



-PAJARRACO'S PAJARERÍA-

PORTAL WEB DE CONSULTA Y VENTA DE AVES

ALUMNOS: CAROLINA ANTÓN CEREZO

MARÍA SÁNCHEZ POSTIGO

TUTOR: FERNANDO DÍAZ



Agradecimientos

Este proyecto ha supuesto para nosotras un largo camino lleno de alti bajos que hemos ido superando gracias a nuestro esfuerzo, acompañado del apoyo de todos los que nos rodean. Queremos rendir en estas líneas un pequeño homenaje a todas las personas que han contribuido de una u otra forma a que hayamos llegado hasta aquí.

A nuestros compañeros de facultad, que no sólo han supuesto una ayuda indispensable para poder superar las diferentes asignaturas de la carrera, sino que a lo largo de los años han dejado de ser compañeros para convertirse en amigos.

A los compañeros de piso, en especial a ti Tere por el apoyo, la ayuda, el cariño, las risas y las partidas al scrabble.

En especial queremos agradecer la confianza que siempre han depositado en nosotras nuestros padres y hermanos. Siempre nos han demostrado su cariño y confianza inquebrantable y sin ellos no habríamos llegado donde estamos hoy en día.

Tampoco queremos olvidarnos de Julio y Pedro, compañeros de vida y fatigas. Siempre dispuestos a echarnos una mano. Puntos fundamentales de ánimo y apoyo que nos han proporcionado el empuje que necesitábamos en todo momento no solo par a finalizar el proyecto sino durante toda la carrera.

Por último y no por ello menos importante, queremos agradecer la paciencia y dedicación que nos ha demostrado nuestro tutor Fernando durante el desarrollo de todo el proyecto. Siempre amable y servicial, para nosotras ha sido una fuente de conocimiento fundamental en la que nos hemos apoyado para poder desarrollar todo el proyecto.

Índice general de contenidos

- I. Memoria del proyecto
- II. Análisis del Sistema de Información
- III. Diseño del Sistema de Información
- IV. Construcción del Sistema de Información
- V. Manual de Usuario

Memoria del proyecto

Índice de Contenido

1. Introducción.....	12
1.1. Identificación del proyecto	12
1.2. Organización de la documentación	12
1.3. Estructura del CD.....	12
2. Descripción general del proyecto.....	14
2.1. Objetivos.....	14
2.2. Cuestiones metodológicas.....	15
2.3. Tecnologías de desarrollo	15
3.- Descripción general de producto	17
3.1.- Funcionalidades del producto	18
3.2. Arquitectura del producto	19
3.2.1. Arquitectura general del producto	19
3.2.2. Arquitectura interna del producto.....	19
3.2.3. Funcionamiento de la aplicación PAJARRACO'S PAJARERÍA	20
3.3. Despliegue del producto	21
4. Planificación y presupuesto	23
4.1. Modelo de gestión del proyecto	23
4.1.1. Estimación mediante puntos de función.....	23
4.1.2. Estimación de costes siguiendo el modelo de COCOMO	29
4.2. Planificación.....	32
4.3. Presupuesto.....	35
5. Consideraciones adicionales	36
5.1. Cuestiones de análisis y diseño	36
5.2.- Cuestiones de implementación reseñables.....	36
5.3. Cuestiones reseñables sobre la base de datos.....	37
6. Conclusiones y posibles ampliaciones.....	38
6.1. Ampliaciones	38
Análisis del Sistema de Información	43
1. Definición del Sistema	43
1.1. Determinación del Alcance del sistema	43
1.2. Ámbito de la aplicación y perspectiva del proyecto	43
1.3. Objetivos del sistema	43
1.4. Identificación del entorno tecnológico	44
1.5. Especificación de estándares y normas	44
1.6. Identificación de usuarios participantes y finales.....	44
2. Establecimiento de Requisitos	46
2.1. Obtención de Requisitos.....	46
2.1.1 Requisitos No Funcionales.....	46
2.1.2 Requisitos Funcionales	47

2.2. Especificación de Casos de Uso	48
3. Identificación de Subsistemas de Análisis	50
4. Análisis de los Casos de Uso	61
4.1. Identificación de Clases Asociadas a un Caso de Uso	61
4.2. Descripción de la Iteración de Objetos	61
5. Análisis de Clases	70
5.1. Identificación de Responsabilidades y Atributos	70
5.2. Identificación de Asociaciones y Agregaciones	70
5.3. Identificación de Generalizaciones	70
6. Definición de las Interfaces de Usuario	72
6.1. Principios generales	72
6.2. Especificación de formatos Individuales de la Interfaz	72
6.3. Formatos de impresión	72
7. Especificación del Plan de Pruebas	73
8. Aprobación del Análisis del sistema de Información	74
Diseño del Sistema de Información	78
1. Antecedentes	78
1.1. Definición del proyecto	78
1.2. Objetivos del sistema	78
1.3. Objetivos del proyecto	79
2. Definición de la Arquitectura del Sistema	80
2.1. Definición de Niveles de Arquitectura	80
2.2. Identificación de Requisitos de Diseño y Construcción	82
2.3 Especificación de Excepciones	83
2.3.1 DNS Caído	83
2.3.2 Servidor Web Caído	83
2.3.3 ISP no da servicio	83
2.2.4 Servidor de Aplicaciones Caído	83
2.2.5 Servidor de Base de datos Caído	83
2.4 Especificación de Estándares y Normas de Diseño y Construcción	84
2.5 Identificación de los Subsistemas de Diseño	85
2.6. Especificación del Entorno Tecnológico	86
2.7. Especificación de Requisitos de Operación y Seguridad	91
3. Diseño de la Arquitectura de Soporte	92
4. Diseño de Casos de Uso Reales	93
4.1.- Diagrama casos de uso tienda virtual	93
4.2.- Diagrama casos de uso gestión del programa Mendel	98
4.3.- Diagrama casos de uso gestión de usuario	100
4.4.- Diagrama casos de uso gestión de productos y proveedores	103
4.5.- Diagrama casos de uso compra on-line	109

5. Matriz de Rastreabilidad	114
6. Diseño de Clases	119
6.1. Identificación de Clases Adicionales	119
6.2. Diagrama de clases.....	126
7. Diseño de la Arquitectura de Módulos del Sistema	144
7.1. Diseño del Modelo Físico de Datos.....	144
7.1.1. Diseño Conceptual	144
7.1.2. Diseño Lógico	156
7.1.3. Diseño Físico de Base de Datos	163
7.2. Optimización del Modelo Físico de Datos (Esquema Normalizado).....	164
8. Generación de Especificaciones de Construcción	176
8.1. Especificación del Entorno de Construcción	176
8.1.1 Entorno Tecnológico	176
8.1.2 Herramientas de construcción.....	176
8.1.3. Planificación.....	176
8.1.4. Requisitos de Operación y seguridad.....	176
9. Especificación Técnica del Plan de Pruebas.....	178
10. Modelo de Comportamiento del Sistema	179
10.1. Diagramas de Secuencia del Sistema.....	179
10.1.1. Diagrama de Secuencia Insertar Entidad.....	179
10.1.2. Diagrama de Secuencia Eliminar Entidad	180
10.1.3. Diagrama de Secuencia Modificar Entidad	180
10.1.4. Diagrama de Secuencia Buscar Información	181
10.1.5. Diagrama de Secuencia Comprar Producto	182
10.1.6. Diagrama de Secuencia Cambiar Credenciales Cliente	184
10.1.7. Diagrama de Secuencia Consultar Información.....	184
10.1.8. Diagrama de Secuencia Identificarse.....	185
10.2. Diagramas de Actividad del Sistema	186
10.2.1. Diagrama de Actividad Insertar Entidad.....	186
10.2.2. Diagrama de Actividad Eliminar Entidad	186
10.2.3. Diagrama de Actividad Modificar Entidad.....	187
10.2.4. Diagrama de Actividad Buscar Información.....	187
10.2.5. Diagrama de Actividad Comprar Producto	188
10.2.6. Diagrama de Actividad Cambiar Credenciales Cliente	189
10.2.7. Diagrama de Actividad Consultar Información	190
10.2.8. Diagrama de Actividad Identificarse	190
1. Preparación del Entorno de Generación y Construcción.....	194
1.1. Implantación de la Base de Datos Física o de Ficheros	194
2. Ejecución de las pruebas unitarias	197
2.1. Página principal de usuario.....	197

2.2. Alta y modificación de usuario.....	197
2.3. Catálogo de productos.....	199
2.4. Contacto.....	199
2.5. Pedidos.....	199
2.6. Formulario de pagos.....	200
2.7. Programa Mendel.....	201
2.8. Gestión de clientes.....	205
2.9 Gestión de cliente.....	206
2.10. Gestión de proveedor.....	206
2.11. Alta y modificación de proveedor.....	207
2.12. Gestión de productos.....	207
2.13. Modificación de productos.....	208
2.14. Pedidos.....	209
2.15. Gestión de pedido.....	209
2.16. Gestión de pedidos a clientes.....	209
2.17. Gestión de pedidos a proveedores.....	210
2.18. Gestión de centros.....	211
2.19. Alta y modificación de centros.....	212
2.20. Consulta de stock.....	213
2.21. Nueva oferta.....	213
2.22. Modificación oferta.....	214
2.23. Visualización de gráficos.....	214
3. Ejecución de las pruebas de integración.....	216
3.1 Diseño pruebas de integración.....	217
3.2. Definición de las pruebas.....	217
3.2.1. Alta usuario.....	217
3.2.2. Modificación datos usuario.....	218
3.2.3. Programa Mendel.....	218
3.2.4. Gestión de proveedores.....	223
3.2.5. Gestión de centros.....	225
3.2.6. Gestión de productos.....	225
3.2.7. Pedidos a proveedores.....	226
3.2.8. Pedidos de clientes.....	226
3.2.9. Consulta de stock.....	227
3.2.10. Generar catálogo.....	227
3.2.11. Gestión clientes.....	228
3.2.12. Visualización de gráficos.....	228
1. Introducción.....	230
2. Requisitos mínimos.....	231
2.1. Requisitos hardware.....	231

2.2. Requisitos software	231
3. Manual de usuario	231
3.1. Presentación de la aplicación.....	231
3.2. Funcionalidades de la aplicación	231
3.3. Accediendo a PAJARRACO'S PAJARERÍA	232
3.4. Menú de usuario	233
3.4.1. Cómo registrarse en la web.....	233
3.4.2. Visualizar catálogo de productos.....	235
3.4.3. Conocer información acerca de numerosas aves	236
3.4.4. Consultar el listado de centros	237
3.4.5. Cómo ponerse en contacto con la empresa.....	237
3.5. Menú cliente.....	238
3.5.1. Cómo realizar un pedido a través de la web.....	239
3.5.2. Gestionar datos de usuario	243
3.5.3. Programa Mendel.....	244
3.6. Menú administrador	250
3.6.1. Gestión de clientes	251
3.6.2. Gestión de proveedores.....	253
3.6.3. Gestión de productos.....	255
3.6.4. Gestión de pedidos	260
3.6.5. Consulta de stock	262
3.6.6. Catálogo de ofertas	263
3.6.7. Visualización gráfico.....	266
4. Manual de administración - parte servidora	267
4.1. Instalación Java.....	267
4.2. Instalación del Servidor de Aplicaciones	267
4.3. Instalación de la herramienta de desarrollo	268
4.4. Instalación del Sistema Gestor de Base de Datos	268
4.5. Instalación del Servidor Web.....	268
4.6. Instalación de la aplicación	269
4.7. Reinicio de la aplicación.....	274
4.8. Export / Import de la Base de Datos	275
4.9. Export / Import de la Base de Datos	275
Bibliografía	277

Índice de Contenido

1. Introducción	12
1.1. Identificación del proyecto	12
1.2. Organización de la documentación.....	12
1.3. Estructura del CD	12
2. Descripción general del proyecto	14
2.1. Objetivos	14
2.2. Cuestiones metodológicas	15
2.3. Tecnologías de desarrollo	15
3.- Descripción general de producto	17
3.1.- Funcionalidades del producto	18
3.2. Arquitectura del producto.....	19
3.2.1. Arquitectura general del producto.....	19
3.2.2. Arquitectura interna del producto	19
3.2.3. Funcionamiento de la aplicación PAJARRACO'S PAJARERÍA	20
3.3. Despliegue del producto.....	21
4. Planificación y presupuesto.....	23
4.1. Modelo de gestión del proyecto	23
4.1.1. Estimación mediante puntos de función	23
4.1.2. Estimación de costes siguiendo el modelo de COCOMO	29
4.2. Planificación	32
4.3. Presupuesto	35
5. Consideraciones adicionales	36
5.1. Cuestiones de análisis y diseño	36
5.2.- Cuestiones de implementación reseñables	36
5.3. Cuestiones reseñables sobre la base de datos	37
6. Conclusiones y posibles ampliaciones	38
6.1. Ampliaciones.....	38

1. Introducción

1.1. Identificación del proyecto

Título: PAJARRACO'S PAJARERIA.

Autores: Carolina Antón Cerezo y María Sánchez Postigo.

Tutor: Fernando Díaz Gómez.

Departamento: Informática.

1.2. Organización de la documentación

La documentación del proyecto PAJARRACO'S PAJARERÍA se distribuye en tres bloques principales bien diferenciados, que pasamos a describir brevemente a continuación:

- **Sección I - Memoria del proyecto:** este documento va dirigido a cualquier público. En él se expone una visión general del producto software desarrollado, así como los objetivos que se han perseguido con la realización de la aplicación. En esta parte de la documentación encontraremos una breve descripción tanto de la arquitectura, como de cuestiones de diseño que hemos seguido para llevar a cabo la aplicación. También se mostrará una planificación detallada de las tareas o hitos en las que se desglosa el proyecto y el presupuesto que hemos estimado que conllevaría realizar cada una de ellas. Por último se detallan las cuestiones metodológicas y tecnológicas más importantes, para finalizar con una serie de conclusiones y ampliaciones futuras.
- **Sección II - Análisis del sistema:** en esta sección obtendremos una especificación detallada del sistema que satisfaga las necesidades de información de los usuarios y sirva de base para el posterior diseño del sistema. En esta etapa se identificará las metas que se persiguen con la aplicación y se analizará las perspectivas, necesidades y requerimientos que puedan ayudar al desarrollo del proyecto.
- **Sección III - Diseño del sistema:** en esta sección se definirá la arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información. En esta parte del proyecto obtendremos un análisis detallado de todo el proceso de construcción sistema de información, la definición de la arquitectura técnica del sistema, un plan de pruebas detallado, la definición de todos los requisitos de implantación y el diseño del procedimiento de migración y la carga inicial del proceso.
- **Sección IV - Construcción del sistema de información:** esta parte de la documentación es muy técnica y específica, por lo que va claramente dirigida a un público concreto con conocimientos técnicos, como pueden ser desarrolladores y analistas informáticos. En ella pretendemos aportar la información necesaria para que el proyecto pudiera ser comprendido y llevado a cabo por un equipo técnico, siguiendo las pautas de análisis que en él se desarrollan. En esta actividad se asegura la disponibilidad de todos los medios necesarios para que se pueda llevar a cabo la construcción del sistema de información. Se incluye la preparación de puestos, creación y carga inicial de la base de datos y las herramientas de generación del código entre otros.
- **Sección V - Manual de usuario:** en el último bloque de documentación hemos elaborado un manual de usuario que, como su nombre indica, irá dirigido a todas aquellas personas que vayan a utilizar la aplicación. En él se explica con detenimiento cada una de las funcionalidades disponibles en la aplicación.

