
Descripción	El lenguaje de programación para las páginas dinámicas será PHP.

Tabla 33: Requisito no funcional de lenguaje de programación

NFR-04	Base de datos
Descripción	El sistema guardará sus datos de forma permanente en una base de datos MySQL.

Tabla 34: Requisito no funcional de base de datos

NFR-05	Seguridad
Descripción	El sistema asegurará que la información queda debidamente protegida y que la información será accesible solo a aquellos usuarios que tengan permisos para visualizarla.

Tabla 35: Requisito no funcional de seguridad

NFR-06	Datos Personales
Descripción	El sistema no revelará información personal acerca de los usuarios del sistema.

Tabla 36: Requisito no funcional de datos personales

NFR-07	Copia de Seguridad
Descripción	El sistema deberá realizar una copia de seguridad de la base de datos.

Tabla 37: Requisito no funcional de copia de seguridad

BLOQUE III. ANÁLISIS

NFR-08	Asincronicidad
Descripción	El sistema deberá mostrar los datos en tiempo real sin necesidad de que los actores tengan que solicitar la información de forma explícita.

Tabla 38: Requisito no funcional de asincronicidad

NFR-09	Usabilidad
Descripción	El sistema deberá ser fácil de usar por una persona con conocimientos básicos en el manejo de dispositivos informáticos.

Tabla 39: Requisito no funcional de usabilidad

NFR-09	Compatibilidad
Descripción	El sistema deberá visualizarse y funcionar correctamente en cualquier navegador web, Mozilla Firefox, Google Chrome y Opera Browser.

Tabla 40: Requisito no funcional de compatibilidad

Capítulo 6.

Casos de uso

6.1 Descripción de los actores

Se han identificado 3 tipos de actores que interactuarán con el sistema. A continuación pasamos a describirlos.

Usuario no registrado

ACT - 002	Usuario no registrado
Versión	1.1 [1/06/2015]
Descripción	Este actor representa a un usuario que no está registrado en el sistema y no tiene toda la funcionalidad del mismo

Tabla 41: Actor usuario no registrado

Usuario registrado

ACT - 001	Usuario registrado
Versión	1.1 [1/06/2015]
Descripción	Este actor representa a un usuario registrado que interactúa con el sistema

Tabla 42: Actor usuario registrado

BLOQUE III. ANÁLISIS

Usuario administrador

ACT - 001	Usuario administrador
Versión	1.1 [1/06/2015]
Descripción	Este actor representa a un usuario administrador que gestiona el portal web.

Tabla 43: Actor usuario administrador

6.2 Diagrama de casos de uso

A continuación se muestra el correspondiente diagrama de casos de uso, generado a través de la herramienta Astah:

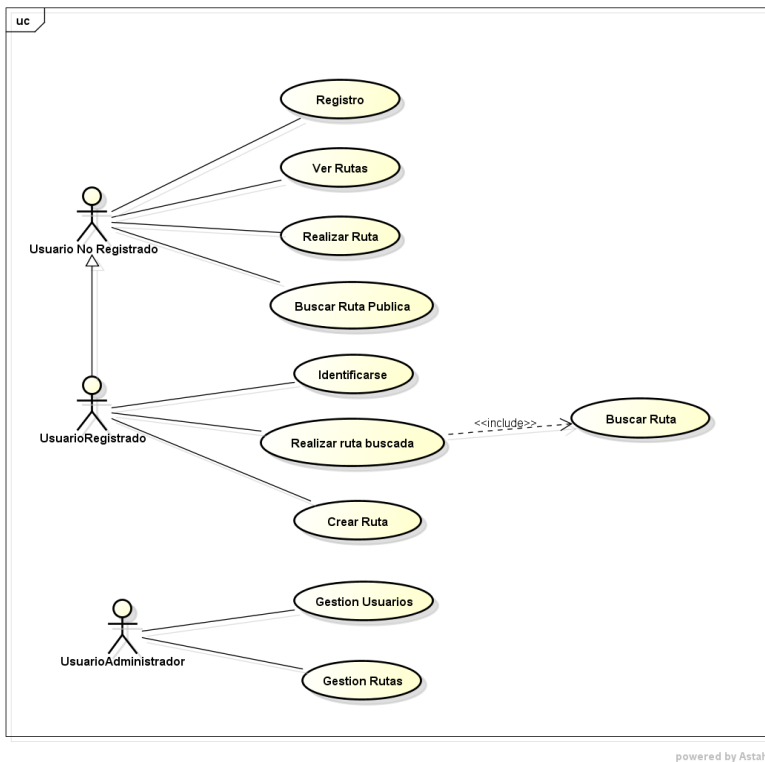


Ilustración 2: Diagrama de casos de uso

BLOQUE III. ANÁLISIS

6.3 Especificación de casos de uso

UC-0001	Crear ruta	
Versión	1.1 [2/06/2015]	
Autor	Álvaro Estébanez Barrena	
Fuentes	Álvaro Estébanez Barrena	
Dependencias		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el actor “usuario registrado” quiera crear una ruta.	
Precondición	El actor “usuario registrado” está identificado en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor “usuario registrado” selecciona la opción crear ruta.
	2	El sistema solicita al actor “usuario registrado” que introduzca los datos de la ruta.
	3	El actor “usuario registrado” introduce los datos que se le han pedido.
	4	El sistema solicita al actor “usuario registrado” que introduzca en el mapa los puntos por donde va transcurrir la ruta.
5	El sistema asigna un identificador a la ruta y la ruta queda registrada en el sistema.	
Postcondición	El actor “usuario registrado” ha creado una nueva ruta.	
Excepciones	Paso	Acción
	2	Si desea salir del proceso, el actor “usuario registrado” cancela la operación y a continuación el caso de uso queda sin efecto.
	3	Los datos introducidos por el actor “usuario registrado” no son válidos, el sistema le indica al actor “usuario registrado” los errores detectados en los datos de entrada y el caso de uso continúa en el paso 2.

Tabla 44: Especificación del caso de uso crear ruta

BLOQUE III. ANÁLISIS

UC-0002	Ver rutas	
Versión	1.1 [2/06/2015]	
Autor	Álvaro Estébanez Barrena	
Fuentes	Álvaro Estébanez Barrena	
Dependencias		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando cualquier actor quiera ver una ruta.	
Precondición	El usuario desea consultar la información de las rutas que se encuentran disponibles y ha realizado una búsqueda.	
Secuencia normal	Paso	Acción
	1	El actor selecciona la opción rutas.
	2	El actor realiza la búsqueda.
	3	El sistema muestra las rutas disponibles.
	4	El actor selecciona la ruta que quiera visualizar.
	5	El sistema muestra en un mapa la ruta deseada.
Postcondición	El actor visualiza la información acerca de la ruta.	
Excepciones	Paso	Acción
	2	Si desea salir del proceso, el actor usuario cancela la operación y a continuación el caso de uso queda sin efecto.
	2	Los datos introducidos por el actor usuario no son válidos, el sistema le indica al actor usuario los errores detectados en los datos de entrada y el caso de uso continúa en el paso 2.

Tabla 45: Especificación del caso de uso ver rutas

UC-0003	Contacto
Versión	1.1 [2/06/2015]
Autor	Álvaro Estébanez Barrena

BLOQUE III. ANÁLISIS

Fuentes	Álvaro Estébanez Barrena	
Dependencias		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando cualquier actor quiera ver el contacto.	
Precondición	El usuario dispone de una dirección de correo electrónico.	
Secuencia normal	Paso	Acción
	1	El actor usuario selecciona la opción contacto.
	2	El sistema solicita al actor usuario que introduzca su nombre, su email, el asunto tratado y el mensaje de la consulta.
	3	El actor usuario introduce los datos que se le han pedido.
	4	El sistema manda un email de consulta al administrador y muestra un mensaje de confirmación de envío.
Postcondición	El administrador gestiona la incidencia.	
Excepciones	Paso	Acción
	2	Si desea salir del proceso, el actor usuario cancela la operación y a continuación el caso de uso queda sin efecto.
	3	Los datos introducidos por el actor usuario no son válidos, el sistema le indica al actor usuario los errores detectados en los datos de entrada y el caso de uso continúa en el paso 2.

Tabla 46: Especificación del caso de uso contacto

UC-0004	Realizar ruta (usuario registrado)
Versión	1.1 [2/06/2015]
Autor	Álvaro Estébanez Barrena
Fuentes	Álvaro Estébanez Barrena
Dependencias	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el actor “usuario registrado” quiera realizar una ruta.

BLOQUE III. ANÁLISIS

Precondición	El actor “usuario registrado” está identificado en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor “usuario registrado” selecciona la opción empezar ruta.
	2	El sistema muestra al actor “usuario registrado” un mensaje de confirmación y comienza la ruta.
	3	El actor “usuario registrado” realiza la ruta.
	4	El actor “usuario registrado” finaliza la ruta.
	5	El sistema solicita al actor “usuario registrado” que introduzca los datos de la ruta.
	6	El actor “usuario registrado” introduce los datos pedidos.
7	El sistema asigna un identificador a la ruta y la ruta queda registrada en el sistema.	
Postcondición	El actor “usuario registrado” ha realizado una ruta.	
Excepciones	Paso	Acción
	3	Si desea salir del proceso, el actor “usuario registrado” cancela la operación y a continuación el caso de uso queda sin efecto.

Tabla 47: Especificación del caso de uso realizar ruta como usuario registrado

UC-0005	Realizar ruta (usuario no registrado)	
Versión	1.1 [2/06/2015]	
Autor	Álvaro Estébanez Barrena	
Fuentes	Álvaro Estébanez Barrena	
Dependencias		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el actor “usuario no registrado” quiera realizar una ruta.	
Precondición		
Secuencia normal	Paso	Acción

BLOQUE III. ANÁLISIS

	1	El actor "usuario no registrado" selecciona la opción empezar ruta.
	2	El sistema muestra al actor "usuario no registrado" un mensaje de confirmación, de aviso que la ruta no puede ser almacenada y comienza la ruta.
	3	El actor "usuario no registrado" realiza la ruta.
	4	El actor "usuario no registrado" finaliza la ruta.
	5	El sistema muestra un mensaje de ruta finalizada y termina la ruta.
Postcondición	El actor "usuario no registrado" ha realizado una ruta.	
Excepciones	Paso	Acción
	3	Si desea pausar la ruta, el actor "usuario no registrado" pulsa pausar y la realización de la ruta queda temporalmente suspendida.
	3	Si desea salir del proceso, el actor "usuario no registrado" cancela la operación y a continuación el caso de uso queda sin efecto.

Tabla 48: Especificación del caso de uso realizar ruta como usuario no registrado

UC-0006	Buscar ruta publica	
Versión	1.1 [2/06/2015]	
Autor	Álvaro Estébanez Barrena	
Fuentes	Álvaro Estébanez Barrena	
Dependencias		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando cualquier actor quiera buscar una ruta.	
Precondición		
Secuencia normal	Paso	Acción
	1	El actor selecciona la opción buscar ruta.
	2	El sistema solicita al actor que introduzca el nombre de la ruta, o la fecha, o la duración, o la dificultad o la valoración.

BLOQUE III. ANÁLISIS

	3	El actor usuario introduce los datos que se le han pedido.
	4	El sistema muestra la información de las rutas que coinciden con los parámetros de búsqueda.
	5	El actor puede hacer click sobre cualquiera de las rutas para verla en el mapa.
Postcondición	El actor usuario ha creado buscado una ruta.	
Excepciones	Paso	Acción
	2	Si desea salir del proceso, el actor cancela la operación y a continuación el caso de uso queda sin efecto.
	3	Los datos introducidos por el actor no son válidos, el sistema le indica al actor los errores detectados en los datos de entrada y el caso de uso continúa en el paso 2.

Tabla 49: Especificación del caso de uso buscar ruta pública

UC-0007	Buscar ruta privada (usuario registrado)	
Versión	1.1 [2/06/2015]	
Autor	Álvaro Estébanez Barrena	
Fuentes	Álvaro Estébanez Barrena	
Dependencias		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el actor "usuario registrado" quiera buscar sus rutas.	
Precondición		
Secuencia normal	Paso	Acción
	1	El actor "usuario registrado" selecciona la opción mis rutas.
	2	El sistema comprueba que existen rutas de dicho actor "usuario registrado" y las muestra.
	3	El actor "usuario registrado" introduce los datos de búsqueda concretos.
	4	El sistema muestra la información de las rutas que coinciden con los parámetros de búsqueda.

BLOQUE III. ANÁLISIS

	5	El actor “usuario registrado” puede hacer click sobre cualquiera de las rutas para verla en el mapa.
Postcondición	El actor “usuario registrado” ha buscado una ruta propia.	
Excepciones	Paso	Acción
	2	Si desea salir del proceso, el actor “usuario registrado” cancela la operación y a continuación el caso de uso queda sin efecto.
	3	Los datos introducidos por el actor “usuario registrado” no son válidos, el sistema le indica al actor los errores detectados en los datos de entrada y el caso de uso continúa en el paso 2.

Tabla 50: Especificación del caso de uso buscar ruta privada usuario registrado

UC-0008	Buscar ruta privada (usuario no registrado)	
Versión	1.1 [2/06/2015]	
Autor	Álvaro Estébanez Barrena	
Fuentes	Álvaro Estébanez Barrena	
Dependencias		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el actor “usuario no registrado” quiera buscar sus rutas (apartado mis rutas).	
Precondición		
Secuencia normal	Paso	Acción
	1	El actor “usuario no registrado” selecciona la opción mis rutas
	2	El sistema avisa al actor “usuario no registrado” que no puede mostrar sus rutas porque no está en el sistema. A continuación le expone el formulario de registro.
	3	El actor “usuario no registrado” introduce los datos que se le han pedido.
	4	El sistema registra al usuario.
Postcondición		

BLOQUE III. ANÁLISIS

Excepciones	Paso	Acción
	2	Si desea salir del proceso, el actor cancela la operación y a continuación el caso de uso queda sin efecto.
	3	Los datos introducidos por el actor no son válidos, el sistema le indica al actor los errores detectados en los datos de entrada y el caso de uso continúa en el paso 2.

Tabla 51: Especificación del caso de uso buscar ruta privada usuario no registrado

UC-0009	Registro usuario	
Versión	1.1 [2/06/2015]	
Autor	Álvaro Estébanez Barrena	
Fuentes	Álvaro Estébanez Barrena	
Dependencias		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el actor “usuario no registrado” quiera registrarse en la aplicación.	
Precondición	El actor “usuario no registrado” no está identificado en el sistema y desea registrarse en la aplicación.	
Secuencia normal	Paso	Acción
	1	El actor “usuario no registrado” selecciona la opción Regístrate!
	2	El sistema muestra un formulario con los datos necesarios.
	3	El actor “usuario no registrado” introduce los datos que se le han pedido.
	4	El sistema valida la información que el actor “usuario no registrado” ha proporcionado.
5	El sistema introduce almacena los datos del nuevo usuario.	
Postcondición	El actor “usuario no registrado” queda registrado en el sistema.	
Excepciones	Paso	Acción

BLOQUE III. ANÁLISIS

	2	Si desea salir del proceso, el actor usuario no registrado cancela la operación y a continuación el caso de uso queda sin efecto.
	3	Los datos introducidos por el actor usuario no registrado no son válidos, el sistema le indica al actor usuario no registrado los errores detectados en los datos de entrada y el caso de uso continúa en el paso 2.

Tabla 52: Especificación del caso de uso registro de usuario

UC-0010	Identificar usuario	
Versión	1.1 [2/06/2015]	
Autor	Álvaro Estébanez Barrena	
Fuentes	Álvaro Estébanez Barrena	
Dependencias		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el actor "usuario registrado" quiera identificarse en el sistema: La aplicación procede a comprobar que los datos del usuario se encuentran almacenados, la contraseña es correcta y puede acceder a la web.	
Precondición	El actor "usuario registrado" está dado de alta en el sistema.	
Secuencia normal	Pa	Acción
	1	El actor "usuario registrado" selecciona la opción identificarse.
	2	El sistema muestra al actor "usuario registrado" un formulario con los datos necesarios.
	3	El actor "usuario registrado" introduce los datos solicitados por el sistema.
	4	El sistema valida la información proporcionada por el usuario.
Postcondición	El actor usuario queda identificado en el sistema.	
Excepciones	Pa	Acción
	4	Si los datos no son correctos, muestra un aviso de error.

Tabla 53: Especificación del caso de uso identificar usuario

BLOQUE III. ANÁLISIS

UC-0011	Gestion usuarios	
Versión	1.1 [2/06/2015]	
Autor	Álvaro Estébanez Barrena	
Fuentes	Álvaro Estébanez Barrena	
Dependencias		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el actor “administrador” quiera gestionar los usuarios.	
Precondición	El actor está identificado en el sistema como admin	
Secuencia normal	Paso	Acción
	1	El actor “administrador” selecciona la opción Gestion Usuarios.
	2	El sistema comprueba que existen usuarios y los muestra.
	3	El actor “administrador” modifica los datos deseados del usuario.
	4	El actor “administrador” confirma los cambios realizados.
	5	El sistema muestra un mensaje de aviso y almacena los cambios.
Postcondición	El actor “administrador” ha actualizado atributos de los usuarios.	
Excepciones	Paso	Acción
	3	Si desea salir del proceso, el actor “administrador” cancela la operación y a continuación el caso de uso queda sin efecto.
	3	Los datos introducidos por el actor “administrador” no son válidos, el sistema le indica al actor los errores detectados en los datos de entrada y el caso de uso continúa en el paso 3.