1.3. Estructura del CD

El proyecto incluye un cd en el que hemos incluido toda la documentación que hemos ido elaborando a lo largo del desarrollo de la aplicación web:

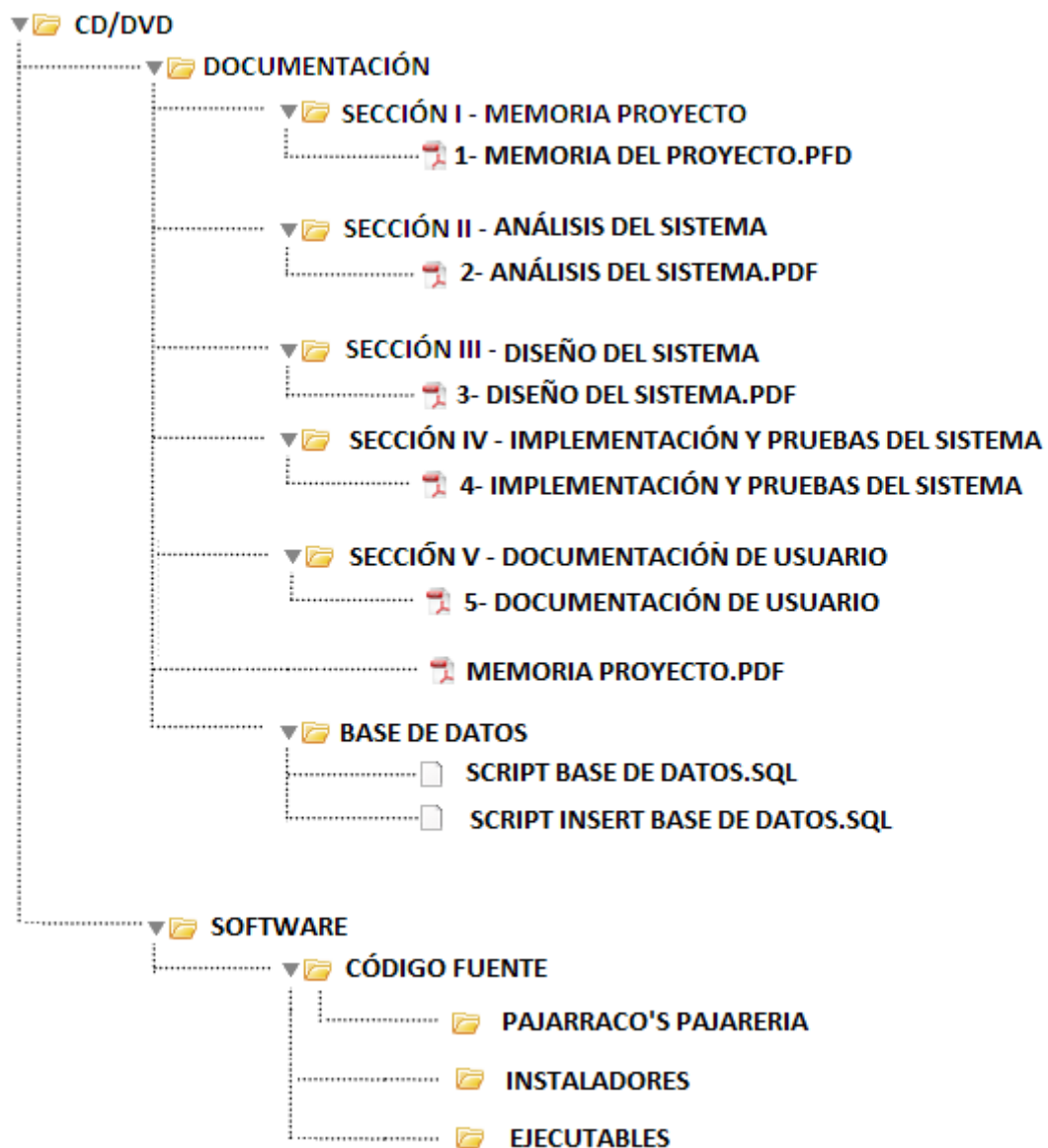


Figura 1. Estructura documentación CD

2. Descripción general del proyecto

El ritmo de vida actual hace que cada vez tengamos menos tiempo disponible. Esto unido al aumento del uso de Internet, ha hecho que proliferen las “empresas online” que permiten hacer compras a través de sus webs de forma rápida cómoda y desde cualquier lugar. Por eso, cada vez un mayor número de empresas utilizan Internet como un escaparate a través de la cual nos acercan y ofrecen sus productos y servicios. También, el uso de aplicaciones web ayuda a las empresas a gestionar de manera más eficiente sus recursos y abarcar una mayor cuota de mercado, de modo que su cartera de clientes se puede extender geográficamente. Esta utilización de las webs como medio de venta se ha hecho especialmente necesaria entre las Pymes (pequeñas y medianas empresas), que se han visto obligadas a buscar alternativas para acercar sus productos al público ante el aumento de grandes superficies comerciales.

Por ello este proyecto surge de la necesidad de incorporar a un negocio tradicional al mundo online, que es cada vez más utilizado por los negocios para vender sus productos. Como ya hemos dicho, la creación de la web para la empresa supondrá una reducción de los costes de venta además de una expansión de negocio, ya que podrá venderse productos en mayor extensión geográfica, sin necesidad de disponer de tiendas físicas. Nuestra aplicación no sólo será capaz de acercar el catálogo de productos a un público mayor, sino que además supondrá una manera más cómoda, fácil y fiable de realizar el mantenimiento y gestión de toda la empresa, ya que el desarrollo de aplicaciones a medida mejora el servicio y las prestaciones. Gracias a esta aplicación, el personal de la empresa podrá realizar entre otras cosas, una gestión del stock de todos los productos disponibles en todas las tiendas, controlar los proveedores que suministran los productos, además de analizar la situación de la empresa mediante un estudio de los productos que se venden en los diferentes centros etc. Además las interfaces gráficas de la web ayudarán a que la aplicación sea más atractiva y amigable para los usuarios por lo que no requerirá de grandes conocimientos informáticos para su manejo.

2.1. Objetivos

El objetivo de este proyecto es desarrollar una aplicación web que sea capaz de satisfacer las necesidades de una empresa de venta de pájaros y productos para pájaros. De esta forma, una venta pasa de ser un proceso costoso en el que se emplea mucho tiempo y muchos recursos humanos, a ser un proceso automatizado, rápido y que permite la realización de múltiples tareas simultáneamente.

La aplicación no solo cumplirá las funciones de una tienda online sino que además ofrecerá una serie de utilidades adicionales para la propia gestión de la empresa. Lógicamente esto nos plantea la necesidad de tener que filtrar las funcionalidades de la web por el tipo de usuario, ya que no queremos que cualquier usuario pueda acceder, por ejemplo, a los datos administrativos de la empresa. Para controlar el acceso a la aplicación se ha desarrollado un sistema de perfiles que se asignarán a los diferentes usuarios y que darán acceso a funcionalidades accesibles por su rol. En concreto se han establecido tres perfiles de usuarios que se corresponderán con tres menús de opciones diferentes.

Estos perfiles son:

- **Perfil usuario:** usuario no registrado en la web.
 - Podrá navegar por la web pero con acceso restringido (consultas de productos)
 - Darse de alta como cliente.

- **Perfil cliente:** usuario registrado en el Área de Clientes.
 - Podrá navegar por toda la web.
 - Realizar pedidos.
 - Tendrá acceso al foro.
 - Podrá modificar sus datos de usuario o darse de baja.
 - Tendrá acceso al uso del programa “Mendel”.

- Podrá recibir en su correo los catálogos personalizados o descargarse en la web el catálogo de productos.
- **Perfil administrador:** usuario encargado de la gestión de la tienda.
 - Podrá navegar por la web.
 - Podrá gestionar altas, bajas y modificación de clientes.
 - Podrá gestionar altas, bajas y modificación de proveedores.
 - Podrá realizar pedidos de productos y gestionar los envíos a clientes.
 - Gestión del stock de las diferentes tiendas.
 - Podrá gestionar el foro.
 - Altas, bajas y modificación de productos.
 - Generar catálogos a usuarios.
 - Visualización de gráficos.

2.2. Cuestiones metodológicas

Como objetivos de este proyecto nos hemos propuesto, por un lado, cubrir todas las funcionalidades que se han definido para esta primera fase y, por otro, tratar de ajustarnos lo máximo posible a los plazos de entrega que se estimarán en la planificación del proyecto. Para poder cumplir dichos objetivos hemos seguido una metodología de desarrollo. Las metodologías de desarrollo software surgieron de la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y documentación para poder desarrollar un producto software.

Nosotros hemos seguido una metodología de carácter “pesado” denominada RUP (Rational Unified Process) ya que, es una metodología que funciona muy bien en la elaboración de sistemas orientados a objetos, como es nuestro caso. Y además se basa en la utilización del lenguaje UML para la definición y diseño del sistema. Las metodologías pesadas se centran en la elaboración detallada de los procesos y tareas a realizar, herramientas que se van a utilizar, y requiere de una preparación extensa de documentación, ya que el objetivo es prever todo de antemano.

La metodología RUP divide las fases de desarrollo de un proyecto en cuatro bloques:

- **Inicio:** el objetivo de esta fase es tener una visión clara de lo que se quiere desarrollar.
- **Elaboración:** es la etapa en la que se escoge la arquitectura óptima para el proyecto.
- **Construcción:** se obtiene la capacidad operacional inicial.
- **Transmisión:** obtenemos el producto acabado y definido.

Por último, para concluir con la metodología empleada vamos a detallar cada una de las fases que hemos seguido a lo largo del desarrollo del proyecto. Estas fases son:

- **Definición de estudio y viabilidad:** se estudia el objetivo que queremos conseguir y las necesidades que pretendemos cubrir con el desarrollo del producto software.
- **Análisis:** elaboración de los documentos de análisis del sistema.
- **Diseño:** etapa destinada al diseño del sistema.
- **Codificación:** desarrollo e implementación del producto software.
- **Pruebas:** batería de pruebas del sistema.
- **Integración del sistema en producción:** estabilización e instalación del sistema en un entorno productivo.
- **Mantenimiento:** elaboración de las tareas necesarias para el correcto funcionamiento de la aplicación en el entorno productivo.

2.3. Tecnologías de desarrollo

La aplicación web PAJARRACO’S PAJARERÍA se ha desarrollado mayoritariamente con tecnologías OpenSource, lo cual quiere decir que no requieren del pago de una licencia para poder utilizarlas. Este es principalmente el motivo por el cual nos hemos decantado por este tipo de programas, el ahorro de costes.

Los programas más destacados que hemos utilizado son:

- **MySql de Oracle:** es un sistema de gestión de bases de datos relacionales. MySql es una base de datos muy rápida en cuanto a lectura de información y muy adecuada en aplicaciones web (como es nuestro caso), ya que, hay muy baja concurrencia en la modificación de datos y sin embargo hay gran cantidad de trabajo de lectura de datos.
- **IBM Webshpere Studio de IBM:** es el entorno de desarrollo que hemos escogido para la implementación de la lógica o código de la aplicación. Está diseñado con tecnologías abiertas y basado en Eclipse. Es compatible con múltiples lenguajes, plataformas y dispositivos. Está diseñado para cumplir los requisitos de programación específicos de empresas, desde interfaces web hasta aplicaciones de servidor, desde programación individual hasta entornos avanzados para equipos, desde programación en java hasta integración de aplicaciones.
- **Pacestar UML Diagrammer v.6.20.2040:** es la herramienta que hemos utilizado para el modelado de los diagramas expuestos en la documentación técnica del proyecto, como son los diagramas de casos de uso, de actividad, de secuencia, de clases de estados, de paquetes... Este programa requiere de licencia comercial para el sistema operativo Windows, pero se ofrece una descarga gratuita de prueba.
- **Microsoft Office 2010:** el paquete de Microsoft Office 2010 ofrece un conjunto de programas de entre los cuales hemos utilizado el Microsoft Office Excel (para el diseño de tablas) y Microsoft Office Word para el desarrollo de toda la documentación del proyecto.

El lenguaje de programación que hemos decidido utilizar para la implementación ha sido Java. Los motivos de la elección son principalmente los siguientes:

- Es un lenguaje que no depende del tipo de plataforma que emplea, lo cual nos permite que sea portable a otros sistemas operativos.
- Java es un lenguaje orientado a objetos lo cual supone una gran ventaja, ya que permite al código emular con mayor exactitud los problemas del mundo real que se desean resolver. Además es un lenguaje dinámico, en el que los pequeños fragmentos de código Java se ensamblan en tiempo de ejecución en el programa y no en el momento en el que se escribe el código.
- Además Java dispone de una serie de funcionalidades ya implementadas y recogidas en un conjunto de paquetes que facilitan el desarrollo de las aplicaciones y que pueden ser utilizadas por cualquier desarrollador.
- Este lenguaje puede ser desarrollado en plataformas OpenSource que es el requisito primordial que buscábamos.

Por último Java tiene la característica de que libera la memoria automáticamente, por lo cual los desarrolladores no se tienen que centrar en la liberación de memoria y como consecuencia mejora notablemente el rendimiento de las aplicaciones. Cabe destacar que, a pesar de que a pesar de que el código de la aplicación está desarrollado en java, el mapa que se visualiza en cuando accedemos a la opción de contacto del menú de usuario está desarrollado en flash.

3.- Descripción general de producto

La aplicación web PAJARRACO'S PAJARERÍA, como ya hemos dicho anteriormente, es una aplicación dirigida a varios públicos algunos de los cuales no tienen porque tener ningún conocimiento informático, ni experiencia con la utilización de aplicaciones web. La interfaz gráfica es el elemento que permite al usuario interactuar con los contenidos, y no sólo se precisa una interfaz atractiva, sino además debe ser funcional.

Con la idea de simplificar el uso de la aplicación para los usuarios, hemos prestado especial atención al desarrollo de la interfaz gráfica para que el usuario establezca un contacto más fácil e intuitivo con la web. Hemos tratado de hacer la aplicación lo más atractiva, sencilla y amigable posible. Uno de los objetivos que hemos perseguido, es que el usuario sea capaz de saber en qué punto del proceso que esté realizando se encuentra en todo momento. La elaboración de una interfaz gráfica clara es fundamental principalmente para hacer que el proceso de compra sea sencillo y con ello que el porcentaje de ventas aumente y no haya dificultades ni dudas por parte del cliente durante dicho proceso.

En el diseño de las páginas de la web PAJARRACO'S PAJARERÍA podemos diferenciar tres partes:

- En función del rol de usuario, se mostrará un menú diferente en la parte izquierda de la pantalla en la que se muestran las opciones a las que tiene acceso.
- En la parte superior de cualquier pantalla se muestra el logo de la empresa y los campos para registrarse en caso de no estar logado o el nombre del usuario en caso de estarlo y el enlace para cerrar la sesión.
- En la parte central de la pantalla se mostrarán los formularios correspondientes a la opción que haya escogido el usuario.

Nuestra interfaz gráfica cumple varias características entre las que destacamos las siguientes:

- Es de fácil comprensión y utilización.
- La tipografía de color y letra es uniforme en todas las pantallas, cuidando el diseño de las formas y la coherencia entre todos los elementos visuales.
- Tiene un diseño claro y eficiente mediante el establecimiento de menús, submenús, enlaces, botones e iconos que facilitan la navegación.
- La acción que se está realizando es identificada claramente en todo momento.
- Las operaciones son rápidas y reversibles en todo momento pudiéndose regresar a la pantalla previa.
- También se ha prestado cuidado con la presentación de mensajes de error de manera que sean comprensibles por el usuario.

Además del cuidado del diseño de las páginas de la web, nuestro desarrollo se ha realizado pensando en cubrir los siguientes objetivos:

- ✓ **Accesibilidad:** desde diferentes navegadores.
- ✓ **Escalabilidad:** la aplicación debe poder adaptarse a un incremento de usuarios, del tamaño de los datos que maneja y debe ser mantenible.
- ✓ **Reutilización de recursos:** se debe programar el código de la aplicación de tal manera que se puedan reutilizar módulos para diferentes partes de la aplicación.
- ✓ **Portable a otros sistemas operativos:** como ya hemos comentado anteriormente la aplicación ha sido desarrollada en una plataforma de desarrollo que no depende del sistema operativo.

Por último, como tareas para abordar en una segunda fase quedarán pendientes varias relacionadas con la mejora de nuestra interfaz gráfica, funcionalidad y manejo de la web. Algunas que se nos ocurren pueden ser la creación de una herramienta de ayuda y soporte poder resolver cualquier pregunta que pueda surgir acerca del manejo de la web, o la elaboración de una sección de FAQs en las que se muestren las preguntas más realizadas por los usuarios. También se debería de mejorar el tratamiento de los errores, intentando especificar al máximo posible el motivo por el cual no se ha podido completar una acción solicitada de manera que sea lo más claro posible para el usuario. Por último se nos ocurre que una buena manera de renovar rápidamente la imagen de la empresa y la web es mediante la actualización de los temas y colores utilizados en las diferentes pantallas de la aplicación.

3.1.- Funcionalidades del producto

A continuación vamos a enumerar las funcionalidades que podemos encontrar en los distintos menús de la web, agrupadas por módulos de gestión:

Gestión de productos:

- Buscar productos.
- Añadir productos.
- Eliminar productos.
- Modificar productos.
- Listar catálogo de productos.
- Consultar el stock de un producto.
- Consultar descendencia Diamantes de Gould.
- Añadir oferta al catálogo de productos.
- Eliminar oferta del catálogo de productos.
- Modificar el precio de una oferta.

Gestión de clientes:

- Alta de usuario.
- Baja de usuario.
- Buscar clientes.
- Modificación de datos de usuario.
- Listar clientes.

Gestión de proveedores:

- Alta de proveedor y de su catálogo de productos.
- Baja de proveedor y de su catálogo de productos.
- Modificación del catálogo de productos de un proveedor.
- Buscar proveedor.
- Listar proveedores.
- Modificar catálogo de proveedor (alta/baja producto).

Gestión del foro:

- Contestar o dar de alta un mensaje.
- Eliminar un mensaje o un hilo de conversación.
- Añadir un nuevo tema.

Gestión de contacto con la empresa:

- Consulta centros.
- Añadir centro.
- Eliminar centro.
- Buscar centro.
- Listar centros.
- Modificación de centro.
- Envío de email.

Gestión de pedidos:

- Realización de pedidos por parte de clientes.
- Realización de pedidos a proveedores.
- Modificación estado pedidos de clientes.
- Buscar pedidos de clientes.
- Buscar pedidos hechos a proveedores.
- Listar pedidos de clientes.
- Listar pedidos a proveedores.
- Visualizar listado productos en pedido.
- Gráfico de ventas (por centro/producto/fecha/familia).

3.2. Arquitectura del producto

En este apartado se pretende dar una visión general de la arquitectura que hemos seguido para la implementación de nuestro proyecto. En primer lugar haremos una descripción a más alto nivel para posteriormente profundizar más en detalle cada componente de dicha arquitectura.

3.2.1. Arquitectura general del producto

La web de PAJARRACO'S PAJARERÍA utiliza varias tecnologías en las cuales nos hemos basado para poder desarrollar la lógica de nuestra aplicación y para poder almacenar, tratar y leer los datos que se manejan en los formularios de ésta.

El desarrollo de la lógica de la aplicación está íntegramente desarrollado en Java, por lo que la plataforma de ejecución Java es la base para el desarrollo de la aplicación web.

En cuanto a la parte visual de la aplicación, los formularios, son JSP (Java Server Pages) que son básicamente una tecnología Java que permite la generación de contenido dinámico web en forma de páginas estáticas HTML que no pueden proporcionar por sí solas esta potencia.

Nuestro modelo de datos o base de datos, será accedido mediante un API que nos proporciona Java llamada JDBC, que permite la ejecución de operaciones sobre la base de datos desde nuestro código java (lógica de nuestra aplicación). Esta comunicación se establecerá con independencia del sistema operativo sobre el que estemos trabajando pero siempre utilizando la sintaxis de SQL. Además esta interfaz Java se encargará de controlar y manejar todas las conexiones que se realicen hacia la base de datos.

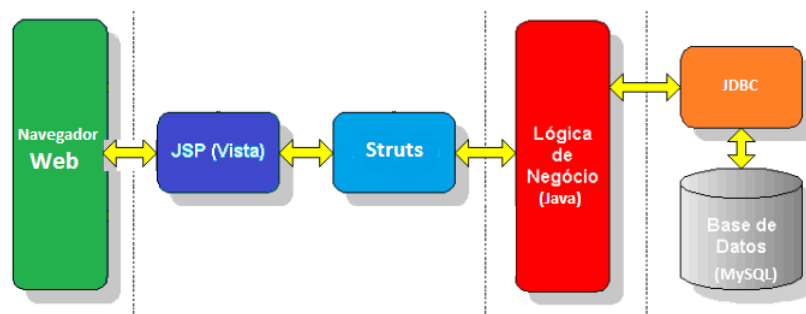


Figura 2. Estructura general y tecnologías utilizadas por nuestra aplicación

3.2.2. Arquitectura interna del producto

En este apartado vamos a centrarnos en describir cómo hemos desarrollado la parte lógica y visual de nuestra aplicación.

A la hora de implementar el core de negocio de nuestra aplicación nos hemos decantado por un patrón de diseño, que podría asemejarse al patrón modelo-vista-controlador con algunas diferencias y que pasamos a detallar a continuación:

- **Modelo-controlador:** en nuestro caso hemos decidido juntar ambas partes ya que no existen servlets como tal pero si contamos con el uso de struts, que será un poco quien se encargue del control de la acción que esté solicitando un usuario (su función será un poco la de intermediario entre las jsp y la parte de las clases java que contienen la lógica). En esta parte se engloba toda la funcionalidad de la aplicación. Estará compuesta por una serie de clases java gestionadas por struts. Struts es un framework que implementa el patrón modelo-vista-controlador en java y que se emplea fundamentalmente como soporte en el desarrollo de aplicaciones web.
- **Vista:** proporcionará una serie de páginas web dinámicamente al cliente, siendo para él simples páginas HTML. Nuestras páginas web serán jsp (java script page), las cuales mediante una serie de tags XML ofrecen un interfaz a

las clases y objetos Java proporcionados por nuestro servidor de aplicaciones. En estas páginas también hemos utilizado un lenguaje de programación llamado js (javascript), el cual se utiliza principalmente para la creación de web's dinámicas. Gracias a él podremos incorporar efectos, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Nuestros archivos js contendrán fundamentalmente las validaciones de los formularios de entrada de datos por parte del usuario, lo cual impedirá que llegue información mal formada a la parte de la lógica de negocio. Además cabe destacar que nuestras jsp utilizan hojas de estilo o css desarrolladas específicamente para dar un aspecto uniforme y atractivo a todos los formularios de la aplicación. La utilización de ficheros .js y .css nos proporcionará una serie de ventajas ya que además de reutilizar código (lo cual proporcionará un mantenimiento rápido y sencillo), liberará de carga a nuestro servidor lo cual repercutirá en el rendimiento de nuestra aplicación.

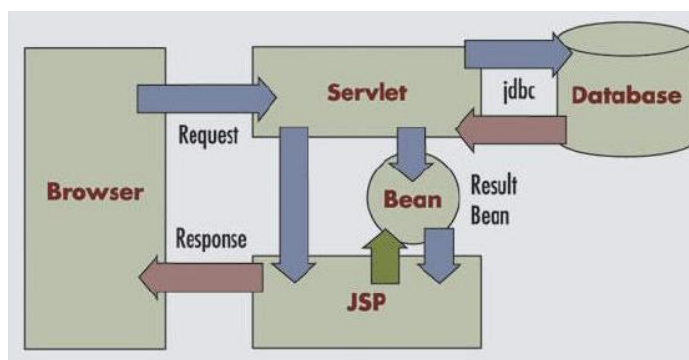


Figura 3. Modelo de arquitectura interna seguida en el desarrollo de la aplicación

3.2.3. Funcionamiento de la aplicación PAJARRACO'S PAJARERÍA

Este apartado tiene como finalidad dar una visión general del funcionamiento de la web desde que el usuario hace una petición hasta que finalmente llega a la base de datos o para leer información o para escribir.

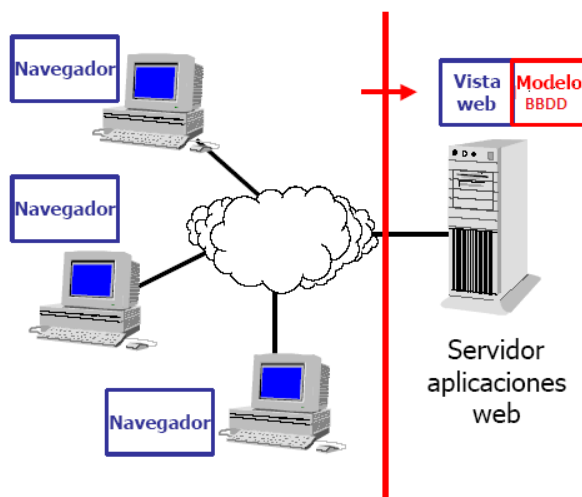


Figura 4. Entorno de ejecución de la aplicación

Un elemento que juega un papel fundamental en el funcionamiento de cualquier web es el servidor de aplicaciones. La arquitectura de un servidor de aplicaciones incluye una serie de subsistemas que pasamos a describir a continuación:

- **Servidor HTTP:** es el encargado de servir las páginas web a los usuarios que hacen una petición http a través de su navegador web.
- **Contenedor de aplicaciones:** contiene los servlet y JSP que componen la aplicación.
- **Contenedor Java:** contiene el código java de la aplicación.

El funcionamiento simplificado de cualquier web es el que se muestra en la siguiente figura:

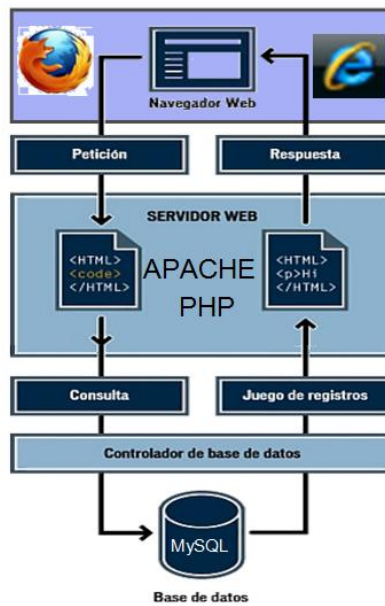


Figura 5. Esquema general de la generación dinámica de contenido

El usuario abre un navegador web y escribe en la barra de ese navegador la URL de la web a la que quiere acceder. La petición llega al servidor de aplicaciones web gracias a la conexión que se establece en base al protocolo de comunicación web HTTP. Es un protocolo de comunicación implementado sobre TCP/IP y que se basa en enviar las solicitudes que recibe de los usuarios hacia el servidor web y de responder a cada una de ellas. Por defecto este protocolo establece la comunicación con el servidor a través de un puerto por defecto 8080 (definido por servidores Tomcat). Cuando el servidor web recibe una petición debe recuperar el recurso solicitado del contenedor web, el cual gestionará su localización y ejecución. La JVM (maquina virtual de java) del contenedor web invocará al objeto solicitado JSP servlet. El servidor no creará una instancia por cada solicitud del mismo recurso sino que da servicio a múltiples solicitudes, las cuales generarán cada una un hilo de ejecución en el servidor, que a su vez se encargara de la gestión de todos los hilos.

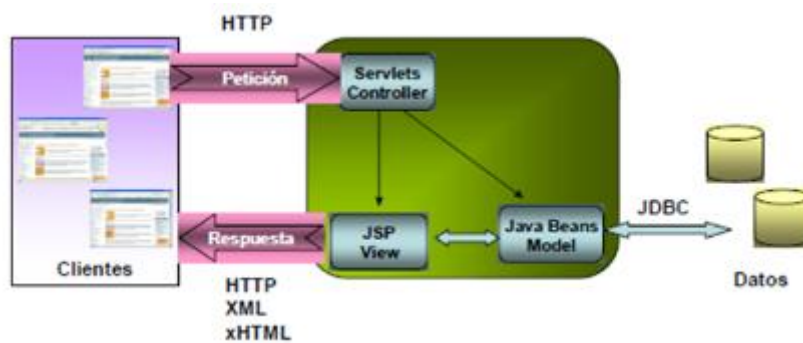


Figura 6. Esquema específico de la generación dinámica de contenido con tecnología JSP

3.3. Despliegue del producto

Las aplicaciones web están formadas por un conjunto de servlets, páginas jsp, ficheros html y clases Java de apoyo empaquetadas o no en ficheros jar, war o ear.



Figura 7. Estructura de directorios y archivos típicos de una aplicación JSP

Durante la etapa de desarrollo estableceremos una determinada estructura de carpetas que definirá la estructura de la aplicación. Posteriormente la aplicación se compilará, utilizando una serie de ficheros de configuración, formando un paquete .war (web application resource) que engloba todo su contenido jsp y clases java. Este paquete contendrá la misma estructura de paquetes que hemos definido durante la etapa de desarrollo. La finalidad es que dicho paquete pueda ser desplegado en diferentes servidores web manteniendo su funcionalidad y sin necesidad de realizar ninguna modificación de código. El paquete compilado se colocará en una ubicación específica definida en función del servidor web que se esté utilizando. Posteriormente se arrancará el servidor web para poder comenzar a utilizar la aplicación web.

4. Planificación y presupuesto

En este capítulo se mostrarán los datos obtenidos para la planificación y presupuesto del proyecto PAJARRACO'S PAJARERÍA.

4.1. Modelo de gestión del proyecto

El ciclo de vida ha marcado la metodología de trabajo, de principio a fin, del proyecto y la planificación del mismo. El ciclo de vida del desarrollo software nos proporciona la orientación que debe seguirse para obtener, a partir de los requisitos de los que partimos a la hora de elaborar una aplicación, sistemas que puedan ser utilizados por los usuarios a los que va dirigido. Todo este proceso comprende un conjunto de fases, procesos y actividades requeridas para ofertar, desarrollar, probar, integrar, explotar y mantener un producto software. El ciclo de vida que se decida utilizar para el desarrollo de un producto software define el orden de las tareas o actividades involucradas, además de la coordinación entre ellas, enlace y realimentación entre las etapas. Como punto de partida en la planificación de nuestro sistema, se deberá tener en cuenta el tamaño que tendrá. Para ello nos hemos decantado por una planificación en puntos de función que detallaremos a continuación.

4.1.1. Estimación mediante puntos de función

La estimación mediante puntos de función es un tipo de métrica utilizada para medir el tamaño de un producto software independientemente de la tecnología empleada para su desarrollo o explotación. Posteriormente se utilizara para evaluar el coste y el esfuerzo que supone el desarrollar el proyecto. Esta métrica consiste en asignar una serie de puntos a una aplicación informática en función de los datos que maneja y de los procesos o funciones que realiza sobre ellos desde el punto de vista del usuario. Los puntos de función que obtienen utilizando una función empírica basando en medidas cuantitativas del dominio de información del software y valoraciones subjetivos de la complejidad del software.

4.1.1.1. Explicación del proceso

Vamos a explicar brevemente en qué consiste este tipo de métrica.

En primer lugar habrá que identificar los componentes que intervienen en este tipo de métrica son:

- **Número de entradas externas en el sistema (EI, External inputs):** Procesos en los que se introducen datos y que suponen la actualización de cualquier archivo interno, es decir por ejemplo la información que introducen los usuarios en el sistema a través de los formularios de la aplicación o selección de opciones de menú etc.
- **Número de salidas del sistema (EO, External outputs):** Procesos en los que se envía datos al exterior de la aplicación, es decir es la información que el sistema aporta a los usuarios que la utilizan.
- **Número de archivos lógicos internos (ILF, Internal logical files):** datos, ficheros y bases de datos de uso exclusivo por parte de la aplicación que residen dentro de los límites del sistema y se mantienen a través de entradas externas.
- **Número de archivos externos (EIF, External interface files):** datos, ficheros y bases de datos que pueden ser utilizados por nuestra aplicación y por otros sistemas. Residen fuera de los límites del sistema y se mantienen por las entradas externas de otras aplicaciones.
- **Número de consultas externas (EQ, External queries):** entradas al sistema que requieren una respuesta por parte de este. Los datos de entrada no actualizan ni mantienen ningún archivo y los datos de salida no contienen datos derivados como por ejemplo las búsquedas que se realizan dentro de un sistema.

A continuación vamos a describir brevemente los pasos a seguir para realizar este tipo de estimación:

- Para el cálculo de los puntos de función, en primer lugar hay que identificar y contar los números de elementos de cada clase.

- Cada uno de estos elementos se clasificará según su grado de complejidad (alta, media o baja). No obstante la determinación de la complejidad es algo subjetivo.
- Por último se obtienen los PFNA (Puntos de Función No Ajustados) mediante una suma ponderada de esas cantidades con los pesos que aparecen a continuación:

Parámetro de medición	FACTOR DE PONDERACIÓN				=	Cuenta
	Cuenta	Simple	Medio	Complejo		
Número de entradas en el sistema	<input type="text"/>	X	3	4	6	<input type="text"/>
Número de salidas del sistema	<input type="text"/>	X	4	5	7	<input type="text"/>
Número de consultas externas	<input type="text"/>	X	3	4	6	<input type="text"/>
Número de ficheros internos	<input type="text"/>	X	7	10	15	<input type="text"/>
Número de ficheros externos	<input type="text"/>	X	5	7	10	<input type="text"/>
CUENTA TOTAL						<input type="text"/>

Figura 8. Pesos puntos de función no ajustados

A continuación presentamos una tabla donde se pueden ver los criterios que se seguirán para determinar la complejidad de los elementos:

Ficheros lógicos externos e internos.	Registros elementales	Datos elementales		
		1-9	20-50	>51
	1	Baja	Baja	Media
	2-5	Baja	Media	Alta
	>6	Media	Alta	Alta
Salidas y consultas	Tipos de ficheros	Datos elementales		
		1-5	6-19	>20
	0-1	Baja	Baja	Media
	2-3	Baja	Media	Alta
	>4	Media	Alta	Alta
Entradas	Tipos de ficheros	Datos elementales		
		1-4	5-15	>16
	0-1	Baja	Baja	Media
	2-3	Baja	Media	Alta
	>3	Media	Alta	Alta

Figura 9. Cálculo de la complejidad de los elementos

Para el cálculo de los PFNA (puntos de función no ajustados), realizaremos la suma de todos los parámetros que hemos calculado previamente:

$$PFNA = (N^{\circ} \text{ Entradas} \times \text{multiplicador (complejidad)}) + (N^{\circ} \text{ Salidas} \times \text{multiplicador (complejidad)}) + (N^{\circ} \text{ Fichero internos} \times \text{multiplicador (complejidad)}) + (N^{\circ} \text{ Ficheros externos} \times \text{multiplicador (complejidad)}) + (N^{\circ} \text{ Consultas externas} \times \text{multiplicador (complejidad)}).$$

Una vez obtenido el resultado deberemos realizar un ajuste mediante un factor de ajuste (FA). Para obtener el valor del factor de ajuste deberemos sumar 14 factores de complejidad siguiendo la siguiente fórmula:

$$FA = (0.01 \times \text{SUM (FC)}) + 0.65$$

Una vez obtenido el factor de ajuste podemos realizar el cálculo definitivo de los puntos de función:

$$PF = PFNA \times FA$$

A continuación vamos a detallar cada uno de los factores de complejidad:

1.- **Comunicación de datos:** los datos o información de control que la aplicación utiliza se envía o recibe a través de las facilidades de comunicación.