Tabla 54: Especificación del caso de uso gestión de usuarios

UC-0012	Gestion rutas
Versión	1.1 [2/06/2015]
Autor	Álvaro Estébanez Barrena

BLOQUE III. ANÁLISIS

Fuentes	Álvaro Estébanez Barrena	
Dependencias		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el actor “administrador” quiera gestionar las rutas.	
Precondición	El actor está identificado en el sistema como admin	
Secuencia normal	Paso	Acción
	1	El actor “administrador” selecciona la opción Gestion Rutas.
	2	El sistema comprueba que existen rutas y las muestra.
	3	El actor “administrador” elimina las rutas deseadas.
	4	El sistema muestra un mensaje de aviso y almacena los cambios.
Postcondición		
Excepciones	Paso	Acción
	3	Si desea salir del proceso, el actor “administrador” cancela la operación y a continuación el caso de uso queda sin efecto.

Tabla 55: Especificación del caso de uso gestión de rutas

6.4 Modelo de dominio

En esta sección se presentan las clases que se han descubierto en una primera aproximación para conocer que debe realizar la aplicación. Una vez se haya mostrado el diagrama se realizara una explicación breve contiendo las clases existen y como se relacionan entre ellas, a la vez que se define las responsabilidades de cada una de ellas.

El diagrama de clases de dominio se puede observar en la ilustración 3.

BLOQUE III. ANÁLISIS

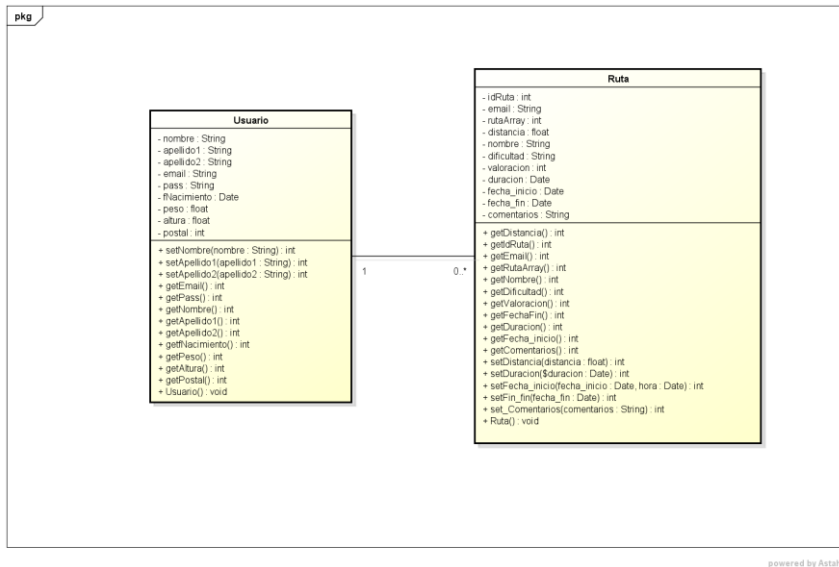


Ilustración 3: Clases del dominio

La clase Usuario, será la encargada de la gestión relativa a los usuarios. Es una clase principal en el sistema ya que se estará instanciada durante toda la funcionalidad del mismo. Un Usuario podrá crear o realizar muchas rutas, y cada una de ellas pertenecerá al usuario que la realizó.

La clase Ruta, es la responsable de cada una de las diferentes rutas. Se trata de una clase genérica que comprende los atributos y métodos básicos y necesarios para componer una ruta. Orientada en un futuro a la posibilidad de aplicar Herencia, para crear futuras clases más específicas dependiendo de la modalidad de deporte o aventura hacia donde la aplicación evolucione.

Bloque IV.

Diseño

BLOQUE IV. DISEÑO

Capítulo 7. Diseño de persistencia

7.1 Decisiones de diseño de la base de datos

A continuación se dará una detallada visión del diseño de la base de datos que se ha escogido y utilizado para la persistencia de los datos.

Durante el desarrollo del diseño de la BBDD se han tenido en cuenta los pilares fundamentales del diseño de BBDD, para tratar de evitar los problemas más frecuentes con el manejo de BBDD, como por ejemplo, asegurar la no redundancia de datos y la integridad de los mismos.

7.2 Diagrama entidad-relación

En esta sección se muestra el diagrama de relaciones entre las tablas que componen la BBDD:

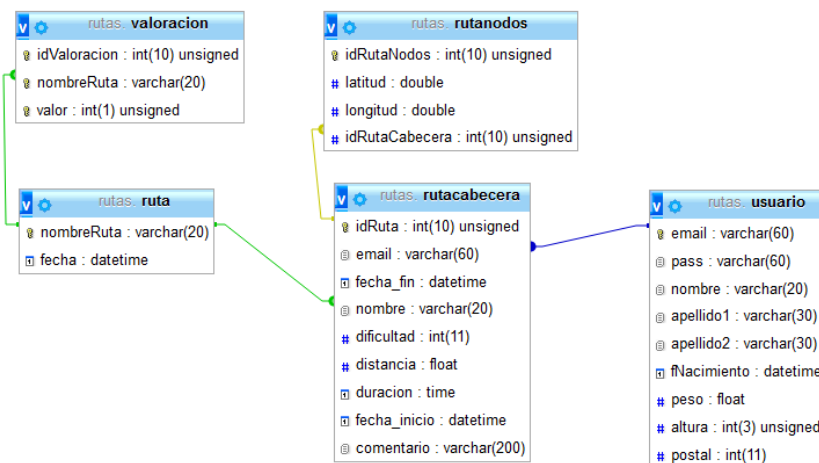


Ilustración 4: Diagrama entidad-relación

BLOQUE IV. DISEÑO

7.2.1 usuario

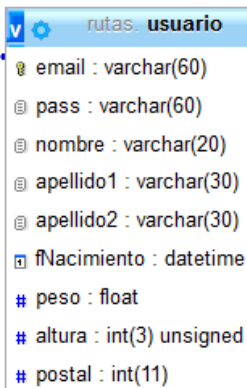


Ilustración 5: Tabla usuario

Descripción	Esta tabla guarda información de los usuarios registrados en el sistema.	
Relaciones	rutacabecera: cada usuario tiene un único email en el sistema.	
Campos	email	Dirección email del usuario e identificador para autenticarse en el sistema.
	pass	Contraseña para poder hacer login en el sistema.
	nombre	Nombre de pila del usuario.
	apellido1	Primer apellido del usuario.
	apellido2	Segundo apellido del usuario.
	fNacimiento	Fecha de nacimiento del usuario.
	peso	Peso en kilogramos del usuario.
	altura	Altura en centímetros del usuario.
	postal	Código postal del lugar de residencia del usuario.

Tabla 56: Especificación tabla usuario

BLOQUE IV. DISEÑO

7.2.2 ruta



Ilustración 6: Tabla ruta

Descripción	Esta tabla guarda información de del nombre y fecha de una ruta.	
Relaciones	rutacabecera: una ruta puede realizarse 0 o muchas veces. valoracion: una ruta tiene una valoración que será la media de todas las valoraciones de esa ruta realizada por los usuarios.	
Campos	nombreRuta	Nombre único que identifica una ruta.
	fecha	Fecha de creación de la ruta.

Tabla 57: Especificación tabla ruta

7.2.3 rutacabecera

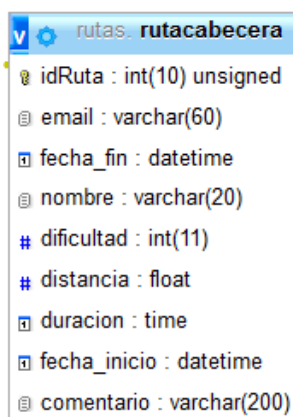


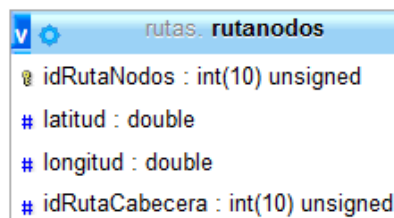
Ilustración 7: Tabla rutacabecera

BLOQUE IV. DISEÑO

Descripción	Esta tabla guarda información acerca del detalle de una ruta.	
Relaciones	usuario: una ruta es realizada por uno o muchos usuarios ruta: el detalle de una ruta pertenece a una única ruta rutanodos: una ruta está compuesta por 1 o muchos nodos	
Campos	idRuta	Identificador único para cada ruta.
	email	Email del usuario que realiza la ruta
	fecha_fin	Fecha de terminación de la ruta.
	nombre	Nombre de la ruta.
	dificultad	Dificultad de la ruta según el usuario.
	valoración	Valoración que el usuario da a una ruta.
	distancia	Distancia total de la ruta.
	duracion	Tiempo total de la ruta.
	comentario	Comentario acerca de la ruta

Tabla 58: Especificación tabla rutacabecera

7.2.4 rutanodo



Column	DataType
idRutaNodos	int(10) unsigned
latitud	double
longitud	double
idRutaCabecera	int(10) unsigned

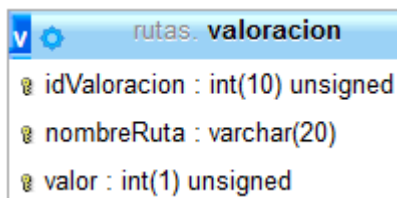
Ilustración 8: Tabla rutanodos

BLOQUE IV. DISEÑO

Descripción	Esta tabla guarda información de los nodos de una ruta.	
Relaciones	rutacabecera: una ruta está compuesta por uno o muchos nodos.	
Campos	idRutaNodos	Identificador único de un nodo.
	latitud	Latitud concreta de un nodo.
	longitud	Longitud concreta de un nodo.
	idRutaCabecera	Un nodo pertenece a una ruta.

Tabla 59: Especificación rutanodos

7.2.5 valoración



```

v  rutas.valoracion
  idValoracion : int(10) unsigned
  nombreRuta   : varchar(20)
  valor        : int(1) unsigned

```

Ilustración 9: Tabla de valoración

Descripción	Esta tabla guarda información acerca de la valoración de las rutas.	
Relaciones	ruta: cada ruta tiene una valoración que será la media aritmética de las veces que dicha ruta ha sido realizada por los usuarios.	
Campos	idValoracion	Identificador único de cada valoración.
	nombre	Nombre de la ruta que se valora.
	valor	Valor de la valoración del 1 al 5.

Tabla 60: Especificación de la tabla valoración

BLOQUE IV. DISEÑO

Capítulo 8.

Arquitectura del sistema

8.1 Introducción

En este capítulo se va a detallar el proceso de diseño de la arquitectura del sistema.

En la primera parte del capítulo se elaborará un diseño arquitectónico del sistema, incluyendo aspectos organizacionales, una subdivisión del sistema completo en subsistemas lógicos que servirán para comprender qué elementos físicos se necesitan para completar el sistema final. Se irán aplicando patrones de diferentes referencias que ayudarán a diseñar el sistema de modo que cumpla con las condiciones de calidad de software de forma probada y documentada.

Una vez identificados los subsistemas, se procederá a la realización de los casos de uso principales.

8.2 Descripción del sistema

8.2.1 Diseño arquitectónico

En esta y en las sucesivas secciones se hará un diseño arquitectónico del sistema milrutas obteniendo una serie de subsistemas que compondrán el sistema completo, junto con las relaciones entre los mismos.

El diseño debe ser realizado de forma que se cumplan las siguientes condiciones de forma que haya un equilibrio entre todas ellas:

- *Rendimiento y mantenimiento*: Se intentará hacer que el rendimiento y el mantenimiento del sistema estén en equilibrio de forma que los componentes sean

BLOQUE IV. DISEÑO

del suficiente grano grueso como para ofrecer un buen rendimiento y al mismo tiempo el suficiente grano fino como para que un cambio o una ampliación del sistema produzca los menores trastornos posibles.

- *Protección:* Se protegerán las partes más sensibles del sistema (los datos) en una capa interna añadiendo niveles de protección para llegar a ella.
- *Seguridad:* El sistema a de asegurar con el mayor grado posible la confidencialidad, integridad y disponibilidad de los datos.
- *Disponibilidad:* el diseño estará orientado a la posible replicación de ciertos componentes tales como los servidores de BBDD y Web en un futuro.

8.2.2 Organización del sistema

Para la organización más elemental del sistema se utilizará uno de los modelos más extendidos: el modelo cliente-servidor. Con los requisitos y los casos de uso de la aplicación es el modelo más adecuado para un sistema de estas características, en el que sus elementos se interrelacionan por medio de internet, así, este modelo tiene la ventaja de ser una arquitectura distribuida, es posible hacer una separación de los elementos diferenciados más sencilla, mantenible y segura. De esta forma ya se diferencian 2 grandes subsistemas: los clientes, y el servidor.

Se intentará integrar el modelo cliente-servidor con un modelo de repositorio de datos, ya que todos los datos compartidos se almacena en una base de datos central a la cual pueden acceder todos los subsistemas directa, o indirectamente.

También se aplicará un modelo de capas a todo el sistema separando la capa de datos, la capa de aplicación y la capa de presentación. De esta forma, cada capa sólo puede comunicarse con sus capas adyacentes añadiéndose un nivel de protección extra al estar los datos en la capa más interna de la aplicación. Así tendremos un modelo cliente-servidor de tres capas con repositorio de datos. De esta forma, el servidor queda descompuesto en otros dos subsistemas, uno que se encargará de la lógica de los datos y el otro de la lógica de aplicación.

Modelo de cliente:

Cliente: cada uno de los clientes que se conecten al servicio por medio de internet, en estos se delega la menor responsabilidad posible, limitándose a implementar la lógica de presentación por medio del navegador web y a obtener la localización del usuario.

Quedan identificados 3 grandes subsistemas que formarán el sistema completo de una forma lógica, que se asociarán a cada una de las capas del modelo cliente-servidor de tres capas:

BLOQUE IV. DISEÑO

- Lógica de presentación: Hará las funciones de intermediario entre los usuarios y el sistema permitiendo el intercambio de información entre ambos. Tiene que ser capaz de mostrar los datos a los usuarios y recoger consultas de los mismos para su procesamiento en la siguiente capa.
- Lógica de negocio: Se encargará de toda la gestión interna del programa. Debe ser capaz de obtener datos de la capa de presentación y llevar a cabo la gestión de todo lo referente al procesamiento de los datos para que se cumplan los objetivos de la especificación.
- Persistencia: Este subsistema será el asociado a la última capa del modelo, contendrá todos los datos que necesita la aplicación para su funcionamiento acorde a las especificaciones.

En la ilustración 10 se observa el diagrama que describe la arquitectura en capas mencionado:

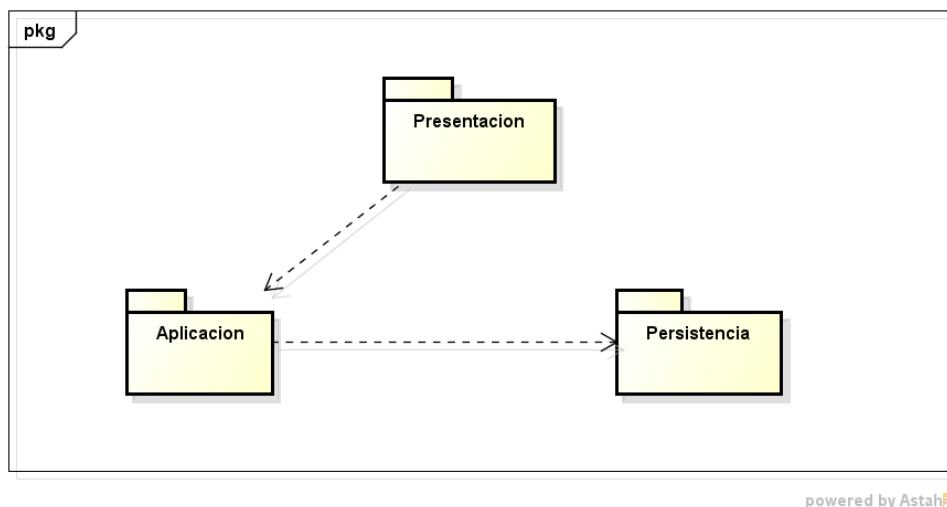


Ilustración 10: Diagrama de capas del sistema

8.2.3 Descomposición modular

Una vez que se han diferenciado los subsistemas básicos a un nivel conceptual del sistema, se verá una descomposición en componentes de los mismos. Estos componentes en ocasiones son fundamentales en el funcionamiento de otros componentes. De esta forma el conjunto de funcionalidades delegadas a los componentes hacen posible el funcionamiento global del sistema.