- 0 Aplicación es batch exclusivamente.
- 1-2 Impresión o entrada de datos remota.
- 3-5 Teleproceso (TP) interactivo.
- 3 TP interface a un proceso batch.
- 5 La aplicación es interactiva predominantemente.

2.- **Datos de procesamiento distribuidos:** "Distribuida" significa que los datos de la aplicación están distribuidos en dos o más procesadores diferentes (esto también implica un incremento del factor de comunicación de datos).

- 0 La aplicación no ayuda a la transferencia de datos o procesamiento entre los componentes de nuestro sistema.
- 1 La aplicación prepara datos para el usuario final de otro procesador.
- 2-4 Los datos se preparan para transferencia, se transfieren y se procesan en otro componente del sistema.
- 5 Las funciones de procesamiento se realizan dinámicamente en el componente más apropiado del sistema.

3. **Objetivos de rendimiento:** para el correcto funcionamiento de cualquier sistema es muy importante la respuesta dentro del sistema.

- 0-3 Análisis y diseño de las consideraciones del rendimiento son estándar. No se precisan requerimientos especiales por parte del usuario.
- 4 En la fase de diseño se incluyen tareas del análisis del rendimiento para cumplir los requerimientos del usuario.
- 5 Además se utilizan herramientas de análisis del rendimiento en el diseño, desarrollo e instalación.

4. **Configuración utilizada masivamente:** referente a la importancia del entorno. Esto es, si hay restricciones de memoria o del hardware.

- 0-3 La aplicación corre en una máquina estándar sin restricciones de operación.
- 4 Restricciones de operación requieren características específicas de la aplicación en el procesador central.
- 5 Además hay restricciones específicas a la aplicación en los componentes distribuidos del sistema.

5. **Tasa de transacción:** cuando un sistema recibe un elevado número de transiciones pueden darse problemas en el sistema (como por ejemplo problemas de rendimiento y lentitud).

- 0-3 Las tasas son tales que las consideraciones de análisis de rendimiento son estándares.
- 4 En la fase de diseño se incluyen tareas de análisis de rendimiento para verificar las altas tasas de transacciones.
- 5 Además se utilizan herramientas de análisis del rendimiento.

6. **Entrada de datos on-line:**

- 0-2 Hasta el 15% de las transacciones tienen entrada interactiva.
- 3-4 15% al 30% tienen entrada interactiva.
- 5 30% al 50% tienen entrada interactiva.

7. **Eficiencia para el usuario:**

- 0-3 No se especifican requerimientos especiales.
- 4 Se incluyen tareas de diseño para la consideración de factores humanos.
- 5 Además se utilizan herramientas especiales o de prototipado para promover la eficiencia.

8. **Actualización on-line:**

- 0 Nada.
- 1-2 Actualización on line de los ficheros de control. El volumen de actualización es bajo y la recuperación fácil.
- 3 Actualización on line de la mayoría de los ficheros internos lógicos.
- 4 Además es esencial la protección contra la pérdida de datos.
- 5 Además se considera el coste de recuperación de volúmenes elevados.

9. Procesamiento complejo: abarca la complejidad interna del sistema más allá de lo referente a la entrada, salida o lógica de procesamiento.

¿Qué características tiene la aplicación?

- Mucho procesamiento matemático y/o lógico.
- Procesamiento complejo de las entradas.
- Procesamiento complejo de las salidas.
- Muchas excepciones de procesamiento, muchas transacciones incompletas y mucho reprocesamiento de las transacciones.
- Procesamiento de seguridad y/o control sensitivo.

- 0 No se aplica nada de esto.
- 1 Se aplica alguna cosa.
- 2 Se aplican dos cosas.
- 3 Se aplican tres cosas.
- 4 Se aplican cuatro cosas.
- 5 Se aplica todo.

10. Reutilización: una de las características que se persigue en la elaboración de cualquier sistema es que el código de ésta pueda ser compartido o utilizable por otras aplicaciones.

- 0-1 Una aplicación local que responde a las necesidades de una organización usuaria.
- 2-3 La aplicación utiliza o produce módulos comunes que consideran más necesidades que las del usuario.
- 4-5 Además, la aplicación se "empaquetó" y documentó con el propósito de fácil reutilización.

11. Facilidad de instalación y conversión:

- 0-1 No se requieren por parte del usuario facilidades especiales de conversión e instalación.
- 2-3 Los requerimientos de conversión e instalación fueron descritos por el usuario y se proporcionaron guías de conversión e instalación.
- 4-5 Además se proporcionaron y probaron herramientas de conversión e instalación.

12. Facilidad de operación:

- 0 No se especifican por parte del usuario consideraciones específicas de operación.
- 1-2 Se requieren, proporcionan y prueban procesos específicos de arranque, backup y recuperación.
- 3-4 Además la aplicación minimiza la necesidad de actividades manuales, tales como instalación de cintas y papel.
- 5 La aplicación se diseña para operación sin atención.

13. Puestos múltiples:

- 0 El usuario no requiere la consideración de más de un puesto.
- 1-3 Se incluyeron necesidades de varios puestos en el diseño.
- 4-5 Se proporciona documentación y plan de apoyo para soportar la aplicación en varios lugares.

14. Facilidad de cambio: esfuerzo específico de diseño para facilitar cambios futuros.

- 0 No hay requerimientos especiales del usuario para minimizar o facilitar el cambio.
- 1-3 Se proporciona capacidad de consulta flexible.
- 4-5 Datos importantes de control se mantienen en tablas que son actualizadas por el usuario a través de procesos on-line interactivos.

Estos factores de complejidad se puntuarán de 0 a 5. Cada valor se asociará a un rango de porcentaje como se detalla en la tabla que mostramos a continuación:

Valor del Factor	Influencia en el Sistema	Porcentaje que afecta o es requerido por la aplicación
0	Ninguna	0%
1	Insignificante	1-20%
2	Moderada	21-40%
3	Media	41-60%
4	Significativa	61-80%
5	Fuerte	81-100%

Figura 10. Porcentaje factores de complejidad

4.1.1.2. Estimación mediante puntos de función

Una vez explicado el modelo de puntos de función que seguiremos, vamos a aplicarlo a nuestro sistema. Esto nos permitirá evaluar la funcionalidad que tendrá nuestro sistema software.

- **Número de entradas externas en el sistema:** en nuestro sistema hemos detectado 8 entradas de complejidad baja que pasamos a enumerar a continuación.

- Datos de usuario.
- Datos de foro.
- Datos de pedido.
- Datos de progenitores para comprobar herencia genética.
- Datos de proveedores.
- Datos de centros.
- Datos de ventas.
- Datos de catálogo.
- Mensajes de entrada al foro (nuevo hilo de conversación, respuestas a otros ya existentes...).

- **Número de salidas del sistema:** las salidas que hemos identificado en nuestro sistema las podemos categorizar de complejidad baja (mensajes mostrados por pantalla), o complejidad media (resto). Son:

- Complejidad baja: 3 salidas.

- ✓ Mensajes de error al no poder realizar una acción concreta por causas ajenas al usuario.
- ✓ Mensajes de advertencia al no poder realizar una acción por errores del usuario.
- ✓ Mensajes de éxito tras finalizar una acción correctamente en el sistema.

- Complejidad media: 10 salidas.

- ✓ Gráficos o informes de ventas por centro o producto.
- ✓ Resumen pedido realizado por el cliente.
- ✓ Apertura de nuevo mensaje de Outlook para enviar un mail de contacto con la empresa.
- ✓ Informe de posible descendencia genética.
- ✓ Resumen de los hilos abiertos en el foro.
- ✓ Información de empresa.
- ✓ Información de aves.
- ✓ Visualización del catálogo de productos tanto de la empresa como de cada uno de los proveedores asociados.
- ✓ Información del stock de los productos del catálogo.

- **Número de archivos lógicos internos:** enumeramos los 4 archivos internos que hemos localizado.

Complejidad baja: 3 archivos internos.

- Fichero de configuración servidor aplicación.

- Fichero de configuración servidor base de datos.
- Configuración PC.

Complejidad alta: 1 archivos internos.

- Base de datos.
- **Número de archivos externos:** nuestra aplicación no generará ningún tipo de fichero o archivo externo. El único fichero externo será el manual de usuario para el manejo de la web que lo categorizamos como complejidad alta.
- Manual de usuario.
- **Número de consultas externas:** como consultas externas hemos incluido una categorizada como complejidad media.
- Consultar manual de usuario.

Una vez que hemos identificado los distintos componentes y los hemos clasificado en función de su complejidad, ya podemos calcular los PFNA de nuestro sistema con la fórmula que hemos explicado anteriormente.

$$PFNA = (8 \times 3) + (3 \times 4) + (10 \times 5) + (3 \times 7) + (1 \times 15) + (1 \times 10) + (1 \times 4) = 24+12+50+21+15+10+4= 136$$

Ahora obtenemos los factores de ajuste valorando los factores de complejidad.

- Comunicación de datos:5
- Datos de procesamiento distribuidos:0
- Objetivos de rendimiento:2
- Configuración utilizada masivamente:2
- Tasa de transacción:3
- Entrada de datos on-line:5
- Eficiencia para el usuario:1
- Actualización on-line:1
- Procesamiento complejo:4
- Reutilización:2
- Facilidad de instalación y conversión:1
- Facilidad de operación:1
- Puestos múltiples:1
- Facilidad de cambio:3

Una vez obtenidos los factores de ajuste, realizamos la suma de todos ellos: en este caso nos da como resultado 31. Aplicamos la fórmula para obtener el FA:

$$FA = FA = (0.01 \times 31) + 0.65 = 0,96$$

Por último obtenemos los puntos de función ajustados:

$$PF = PFNA \times FA = 136 \times 0,96 = 130,56$$

Para finalizar realizamos un cálculo aproximado de las líneas de código Java que habrá que programar por cada punto de función estimado. Para ello seguiremos la relación de equivalencia de Casper Jones que establece la siguiente equivalencia:

1 PF equivale a 53 líneas de código Java

Por lo tanto en nuestro caso tendremos aproximadamente:

$$130,56 \times 53 = 6919,68 \text{ líneas de código.}$$

4.1.2. Estimación de costes siguiendo el modelo de COCOMO

La planificación de un proyecto software implica la estimación para poder determinar cuánto dinero, esfuerzo, recursos y tiempo supondrá construir un sistema o producto software. Para ello podemos basarnos en el modelo de estimación de costes COCOMO. Se basa en un modelo algorítmico que produce una estimación en función de un conjunto de variables como pueden ser:

- Líneas de código fuente.
- Capacidad del equipo de desarrolladores del producto software.
- Restricciones de hardware (equipos, memoria ...)
- Restricciones de tiempo.
- Fiabilidad de la aplicación.

4.1.2.1. Explicación del algoritmo COCOMO

El modelo COCOMO es una jerarquía de modelos de estimación de costes software que incluye los submodelos básico, intermedio y detallado.

$$E = a_i S^{b_i} m(X)$$

Las ecuaciones de estimación del esfuerzo de desarrollo tienen la forma:

donde S es el número de líneas de código fuente y $m(X)$ es un multiplicador que depende de 15 atributos. Estos atributos que se utilizan para el cálculo del esfuerzo se obtienen de la siguiente tabla:

FACTORES	VALOR DE LOS FACTORES					
	MUYBAJO	BAJO	MEDIO	ALTO	MUYALTO	EXTRA
Fiabilidad Requerida	0.75	0.88	1	1.15	1.4	
Tamaño de la base de datos		0.94	1	1.08	1.16	
Complejidad software	0.70	0.85	1	1.15	1.30	1.65
Restricciones de tiempo de ejecución			1	1.11	1.30	1.66
Restricciones de memoria			1	1.06	1.21	1.56
Volatilidad del hardware		0.87	1	1.15	1.30	
Restricciones de tiempo de respuesta		0.87	1	1.07	1.15	
Calidad de los analistas	1.46	1.19	1	0.86	0.71	
Experiencia con el tipo de aplicación	1.29	1.13	1	0.91	0.82	
Experiencia con el hardware	1.21	1.10	1	0.90		
Experiencia con el lenguaje de programación	1.14	1.07	1	0.95		
Calidad de los programadores	1.42	1.17	1	0.86	0.70	
Técnicas modernas de programación	1.24	1.10	1	0.91	0.82	
Empleo de herramientas	1.24	1.10	1	0.91	0.83	
Restricciones a la duración del proyecto	1.23	1.08	1	1.04	1.10	

Figura 11. Factores de complejidad

A continuación presentamos la siguiente tabla en la que se muestran los coeficientes de los nodos:

Modo	Básico		Intermedio	
	a_i	b_i	a_i	b_i
Orgánico	2.4	1.05	3.2	1.05
Semiencajado	3.0	1.12	3.0	1.12
Empotrado	3.6	1.2	2.8	1.2

Figura 12. Coeficientes de los nodos

Vamos a exponer brevemente los submodelos mencionados anteriormente.

1.- Modelo Básico: calcula el esfuerzo y el coste del desarrollo de un producto software pequeño y mediano, en función del tamaño del programa, expresado en las líneas estimadas de código (LDC). Dentro de este modelo podemos encontrar tres modelos que se diferenciarán en función de los distintos prototipos proyectos software: orgánico, semiacoplado y empotrado.

Los modelos COCOMO están establecidos para tres prototipos de proyectos software que son:

- **Modo orgánico:** en este grupo se engloban los proyectos software pequeños y sencillos en donde trabajan equipos pequeños sobre requisitos poco rígidos. Las ecuaciones que siguen este modelo son:
Ecuación para determinar el esfuerzo personal:

$$K_m = 2.4 S_k^{1.05}$$

Donde K_m se mide en personas-mes y S_k es el tamaño expresado en miles de líneas de código fuente.
Ecuación para determinar el tiempo de desarrollo:

$$t_d = 2.5 K_m^{0.38}$$

K_m se obtiene de la ecuación anterior y t_d es el tiempo de desarrollo en meses.

- **Modo semiacoplado:** proyectos software de complejidad media en los que intervienen equipos de distintos niveles de experiencia que trabajan sobre requisitos poco o medio rígidos.
Las ecuaciones de este modelo son:
Ecuación para determinar el esfuerzo personal:

$$K_m = 3.0 S_k^{1.12}$$

Ecuación para determinar el tiempo de desarrollo:

$$t_d = 2.5 K_m^{0.35}$$

- **Modo empotrado:** engloba a los proyectos software que deben ser desarrollados en un conjunto de hardware, software y restricciones operativas muy restringido.
Las ecuaciones de este modelo son:
Ecuación para determinar el esfuerzo personal:

$$K_m = 3.6 S_k^{1.20}$$

Ecuación para determinar el tiempo de desarrollo:

$$t_d = 2.5 K_m^{0.32}$$

2.- Modelo Intermedio: En este modelo se introducen 15 atributos de coste a tener en cuenta en el entorno de trabajo. Estos atributos se utilizan para ajustar el coste nominal del proyecto al entorno real, incrementando la precisión de la estimación.

Ecuaciones para determinar el coste: Para cada modo de desarrollo, los 15 atributos del coste intervienen como multiplicadores en el coste nominal, K_n , para producir el coste ajustado.

Las ecuaciones nominales de coste para el modelo intermedio son:

- **Modo orgánico:**

$$K_n = 3.2 S_k^{1.05}$$

- **Modo semiacoplado:**

$$K_n = 3.0 S_k^{1.12}$$

- **Modo empotrado:**

$$K_n = 2.8 S_k^{1.20}$$

Los atributos de coste que intervienen en el cálculo del esfuerzo son:

- Atributos del producto
 - RELY: garantía de funcionamiento requerida al software
 - DATA: tamaño de la base de datos
 - CPLX: complejidad del producto
- Atributos del ordenador
 - TIME: restricción de tiempo de ejecución
 - STOR: restricción del almacenamiento principal
 - VIRT: volatilidad de la máquina virtual
 - TURN: tiempo de respuesta del ordenador
- Atributos del personal
 - ACAP: capacidad del analista
 - AEXP: experiencia en la aplicación
 - PCAP: capacidad del programador
 - VEXP: experiencia en máquina virtual
 - LEXP: experiencia en lenguaje de programación
- Atributos del proyecto
 - MODP: prácticas de programación modernas
 - TOOL: utilización de herramientas software
 - SCED: plan de desarrollo requerido.

3.- Modelo Detallado: Este modelo puede procesar todas las características del proyecto para construir una estimación. Introduce dos características principales:

- Multiplicadores de esfuerzo sensitivos a la fase. Algunas fases se ven más afectadas que otras por los atributos. El modelo detallado proporciona un conjunto de multiplicadores de esfuerzo para cada atributo. Esto ayuda a determinar la asignación del personal para cada fase del proyecto.
- Jerarquía del producto a tres niveles. Se definen tres niveles de producto. Estos son módulo, subsistema y sistema. La cuantificación se realiza al nivel apropiado, esto es, al nivel al que es más susceptible la variación.

4.1.2.2. Aplicación del algoritmo COCOMO a nuestro proyecto

Nuestro proyecto encaja dentro de las especificaciones de COCOMO para el modelo intermedio orgánico ya que, como hemos explicado anteriormente, en este grupo se engloban los proyectos software pequeños y sencillos en donde trabajan equipos reducidos sobre requisitos poco rígidos.

Anteriormente hemos obtenido, mediante la técnica de puntos de función, una estimación aproximada del número de líneas de código que requerirá el desarrollo de nuestra aplicación, en concreto 6919,68 líneas = 6,92 KI.

Por tanto si aplicamos la fórmula correspondiente para calcular el número de personas que necesitamos obtendremos el siguiente resultado:

$$\mathbf{N^{\circ} personas = 3.2 \times (6,92)^{1,05} = 24,391 personas-mes \sim 25 personas-mes}$$

A continuación calculamos el esfuerzo que será necesario:

Esfuerzo = N° personas X Factores de coste = 24,391 x 1,4 (Fiabilidad Requerida) x 0,94 (Tamaño de la base de datos) x 0,85 (Complejidad de la base de datos) x 0,87 (Restricciones de tiempo de respuesta) x 1,13 (Experiencia con el tipo de aplicación) x 0,90 (Experiencia con el hardware) x 1,07 (Experiencia con el lenguaje de programación) x 0,86 (Calidad de los programadores) x 0,86 (Calidad de los analistas) x 0,83 (Empleo de herramientas) = 15,85 personas-mes \sim 16 personas-mes.

El siguiente cálculo que tenemos que realizar es el tiempo que tardaríamos en desarrollar el proyecto teniendo en cuenta el número de personas que hemos calculado que hacen falta:

Tiempo estimado de desarrollo = $2.5 \times 15,85^{0.38} = 7,144$ meses

Por último, una vez que tenemos los cálculos del tiempo de desarrollo de la aplicación y las personas que serán necesarias por mes, podemos calcular las personas que se necesitarían para poder implementar la aplicación por mes. Entonces obtendríamos:

Nº personas por mes = $15,85$ (personas *mes) / $7,144$ (meses) = $2,18 \sim 3$ personas

Hemos deducido que para completar el desarrollo en el tiempo calculado serían necesarias tres personas trabajando en la aplicación cada mes.

Teniendo en cuenta que este proyecto ha sido desarrollado por dos personas el tiempo será ligeramente superior al obtenido.

4.2. Planificación

El proceso de gestión de un proyecto de software comienza con un conjunto de actividades que globalmente se denominan planificación del proyecto. La fase de planificación aporta una aproximación lo más real posible del orden en el que se van a desarrollar las distintas fases del proyecto y el tiempo que abarcará cada una de ellas. El objetivo que se persigue cuando se realiza una planificación de un proyecto es el proporcionar un marco de trabajo que permita, al gestor del proyecto, hacer estimaciones razonables de recursos, coste y planificación temporal. Las estimaciones se hacen dentro de un tiempo limitado al comienzo de un proyecto software, y se deben ir actualizando conforme se van avanzando por las distintas fases, ya que es bastante frecuente que se tengan que ir realizando ajustes. Dicha planificación nos ayudará a realizar un seguimiento del desarrollo de todo el proyecto. Esto nos permitirá que en el caso de que una tarea no se cumpla en los plazos determinados, puedan realizarse ajustes en las tareas posteriores.

Vamos a describir cada una de las fases identificadas en la planificación de nuestro proyecto PAJARRACO'S PAJARERÍA.

- 1.- Definición.
- 2.- Planificación.
- 3.- Iniciación.
- 4.- Control.
- 5.- Finalización o Cierre.

1.- Definición y estudio de la viabilidad:

El objetivo de esta fase consiste en especificar, con claridad, aquello que se espera conseguir con el proyecto. Toda definición de un proyecto debe identificar los propósitos y los objetivos del trabajo. En definitiva, el estudio del problema que queremos abordar y las herramientas que se usarán para poder llevarlo a cabo. Las tareas que realizaremos durante esta etapa son:

- Escribir una descripción de los requisitos a alto nivel, que tendrán nuestro sistema en función de los objetivos que se pretenden alcanzar con el desarrollo. Ello significa comprender el problema o propósito que se persigue. Tiempo estimado: 15 días.
- Estimar el tamaño del sistema, la planificación y los costos que conllevan el desarrollo de la aplicación. Tiempo estimado: 5 días.

En nuestro caso esta fase ha sido la más breve. Se definió brevemente los objetivos que queríamos perseguir con este proyecto y la plataforma de desarrollo que queríamos emplear para llevarlo a cabo.

Nuestra aplicación será una primera versión de una web de una pequeña empresa con lo que inicialmente no requerirá de grandes infraestructuras ni un gran desarrollo. El objetivo que hemos buscado con este proyecto es realizar una versión inicial de la web de una empresa que comienza a darse a conocer a través de la web.

Como ya hemos comentado anteriormente, para este proyecto decidimos desarrollar nuestra aplicación con un lenguaje java por varios motivos. Se puede utilizar una plataforma de desarrollo gratuita y además este lenguaje de programación dispone de una amplia API que nos facilitará en gran medida el desarrollo de aplicaciones web y la gestión con nuestro sistema de

persistencia de datos.

En esta primera fase emplearemos un total de 20 días para realizar el estudio.

2.- Análisis:

Esta fase tiene lugar una vez que el estudio de viabilidad de un proyecto ha sido favorable, y se ha decidido llevar a cabo el proyecto. En esta etapa de la planificación hemos incluido las siguientes tareas:

- Realizaremos de una estimación más detallada y revisada de costes, planificación, recursos necesarios, etc. Es en este punto en el que en función de los resultados obtenidos en esta etapa se deberá tomar la decisión de continuar o no con el proyecto. Tiempo estimado: 5 días.
- En esta fase además se revisarán los requisitos para ofrecer un mayor detalle de cada uno de ellos de cara a la elaboración del funcional. 15 días.
- A continuación elaboraremos el documento de análisis funcional. Este documento es una transformación de los requerimientos del usuario de la fase anterior en unas especificaciones funcionales. Definiremos los usuarios, dimensiones y restricciones que tendrá nuestro sistema. Incluirá además un análisis del modelo de datos que se quiere utilizar como método de persistencia de la información que se maneje en la web. Tiempo estimado:15 días.

En nuestro caso el tiempo total empleado en realizar todas estas tareas será de 35 días.

3.- Diseño:

Esta etapa dentro de la planificación de un proyecto en la que los requisitos empiezan a tomar forma. Las tareas que hemos realizado en esta etapa de la planificación del proyecto son:

- Evaluaremos diferentes posibilidades de diseño para escoger la que mejor se adapta a nuestro proyecto. Tiempo estimado:5 días.
- Desarrollo de un diseño detallado del sistema (interfaz del sistema) y de la base de datos. Tiempo estimado:10 días.
- Posterior elaboración del documento de diseño definitivo del sistema. Tiempo estimado:12 días.
- Identificación de las necesidades de conocimiento y documentación de los usuarios que van a utilizar la herramienta; definición de las guías finales. Tiempo estimado:3 días.

El tiempo total que hemos empleado en esta fase ha sido de 30 días.

4.- Codificación:

Esta etapa es la más larga de todo el proceso de planificación. En ella se desarrollará todo lo que se ha ido definiendo y tratando en las anteriores etapas de análisis. Una vez que se llega a esta fase no debe quedar ninguna duda en la definición de ninguna funcionalidad.

- Implementación tanto del código de la aplicación web como de la base de datos donde se almacenará la información. Tiempo estimado: 120 días.

5.- Pruebas:

Esta etapa se encamina a probar la consistencia, fiabilidad y robustez del producto que se ha desarrollado. Los hitos que hemos realizado en esta etapa son:

- Realización del test del sistema. Normalmente las pruebas se dividen en pruebas unitarias, integradas y de usuario. Como en este caso tanto los desarrolladores como los usuarios finales de la aplicación son los mismos todas las pruebas se realizarán como un solo plan de pruebas. Este test incluirá la elaboración de una amplia batería de pruebas tanto de las funcionalidades de la aplicación como del rendimiento y capacidad del mismo. Tiempo estimado:21 días.

- Finalización del sistema completamente probado. Modificación de mensajes a usuario y retoques de interfaz. Tiempo estimado: 7 días.

En total en esta fase hemos empleado un total de 28 días.

6.- Integración del sistema en producción:

Una vez pasadas todas las pruebas del sistema, el siguiente paso es su explotación en un entorno de producción. Este es el objetivo que se pretende en esta etapa:

- Instalación del hardware y software para el entorno de explotación. En nuestro caso, en principio, necesitaremos un servidor de aplicaciones que soporte todo el despliegue del código de la aplicación y una base de datos que sea capaz de hacer persistir el contenido de la aplicación web.
- Llevar a cabo la configuración de los servidores (servidor de la aplicación y de la base de datos), el despliegue del código en el servidor de aplicación y la creación de la base de datos con todos los datos que inicialmente se necesiten para el correcto funcionamiento de la aplicación, como puede ser el catálogo de productos o la creación de usuarios con perfiles de administración del sistema
- Estabilización de la plataforma en el entorno de producción. Con esto se persigue el mantener la cuarentena del desarrollo para poder reaccionar de manera rápida y eficaz ante posibles incidencias que pudieran detectarse en dicho entorno.

Esta etapa no la hemos contemplado ya que este proyecto no se desplegará en ningún entorno de producción.

7.- Mantenimiento:

Al igual que la etapa anterior, esta fase la hemos incluido a título informativo como una fase más dentro de la planificación de un proyecto. El objetivo que se persigue en esta fase es ir adaptando la aplicación a las distintas necesidades que puedan surgir, tanto a nivel de funcionalidad, como a nivel de aumentar el rendimiento de la plataforma de explotación mediante la inclusión de nuevos servidores. Algunas de las tareas que podemos incluir en esta etapa pueden ser:

- Implementar los cambios y evoluciones funcionales que vaya requiriendo el sistema.
- Asegurarse de que el sistema soporta adecuadamente el tráfico que recibe la aplicación.

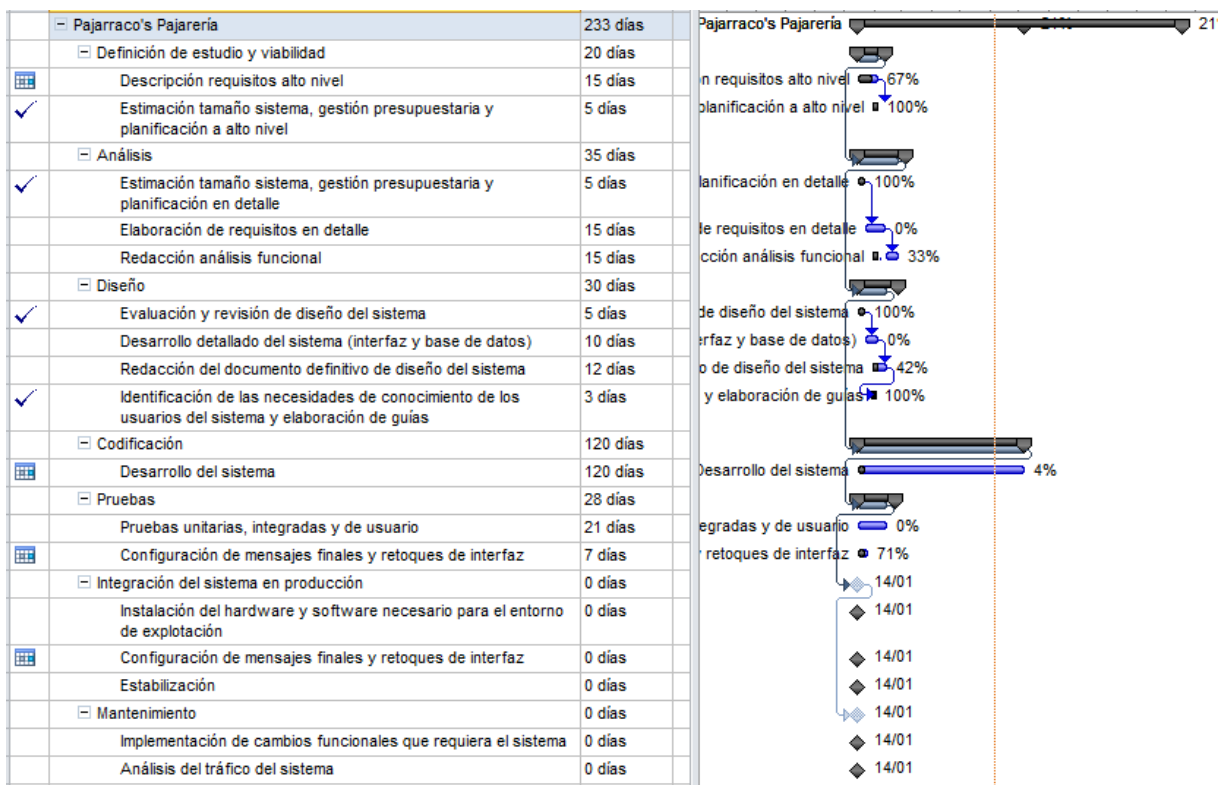


Figura 12. Diagrama de Gantt

4.3. Presupuesto

Para calcular el presupuesto de un proyecto hay que tener en cuenta tanto los recursos humanos como los materiales que han sido necesarios para el desarrollo del mismo.

En cuanto a los recursos humanos empleados para la elaboración de la web PAJARRACO'S PAJARERÍA hemos dividido el trabajo por perfiles. Cada uno de estos perfiles se encargaran de la elaboración de diferentes etapas dentro de la planificación del proyecto. Suponemos que cada día empleado supone 8 horas de trabajo.

Por un lado será necesario un perfil de analista y otro de programador. A continuación mostramos el desglose de las tareas y el coste humano total que supondrá el desarrollo de todas.

Tarea	Perfil	Coste perfil(Eur/h)	Horas tarea(h)	Coste Tarea(Eur)
Definición y estudio de la viabilidad	Analista	10 Eur/h	20 días X 8= 160h	160 X 10 = 1600 Eur
Análisis	Analista	10Eur/h	20 días X 8 =160h	160 X 10= 1600 Eur
Diseño	Analista	10Eur/h	30 días X 8 = 240h	240 X 10 = 2400Eur
Codificación	Programador	5Eur/h	120 días X 8 =960h	960 X 5 = 4800 Eur
Pruebas	Programador	5Eur/h	28 días X 8 = 224h	224 X 5 = 1120 Eur
			TOTAL	11520Eur

Figura 13. Coste recursos humanos

En cuanto al presupuesto en recursos materiales, hardware y software lo hemos desglosado en la siguiente tabla:

Recurso	Uso (%)	Coste total(Eur)	Coste para el proyecto(Eur)
Ordenador Personal	25	800 Eur	200 X 2(unidades) = 400 Eur
Impresora	10	80 Eur	8 Eur
Internet ADSL 12 meses.	25	240 Eur	60 X 2(conexiones) = 120 Eur
Microsoft Windows Vista (incluido en coste del ordenador personal y servidor).	100	0 (incluido en el PC) Eur	0 Eur
Maquina virtual Java	100	0 Eur	0 Eur
IBM Webshpere Studio con servidor de aplicaciones integrado	100	0 Eur	0 Eur
Microsoft Office 2010	20	700 Eur	140 X 2 (unidades) = 280 Eur
StarUML 5.0	100	0 Eur	0 Eur
MySql	100	0 Eur	0 Eur
Adobe Reader	100	0 Eur	0 Eur
Material de oficina	100	50 Eur	50 Eur
		TOTAL	850 Eur

Figura 14. Coste recursos materiales

Por tanto el coste total del desarrollo de la aplicación web es aproximadamente $11520 + 850 = 12370$ Eur.

5. Consideraciones adicionales

En este apartado vamos a exponer cuestiones reseñables que hemos tenido en cuenta a la hora del desarrollo de nuestra aplicación y que merece la pena mencionar.

5.1. Cuestiones de análisis y diseño

Jerarquía de usuarios: nosotras hemos enfocado el diseño de la web en base a cubrir las necesidades de cualquier usuario que acceda a la web. Lógicamente no todos los usuarios que navegan por la web realizan un pedido, a pesar de ser ese el principal objetivo. Es por ello por lo que nos surgió la necesidad de establecer un modelo de jerarquía de usuarios, que nos permitiera obtener diferentes niveles de seguridad requeridos para cada funcionalidad. Por tanto decidimos crear tres roles o perfiles de usuario que son: usuario, cliente y administrador.

Pago del pedido realizado: la implementación de un proceso real de pago complicaba en gran medida el desarrollo de la aplicación ya que habría que establecer una comunicación entre nuestra aplicación y una pasarela de pagos a través de la cual poder realizar los pagos de los pedidos. El objetivo del proyecto es la realización de una primera versión de la web de una pajarería con las funcionalidades más básicas. Es por ello por lo que hemos descartado la implementación de la pasarela. En versiones posteriores se tendrían que ir realizando ajustes y modificaciones para ir perfeccionando la web.