BLOQUE IV. DISEÑO

Esta división del sistema al completo se representa en la ilustración 11, que no es más que diagrama de componentes siguiendo una estrategia de descomposición orientada a objetos. Se trata de un conjunto componentes que se comunican entre sí.

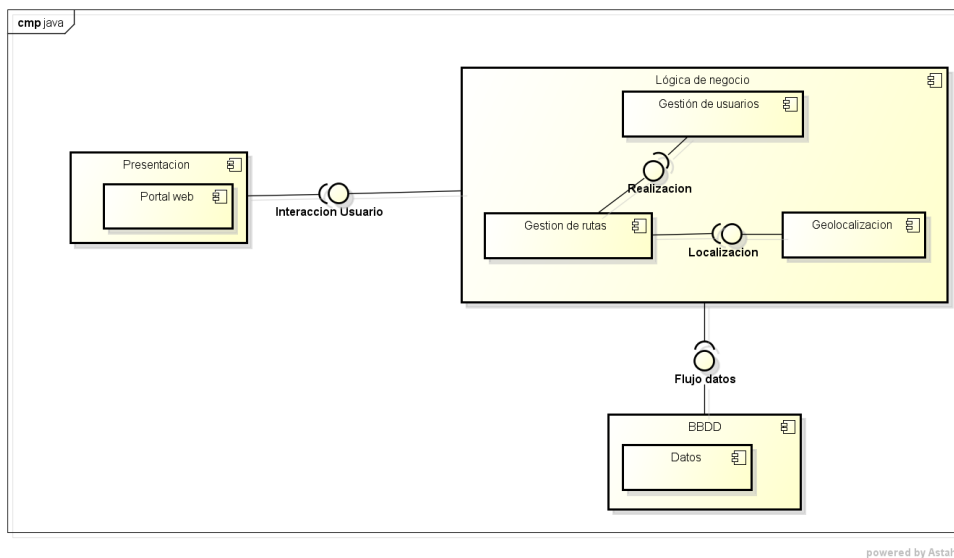


Ilustración 11: Diagrama de componentes

A continuación detallamos cada uno de los componentes que aparecen en la ilustración 11:

- *Presentación* Se encarga de interactuar con el usuario de la aplicación. Se comunica con la lógica de la aplicación informando de la petición que el cliente lanza e informando al usuario una vez se haya procesado dicha petición.
- *BBDD* Este componente contiene los datos que deben mantenerse de forma persistente en la aplicación. Se encuentra dentro del componente de persistencia
- *Lógica de negocio* Administra y se encarga de la funcionalidad de la aplicación, separado en 3 componentes. Gestión de usuarios será el responsable del manejo de las actividades básicas de los clientes de la aplicación; gestión de rutas realiza las tareas con los diferentes itinerarios que los usuarios pueden realizar y por último, geolocalización es el conjunto de todas las APIs que se encargaran de obtener la posición del usuario.

8.2.4 Flujo de control

BLOQUE IV. DISEÑO

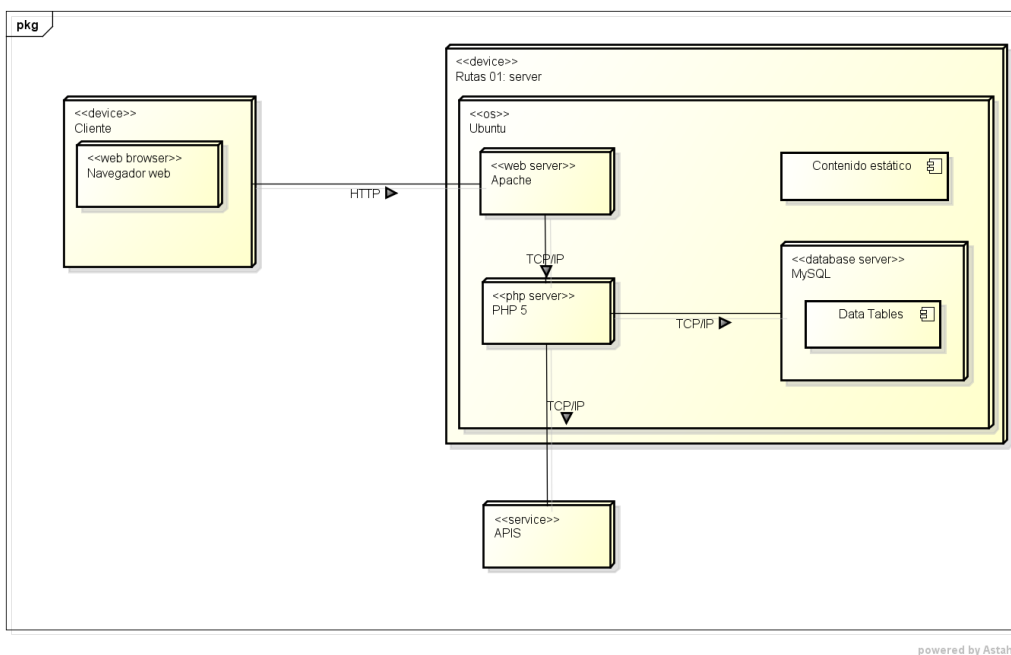
Para la parte del sistema encargada de llevar la gestión de las transacciones realizadas desde el portal web se utilizará un modelo de control centralizado basado en un modelo del gestor al tratarse de un sistema concurrente.

8.2.5 Diseño físico

Con el análisis a nivel conceptual obtenido de las anteriores secciones, ya se dispone de un punto de partida más cercano a la solución del problema con una serie de elementos bien diferenciados y definidos y sus interrelaciones. Se intentará por medio de la tecnología disponible ajustarse de la forma más precisa al problema.

En el siguiente diagrama de despliegue se muestra la propuesta para los diferentes dispositivos que conformarán el sistema y los protocolos que se utilizarán para comunicar

A continuación pasamos a detallar los elementos presentados en la ilustración 12:



powered by Astah

Ilustración 12: Diagrama de despliegue

- *Servidor web*: es el encargado de recibir las peticiones por parte del usuario y de enviar al servidor PHP los ficheros con los que tiene que trabajar.

BLOQUE IV. DISEÑO

- *Servidor PHP*: este elemento se encargará de interpretar el código PHP para producir respuestas de forma dinámica al usuario.
- *Servidor de base de datos*: este elemento es el encargado de almacenar y gestionar los datos de forma persistente, tales como rutas o usuarios.
- *Contenido estático*: el contenido estático es aquel que no cambiará, como por ejemplo las imágenes de la web, o las plantillas para las vistas del interfaz de usuario.
- *Servicios API*: es el responsable de las gestiones con los servidores externos para la localización, redes sociales y conexiones asíncronas.

Nota: Vemos que en la parte del servidor hemos optado por introducir, el servidor web, el servidor PHP y el servidor MySQL en la misma máquina, sin embargo, nada impediría que esos elementos estuviesen en máquinas físicas diferentes.

8.3 Modelos del sistema

Se verán en detalle los modelos del sistema resultantes. En primer lugar, un modelo estático con las clases que formarán la aplicación para así tener un punto de partida en los modelos dinámicos, que definirán cómo se interrelacionan los objetos de estas clases en tiempo de ejecución para implementar una determinada funcionalidad.

8.3.1 Patrones utilizados

A continuación enumeramos los patrones utilizados durante la etapa de diseño y damos una breve explicación de por qué hemos utilizado estos patrones:

MVC (Modelo-vista-controlador) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

BLOQUE IV. DISEÑO

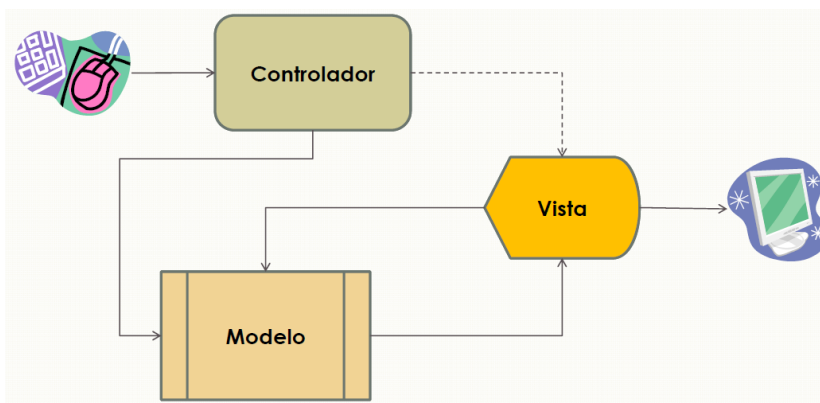


Figura 6: Modelo Vista Controlador

En la figura 6 podemos ver como se relacionan el modelo, la vista y el controlador. En un caso típico la secuencia de llamadas entre estos elementos es la siguiente: El usuario solicita hacer cierta acción, el controlador recibe esta petición y llama a la función del modelo que corresponda para después pasar a la vista los datos obtenidos desde el modelo.

Se ha utilizado este patrón de diseño ya que es uno de los más utilizados para la creación de páginas web y además permite diferenciar perfectamente entre el modelo de dominio, la persistencia de los datos y la interfaz gráfica de usuario.

Controlador frontal. Es un patrón de diseño que se basa en usar un controlador como punto inicial, que denominaremos controlador frontal, para la gestión de las peticiones. El controlador frontal gestiona estas peticiones, y realiza algunas funciones como: comprobación de restricciones

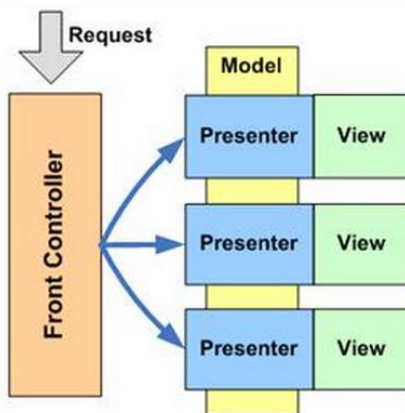


Figura 7: Controlador frontal

BLOQUE IV. DISEÑO

de seguridad, manejo de errores, mapear y delegación de las peticiones a otros componentes de la aplicación que se encargarán de generar la vista adecuada para el usuario. La figura 7 muestra un esquema de ello.

DataMapper El patrón datamapper es un patrón que nos permite separar de forma clara, el modelo de dominio de la persistencia de datos. Se basa en la idea de que cada clase del modelo de dominio es representada por una clase datamapper para gestionar los datos entre la capa de lógica de la aplicación y la persistencia de los datos. Las clases datamapper son las encargadas de insertar, modificar, recuperar... objetos desde la base de datos hacia la memoria de la aplicación.

Para entender mejor este último párrafo pondremos un sencillo ejemplo. Tenemos una clase llamada Usuario que modela la información de un usuario de la aplicación. Al tener esta clase tendremos otra que podemos llamar DataMapperUsuario que contendrá métodos que permitan mover objetos de tipo Usuario desde/hacia la base de datos. La clase Usuario no es consciente de que existe una clase que está moviendo objetos de su tipo desde o hacia la base de datos y es esto precisamente lo que da potencia a este patrón, ya que estamos separando la lógica de la aplicación del acceso a la capa de persistencia de datos. El datamapper es el encargado de que exista una coherencia entre los objetos en memoria y la base de datos.

En la ilustración 13 vemos un diagrama que muestra el concepto del patrón DataMapper.

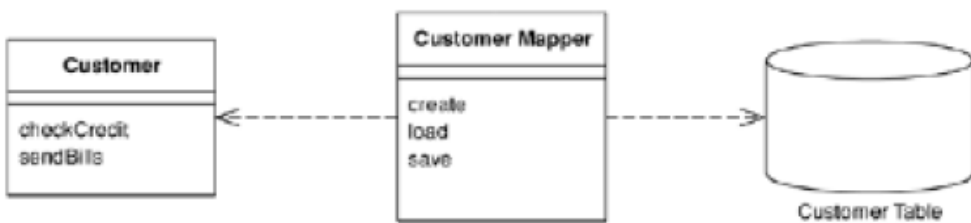


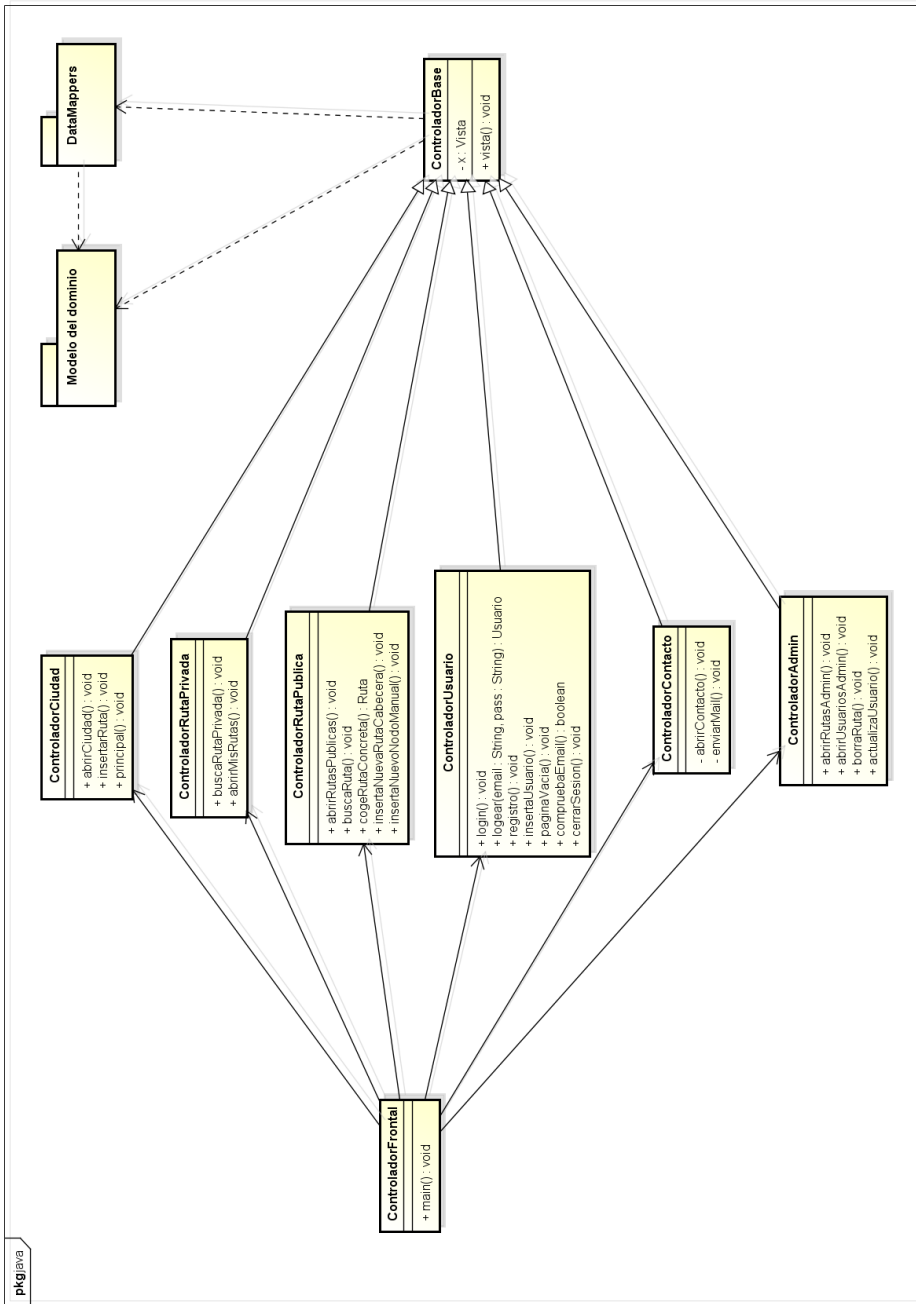
Ilustración 13: Patrón DataMapper

8.3.2. Modelos estáticos

En este apartado se documentan los diagramas de las clases de diseño. Para una mejor interpretación, se ha optado por dividir el diagrama global en dos, en la ilustración 22 se muestra los datamappers de la aplicación y en la ilustración 15 se detallan los controladores.

BLOQUE IV. DISEÑO

Diagrama de clases de los controladores



powered by Astah

Ilustración 15: Diagrama de clases de los controladores

BLOQUE IV. DISEÑO

A continuación se describen con más detalle cada uno de los controladores.

ControladorUsuario:

ControladorUsuario
+ login() : void + logear(email : String, pass : String) : Usuario + registro() : void + insertaUsuario() : void + paginaVacía() : void + compruebaEmail() : boolean + cerrarSesion() : void

Ilustración 16: ControladorUsuario

Explicación de la clase:

Definición	La clase ControladorUsuario es la responsable de la gestión relacionada con los usuarios del sistema	
Métodos	login()	Recoge los datos introducidos en la vista menuLogin (formulario) y llama a logear(), para permitir la autenticación en caso de que los datos sean correctos.
	logear()	Comprueba que si el usuario está registrado, y en tal caso, si el pass es correcto, nos devuelve el Usuario.
	registro()	Llama a la vista para mostrar el registro de usuario.
	insertaUsuario()	Recoge los parámetros del registro de usuario, encripta la pass, crea un nuevo Usuario y lo almacena.
	paginaVacía()	Muestra página vacía.
	compruebaEmail()	Verifica que el email de registro de usuario no está en el sistema.
	cerrarSesion()	Destruye el objeto sesión y muestra la página principal.

Tabla 61: Descripción de la clase ControladorUsuario

BLOQUE IV. DISEÑO

ControladorRutaPublica:

ControladorRutaPublica
+ abrirRutasPublicas() : void + buscaRuta() : void + cogeRutaConcreta() : Ruta + insertaNuevaRutaCabecera() : void + insertaNuevoNodoManual() : void

Ilustración 17: ControladorRutaPublica

Explicación de la clase:

Definición	La clase ControladorRutaPublica se encarga de gestión de las rutas que todos los usuarios pueden ver y realizar.	
Métodos	abrirRutasPublicas	Muestra las rutas existentes en el sistema.
	buscaRuta	Recoge los parámetros de búsqueda (del buscador), y comprueba si existe alguna ruta, mostrándola en tal caso.
	cogeRutaConcreta	Obtiene los detalles de una ruta (nodos) y se los manda a Ajax por medio de Json.
	insertaNuevaRutaCabecera	Recoge los parámetros de una nueva ruta, asigna los valores correspondientes y almacena la ruta.
	insertaNuevoNodoManual	Recoge un nodo introducido manualmente (en el mapa) y lo guarda.

Tabla 62: Descripción de la clase ControladorRutaPublica

ControladorRutaPrivada:

ControladorRutaPrivada
+ abrirMisRutas() : void + buscaRutaPrivada() : void

Ilustración 18: ControladorRutaPrivada

BLOQUE IV. DISEÑO

Explicación de la clase:

Definición	La clase ControladorRutaPrivada se encarga de las rutas concretas de cada usuario	
Métodos	abrirMisRutas	Muestra las rutas de un usuario registrado.
	buscarRutaPrivada	Busca las rutas de un usuario registrado.

Tabla 63: Descripción de la clase ControladorRutaPrivada

ControladorCiudad:

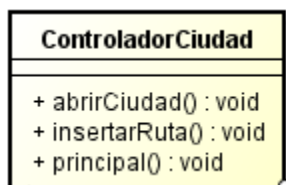


Ilustración 19: ControladorCiudad

Explicación de la clase:

Definición	La clase ControladorCiudad se encarga de la gestión de las ciudades (no visible) y de la creación de las rutas en cada una.	
Métodos	abrirCiudad	Nos abre la vista correspondiente a la ciudad.
	insertaRuta	Guarda una ruta completa.
	principal	Nos muestra la vista principal.

Tabla 64: Descripción de la clase ControladorCiudad

ControladorContacto:

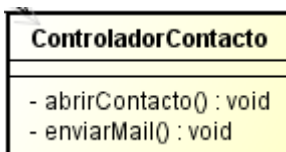


Ilustración 20: ControladorContacto

BLOQUE IV. DISEÑO

Explicación de la clase:

Definición	La clase ControladorContacto administra el contacto del sistema.	
Métodos	abrirContacto	Nos muestra el contacto con el formulario para el envío del email.
	enviarMail	Envía un email al encargado del contacto.

Tabla 65: Descripción de la clase ControladorContacto

ControladorAdmin:

ControladorAdmin
+ abrirRutasAdmin() : void + abrirUsuariosAdmin() : void + borraRuta() : void + actualizaUsuario() : void

Ilustración 21: ControladorAdmin

Explicación:

Definición	La clase ControladorAdmin gestiona la parte del administrador.	
Métodos	abrirRutasAdmin	Nos muestra las rutas del sistema.
	abrirUsuariosAdmin	Muestra los usuarios del sistema.
	borrarRuta	Elimina la ruta seleccionada por el administrador.
	actualizaUsuario	Actualiza los datos modificados de un usuario en concreto

Tabla 66: Descripción de la clase ControladorAdmin

Diagrama de clases de los Datamappers

En este punto se detallan cada uno de los datamappers.

BLOQUE IV. DISEÑO

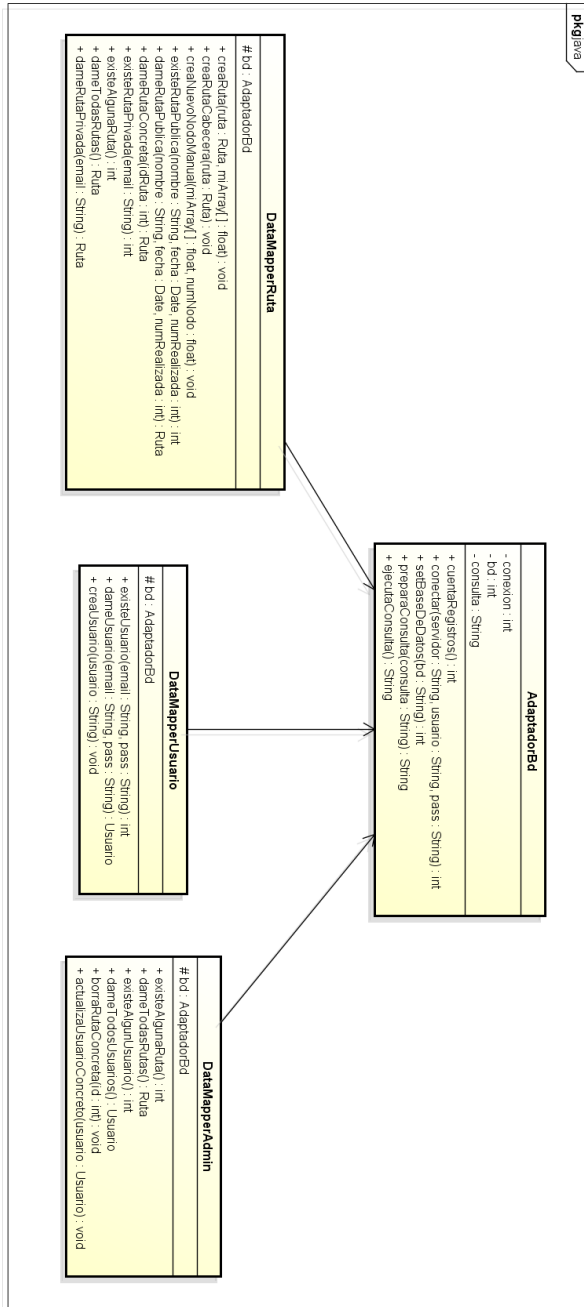


Ilustración 22 : Diagrama de clases de los DataMappers

BLOQUE IV. DISEÑO

DataMapperRuta

DattaMapperRuta
bd : AdaptadorBd
+ creaRuta(ruta : Ruta, miArray[] : float) : void + creaRutaCabecera(ruta : Ruta) : void + creaNuevoNodoManual(miArray[] : float, numNodo : float) : void + existeRutaPublica(nombre : String, fecha : Date, numRealizada : int) : int + dameRutaPublica(nombre : String, fecha : Date, numRealizada : int) : Ruta + dameRutaConcreta(idRuta : int) : Ruta + existeRutaPrivada(email : String) : int + existeAlgunaRuta() : int + dameTodasRutas() : Ruta + dameRutaPrivada(email : String) : Ruta

Ilustración 23: DataMapperRuta

Explicación de la clase:

Definición	La clase DataMapperRuta se encarga de las transacciones de objetos de tipo Ruta hacia/desde la base de datos y, en general, realizar acciones que requieren acceso a la base de datos para objetos de tipo Ruta.	
Métodos	creaRuta	Inserta en las tablas correspondientes de la BBDD los datos de una ruta pasados por parámetro adaptando las fechas al tipo de MySQL
	creaRutaCabecera	Inserta los parámetros de una ruta creada manualmente (aún no se ha realizado) en la BBDD.
	creaNuevoNodoManual	Inserta un nodo manual (no realizando una ruta) en la BBDD.
	existeRutaPublica	Comprueba si existe una ruta (o conjuntos de rutas) en la BBDD.
	dameRutaPublica	Devuelve los parámetros almacenados en la BBDD de una ruta (o conjuntos de rutas).
	dameRutaConcreta	Devuelve los nodos (detalles de una ruta) concretos.
	existeRutaPrivada	Comprueba si existen rutas de un usuario determinado.

BLOQUE IV. DISEÑO

	existeAlgunaRuta	Comprueba si existe alguna ruta.
	dameTodasRutas	Devuelve todas las rutas del sistema.
	dameRutaPrivada	Nos devuelve los parámetros de las rutas concretas de un usuario.

Tabla 67: Descripción DataMapperRuta

DataMapperUsuario:

DattaMapperUsuario
bd : AdaptadorBd
+ existeUsuario(email : String, pass : String) : int
+ dameUsuario(email : String, pass : String) : Usuario
+ creaUsuario(usuario : String) : void

Ilustración 24: DataMapperUsuario

Explicación:

Definición	La clase DataMapperUsuario es la encargada de mover objetos de tipo Usuario hacia/desde la base de datos y, en general, realizar acciones que requieren acceso a la base de datos para objetos de tipo Usuario.	
Métodos	existeUsuario	Devuelve un tipo boolean dependiendo si el usuario existe en la BBDD.
	dameUsuario	Nos devuelve un objeto Usuario determinado.
	creaUsuario	Nos inserta un usuario en la BBDD.

Tabla 68: Descripción DataMapperUsuario

DataMapperAdmin:

BLOQUE IV. DISEÑO

MapperAdmin
bd : AdaptadorBd
+ existeAlgunaRuta() : int + dameTodasRutas() : Ruta + existeAlgunUsuario() : int + dameTodosUsuarios() : Usuario + borraRutaConcreta(id : int) : void + actualizaUsuarioConcreto(usuario : Usuario) : void

Ilustración 25: MapperAdmin

Explicación:

Definición	La clase MapperAdmin es la encargada de mover objetos de tipo Usuario y Ruta hacia/desde la base de datos para las opciones del administrador.	
Métodos	existeAlgunaRuta	Devuelve el número de rutas que existen en la BBDD.
	dameTodasRutas	Nos devuelve todas las rutas de la BBDD.
	existeAlgunUsuario	Nos retorna el número de usuarios que están en la BBDD.
	dameTodosUsuarios	Nos devuelve todos los Usuarios.
	borraRutaConcreta	Borra una ruta específica de la BBDD, incluyendo los detalles y los nodos que la componen.
	actualizaUsuarioConcreto	Actualiza los datos de un usuario específico.

Tabla 69: Descripción MapperAdmin

8.3.3 Modelos dinámicos

En este apartado se describe el sistema desde una perspectiva dinámica, es decir, se mostrará que secuencia de mensaje desatan las peticiones de los actores las clases que lo forman para dar la respuesta adecuada a cada una de estas peticiones

A continuación se verá los diagramas de secuencia de los casos de uso principales de la aplicación. En estos diagramas se muestra como se comunican o relacionan las clases para construir una respuesta ante una petición por parte del usuario.