Modelo de datos: nuestro modelo de datos utiliza bastantes relaciones entre las diferentes entidades. Esto es debido a que la mayoría están directamente relacionadas y la eliminación de algún registro de una de ellas hace que sea lógica la eliminación del registro relacionado de otra, ya que dejaría de tener sentido por sí solo.

Diseño de la interfaz: hemos tenido especial cuidado en el desarrollo de la interfaz web. Para el éxito de cualquier página web es fundamental que la interfaz de la misma sea atractiva e intuitiva para el usuario.

En nuestro caso hemos optamos por la modularidad de las funcionalidades y los recursos gráficos para hacer la interfaz más amigable. Además se diseñó un controlador de dialogo que atiende a las peticiones para cambiar de interfaz y, opcionalmente, admite guardar un histórico de las interfaces de las que se procede, permitiendo regresar a ellas en cualquier momento, mediante los botones de navegación oportunos.

Patrón o modelo de nuestro sistema: para la implementación de nuestro modelo hemos utilizado el patrón de diseño MVC(modelo-vista-controlador). Nos hemos decantado por este modelo debido a una serie de razones entre las que destacamos:

- El mantenimiento de las aplicaciones que siguen este patrón es mucho más sencillo.
- Se separa la parte de la lógica de negocio y la presentación de datos.
- Se pueden reutilizar componentes.

Se basa principalmente en la descomposición de la lógica de negocio (que es lo que llamamos modelo) y la presentación de la información (la vista). El controlador es el encargado de realizar la abstracción de los datos, haciendo que la vista y las acciones sean independientes.

5.2.- Cuestiones de implementación reseñables

Métodos de una clase: todas las clases que hemos implementado, que necesitan de un listado de elementos en la aplicación y una visualización individual por pantalla, incorporaron en su implementación métodos para poder cumplir este objetivo de visualización de elementos almacenados en la base de datos, sin necesidad de instanciar objetos de dicha clase.

Clases de acceso a base de datos: a la hora de implementar nuestra aplicación, hemos creado varias clases para establecer el acceso y conexión de las distintas clases con la base de datos. Hemos creado una clase de acceso a la base de datos en función de los objetos con los que estemos trabajando. Es una manera de ordenar las operaciones por objeto. Por ejemplo, todas las operaciones de acceso a base de datos que estén relacionadas con el objeto *centro* se realizaran en una clase denominada *CentroBO*.

Utilización de struts: para la implementación de la web nos hemos basado en la utilización de struts. Struts es un framework que implementa el modelo patrón vista controlador en Java que es el modelo que hemos seguido para el desarrollo de la web. El motivo por el que nos decantamos a utilizarlo es que simplifica en gran medida la implementación de una arquitectura según el patrón MVC. Nos ayuda a gestionar el workflow de la aplicación, del modelo de objetos y de la generación de la interfaz. Para ello utilizaremos dentro de las JSP's etiquetas propias que proporciona el framework de struts.

A continuación vamos a explicar brevemente como funciona struts:

- Un usuario realiza una petición.
- Cada petición se asocia con una determinada regla de negocio.
- La acción necesita de un conjunto de datos que se encapsulan en un objeto llamado *Form*. Si no se pasan las validaciones de este formulario se mostrará de nuevo, para que el usuario vuelva a introducir los datos.
- Una vez que se han pasado las validaciones del formulario, se ejecutará la acción que corresponda. El resultado de ejecutar esta acción puede ser favorable o no. En función de dicho resultado se mostrará diferentes páginas de presentación. Los mensajes mostrados se almacenarán en un fichero de recursos que ayuda a facilitar el mantenimiento y la internacionalización de las aplicaciones.
- Cada elemento se construye por separado y se relacionan en el fichero struts-config.xml que es el que define la lógica.

5.3. Cuestiones reseñables sobre la base de datos

Relaciones 1:N y 1:1: en nuestra base de datos existen relaciones de diferentes tipos. Las relaciones tipo 1:1 y 1:N se transformaron en tablas al igual que las relaciones de tipo N:M.

Identificadores principales auto numéricos: en nuestra base de datos existen identificadores principales que son auto numéricos.

Único usuario sobre la base de datos: nuestra base de datos ha sido diseñada con un único usuario. Los credenciales de este usuario se utilizarán desde la parte de la lógica de la aplicación para establecer la conexión con la base de datos.

6. Conclusiones y posibles ampliaciones

Este proyecto cumple los objetivos que perseguíamos con esta primera fase. Esta página web es la manera de extender y acercar un negocio a un mayor número de posibles compradores. Además supone el realizar un seguimiento y mantenimiento del negocio de una manera más sencilla y fiable. Todo esto se logra gracias a un interfaz muy sencillo y amigable que no requiere de grandes esfuerzos para aprender a manejar la aplicación web. Por eso hemos dedicado un gran esfuerzo en cuidar su desarrollo.

Para poder realizar el proyecto tal y como queríamos, resultó indispensable el majeo de algunos conocimientos y lenguajes, que no habíamos visto a lo largo de la carrera. El uso del lenguaje java y del framework de struts son algunos de los conocimientos que hemos necesitado para llevar a cabo nuestro desarrollo. No obstante, nosotras contábamos con la ventaja de tener ya gran parte de los conocimientos que necesitábamos adquiridos. Esto nos ha facilitado enormemente la implementación del código de la web. No obstante pese a ello, el hecho de iniciar una aplicación desde cero no ha sido tarea fácil ya que además hay que establecer la configuración de todas las herramientas que hemos utilizado para el desarrollo. También hemos recurrido a la utilización de conocimientos adquiridos en las distintas asignaturas de la carrera. Algunos de estos conocimientos son el lenguaje de modelado UML, patrones de diseño, creación y manejo de bases de datos etc.

Otra cosa que nos ha resultado difícil has sido el realizar una estimación del tiempo y los costes que nos llevarían realizar el proyecto por varios motivos:

- La inexperiencia a la hora de hacer estimaciones de proyectos.
- El no saber a priori el trabajo que supondría la realización del proyecto, y la dedicación al mismo, lo que nos supuso la realización de una estimación “a ciegas”.

Por último tenemos que destacar que para nosotras el mayor obstáculo ha sido el ser fiel al cumplimiento de los plazos fijados en las distintas fases del proyecto. El no tener exclusiva dedicación en el desarrollo del proyecto ha hecho que las fechas iniciales se fueran dilatando en el tiempo más de lo que nos hubiera gustado.

6.1. Ampliaciones

Una de las características de las páginas web son la posibilidad de que se reinventen, es decir de añadir, eliminar y modificar funcionalidad a medida que se requiera. Entre las posibles ampliaciones que se podrían realizar en el futuro podemos pensar en las siguientes opciones:

- En el futuro habría que mejorar el foro como herramienta para conocer la opinión de los usuarios con respecto a la web y al negocio. Esto proporcionaría una visión diferente del negocio y nos permitirá mejorar la situación de nuestro negocio en el mercado.
- Igualmente habría que añadir funcionalidad a la página web a medida que fuera creciendo el negocio.
- Una de las maneras de hacer mejor una web es mejorando al máximo posible su rendimiento. Para ello tendríamos que estudiar la volumetría de tráfico que tiene la web y así determinar el número de recursos que son necesarios para dar un soporte óptimo.
- Una manera de mejorar la contabilidad de la empresa es mediante su automatización. Esto podría realizarse mediante el control de los pedidos que se vayan pagando a través de la web y los pedidos que la empresa realice a los diferentes proveedores. Así podemos obtener el resultado de los ingresos y los gastos y saber de una manera más rápida el balance de cuentas de la empresa.
- La implementación de conectividad entre la aplicación y una pasarela de pagos real es una de las funcionalidades que se nos ha quedado pendientes en esta primera fase. Así el pago de los pedidos de los clientes se re direccionará al servidor seguro del banco, que será el encargado de aceptar o rechazar el pago y devolver el resultado de la transacción a la web.
- Una funcionalidad que se nos ocurre que se podría implementar en una segunda fase es la posibilidad de ofrecer al cliente ciertos productos tras realizar una compra y en función de lo que acabase de comprar. Es una manera de aprovechar el momento de compra impulsiva de los clientes para aumentar las ventas además de proporcionar un asesoramiento personalizado a los clientes.
- A medida que aumentara el negocio se podría pensar en crear más perfiles de administrador y distribuir las tareas entre todos los perfiles. Por ejemplo uno de los perfiles podría ser el encargado de estudiar las ventas que se van

registrando mientras que otro podría dedicarse exclusivamente a controlar el stock de los productos del catálogo. Cada perfil vería exclusivamente las opciones correspondientes a sus funciones. Al mismo tiempo existiría un perfil “maestro” que sería el encargado de distribuir las funciones y gestionar los diferentes roles administrativos de la web.

- Adaptar el diseño de la web a los nuevos recursos gráficos que vayan surgiendo en el futuro. Esto nos proporcionará una web más vistosa atractiva e interactiva para los usuarios.
- Por último sería muy útil e interesante el crear un microsite de la web para poder acceder a la aplicación desde dispositivos móviles como tablets o smartphones, ya que actualmente cada vez se utiliza más dichos dispositivos para conectarse a internet.

Índice de Contenido

Análisis del Sistema de Información	43
1. Definición del Sistema	43
1.1. Determinación del Alcance del sistema	43
1.2. Ámbito de la aplicación y perspectiva del proyecto	43
1.3. Objetivos del sistema	43
1.4. Identificación del entorno tecnológico	44
1.5. Especificación de estándares y normas	44
1.6. Identificación de usuarios participantes y finales.....	44
2. Establecimiento de Requisitos	46
2.1. Obtención de Requisitos.....	46
2.1.1 Requisitos No Funcionales.....	46
2.1.2 Requisitos Funcionales	47
2.2. Especificación de Casos de Uso.....	48
3. Identificación de Subsistemas de Análisis	50
4. Análisis de los Casos de Uso.....	61
4.1. Identificación de Clases Asociadas a un Caso de Uso	61
4.2. Descripción de la Iteración de Objetos	61
5. Análisis de Clases	70
5.1. Identificación de Responsabilidades y Atributos	70
5.2. Identificación de Asociaciones y Agregaciones	70
5.3. Identificación de Generalizaciones	70
6. Definición de las Interfaces de Usuario	72
6.1. Principios generales	72
6.2. Especificación de formatos Individuales de la Interfaz	72
6.3. Formatos de impresión.....	72
7. Especificación del Plan de Pruebas	73
8. Aprobación del Análisis del sistema de Información.....	74

Análisis del Sistema

Índice de Contenido

Análisis del Sistema de Información	43
1. Definición del Sistema	43
1.1. Determinación del Alcance del sistema	43
1.2. Ámbito de la aplicación y perspectiva del proyecto	43
1.3. Objetivos del sistema	43
1.4. Identificación del entorno tecnológico	44
1.5. Especificación de estándares y normas	44
1.6. Identificación de usuarios participantes y finales.....	44
2. Establecimiento de Requisitos	46
2.1. Obtención de Requisitos.....	46
2.1.1 Requisitos No Funcionales.....	46
2.1.2 Requisitos Funcionales	47
2.2. Especificación de Casos de Uso.....	48
3. Identificación de Subsistemas de Análisis	50
4. Análisis de los Casos de Uso.....	61
4.1. Identificación de Clases Asociadas a un Caso de Uso	61
4.2. Descripción de la Iteración de Objetos	61
5. Análisis de Clases	70
5.1. Identificación de Responsabilidades y Atributos	70
5.2. Identificación de Asociaciones y Agregaciones	70
5.3. Identificación de Generalizaciones	70
6. Definición de las Interfaces de Usuario	72
6.1. Principios generales	72
6.2. Especificación de formatos Individuales de la Interfaz	72
6.3. Formatos de impresión.....	72
7. Especificación del Plan de Pruebas	73
8. Aprobación del Análisis del sistema de Información.....	74

Análisis del Sistema de Información

El objetivo del Análisis del Sistema de Información es la obtención de una especificación detallada del sistema que satisfaga las necesidades de información de los usuarios y sirva de base para el posterior diseño del sistema.

1. Definición del Sistema

1.1. Determinación del Alcance del sistema

El ritmo de vida actual hace que cada vez tengamos menos tiempo disponible, y la creciente expansión de Internet ha hecho que proliferen las “empresas online” que permiten hacer compras a través de sus webs de forma rápida cómoda y desde cualquier lugar.

El uso de aplicaciones web ayuda a las empresas a gestionar de manera más eficiente sus recursos y abarcar una mayor cuota de mercado, de modo que su cartera de clientes se puede extender geográficamente. Además el aumento de grandes áreas comerciales hace necesario que la Pymes (pequeñas y medianas empresas), busquen nuevas alternativas para acercar sus productos al público.

El desarrollo de aplicaciones “a medida” mejora el servicio y prestaciones amoldándose al perfil y a las necesidades de la empresa, tanto para su gestión como para la comercialización de los productos. Además es una buena manera de darse a conocer con un mínimo coste en infraestructuras.

Las interfaces gráficas ayudan a que estos programas sean más atractivos y amigables para los usuarios y minimiza la necesidad de tener grandes conocimientos informáticos para su uso.

1.2. Ámbito de la aplicación y perspectiva del proyecto

El objetivo del proyecto es realizar una aplicación web con la cual poder realizar la gestión interna de una pajarería y que además ofrezca la posibilidad de comprar los productos que se ofrecen, incluyendo información acerca de los mismos.

También será una aplicación interactiva donde los usuarios intercambien información a través de un foro y puedan, entre otras cosas, conocer las necesidades, cuidados y habilidades que tienen los animales que pueden adquirir.

Nuestra aplicación va a incluir una funcionalidad que permitirá al usuario conocer la posible descendencia de los *diamantes de gould* (ave paseriforme de origen australiano), mediante un estudio de probabilidades genéticas, antes de adquirirlos.

El usuario podrá conocer los productos a través de un catálogo y también recibirá ofertas personalizadas. Además la aplicación generará unos informes para poder conocer el desarrollo y alcance de la empresa y prever su evolución en el mercado.

Pretendemos aprovechar las nuevas tecnologías para poder expandir la empresa en el mercado con un mínimo coste, aumentando su competitividad y ofreciendo mayor servicio y de manera más cómoda.

1.3. Objetivos del sistema

El objetivo de nuestro proyecto es realizar el análisis, desarrollo e implantación de toda la plataforma capaz de poder ofrecer distintas funcionalidades basándose en el perfil del usuario que se registre en la web.

Las etapas necesarias para llegar a estos objetivos son:

- Análisis del sistema de información
- Catalogación de los requisitos
- Diseño
- Desarrollo
- Pruebas
- Implantación

- Mantenimiento

1.4. Identificación del entorno tecnológico

La infraestructura tecnológica se hará de acuerdo a la especificación **J2EE**. Esta es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en lenguaje de programación java, con arquitectura distribuida de n niveles, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. En nuestro caso, emplearemos el servidor de aplicaciones **Websphere 6.1**.

La infraestructura tecnológica seguirá el patrón Modelo Vista Controlador y para la implementación de este modelo usaremos **Struts Apache Framework**. Struts está diseñado para ayudar al desarrollo de aplicaciones web. Divide el modelo de negocio o acceso a base de datos, la vista, es decir, la capa de presentación, y el controlador, que representa el código de navegación. Struts funciona para todos los servidores Java EE y ofrece una arquitectura sólida y estable que se adapta a aplicaciones Web de gran tamaño y permite descomponer aplicaciones complejas en componentes más simples. Además, facilita el desarrollo permitiendo que este sea rápido y poco costoso.

En cuanto al sistema gestor de base de datos usaremos **MySQL**.

Como herramienta de desarrollo usaremos **Websphere Application Server Toolkit**.

En cuanto al Servidor Http Web utilizaremos **Apache**.

El sistema operativo será **Windows Server**

1.5. Especificación de estándares y normas

La metodología adoptada en el análisis, diseño e implementación de nuestra aplicación será la metodología Métrica Versión 3.0. Esta metodología fue definida en España por el Ministerio de Administraciones Públicas, país que se ha especializado en estudio y desarrollo de sistemas de información, siendo probada y adoptada por diferentes entidades públicas y privadas en diferentes países del mundo.

En cuanto a la metodología y estándares utilizados en el desarrollo de la aplicación:

- Uso de la librería tiles para la implementación de la interfaz gráfica. Estas librerías permiten definir plantillas sobre las que construir la parte visual de la aplicación.
- Uso de los taglib para el desarrollo de jsp.
- Uso de ficheros .properties para la internalización de los textos de la aplicación.
- Control de la navegación a través de Dispath Action.

1.6. Identificación de usuarios participantes y finales

En esta aplicación tendrá tres perfiles de usuario:

Perfil Usuario – Usuario no logado.

- Podrá navegar por la web pero con acceso restringido (consultas de productos).
- Podrá registrarse como cliente.

Perfil Cliente – Usuario logado.

- Podrá navegar por toda la web.
- Podrá realizar pedidos.
- Tendrá acceso al foro.
- Podrá modificar sus datos de usuario o darse de baja.
- Tendrá acceso al uso del programa “Mendel”.