BLOQUE IV. DISEÑO

8.3.4.2 Diagrama de secuencia, Empieza ruta

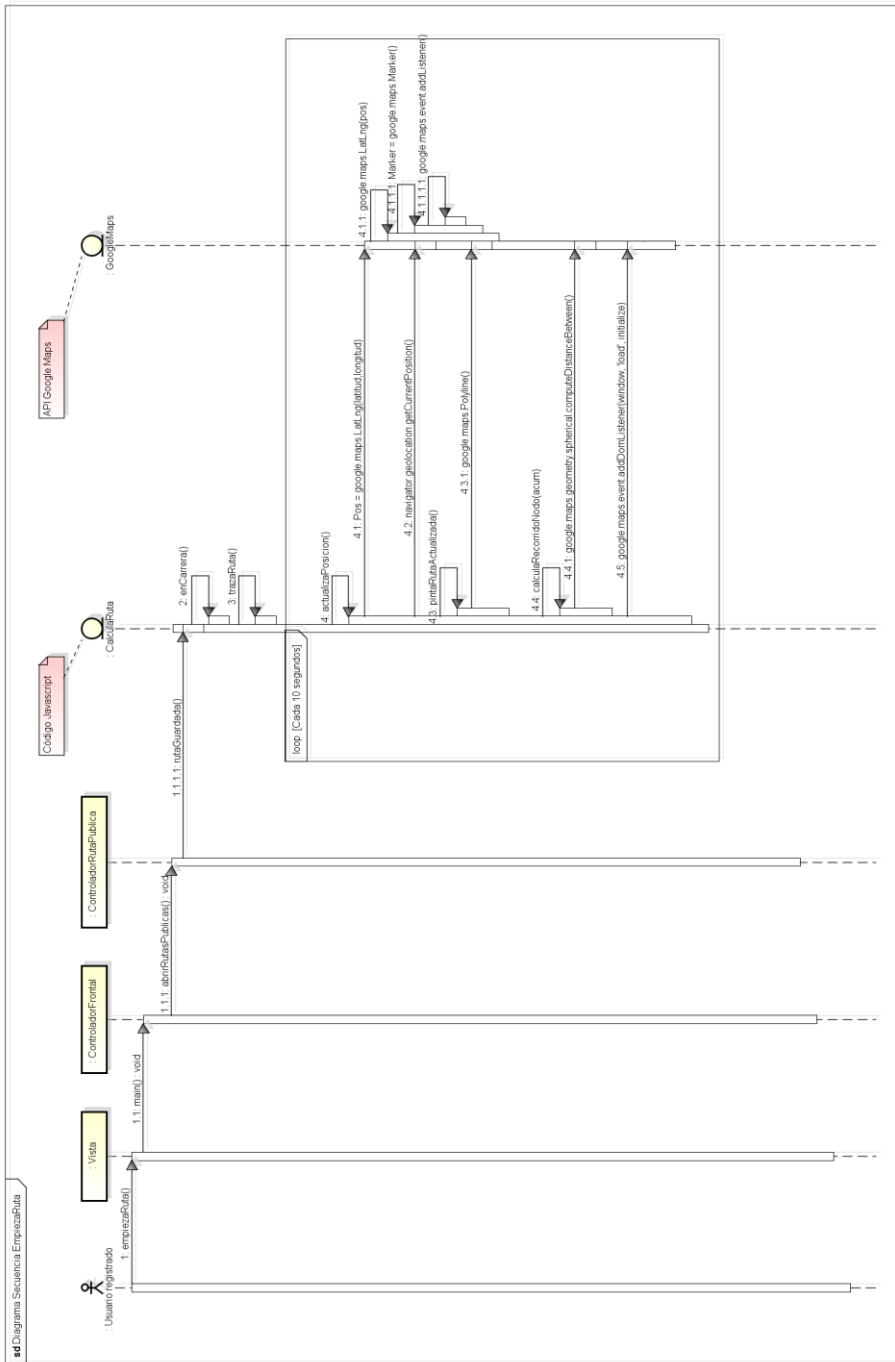


Ilustración 27: Diagrama de secuencia empezar ruta

BLOQUE IV. DISEÑO

8.3.4.3 Diagrama de secuencia, Termina ruta

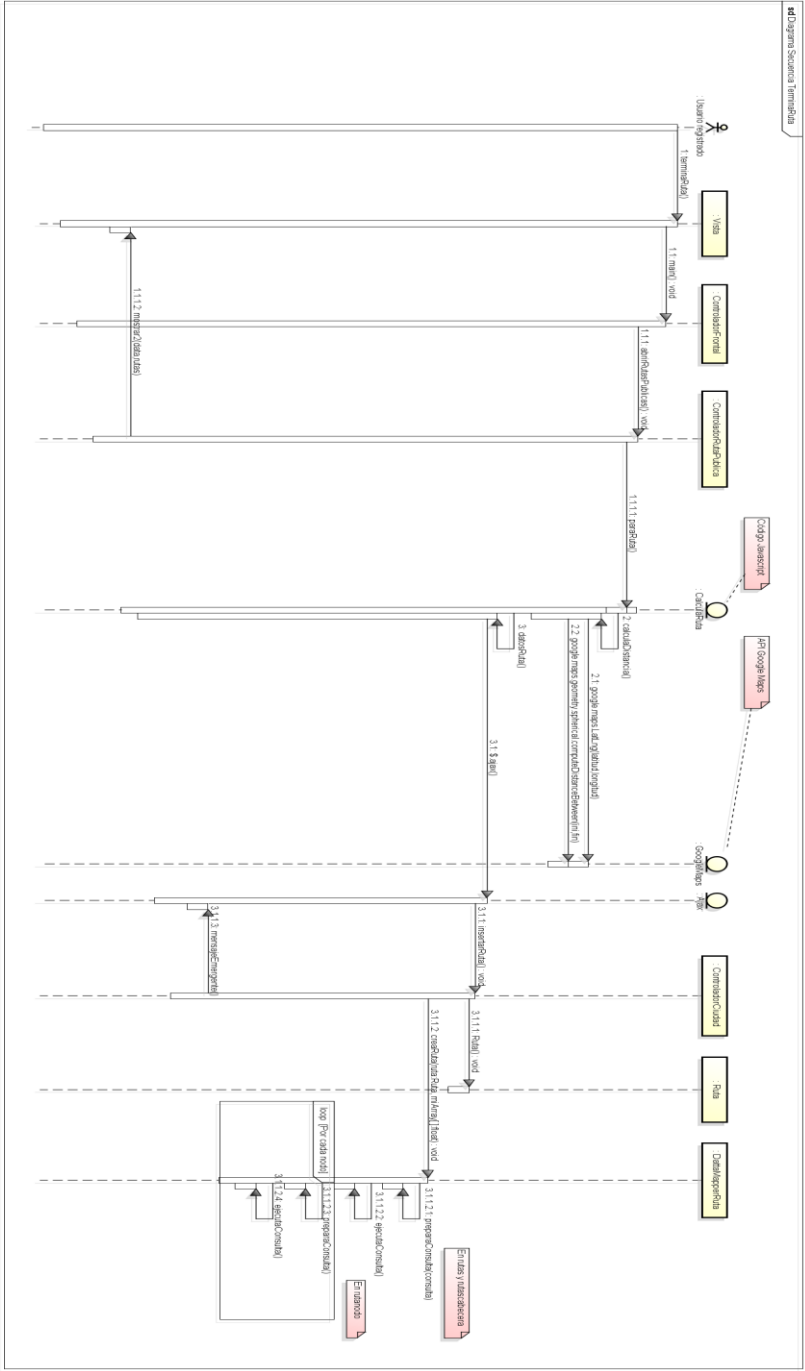


Ilustración 28: Diagrama de secuencia terminar ruta

BLOQUE IV. DISEÑO

8.3.4.4 Diagrama de secuencia, Registro

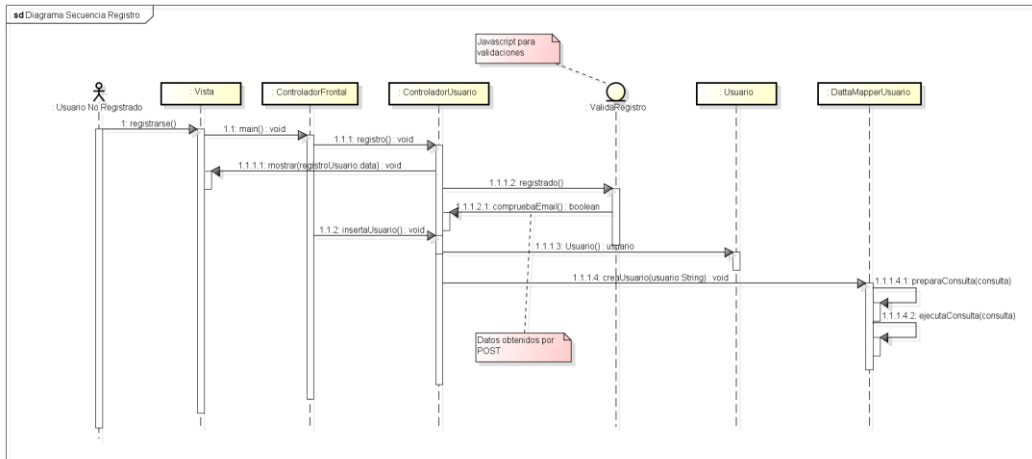


Ilustración 29: Diagrama de secuencia registro

8.3.4.5 Diagrama de secuencia (Login)

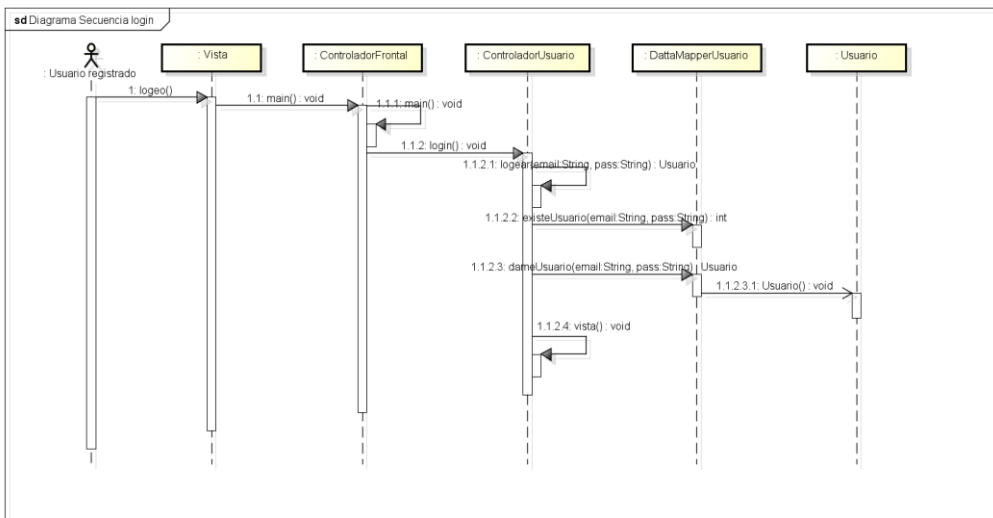


Ilustración 30: Diagrama de secuencia login

BLOQUE IV. DISEÑO

Los diagramas de secuencia correspondientes a contacto, misrutas, administración, búsqueda o redes sociales, son análogos a los descritos anteriormente, variando el controlador, la vista, el modelo y las APIS correspondientes en cada caso.

Bloque V.

Implementación

BLOQUE V. IMPLEMENTACIÓN

Capítulo 9.

Estructura de directorios

9.1 Introducción

El objetivo de este capítulo es describir y reflejar el proceso de implementación seguido junto con las soluciones adoptadas a la hora de implementar el proyecto tras las fases de análisis y diseño.

9.2 Estructura

La estructura de directorios de la página web implementada se puede describir en la siguiente ilustración:

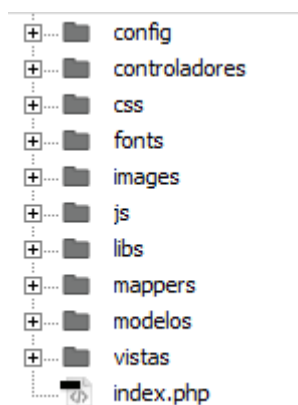


Ilustración 31: Estructura de directorios

BLOQUE V. IMPLEMENTACIÓN

A continuación pasamos a describir la estructura de directorios de la página web que gestiona milrutas:

1. **config:** contiene un fichero de configuración en el que se definen constantes de configuración, tales como las claves de acceso a la base de datos o el nombre de la base de datos sobre las que trabajará la aplicación web. También contiene las constantes necesarias para construir las vistas y controladores.

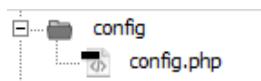


Ilustración 32: Implementación: Configuraciones

2. **controladores:** Dado que en las etapas anteriores decidimos usar el patrón MVC (Modelo-Vista-Controlador), este directorio contiene todos los controladores de la herramienta

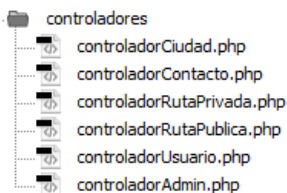


Ilustración 33: Implementación:

3. **css:** contiene los ficheros css que dan forma a la interfaz gráfica de usuario, incluye bootstrap para que la web sea responsive (adaptable a los diferentes tipos de dispositivos), jquery para la interfaz de usuario y archivos necesarios para el uso de plugins que hacen más vistosa a la aplicación

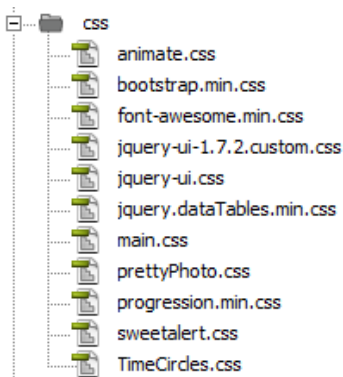


Ilustración 34: Implementación: css

BLOQUE V IMPLEMENTACION

4. fonts: los archivos relacionados con el tipo de tipografía.

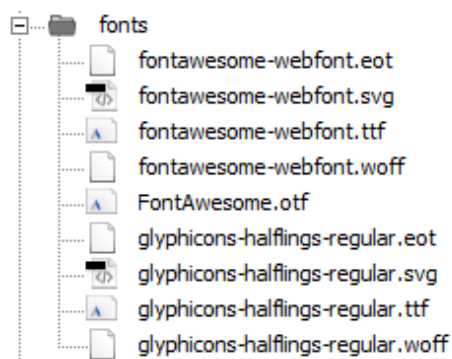


Ilustración 35: Implementación: fuentes

5. images: directorio que abarca las imágenes necesarias.

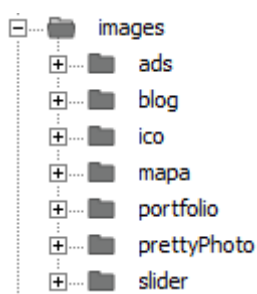


Ilustración 36: Implementación: imágenes

6. js: contiene los ficheros Javascript.

Explicación: incluye los ficheros de bootstrap y jquery. También, los archivos necesarios para el tratamiento de la geolocalización, Facebook, registro, elementos de ruta y validaciones.

BLOQUE V. IMPLEMENTACIÓN

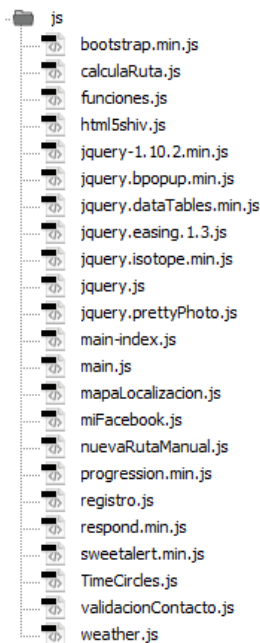


Ilustración 37 Implementación:

7. **libs:** Contiene algunas clases de propósito general, tales como el motor de plantillas o el controlador frontal, el cual es el encargado de recibir las solicitudes y llamar al método adecuado del controlador correspondiente.

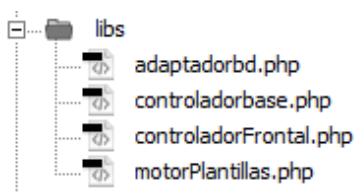


Ilustración 38: Implementación: Libs

Explicación: son los archivos necesarios para la implementación del patrón MVC así como para el datamapper y el controlador frontal.

8. **mappers:** En las fases previas, se decidió usar como patrón de acceso al almacén de datos el patrón DataMapper. Este directorio contiene los diferentes datamapper que encapsulan el acceso a la base de datos, de tal forma que desacoplamos el modelo de dominio de nuestra herramienta del acceso a la BBDD.

BLOQUE V IMPLEMENTACION

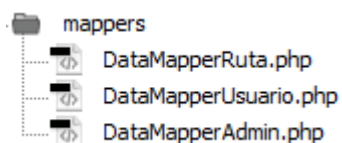


Ilustración 39 Implementación:

9. modelos: Contiene las clases del modelo de dominio de la aplicación.

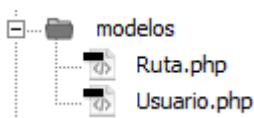


Ilustración 40: Implementación: Modelos

10. vistas: Contiene las diferentes vistas de la aplicación.

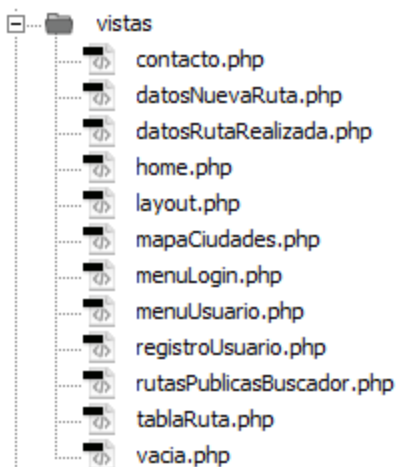


Ilustración 41: Implementación: Vistas

Explicación: incluye el conjunto de todas las vistas de la que consta la aplicación, dependiendo de la funcionalidad en cada momento.

11. index: índice de la aplicación, carga las configuraciones, el controlador frontal y finalmente ejecuta el main.

Bloque VI.

Pruebas y resultados

BLOQUE VI. PRUEBAS Y RESULTADOS

Capítulo 10.

Pruebas de

funcionalidad y

rendimiento

10.1 Pruebas de instalación

Sistema Operativo: Windows 7, Windows 8.1 y Linux en sus variantes Debian y Ubuntu.

Servidor: XAMPP tanto en Windows como en Linux.

Navegador: Mozilla Firefox, Google Chrome y Safari.

Instalación Base de Datos con éxito.

10.2 Pruebas de sistema

ROBUSTED

Problemas con el servidor que harán que la aplicación deje de funcionar:

- Apagado del servidor.
- Avería en la tarjeta de red.
- Servidor bloqueado.

SEGURIDAD

Tiempo de expiración de sesiones y opciones controlado por la aplicación.

Utilización de permisos dentro de las tablas usuarios en MySQL.

BLOQUE VI. PRUEBAS Y RESULTADOS

Encriptación de contraseñas dentro del sistema de bases de datos.

10.3 Pruebas de caja negra

Son utilizadas para verificar una salida de datos correcta de la aplicación.

Datos de usuario

Email

Condiciones de entrada	Clases de equivalencia válidas	Clases de equivalencia no válidas
Entre 1 y 60 caracteres	A $1 \leq n^{\circ} \text{ caracteres} \leq 60$	C $n < 1$
	B usuario@dominio.xxx	D $n > 60$
		E usuario
		F usuario@
		G usuario@dominio
		H @dominio.xxx

Tabla 70: Pruebas de caja negra: email de usuario

Identificador	Clases de equivalencia cubiertas	Resultado
"aebmmb@gmail.com"	A, B	Correcto
""	C	Error
"usuario@dominioDelSitio..."	D	Error
"Pedro"	E	Error
"alberto@"	F	Error

Tabla 71: Pruebas de caja negra: comprobación email de usuario

Password

Condiciones de entrada	Clases de equivalencia válidas	Clases de equivalencia no válidas
Entre 1 y 60 caracteres	A $1 \leq n^{\circ} \text{ caracteres} \leq 60$	B $n < 1$
		C $n > 60$

Tabla 72: Pruebas de caja negra: password

Identificador	Clases de equivalencia cubiertas	Resultado
"1234"	A	Correcto
""	B	Error

BLOQUE VI. PRUEBAS Y RESULTADOS

"123456789..."	C	Error
----------------	---	-------

Tabla 73: Pruebas de caja negra: comprobación password

Nombre

Condiciones de entrada	Clases de equivalencia válidas	Clases de equivalencia no válidas
Entre 1 y 20 caracteres	$A \ 1 \leq n^{\circ} \text{ caracteres} \leq 20$	$B \ n < 1$
		$C \ n > 20$

Tabla 74: Pruebas de caja negra: nombre

Identificador	Clases de equivalencia cubiertas	Resultado
"Juan"	A	Correcto
""	B	Error
"Luis Alberto Javier Felipe"	C	Error

Tabla 75: Pruebas de caja negra: comprobación nombre

Apellidos

Condiciones de entrada	Clases de equivalencia válidas	Clases de equivalencia no válidas
Entre 1 y 30 caracteres	$A \ 1 \leq n^{\circ} \text{ caracteres} \leq 30$	$B \ n < 1$
		$C \ n > 30$

Tabla 76: Pruebas de caja negra: apellidos

Identificador	Clases de equivalencia cubiertas	Resultado
"Barrena"	A	Correcto
""	B	Error
"Merino del Palacio González..."	C	Error

Tabla 77: Pruebas de caja negra: comprobación apellidos

Análogos a los anteriores y cuyo rango permitido no se puede introducir son el peso, la altura, la fecha de nacimiento y el código postal.

Datos de ruta

Nombre ruta

BLOQUE VI. PRUEBAS Y RESULTADOS

Condiciones de entrada	Clases de equivalencia válidas	Clases de equivalencia no válidas
Entre 1 y 20 caracteres	$A 1 \leq n^{\circ} \text{ caracteres} \leq 20$	B $n < 1$
		C $n > 20$

Tabla 78: Pruebas de caja negra: nombre de la ruta

Identificador	Clases de equivalencia cubiertas	Resultado
“Pinar de antequera”	A	Correcto
“”	B	Error
“Canal Medina de Rioseco por el camino forestal...”	C	Error

Tabla 79: Pruebas de caja negra: comprobación nombre de la ruta

Comentarios

Condiciones de entrada	Clases de equivalencia válidas	Clases de equivalencia no válidas
Entre 0 y 200 caracteres	$A 0 \leq n^{\circ} \text{ caracteres} \leq 200$	B $n > 200$

Tabla 80: Pruebas de caja negra: comentarios de una ruta

Identificador	Clases de equivalencia cubiertas	Resultado
“Ruta para principiantes, patinadores, 80% en llano”	A	Correcto
“”	A	Correcto
“Senderismo avanzado, requiere equipo de montaña, ... ”	B	Error

Tabla 81: Pruebas de caja negra: comprobación comentarios de una ruta

Los datos del resto del formulario de una ruta; dificultad, fecha y valoración, son entradas prefijadas y también análogas a las anteriores.

10.4 Casos de prueba

Un caso de prueba o test case es, en ingeniería del software, un conjunto de condiciones o variables bajo las cuáles un se determinará si una aplicación, un sistema software (software system), o una característica de éstos es parcial o completamente satisfactoria.

Se pueden realizar muchos casos de prueba para determinar que un requisito es completamente satisfactorio.

BLOQUE VI. PRUEBAS Y RESULTADOS

Para este proyecto se realizarán los casos de prueba relacionados con la localización, ya que son laboriosos de llevar a cabo pues es necesario la realización de las rutas junto con otro dispositivo externo para la comprobación exitosa del caso de prueba. Cada una de estas pruebas se realizará con un GPS TomTom auxiliar y un velocímetro siempre que sea posible.

Para la verificación correcta es necesario que el usuario esté registrado para que quede constancia en la BBDD de la ruta realizada. Si bien podemos extrapolar a un usuario no registrado, con la salvedad de que no se guardará la ruta.

En este primer caso de prueba se quiere comprobar el correcto funcionamiento y el grado de precisión al realizar una ruta:

Caso de prueba 1	
Objetivo	Realizar una ruta Valladolid – Simancas
Condiciones de la prueba	El usuario lleva a cabo una ruta en bici desde la junta de Castilla y León hasta el ayuntamiento de Simancas. Dispone de un dispositivo móvil con conexión a internet y GPS. Se encuentra registrado y autenticado en el sistema
Entradas	Los datos de la ruta y los nodos que la componen.
Salida deseada	Muestra los datos actualizados, mensajes emergentes y guarda la ruta correctamente

Tabla 82: Caso de prueba: Realizar ruta nueva

En este segundo caso, se testeará la aplicación realizando una ruta previamente creada:

Caso de prueba 1	
Objetivo	Realizar una ruta Valladolid – Simancas
Condiciones de la prueba	El usuario lleva a cabo una ruta running que ya ha sido realizada o creada por un usuario. Dispone de un dispositivo móvil con conexión a internet y GPS. Se encuentra registrado y autenticado en el sistema
Entradas	Búsqueda de la ruta en concreto y realización de la misma (obtención de los nodos y parámetros que la forman).
Salida deseada	Muestra los datos actualizados y mensajes emergentes oportunos

Ilustración 42: Caso de prueba: Realizar una ruta previamente creada

Bloque VII.

Manual de instalación y usuario

BLOQUE VII. MANUALES

Capítulo 11.

Manual de instalación

11.1 Introducción

El objetivo de este capítulo consiste en dar a conocer el proceso necesario para realizar una correcta instalación del software presentado en capítulos anteriores.

11.2 Requisitos de instalación

Este software, como se ha indicado, está escrito en lenguaje PHP, se ejecuta bajo un servidor web Apache y requiere un servidor MySQL para poder almacenar de forma persistente los datos, por ello es necesario disponer de un sistema que al menos tenga instalado un servidor web, un intérprete de PHP y un sistema gestor de bases de datos MySQL.

Características mínimas requeridas:

- Hardware:
 - Servidor con acceso a la red.
 - 4 GB de memoria RAM.
 - Un procesador de más de 1'5 GHz.
 - Disco duro de 1 GB.
- Software:
 - El sistema operativo puede ser tanto Linux como Windows.
 - Un navegador, con JavaScript y localización habilitado (en la parte de cliente).
 - XAMPP.

BLOQUE VII. MANUALES

11.3 Instalación

Instalación previa

Dado que Windows y Linux son dos de los sistemas operativos más utilizados, presentaremos a modo de ejemplo como instalar todos los elementos mencionados en el párrafo anterior en el SO Linux Ubuntu 14.04, pues como se ha documentado a lo largo de esta memoria, siempre se ha priorizado el uso de herramientas libres.

Si ya se dispone de un servidor web, de un intérprete de PHP y de un sistema gestor de bases de datos no es necesario que siga leyendo este punto.

Existe una herramienta denominada XAMPP que incluye entre otras cosas Apache, PHP y MySQL. Para instalarla debemos visitar su página oficial y acceder al apartado de descargas. Una vez ahí seleccionamos la última versión y la descargamos.

La URL de descarga es la siguiente: <https://www.apachefriends.org/es/download.html>

Después debemos seleccionar el instalador de 32 o 64 bits en función del tipo de máquina en la que deseemos realizar la instalación. Si no sabemos qué tipo de máquina tenemos podemos ejecutar el comando “uname -a” desde un terminal.

Si en la salida del comando aparece “x86_64 GNU/Linux” estaremos frente a una máquina de 64 bits, sin embargo, si la salida es “i386/i486/i586/i686” la máquina será de 32 bits.

Una vez conozcamos la arquitectura de nuestro procesador procederemos a descargar el instalador correspondiente.

A continuación descargamos e instalamos el archivo correspondiente.

Para comprobar que todo ha ido correctamente basta con introducir en la barra del navegador: <http://localhost/> o <http://127.0.0.1/> y aparecerá el logotipo de Xampp

Instalación del sitio web

Explicaremos los sencillos pasos para la puesta en marcha de milrutas en un SO Linux.

Lo primero que tenemos que hacer es movernos al directorio /opt/lampp/htdocs que es donde se encuentran las carpetas de nuestro servidor web

En dicha carpeta, descomprimir el archivo milrutas.zip que se encuentra en el CD adjunto a la memoria.

Pasamos a la instalación de la BBDD, para ello en la barra de navegación del navegador escribimos <http://localhost/phpmyadmin> . El siguiente paso es la creación de la BBDD de nombre rutas en formato utf8_general_ci. Y por último importamos la BBDD contenida en el CD

BLOQUE VII. MANUALES

Para finalizar, tenemos que adaptar los parámetros de configuración a nuestra máquina. Toda la gestión relacionada con la configuración se encuentra en el fichero `/opt/lampp/htdocs/milrutas/config/config.php` y modificaremos los siguientes parámetros (*cursiva*):

- `define("HOSTBD","localhost");` //nombre del servidor
- `define("USERBD","rutas");` //nombre del usuario
- `define("PASSBD","123456");` //contraseña fijada en la BBDD para el usuario

NOTA: La instalación del servidor Xampp crea un usuario con login “root” y contraseña vacía para MySQL. Es muy recomendable crear una contraseña para el usuario root con la finalidad de proteger nuestra base de datos ante posibles atacantes.

Para comprobar que todo ha ido de manera correcta, nos dirigimos a la siguiente URL en el navegador: <http://localhost/milrutas> si ha llegado hasta este punto y aparece la página principal de milrutas, ¡Felicidades! El sistema se instaló correctamente.

BLOQUE VII. MANUALES

Capítulo 12.

Manual de usuario

12.1 Usuario no registrado

Introducción

Dado que la aplicación está diseñada para ser responsive, mostraremos las principales funcionalidades en un dispositivo de escritorio y un dispositivo móvil.

Al acceder a la dirección: <http://vps150302.ovh.net/Alvaro/milrutas> nos encontramos con la pantalla principal de la aplicación:

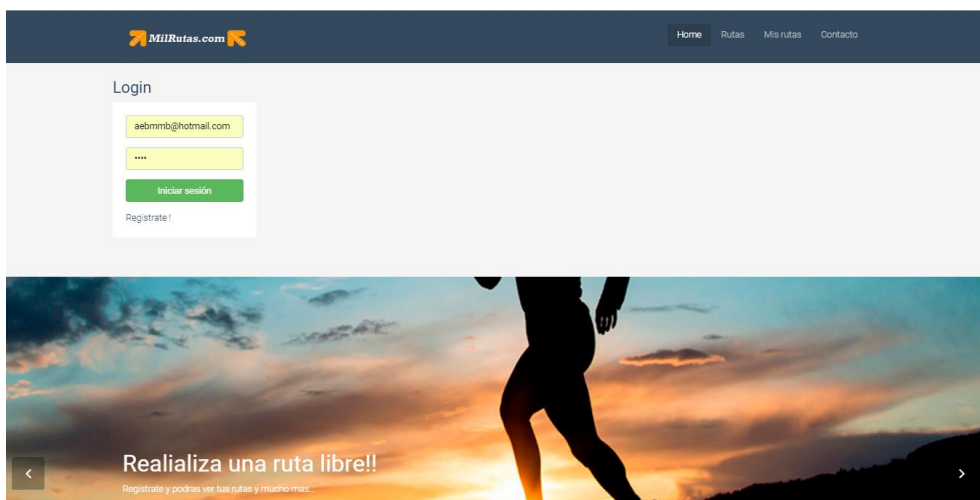


Ilustración 43: Página principal escritorio1/2, usuario no registrado

BLOQUE VII. MANUALES

En la parte superior del sitio web, denominado comúnmente header, aparece el logo de la aplicación en la parte izquierda y el menú de navegación en la parte derecha.

Al hacer click en el logo, estando en cualquier parte del sistema, nos llevara a la página principal. El menú nos permite ir navegando en las diferentes partes del mismo.

- Home: nos lleva a la página principal.
- Rutas: nos muestra las rutas públicas y permite realizar una ruta
- Mis rutas: al no estar autenticado, nos lleva al registro.
- Contacto: muestra el formulario típico de contacto.

A continuación del header, encontramos lo que comúnmente conocemos como “pantalla de login” donde los usuarios de milrutas pueden iniciar sesión o registrarse.

La siguiente sección es un “carrusel” de imágenes principales cambiantes en el tiempo.

Si continuamos bajando en la página principal (scroll), vemos la siguiente imagen:

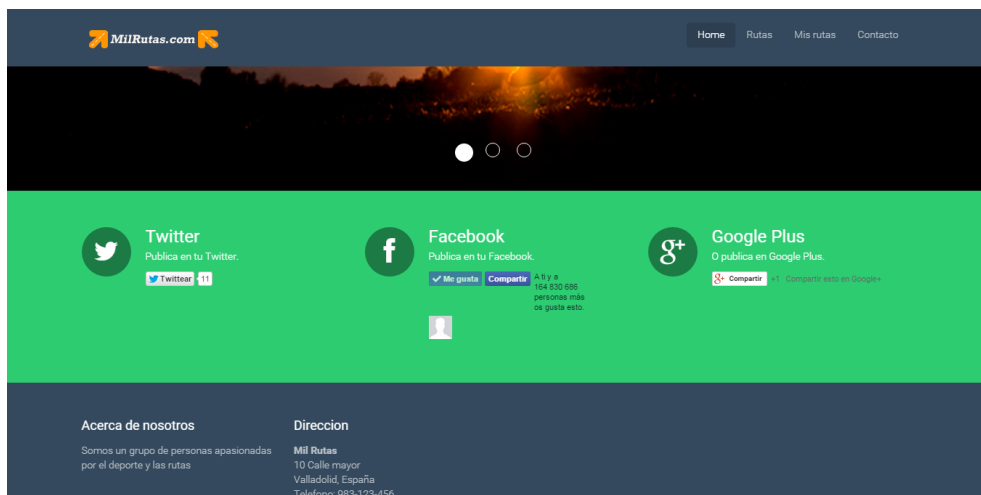


Ilustración 44: Página principal escritorio 2/2, usuario no registrado

En esta segunda parte de la página principal vemos los enlaces a las redes sociales de Twitter, Facebook y Google Plus.

En la siguiente sección vemos la información acerca de y la dirección física de milrutas.

En la última parte de la aplicación, el footer, se observa el mismo menú que en el header y con las mismas funcionalidades, así como un icono para volver al principio de la página.

BLOQUE VII. MANUALES

Estas tres últimas opciones estarán siempre disponibles excepto cuando se esté realizando una ruta

En dispositivo móvil:

Primera parte:

menú

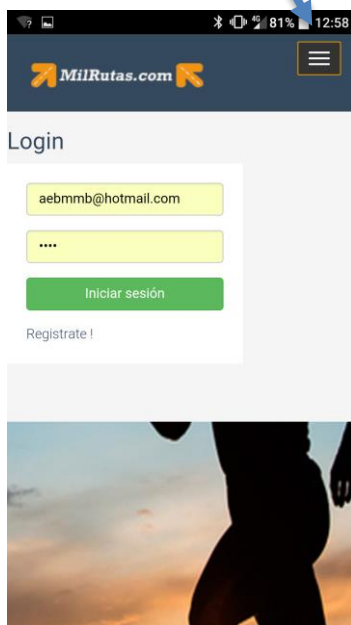


Ilustración 46: Página principal móvil 1/2, usuario no registrado

Segunda parte:

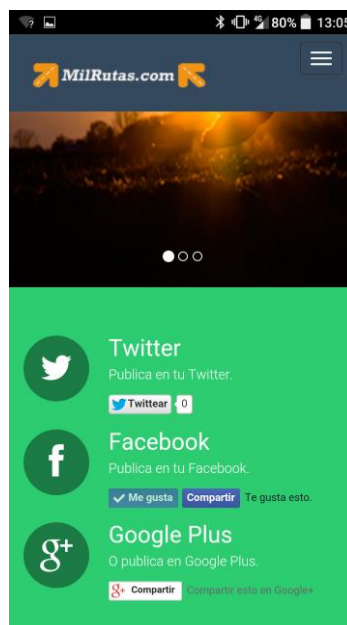


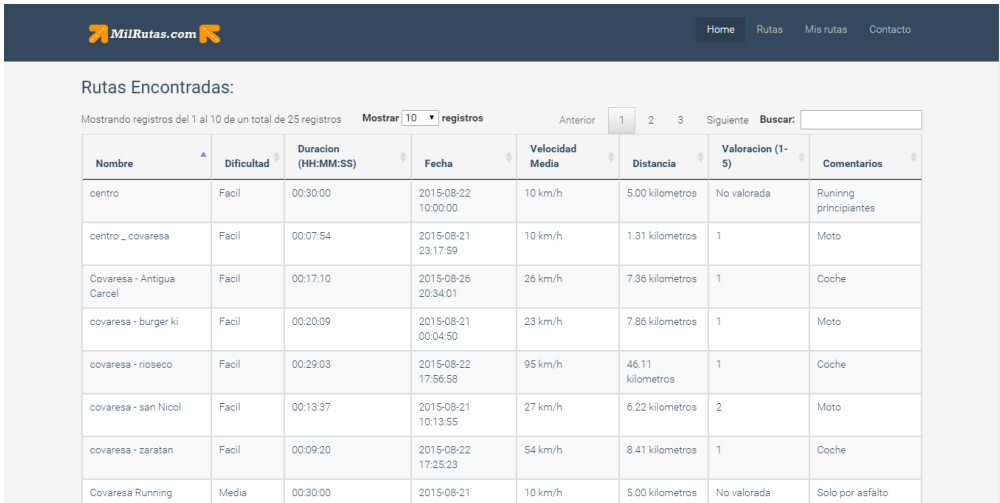
Ilustración 45: Página principal móvil 2/2, usuario no registrado

Como se expone en las ilustraciones anteriores, podemos realizar la misma funcionalidad que en un dispositivo de escritorio. En este caso, el menú pasa a ser un boto que despliega las opciones, y el resto de opciones se acumulan en vertical.

Ver rutas

Para ver las rutas que hay disponibles en el sistema, se debe hacer click en “Rutas”. Si existen rutas, aparecerán en una tabla con los detalles de las mismas, siendo éstas clicables para mostrar en el mapa la ruta en concreto.