Perfil Administrador – Usuario logado encargado de gestionar la tienda.

- Podrá consultar en la web el catálogo de productos.
- Podrá navegar por la web.
- Podrá gestionar altas, bajas y modificación de clientes.
- Podrá gestionar altas, bajas y modificación de proveedores.
- Podrá realizar pedidos de productos y gestionar los envíos a clientes.
- Gestión del stock de las diferentes tiendas.
- Podrá gestionar el foro.
- Altas, bajas y modificación de productos.
- Generar catálogos a usuarios.
- Visualización de gráficos.

2. Establecimiento de Requisitos

2.1. Obtención de Requisitos

En esta tarea comienza la obtención detallada de información mediante sesiones de trabajo con los usuarios.

2.1.1 Requisitos No Funcionales

Los tipos de requisitos que vamos a especificar son:

Requisitos Interfaz de usuario

Nuestro sistema tiene que ser una plataforma disponible para un gran número de usuarios ubicados en distintas partes, por tanto la interfaz de usuario será de tipo Web.

El sistema debe ser sencillo, intuitivo y contener toda la información relevante para el usuario. Los módulos debe estar ordenados informado en todo momento del punto en el que se encuentra el estado del pedido. Dentro de este tipo de requisitos se definen aquellos tales como la manera en la que se presentará la información en las distintas pantallas, los colores que se utilizarán y el estilo de letra. Si trasladamos esto a la aplicación de la pajarería algunos de estos requisitos de interfaz de usuario son:

- El color verde elegido para el fondo de las pantallas.
- La estructuración general de las pantallas en: menú (ocupa el lateral izquierdo), submenú situado en la parte superior debajo del encabezado, el encabezado (situado en el margen superior) y la información principal que ocupa el resto de las pantallas.
- El color amarillo y el estilo de la fuente.

Requisitos de Interfaz de Hardware y Rendimiento

Se definen los elementos hardware mínimos necesarios para que la aplicación pueda funcionar de manera adecuada y mejorar al máximo su capacidad y rendimiento. Para establecer estos requisitos habría que disponer de información relativa al volumen de carga que se espera manejar (por si se necesitan varios servidores en cluster), tiempos de respuesta, puertos de conexión, velocidad de transmisión de datos etc...

Para nuestro proyecto únicamente necesitaremos un único servidor al tratarse de una aplicación que en principio no tendrá un volumen muy alto de carga de trabajo. Por tanto, habrá que tener en cuenta la capacidad que se espera del servidor donde estará alojada nuestra aplicación y la base de datos con la que interactuará. Para ello se realiza un análisis de la capacidad máxima que puede soportar el servidor para dar servicio al usuario de manera adecuada (es decir se analizará el número de usuarios que pueden estar conectados simultáneamente), además del número de transacciones por segundo que soportará la base de datos. También se deberán tener en cuenta la frecuencia de uso de la información almacenada en la base de datos, el tiempo de conexión, respuesta y el volumen de información que se espera almacenar. Todo esto influirá también a la hora de realizar el esquema de nuestra base de datos.

Requisitos de Interfaz de Software

Se establecen las condiciones de software mínimas para que la aplicación pueda funcionar correctamente. Algunos de estos requisitos pueden ser la versión del sistema operativo, la versión del navegador, formato de archivos...

Como ya hemos mencionado, uno de los requisitos software fundamental a tener en cuenta para el desarrollo nuestra aplicación es la versión del navegador. Las aplicaciones web se comportan, en ocasiones, de manera diferente si se cambia la versión o el tipo del explorador web para el que se desarrolló inicialmente. Nuestra aplicación web se ha desarrollado sobre Internet Explorer 6.0 con lo cual sólo se garantiza el total funcionamiento de la misma en aquellos clientes que usen este tipo y versión de explorador.

Requisitos Usabilidad

La aplicación debe ser intuitiva y fácil de usar además debe proveer de de todas las herramientas necesarias para llevar a cabo las tareas de gestión y generación de informes de una tienda on-line.

Requisitos Internalización

La aplicación debe poder ser traducida a varios idiomas.

Requisitos Disponibilidad

La aplicación requiere una alta disponibilidad, alto rendimiento y fácilmente escalable en el sentido que permita añadir nuevas funcionalidades con un bajo impacto sobre nuestro el conjunto del sistema.

Requisitos Seguridad

Nuestro sistema deberá proveer de un sistema de protección de datos además de un sistema de control de acceso a los distintos perfiles de usuario.

En nuestra aplicación se definen unos niveles de acceso a la información del sistema para los distintos usuarios (control de usuarios por perfiles), validación de datos de entrada por pantalla, gestión de errores y gestión de sesión. Todos ellos formarán el conjunto de nuestros requisitos de seguridad. A continuación pasamos a exponer brevemente cada uno de ellos.

La aplicación web identificará a cada usuario que se registre de manera única. Una vez logado en el sistema se le mostrará el menú con las posibles acciones que podrá realizar en la web dependiendo del perfil que tenga. Además la aplicación asegura que los accesos del usuario se realizan mediante un formulario de login el cual no puede ser “vulnerado”, accediendo por URL directamente desde la barra del navegador. Esto es así debido a que al hacerse el login se establece una sesión de usuario la cual no se obtendría de otra manera.

Otra peculiaridad de nuestro sistema será que en el momento en el cual se realice un pedido el sistema enviará al usuario automáticamente a la pantalla inicial de “pedidos”, impidiéndole volver atrás una vez realizado dicho pedido.

Los formularios que aparecen en numerosas pantallas de la aplicación web, cuenta con diferentes validaciones en los datos que puede introducir el usuario (por ejemplo en cuanto al formato, longitud...), que aseguran que la información que se le está mandando al sistema sea adecuada en cada caso.

Por último hay que destacar el control de errores que tiene la aplicación y que impide que las llamadas a métodos y funciones que devuelve valores o no, están controladas de manera que si en algún momento fallara la conexión con la BBDD o no se devolvieran datos, no se producirían pantallazos de errores no controlados.

Requisitos de Fiabilidad

Los requisitos de fiabilidad deben establecer los factores que se requieren para la fiabilidad del software en tiempo de explotación. La fiabilidad mide la probabilidad que tiene el sistema de responder satisfactoriamente a las demandas del usuario.

Nuestro sistema debe tener un alto grado de fiabilidad ya que un mal funcionamiento de este podría suponer, por ejemplo, desde la realización de pedidos involuntarios o la vulnerabilidad de los datos del cliente. Todo ello queda cubierto mediante el cumplimiento de los requisitos de seguridad, software y hardware definidos anteriormente.

Requisitos de Mantenimiento

El sistema es susceptible de ser ampliado o modificado. Por tanto, debe ser de fácil mantenimiento utilizando para ello las metodologías de desarrollo necesarias además de un código comprensible y explicado con los comentarios que se crean oportunos. Los tipos de mantenimiento del software pueden ser perfectivo o evolutivo (mejoras en el software para evolucionar el sistema o mejorar sus tiempos de respuesta o rendimiento), adaptativo (para acondicionar el sistema a los nuevos avances tecnológicos), correctivo (para solventar aquellas incidencias detectadas en la aplicación) y por ultimo preventivo (todo desarrollo orientado a evitar posibles fallos en el sistema).

2.1.2 Requisitos Funcionales

En función del tipo de usuario que va a acceder al sistema, y de las distintas actividades que pueden realizar se establece una serie de requisitos funcionales que dictan cuál va a ser la funcionalidad básica del sistema.

En la aplicación se distinguen tres actores asociados a cada uno de los roles que pueden tomar los usuarios que navegan a través de la web de la aplicación. Entre ellos se establece una jerarquía de actores de manera que cualquier usuario con perfil Cliente o Administrador podrá realizar las acciones de un Usuario, y cualquier usuario Administrador podrá realizar las funciones de un Cliente (con flujos adaptados a la finalidad de cada perfil, es decir, por ejemplo ambos usuarios podrán realizar pedidos, pero un Cliente realizará un pedido para él, mientras que un Administrador realizará un pedido para la tienda, con lo cual el flujo variará en ambos procesos).

- **Actor Usuario:** cualquier persona anónima que accede a la web de la aplicación tiene acceso a funcionalidades sin necesidad de darse de alta en el sistema. Podrá realizar acciones como consultar datos acerca de la empresa, ver las diferentes tiendas que tiene en España, realizar cualquier consulta...
- **Actor Cliente:** usuario registrado en el sistema con una determinada nomenclatura y que tendrá acceso a funcionalidades restringidas adaptadas a ese perfil de usuario tales como la realización de un pedido, acceso al programa Mendel de descendencia genética etc.
- **Actor Administrador:** usuario registrado en la aplicación con un código específico que puede realizar las operaciones propias de mantenimiento de la web y gestión de la pajarería. Entre sus funcionalidades podemos destacar el poder realizar pedidos de productos a proveedores o la modificación del catálogo de productos.

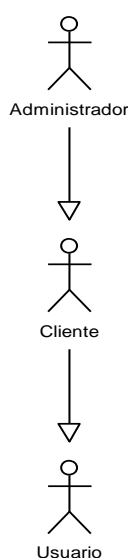


Figura 1. Jerarquía de usuarios

2.2. Especificación de Casos de Uso

Un caso de uso es una secuencia de transacciones que son realizadas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas, es decir, es un diagrama que muestra la relación entre los actores y los casos de uso en un sistema.

Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona una respuesta a eventos que se producen en el mismo.

Por cada caso de uso es necesario especificar información relativa a:

- **Actores:** es toda entidad externa que interactúa con el sistema y que demanda una funcionalidad de este.
- **Descripción del escenario:** es decir, cómo un usuario interactúa con el sistema y cuáles son los distintos caminos por los que puede evolucionar nuestro caso de uso.
- **Precondiciones y postcondiciones:** son las condiciones que han de darse antes para que se realice un caso de uso y cuáles son las condiciones que se cumplen posteriormente.
- **Identificación de las interfaces de usuario,** es la especificación de la interfaz de usuario donde se desarrolla el caso de uso.
- **Flujo de eventos y condiciones de fallo,** especifica el flujo de eventos que se desarrollan dentro del caso de uso en condiciones óptimas, también debe especificar el flujo del tratamiento que realiza el sistema en condiciones de fallo.

En esta fase cada caso de uso recoge información general de sistema, de modo que cada caso de uso puede englobar uno o varios requisitos del sistema.

3. Identificación de Subsistemas de Análisis

El objetivo de esta actividad es facilitar el análisis del sistema de información llevando a cabo la descomposición del sistema en subsistemas. El análisis de estos subsistemas está orientado a identificar los procesos de negocio. Para identificar cada subsistema, vamos a agrupar las funcionalidades de nuestro sistema por perfil del posible usuario que puede acceder a la web y además por proceso de negocio. Siguiendo este razonamiento hemos descompuesto nuestro sistema en tres paquetes funcionales:

- Paquete funcional público
- Paquete funcional cliente
- Paquete funcional administrador

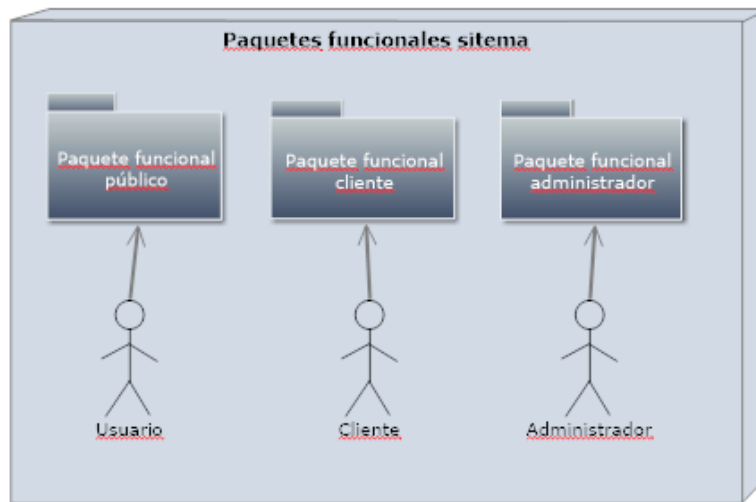


Figura 2. Paquetes funcionales del sistema

Paquete funcional público

El paquete funcional público abarca todas las operaciones que pueden ser realizadas por cualquier usuario que acceda a la web.

Las operativas que hemos identificado dentro de este paquete son:

- Acceso al formulario de alta en el sistema para poder acceder con un perfil Cliente.
- Consulta del catálogo de productos ofrecidos por la tienda on-line. De cada producto se podrá consultar además de su código y nombre, su precio unitario y el número de unidades disponibles en el almacén. El usuario podrá buscar un producto concreto para visualizar únicamente la información de dicho producto.
- Consulta de características de las distintas especies. El usuario podrá consultar distintos aspectos como cuidados, habilidades, enfermedades y curiosidades de las especies de pájaros que existen en la tienda.
- Consulta de centros. El usuario podrá conocer la localización, teléfono e email de las distintas tiendas que existen en España. De esta forma el usuario puede conocer cuál es su centro más cercano.
- Acceso al formulario de contacto. El usuario tendrá la posibilidad de contactar con una tienda concreta enviando un e mail. Así puede preguntar las dudas que le surgen acerca de cualquier producto, compra online etc. al centro que más le interese.

Casos de uso para el paquete funcional público

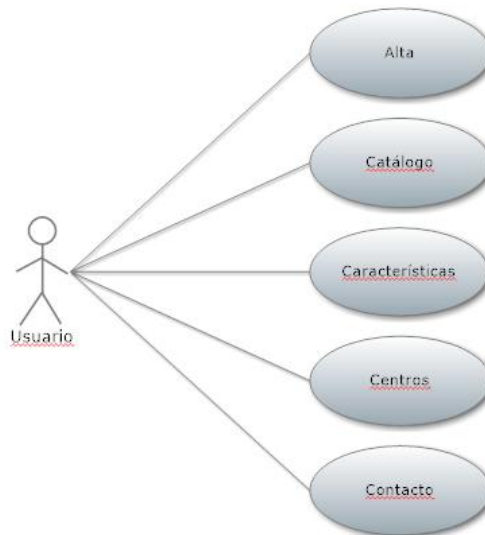


Figura 3. Paquete funcional público

1.- Caso de uso Alta:

Actor: Usuario no registrado.

Descripción: Cualquier usuario que acceda a la web de la pajarería puede registrarse rellendo un formulario de alta con sus datos personales.

Precondiciones: Seleccionar el enlace que accede al formulario para registrarse en el sistema.

Postcondiciones: Se da de alta al usuario en la web de la pajarería.

Flujo de eventos.

1. El usuario accede a la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario.
3. El usuario solicita el acceso al registro en la aplicación.
4. El sistema accede, y muestra el formulario que el usuario deberá rellenar para darse de alta en la web.
5. El usuario rellena el formulario de alta.
6. El sistema recoge el formulario relleno y guarda la información en la base de datos. A partir de ese momento el usuario pasará a ser un cliente de la aplicación y por tanto podrá acceder a la funcionalidad asociada a este perfil.

Condiciones de fallo.

1. Si algún campo del formulario no se ha relleno correctamente el sistema impedirá el alta y mostrará al usuario un mensaje de error indicándolo.
2. En caso de que se produzca un error al tratar de guardar los datos del usuario, el sistema también mostrará un mensaje de error indicándolo.

Interfaz: Paquete funcional público.

2.- Catálogo:

Actor: Usuario no registrado.

Descripción: Cualquier usuario que acceda a la web de la pajarería puede consultar el catálogo de productos de la tienda.

Precondiciones: Seleccionar la búsqueda de productos.

Postcondiciones: devuelve el listado de productos que se ajusten a los criterios de búsqueda.

Flujo de eventos.

1. El usuario accede a la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario.
3. El usuario solicita la opción de menú de catálogo de productos.
4. El sistema accede, y muestra el formulario que el usuario deberá rellenar con sus criterios de búsqueda.
5. El sistema realiza la búsqueda según los criterios de búsqueda.

Interfaz: Paquete funcional público.

3.- Caso de uso Características:

Actor: Usuario no registrado.

Descripción: Contiene información de diferentes aspectos acerca de varias especies de aves.

Precondiciones: Elegir un aspecto del cual desea conocer información y la especie sobre la que desea informarse.

Postcondiciones: Se muestran por pantalla la información solicitada de la especie elegida.

Flujo de eventos.

1. El usuario accede a la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario.
3. El usuario solicita el acceso las características de las aves.
4. El sistema accede, y muestra el menú horizontal con las opciones de: cuidados, habilidades, enfermedades y curiosidades. Dentro de cada opción se muestra una lista desplegable con las distintas especies de aves y variedades dentro de cada una de ellas.
5. El usuario elige una opción del menú horizontal y una especie (corresponde al caso de uso Selección especie).
6. El sistema recoge la elección del usuario y muestra la información correspondiente a la opción seleccionada y a la especie y variedad de ave.

Interfaz: Paquete funcional público.

4.- Caso de uso Consulta de Centros:

Actor: Usuario no registrado.