BLOQUE VII. MANUALES



MilRutas.com Home Rutas Mis rutas Contacto

Rutas Encontradas:

Mostrando registros del 1 al 10 de un total de 25 registros. Mostrar 10 registros. Anterior 1 2 3 Siguiente. Buscar:

Nombre	Dificultad	Duracion (HH:MM:SS)	Fecha	Velocidad Media	Distancia	Valoracion (1-5)	Comentarios
centro	Facil	00:30:00	2015-08-22 10:00:00	10 km/h	5.00 kilometros	No valorada	Runing principiantes
centro_covaresa	Facil	00:07:54	2015-08-21 23:17:59	10 km/h	1.31 kilometros	1	Moto
Covaresa - Antigua Carcel	Facil	00:17:10	2015-08-26 20:34:01	26 km/h	7.36 kilometros	1	Coche
covaresa - burger ki	Facil	00:20:09	2015-08-21 00:04:50	23 km/h	7.86 kilometros	1	Moto
covaresa - rioseco	Facil	00:29:03	2015-08-22 17:56:58	95 km/h	46.11 kilometros	1	Coche
covaresa - san Nicol	Facil	00:13:37	2015-08-21 10:13:55	27 km/h	6.22 kilometros	2	Moto
covaresa - zaratan	Facil	00:09:20	2015-08-22 17:25:23	54 km/h	8.41 kilometros	1	Coche
Covaresa Running	Media	00:30:00	2015-08-21	10 km/h	5.00 kilometros	No valorada	Solo por asfalto

Ilustración 47: Página rutas disponibles escritorio 1/2, usuario no registrado

La versión móvil se diferencia en que la tabla está fijada, y para poder ver los detalles es necesario hacer scroll de izquierda a derecha. En la siguiente ilustración vemos un ejemplo:



MilRutas.com

Rutas Encontradas:

Mostrar 10 registros. Buscar: covaresa

Nombre	Dificultad	Duracion (HH:MM:SS)	
covaresa - burger ki	Facil	00:20:09	20
covaresa - san Nicol	Facil	00:13:37	21
Covaresa Running	Media	00:30:00	21
la rubia covaresa	Facil	00:04:51	22
Monasterio covaresa	Facil	00:04:14	21

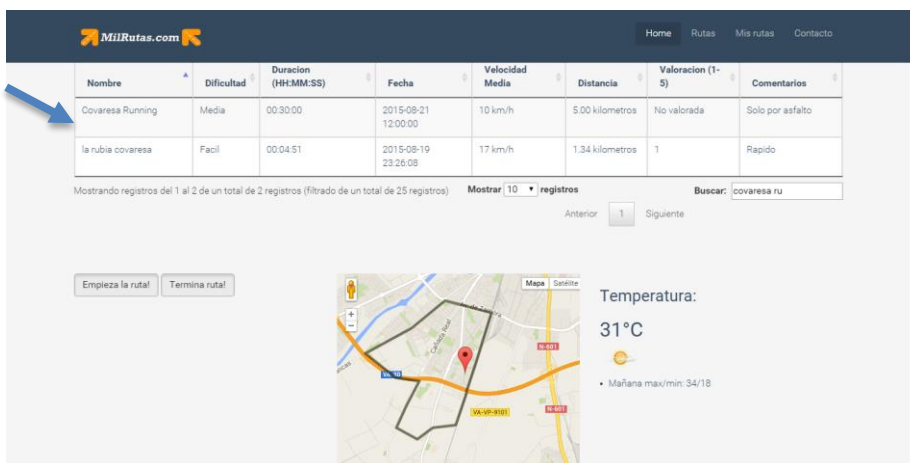
Mostrando registros del 1 al 5 de un total de 5 registros (filtrado de un total de 15 registros)

Ilustración 48: Página rutas disponibles móvil, usuario no registrado

BLOQUE VII. MANUALES

Cargar Rutas

Una vez que se localice la ruta, haciendo click sobre ella aparecerá en el mapa:



Nombre	Dificultad	Duracion (HH:MM:SS)	Fecha	Velocidad Media	Distancia	Valoracion (1-5)	Comentarios
Covaresa Running	Media	00:30:00	2015-08-21 12:00:00	10 km/h	5.00 kilometros	No valorada	Solo por asfalto
la rubia covaresa	Facil	00:04:51	2015-08-19 23:26:08	17 km/h	1.34 kilometros	1	Rapido

Mostrando registros del 1 al 2 de un total de 2 registros (filtrado de un total de 25 registros) Mostrar 10 registros Buscar: covaresa ru

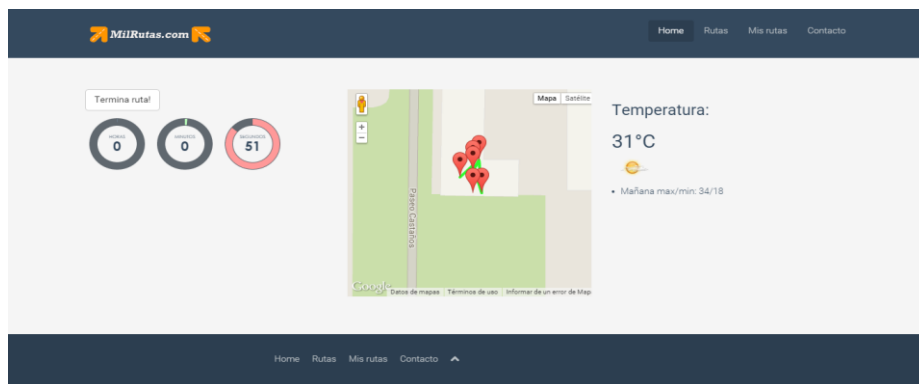
Empezar la ruta! Termina ruta!

Temperatura: 31°C
Mañana max/min: 34/18

Ilustración 49: Página cargar rutas, usuario no registrado

Realizar ruta

Para empezar una ruta, damos al botón “Empezar la ruta”, nos mostrara un mensaje emergente indicando que no es posible guardar la ruta sin estar registrado en el sistema y a continuación la siguiente pantalla. En ella podemos observar el tiempo de duración y un mapa que nos indica la ruta realizada hasta el momento.



Termina ruta!

0 0 51

Temperatura: 31°C
Mañana max/min: 34/18

Ilustración 50: Realizando ruta, usuario no registrado

BLOQUE VII. MANUALES

Al finalizar la ruta, pulsando el botón terminar ruta nos aparece el siguiente mensaje:

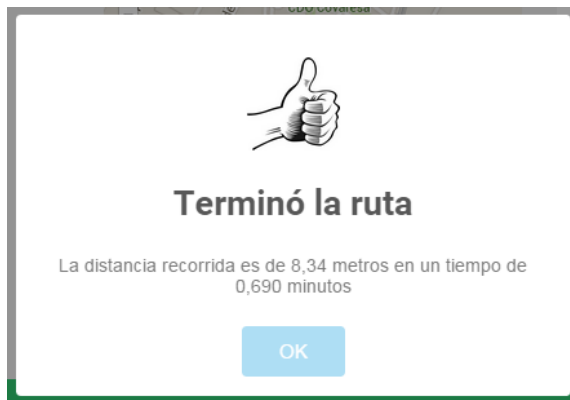


Ilustración 51: Fin de ruta, usuario no

En el cual nos muestra información acerca de la distancia recorrida y el tiempo empleado.

[Ver mis rutas \(registro\)](#)

Al tratarse de un usuario no registrado, no tiene rutas realizadas, por lo que se muestra el registro de usuario al hacer click en “Mis Rutas”:

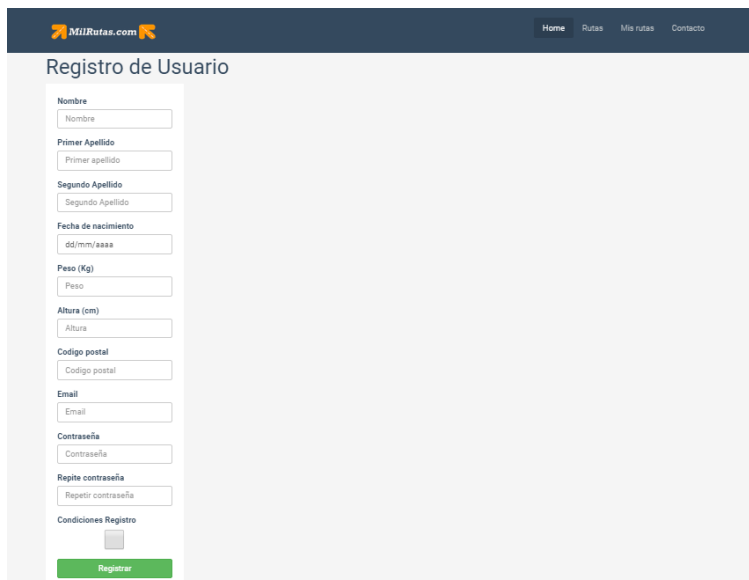
A screenshot of a web browser showing the registration page for "MiRutas.com". The page title is "Registro de Usuario". The form contains the following fields: "Nombre" (text input), "Primer Apellido" (text input), "Segundo Apellido" (text input), "Fecha de nacimiento" (date picker with format dd/mm/aaaa), "Peso (Kg)" (text input), "Altura (cm)" (text input), "Codigo postal" (text input), "Email" (text input), "Contraseña" (password input), and "Repite contraseña" (password input). There is a checkbox for "Condiciones Registro" and a green "Registrar" button at the bottom.

Ilustración 52: Registro de usuario, usuario no registrado

BLOQUE VII. MANUALES

Siendo idéntico a la versión móvil.

Contacto

Para ver el contacto, debemos hacer click en Contacto, y observaremos la siguiente

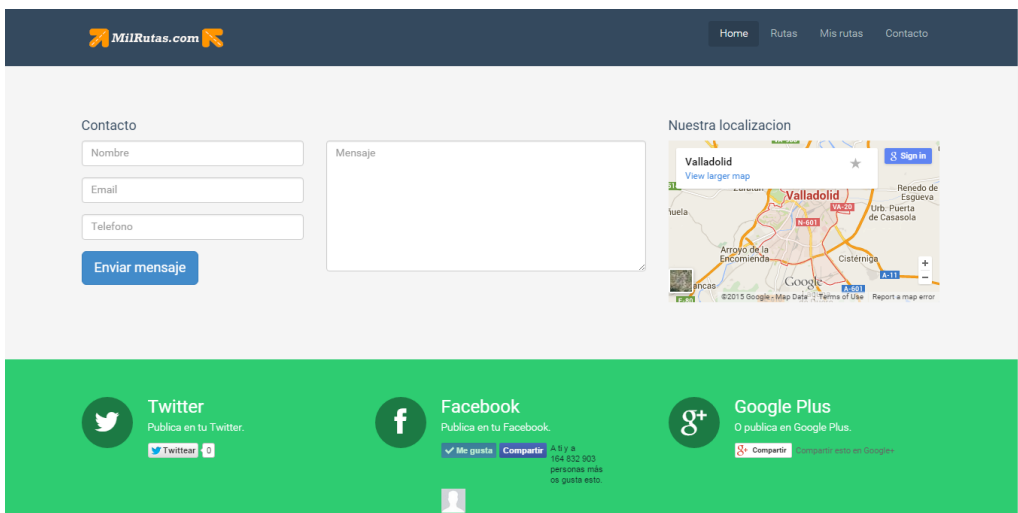


Ilustración 53: Contacto, usuario no registrado

ilustración:

Sí se quisiera mandar una cuestión, se rellenan los campos solicitados y pulsamos sobre el botón Enviar mensaje, a continuación si todo ha ido de manera correcta, se muestra un mensaje satisfactorio.

Redes Sociales

El apartado de redes sociales es común para los usuarios registrados como para los que no lo están. Las opciones de compartir las vemos a continuación:



Ilustración 54: Redes sociales

BLOQUE VII. MANUALES

Twitter

Si queremos twitear milrutas, basta con hacer click en el botón Twitear y nos emergerá la siguiente pantalla (si estamos autenticados previamente en Twitter):

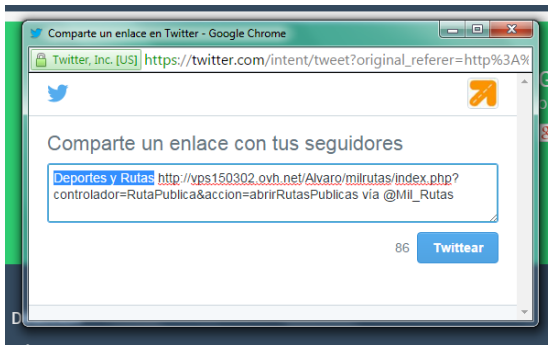


Ilustración 55: Twitear

Facebook

Compartir en Facebook:



Ilustración 56: Compartir en Facebook

Para el caso de Google Plus se realiza de manera análoga.

12.2 Usuario registrado

Introducción

BLOQUE VII. MANUALES

La página principal es la misma para ambos usuarios, en este caso, introducimos el email y la pass para identificarnos en la aplicación.

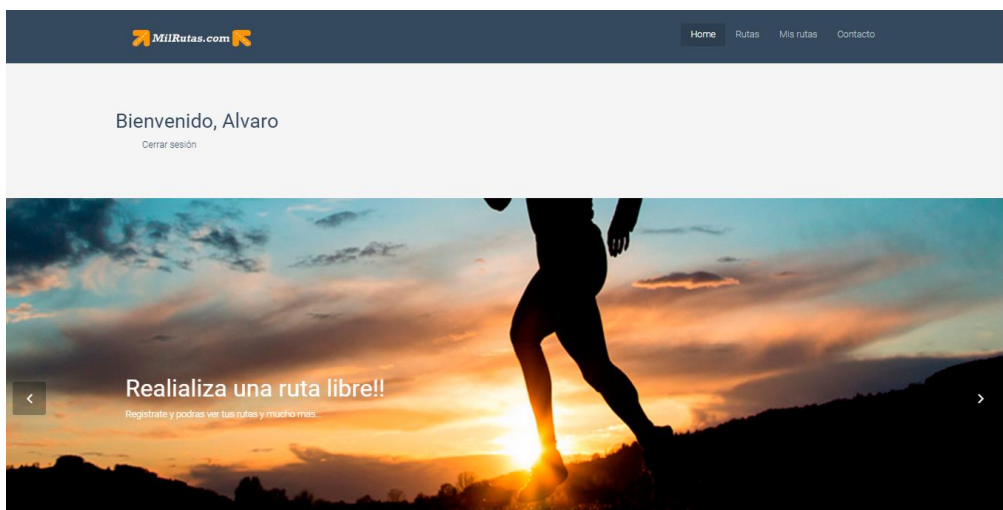


Ilustración 57: Página principal, usuario registrado

Realizar Ruta

Al realizar una ruta, la forma es la misma que si se trata de un usuario no registrado, con la salvedad de que en este caso nos pedirá información acerca de la ruta al terminar la misma:

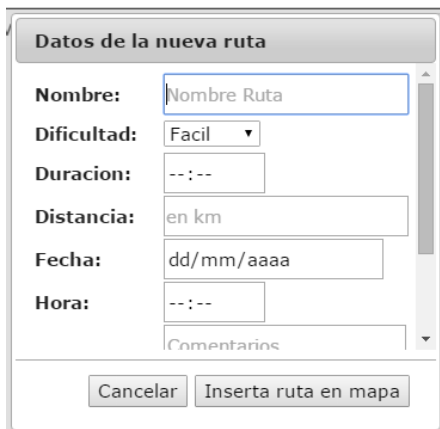
The image shows a modal form titled 'Datos de la ruta'. It contains the following fields: 'Nombre:' with a text input field containing 'Nombre Ruta'; 'Dificultad:' with a dropdown menu set to 'Facil'; 'Valoracion:' with a dropdown menu set to '1'; and 'Comentarios:' with a large text area. At the bottom of the form, there are two buttons: 'Cancelar' and 'Aceptar'.

Ilustración 58: Datos ruta realizada, usuario registrado

BLOQUE VII. MANUALES

Por último, si todo ha ido correctamente, nos muestra un mensaje confirmando la ruta guardada.

Crear ruta manual



Datos de la nueva ruta

Nombre:

Dificultad:

Duración:

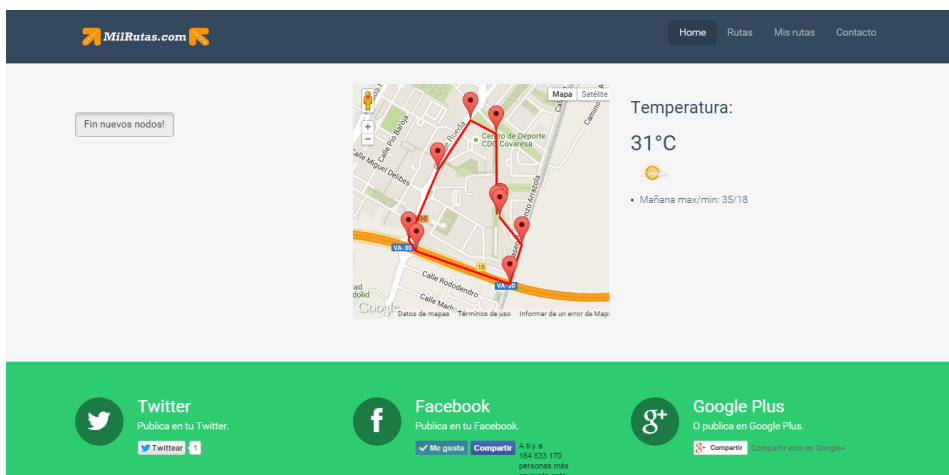
Distancia:

Fecha:

Hora:

Ilustración 59: Datos ruta manual, usuario registrado

Insertamos los nodos que compondrán la ruta:



Fin nuevos nodos!

Mapa | Satélite

Temperatura: 31°C

• Mañana max/min: 35/18

Twitter Publica en tu Twitter.

Facebook Publica en tu Facebook.

Google Plus O publica en Google Plus.

Ilustración 60: Introducción de nodos manual, usuario registrado

BLOQUE VII. MANUALES

Ver mis rutas

Para poder ver las rutas realizadas por un usuario, nos iremos a la opción Mis Rutas, como podemos ver en la siguiente ilustración.

The screenshot shows the MilRutas.com website interface for a user named Juan. The page title is "Bienvenido, Juan" with a "Cerrar sesión" link. Under the heading "Rutas Encontradas:", it indicates "Mostrando registros del 1 al 8 de un total de 8 registros". There are navigation controls for "Mostrar 10 registros", "Anterior 1", and "Siguiente", along with a "Buscar:" input field. A table lists the routes with columns for Nombre, Dificultad, Duracion (HH:MM:SS), Fecha, Velocidad Media, Distancia, Valoracion (1-5), and Comentarios. The table contains 8 rows of route data.

Nombre	Dificultad	Duracion (HH:MM:SS)	Fecha	Velocidad Media	Distancia	Valoracion (1-5)	Comentarios
centro_covaresa	Facil	00:07:54	2015-08-21 23:17:59	10 km/h	1.31 kilometros	1	Moto
Covaresa - Antigua Carcel	Facil	00:17:10	2015-08-26 20:34:01	26 km/h	7.36 kilometros	1	Coche
covaresa - burger ki	Facil	00:20:09	2015-08-21 00:04:50	23 km/h	7.86 kilometros	1	Moto
covaresa - rioseco	Facil	00:29:03	2015-08-22 17:56:58	95 km/h	46.11 kilometros	1	Coche
covaresa - zaratan	Facil	00:09:20	2015-08-22 17:25:23	54 km/h	8.41 kilometros	1	Coche
Junta de Castilla y leon	Facil	00:05:34	2015-08-27 11:14:05	7 km/h	0.63 kilometros	1	Andando
san Nicolás - covaresa	Facil	00:16:29	2015-08-21 14:19:55	29 km/h	8.01 kilometros	1	Moto
zamudia_terraza	Facil	00:01:35	2015-08-21 20:16:25	1 km/h	0.02 kilometros	1	

Ilustración 61: Ver rutas privadas, usuario registrado

The screenshot shows the MilRutas.com mobile app interface for a user named Juan. The page title is "Bienvenido, Juan" with a "Cerrar sesión" link. Under the heading "Rutas Encontradas:", it indicates "Mostrando registros del 1 al 8 de un total de 8 registros". There are navigation controls for "Mostrar 10 registros", "Anterior 1", and "Siguiente", along with a "Buscar:" input field. A table lists the routes with columns for Nombre, Dificultad, and Duracion (HH:MM:SS). The table contains 3 rows of route data.

Nombre	Dificultad	Duracion (HH:MM:SS)
centro_covaresa	Facil	00:07:54
Covaresa - Antigua Carcel	Facil	00:17:10
covaresa -	Facil	00:20:09

Ilustración 62: Ver rutas privadas móvil

BLOQUE VII. MANUALES

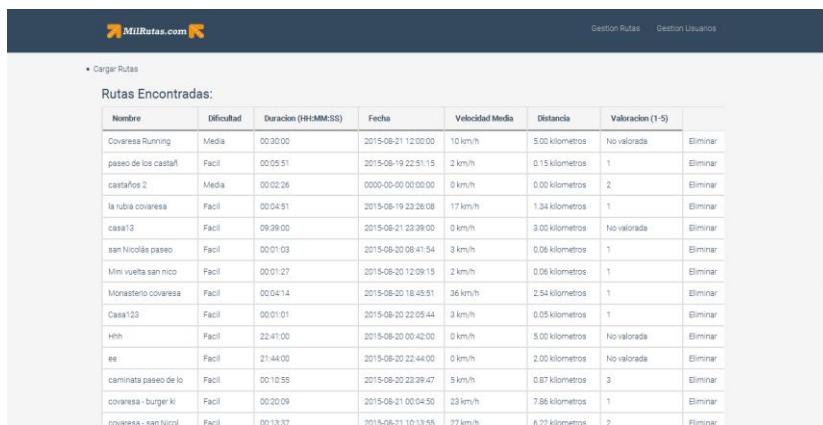
12.3 Usuario Administrador

Las funcionalidades del administrador son gestionar las rutas y los usuarios del sistema, para identificarse como administrador, el usuario es: admin y la contraseña correspondiente: admin1234

Una vez identificados en el sistema, tendremos las siguientes opciones:

Gestion Rutas

Al hacer click en Gestion Rutas, nos aparece la siguiente ilustración, en la cual sí se quiere eliminar una ruta, se seleccionaría la opción de Eliminar de la ruta deseada

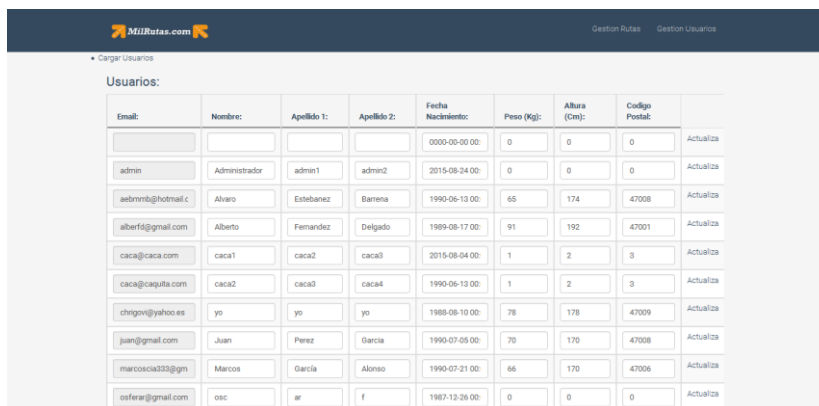


Nombre	Dificultad	Duracion (HH:MM:SS)	Fecha	Velocidad Media	Distancia	Valoracion (1-5)	
Covaresa Running	Media	00:30:00	2015-08-21 12:00:00	10 km/h	5.00 kilometros	No valorada	Eliminar
paseo de los castaños	Facil	00:05:51	2015-08-19 22:51:15	2 km/h	0.15 kilometros	1	Eliminar
castaños 2	Media	00:02:26	0000-00-00 00:00:00	0 km/h	0.00 kilometros	2	Eliminar
la nubia covaresa	Facil	00:04:51	2015-08-19 23:26:08	17 km/h	1.24 kilometros	1	Eliminar
casa13	Facil	09:39:00	2015-08-21 23:39:00	0 km/h	3.00 kilometros	No valorada	Eliminar
san Nicolás paseo	Facil	00:01:03	2015-08-20 08:41:54	3 km/h	0.06 kilometros	1	Eliminar
Mini vuelta san nico	Facil	00:01:27	2015-08-20 12:09:15	2 km/h	0.06 kilometros	1	Eliminar
Monasterio covaresa	Facil	00:04:14	2015-08-20 18:45:51	36 km/h	2.54 kilometros	1	Eliminar
Casa123	Facil	00:01:01	2015-08-20 22:05:44	3 km/h	0.05 kilometros	1	Eliminar
Héh	Facil	22:41:00	2015-08-20 00:42:00	0 km/h	5.00 kilometros	No valorada	Eliminar
ee	Facil	21:44:00	2015-08-20 22:44:00	0 km/h	2.00 kilometros	No valorada	Eliminar
caminata paseo de lo	Facil	00:10:55	2015-08-20 23:39:47	5 km/h	0.87 kilometros	3	Eliminar
covaresa - burger ki	Facil	00:20:09	2015-08-21 00:04:50	23 km/h	7.86 kilometros	1	Eliminar
covaresa - san Nicol	Facil	00:13:37	2015-08-21 10:13:55	27 km/h	6.22 kilometros	2	Eliminar

Ilustración 63: Ver gestión de rutas, administrador

Gestion Usuarios

En este apartado veremos cómo modificar los datos de un usuario. La siguiente ilustración muestra la interfaz para poderlo llevar a cabo:



Email:	Nombre:	Apellido 1:	Apellido 2:	Fecha Nacimiento:	Peso (Kg):	Altura (Cm):	Código Postal:	
				0000-00-00 00:00:00	0	0	0	Actualizar
admin	Administrador	admin1	admin2	2015-08-24 00:00:00	0	0	0	Actualizar
albrnmb@hotmail.c	Alvaro	Estebanez	Barrena	1990-06-12 00:00:00	65	174	47008	Actualizar
alberfd@gmail.com	Alberto	Fernandez	Delgado	1989-08-17 00:00:00	91	192	47001	Actualizar
caca@caca.com	caca1	caca2	caca3	2015-08-04 00:00:00	1	2	3	Actualizar
caca@caquita.com	caca2	caca3	caca4	1990-06-13 00:00:00	1	2	3	Actualizar
chrgnm@yahoo.es	yo	yo	yo	1988-08-10 00:00:00	78	178	47009	Actualizar
juan@gmail.com	Juan	Perez	Garcia	1990-07-05 00:00:00	70	170	47008	Actualizar
marcoacia33@gn	Marcos	García	Alonso	1990-07-21 00:00:00	66	170	47006	Actualizar
osferar@gmail.com	osc	ar	f	1987-12-28 00:00:00	0	0	0	Actualizar

Ilustración 64: Ver gestión de usuarios, administrador

BLOQUE VII. MANUALES

Para ello, modificamos los atributos deseados y para finalizar pulsamos en actualizar, tras lo cual el sistema mostrara un mensaje de confirmación.

12.4 Optimización al realizar rutas

En cuanto a la ubicación del dispositivo, es suficiente con el GPS, ya que el grado de fallo es muy pequeño y el consumo de datos es el menor posible. Si bien, también se puede mejorar la precisión con el uso de datos y/o WIFI, aunque ello conlleva un mayor consumo de datos.

A continuación se muestra una imagen de la configuración de la posición en un sistema Android, y como llegar a él:

Ajustes – Ubicación – Modo:

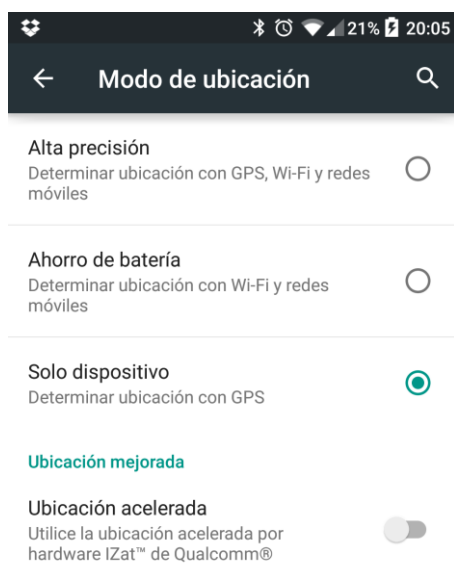


Ilustración 65: Configuración ubicación

Bloque VIII.

Conclusiones

BLOQUE VIII. CONCLUSIONES

Capítulo 13.

Conclusiones

13.1 Objetivos logrados

El objetivo principal alcanzado al finalizar el proyecto ha sido la implementación y puesta en marcha de una aplicación web móvil para poder crear una ruta manualmente o realizar una ruta libremente con versión escritorio y responsive, y un contacto vía email.

Se ha conseguido determinar aquellas herramientas de libre utilización que han sido útiles al presente trabajo y se han utilizado éstas con la mayoría de sus funcionalidades, lo que ha obligado a estudiar su documentación y sus estructuras. Este apartado es el que ha tenido una especial complejidad pues como se mencionó en la introducción, el prescindir deliberadamente de herramientas profesionales, ha obligado a buscar alternativas “free” y solucionar en la medida de lo posible sus carencias o limitaciones. Como ejemplo concreto de este comentario, cabe mencionar la necesidad de trabajar simultáneamente con herramientas dispares (del lado del cliente y del lado del servidor) tales como jQuery, PHP, Ajax. Un gran logro ha sido, por tanto, el interactuar con tales herramientas, buscar su compatibilidad y obtener lo mejor de cada una de ellas.

Se ha puesto de manifiesto una vez más como la informática se acerca al mundo cotidiano aportando al “usuario de a pie” una herramienta que complementa sus momentos de ocio.

Si bien, el mayor logro ha sido el poner en práctica muchas ideas y conocimientos adquiridos durante la carrera y plasmarles en un trabajo, y sobre todo, desarrollar este conforme a una estructura predeterminada, en un tiempo determinado y con objetivos concretos

13.2 Análisis global

El primer obstáculo (afortunadamente superado en los inicios del presente trabajo) ha sido el partir de cero. Inicialmente me pareció un reto excesivamente complejo. Fue en el día a día y conforme el presente trabajo comenzaba a tomar forma, como el camino se me allanaba.

BLOQUE VIII. CONCLUSIONES

Una dificultad añadida ha sido la necesidad ineludible de profundizar en algunos de los temas abordados durante la carrera. Sin ir más lejos, ha sido necesario un aprendizaje en profundidad en lo referente a los lenguajes de programación utilizados. Adquirir y asimilar la documentación de la ingente cantidad de herramientas (APIS, Frameworks, script...) existentes. Determinar cuáles eran las más idóneas, las más estables, las más compatibles,...

Sin duda alguna, un hándicap ha sido la inicial pretensión de que la aplicación desarrollada fuera compatible para la mayoría de los dispositivos móviles existente en el mercado, con su enorme variedad de características (tamaños de pantalla,...)

13.3 Trabajo futuro

Respecto a este apartado, insistir en la renuncia expresa a cualquier aplicación de índole profesional y reiterar el carácter pedagógico de la misma. Si bien, humildemente expresar que quedan sentadas las bases para distintas evoluciones y mejoras que pudieran dar como resultado aplicaciones profesionales de gran calado relacionadas con la geolocalización y control de desplazamientos. Me viene a la mente pensar en el mundo del transporte, de la distribución, etc.

Posibles mejoras:

- Login en la aplicación a través del login de Facebook, Twitter o Google Plus.
- Separación por tipo de deportes.
- Avisos acústicos y música durante las rutas.
- Estadísticas y gráficos de las rutas y el trabajo físico realizado.
- Versión de App para Android e IOS.
- Optimización de la geolocalización dependiendo de diferentes parámetros.
- Creación de URL con rutas específicas.
- Implementación de https.

Bloque IX.

Apéndice

BLOQUE IX. APÉNDICE

Capítulo 14.

Apéndices

14.1 Contenido del CD

El CD está estructurado con los siguientes directorios:

- Código Fuente: comprende los ficheros fuentes de la aplicación según estructurados en el capítulo 9
- Instalacion: contiene todo el código fuente comprimido y la estructura de la base de datos, ambos necesarios para la instalación del proyecto.
- Memoria: memoria del trabajo fin de grado.

14.2 Referencias

14.2.1 Libros y manuales

[1] I. Sommerville, *Software Engineering*, 9th ed., Addison-Wesley, 2011.

[2] M. Fowler, *Patterns of Enterprise Application Architecture*

[3] Gauchat, Juan Diego. *El gran libro de HTML5, CSS3 y Javascript*. Barcelona: MARCOMBO S.A., 2012

[4] Project Management Institute. *Project Management Body of Knowledge*.

[5] Philippe Kruchten. *The Rational Unified Process. An introduction*.

[6] Ivar Jacobson, Grady Booch, James Rumbaugh. *El Proceso Unificado de Desarrollo Software*. Addison Wesley, 1999

[7] Isabel Ramos Román, José Javier Dolado Cosín *Técnicas cuantitativas para la gestión en la ingeniería del software*

BLOQUE IX. APÉNDICE

[8] David Cochran. Twitter Bootstrap Web Development How-To. Starting, noviembre 2012

14.2.2 Documentación web

- Mozilla Developer Network [en línea].HTML5. [Consulta: 15 de marzo 2015]. Disponible en: <https://developer.mozilla.org/es/docs/HTML/HTML5>
- Google Developers [en línea] Google Maps JavaScript API. [Consulta: 29 de junio 2015]. Disponible en: <https://developers.google.com/maps/documentation/javascript/3.exp/reference>
- JQuery [en línea] API Documentation. [Consulta 2 de julio 2015]. Disponible en: <http://api.jquery.com/>
- JQuery TimeCircles [en línea] API Documentation. [Consulta 4 de julio 2015]. Disponible en: <https://plugins.jquery.com/timecircles/>
- Php [en línea] Manual de PHP [Consulta 5 de julio 2015]. Disponible en: <https://secure.php.net/manual/es/index.php>
- MySQL [en línea] MySQL Reference Manual [Consulta 9 de julio 2015]. Disponible en: <http://dev.mysql.com/doc/refman/5.5/en/>
- Apache [en línea] Documentación del servidor Http Apache [Consulta 9 de julio 2015]. Disponible en: <http://httpd.apache.org/docs/2.4/>
- JQuery User Interface [en línea] API Documentation. [Consulta 11 de julio 2015]. Disponible en: <https://jqueryui.com/dialog/>
- JQuery DataTables [en línea] API Documentation. [Consulta 12 de julio 2015]. Disponible en: <https://www.datatables.net/reference/api/>
- SDK de PHP Version [en línea] FaceBook Developers [Consulta 16 de julio 2015]. Disponible en: <https://developers.facebook.com/docs/php/api/5.0.0>
- Tweet Button [en línea] Twitter Developers [Consulta 17 de julio 2015]. Disponible en <https://dev.twitter.com/web/tweet-button>
- Google+ Platform [en línea] API de JavaScript de Google [Consulta 18 de julio 2015]. Disponible en: <https://developers.google.com/+/web/api/javascript?hl=es>

14.3 Glosario de términos

BBDD es la abreviatura de base de datos. Un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

BLOQUE IX. APÉNDICE

CSS lenguaje de hojas de estilo en cascada utilizado para describir el aspecto y el formato de un documento escrito en un lenguaje de marcas

MVC es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

API conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

JS abreviatura de JavaScript. Es un lenguaje de programación interpretado.