Descripción: Cualquier usuario que acceda a la web de la pajarería puede consultar información de los centros dados de alta en el sistema.

Precondiciones: Realizar una búsqueda de centros. Seleccionar una provincia dentro del mapa.

Postcondiciones: Se muestra la información de contacto (dirección, teléfono de contacto e email) de la tienda elegida por el usuario.

Flujo de eventos.

1. El usuario accede a la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario.
3. El usuario solicita el acceso a la consulta de centros.
4. El sistema accede, y muestra un mapa de España con todas sus provincias delimitadas.
5. El usuario elige una provincia y pincha sobre ella (corresponde al caso de uso Selección provincia).
6. El sistema muestra todas las tiendas ubicadas en esa provincia (con su dirección, teléfono y correo electrónico para poder contactar) o un mensaje informativo de que no todavía no existen tiendas en la provincia seleccionada.

Interfaz: Paquete funcional público.

5.- Caso de uso Formulario de Contacto:

Actor: Usuario no registrado.

Descripción: La aplicación contiene un formulario a través del cual el usuario puede formular cualquier pregunta a la empresa a través de la web.

Precondiciones: Posibilidad de elegir el centro al que se quiere enviar la consulta introduciendo su dirección de correo electrónico.

Postcondiciones: Se muestra un mensaje de éxito si todo ha ido bien y se envía el email al centro elegido.

Flujo de eventos principal.

1. El usuario accede a la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario.
3. El usuario solicita el acceso al formulario de contacto con la empresa.
4. El sistema accede, y muestra el formulario de contacto.
5. El usuario rellena el formulario, elige una provincia y posteriormente un centro al que enviará la pregunta y lo envía al sistema (corresponde al caso de uso enviar pregunta).
6. El sistema recibe la información, y abre una ventana de Outlook rellena con los datos del formulario para ser enviado al centro elegido.

Condiciones de fallo.

1. El sistema mostrará un mensaje de error en caso de que algún campo del formulario de contacto sea incorrecto o no se haya rellenado la dirección de correo electrónico de la tienda a la cual se enviara el formulario.
2. También se mostrará un mensaje de error cuando por algún motivo (caída del sistema, pérdida de conexión con las Base de Datos etc...) no se haya podido enviar el correo.

Interfaz: Paquete funcional público.

Paquete funcional cliente

El paquete funcional cliente abarca todas las operaciones que pueden ser llevadas a cabo por un usuario que ya ha sido dado de alta previamente con el perfil "Cliente".

Las operativas que hemos detectado dentro de este paquete funcional son:

- Identificarse/login. El usuario podrá introducir su login de usuario y password y el sistema le permitirá acceder a su cuenta personal. Todos los casos de uso relacionados con el paquete funcional cliente y administrador requieren la comprobación del usuario para poder llevarse a cabo.
- Realizar pedidos. El cliente tiene la posibilidad de hacer compras por internet. Podrá realizar búsquedas de los productos que desea comprar (por su código, familia, nombre o nada). De esta forma puede seleccionar los productos que se quieren adquirir y la cantidad de cada uno. También puede ver un desglose de toda la compra antes de realizarla para poder quitar productos de su cesta, añadir otros nuevos o modificar el pedido de alguno de sus productos. Visualizar información de los productos como su código, nombre, número de unidades existentes en el almacén y el precio por unidad. El cliente tiene la posibilidad de elegir la forma de pago de su pedido (contra reembolso o por transferencia bancaria). En caso de que todo el proceso de compra sea correcto, el pedido se realizará correctamente.
- Acceso al foro. El usuario podrá intercambiar información con los distintos usuarios. El cliente puede crear nuevos mensajes para los distintos temas en los que se divide el foro; y para una especie concreta También puede responder a los mensajes que existen.
- Modificación de los datos de usuario. El cliente puede revisar sus datos y cambiarlos o actualizarlos. En caso de modificar los datos de acceso como cliente puede verificar si sus nuevos datos no se han cogido todavía por otros clientes anteriormente. En este apartado también tendrá la posibilidad de darse de baja como cliente de la aplicación.
- Acceso al Programa Mendel. El usuario podrá realizar un pronóstico de la posible descendencia de pájaros Diamante de Gould, en función de los genes del padre y madre. Se podrá visualizar un gráfico y tabla con los resultados en porcentajes.

Casos de uso para el paquete funcional cliente

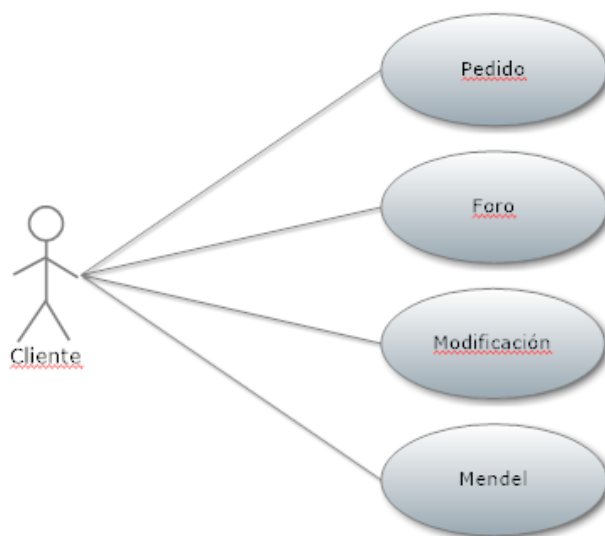


Figura 4. Paquete funcional cliente

1.- Caso de uso Pedido:

Actor: Usuario no registrado.

Descripción: Cualquier usuario que este dado de alta en el sistema como usuario de tipo cliente puede realizar pedidos de productos.

Precondiciones: Seleccionar el enlace que accede al formulario para registrarse en el sistema.

Postcondiciones: Se da de alta al usuario en la web de la pajarería.

Flujo de eventos.

1. El usuario se loga en la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso el usuario cliente.
3. El usuario hace una búsqueda de productos a través del popup de búsqueda de productos.
4. El sistema devuelve el listado de productos.
5. El usuario selecciona los productos que desea y los añade al listado de productos seleccionado.
6. El usuario continúa con el proceso de compra.
7. El usuario selecciona la forma de pago (contra reembolso o transferencia bancaria).
8. El sistema almacena la información relativa al pedido y devuelve el resultado de la operación.

Condiciones de fallo.

1. Si algún campo del formulario no se ha rellenado correctamente el sistema impedirá el alta y mostrará al usuario un mensaje de error indicándolo.
2. El sistema no debe permitir añadir una cantidad de productos superior al stock existente.
3. En caso de que se produzca un error al tratar de guardar los datos del usuario, el sistema también mostrará un mensaje de error indicándolo.

Interfaz: Paquete funcional cliente.

2.- Caso de uso Foro:

Actor: Usuario no registrado.

Descripción: Cualquier usuario que acceda a la web podrá hacer consultar e interactuar con otros usuarios a través del foro.

Precondiciones: Seleccionar el enlace que accede al apartado del foro.

Postcondiciones: Muestra los distintos temas del foro.

Flujo de eventos.

1. El usuario accede a la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario.
3. El usuario solicita la opción del menú del foro.
4. El sistema accede, y muestra los mensajes del foro agrupados por temas.
5. El usuario da de alta un mensaje sobre un tema.
6. El sistema da de alta el mensaje de modo que estará disponible para el resto de usuarios.

Condiciones de fallo.

1. Si algún campo del formulario no se ha rellenado correctamente el sistema impedirá el alta y mostrará al usuario un mensaje de error indicándolo.

Interfaz: Paquete funcional cliente.

3.- Caso de uso Modificación:

Actor: Usuario no registrado.

Descripción: Cualquier usuario que acceda a la web de la pajarería puede modificar sus datos personales.

Precondiciones: El usuario tiene que estar dado de alta en el sistema.

Postcondiciones: Se modifican los datos de usuario.

Flujo de eventos.

1. El cliente accede a la opción del menú correspondiente a la Gestión de datos de Usuario tras logarse en la web (corresponde al caso de uso Identificarse).
2. El sistema muestra el formulario de modificación de los datos de usuario.
3. El cliente puede consultar, modificar o borrar (darse de baja) sus datos de usuario en el sistema.
4. En caso de que el cliente solicite al sistema guardar modificaciones sobre sus datos de usuario, éste validará que los nuevos datos introducidos por el cliente son correctos y en caso de serlo los guardará en el sistema.
5. En caso de que lo que quiera el cliente sea darse de baja, el sistema pide confirmación por parte del cliente para continuar con el proceso de baja. Si el cliente confirma su decisión, el sistema eliminará sus

datos de alta y toda la información referente a él. A continuación la aplicación mostrará al cliente el menú de usuario ya que no está ya dado de alta en el sistema.

Condiciones de fallo.

1. Si el cliente tiene pedidos pendientes, el sistema emitirá un mensaje advirtiéndolo de esto e impedirá que el cliente pueda darse de baja en la web.
2. En caso de que se produzca algún fallo al actualizar la información en la base de datos se mostrará un mensaje de error y no se guardarán las modificaciones.

Interfaz: Paquete funcional cliente.

4.- Caso de uso Mendel:

Actor: Usuario no registrado.

Descripción: La aplicación muestra un formulario que le permite al cliente introducir los datos de los pájaros macho y hembra de los cuales se quiere comprobar la posible descendencia genética que tendrían.

Precondiciones: El cliente debe ser un usuario válido, rellenar el formulario con los fenotipos de los progenitores y solicitar al sistema que calcule la descendencia con esos datos.

Postcondiciones: El sistema calculará y mostrará la posible descendencia a partir de los datos proporcionados por el cliente

Flujo de eventos.

1. El cliente accede a la opción del menú correspondiente al Programa Mendel tras logarse en la web (corresponde al caso de uso Identificarse).
2. El cliente introduce los fenotipos del macho y la hembra.
3. El sistema validará que los fenotipos introducidos por el usuario son válidos y calculará la posible descendencia.

Condiciones de fallo.

1. Si el sistema detecta algún error durante el proceso de cálculo de la descendencia mostrará un mensaje indicándolo y volverá a la página principal del programa Mendel.

Interfaz: Paquete funcional cliente.

Paquete funcional administrador

El paquete funcional administrador abarca todas las operaciones que pueden ser llevadas a cabo por un usuario que ya ha sido dado de alta previamente con perfil "Administrador".

Las operativas que pueden realizarse dentro de este módulo son:

- Identificarse/login. El usuario podrá introducir su login de usuario y password y el sistema le permitirá acceder a su cuenta personal. Todos los casos de uso relacionados con el paquete funcional cliente y administrador requieren la comprobación del usuario para poder llevarse a cabo.
- Gestionar clientes. El administrador podrá realizar búsquedas de clientes concretos en función de distintos filtros como pueden ser su código de cliente, nombre, alguno de sus apellidos, el estado de un determinado pedido del cliente o la fecha en que se realizó alguno de sus pedidos. En caso de que no introduzca ningún parámetro de búsqueda, visualizará un listado con todos los clientes dados de alta en el sistema. De cada uno de ellos obtendrá como información su código de cliente, nombre, apellidos, estado de pedido y fecha de cada uno de los pedidos que cada cliente haya realizado. El administrador podrá elegir un pedido concreto de cualquier cliente y modificar su estado, en función de si el pedido ha avanzado o no (por ejemplo el pedido ya ha sido entregado al cliente o ha sido pagado) y guardar los cambios en el sistema. Por otro lado también tiene la posibilidad de dar de baja del sistema a clientes.
- Gestionar proveedores. Al igual que en el apartado anterior el administrador podrá realizar búsquedas de proveedores en función de algunos parámetros (código, nº documento, nombre, apellidos). El administrador puede eliminar uno o varios proveedores del sistema, también puede elegir un proveedor y modificar alguno de sus datos como pueden ser su nombre, país etc... o puede simplemente elegir un proveedor para ver su ficha personal que contendrá toda la información suya. Por último podrá dar de alta a nuevos proveedores en el sistema y modificar sus catálogos de productos pudiendo eliminar o añadir nuevos productos.
- Gestión de productos. El usuario podrá realizar un control de los productos existentes en el almacén. Se pueden realizar búsquedas de productos concretos o mostrar todos los productos existentes. Se podrán realizar pedidos de productos que estén desabastecidos o añadir nuevos productos en base de datos que entren en la tienda. El administrador también puede modificar alguno de los aspectos de un producto ya existente como puede ser su precio o por último eliminar productos que ya estén descatalogados.
- Gestión de envíos a clientes. El administrador podrá obtener un informe de los pedidos que se han realizado en unas fechas concretas.

- Gestión de pedidos. Se podrán consultar por un lado los pedidos que ha realizado los clientes a la tienda, y por otro los pedidos que el almacén ha hecho a proveedores. En ambos casos se pueden realizar búsquedas de pedidos concretos (filtrando por tipo de producto, fecha de realización del pedido etc...). Los resultados de las búsquedas se podrán guardar en el sistema.
- Consulta de stock. El administrador puede saber o consultar en todo momento la cantidad de cada producto que hay disponible en el almacén o en una tienda determinada. También se pueden realizar búsquedas de productos para conocer únicamente las unidades de un producto determinado en una tienda.
- Gestión de foro. Una de las funciones que tendrá un administrador será la gestión del foro de la web. Será el encargado de poder responder dudas o preguntas que puedan surgir a través de los diferentes hilos de conversación. Además también tendrá potestad para eliminar cualquier comentario o hilo de conversación que considere inadecuado o no siga la política del foro.
- Generar Catálogo. Crear un folleto virtual con los productos que se ofrecen en un período determinado a precios más bajos de los habituales. El administrador puede dar de baja ofertas descatalogadas o añadir nuevos productos al catálogo. También puede visualizar el catálogo de ofertas disponible en ese momento con una breve descripción de las características de cada producto y una imagen de estos.
- Visualización de gráficos. se podrán visualizar distintos gráficos en función de unos parámetros que se seleccionen como pueden ser tipo de producto, tienda y fechas... Los tipos de gráficos que se podrán consultar son:
 - Ventas por producto en un determinado periodo de tiempo.
 - Ventas por cada tienda.

Casos de uso para el paquete funcional administrador

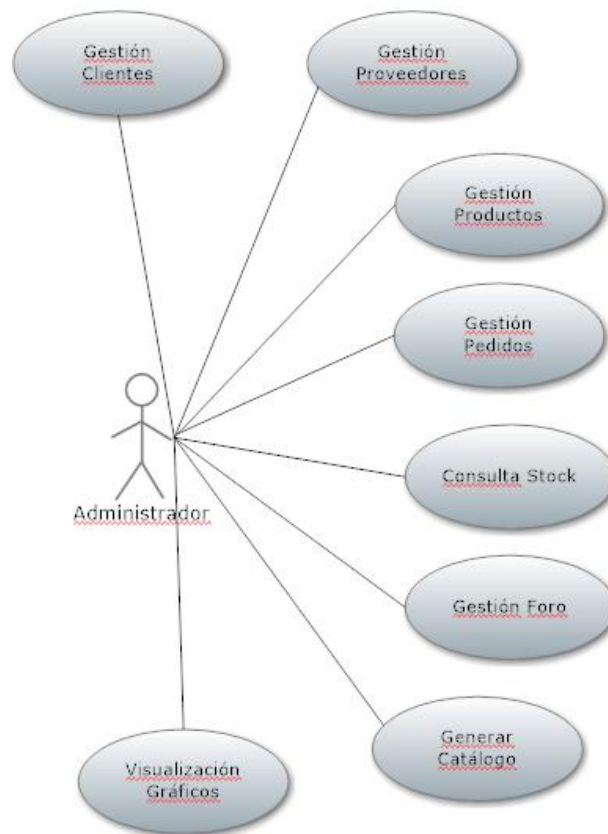


Figura 5. Paquete funcional administrador

1.- Caso de uso Gestión Clientes:

Actor: Usuario registrado con perfil administrador.

Descripción: Un usuario con perfil administrador puede gestionar los clientes del sistema.

Precondiciones: Seleccionar la opción de menú de gestión de usuarios.

Postcondiciones: Se modifican los datos relativos al cliente.

Flujo de eventos.

1. El usuario accede a la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario.
3. El usuario selecciona la opción de menú de gestión de clientes.
4. El sistema muestra los datos relativos al cliente, tanto datos personales como de pedidos.
5. El usuario modifica datos de usuario.
6. El sistema recoge la información y la guarda en la base de datos.

Condiciones de fallo.

1. Si algún campo del formulario no se ha rellenado correctamente el sistema impedirá el alta y mostrará al usuario un mensaje de error indicándolo.
2. En caso de que se produzca un error al tratar de guardar los datos del usuario, el sistema también mostrará un mensaje de error indicándolo.

Interfaz: Paquete funcional administrador.

2.- Caso de uso Gestión Proveedores:

Actor: Usuario registrado con perfil administrador.

Descripción: Un usuario con perfil administrador puede gestionar los proveedores del sistema.

Precondiciones: Seleccionar la opción de menú de gestión de proveedores.

Postcondiciones: Se modifican los datos relativos al proveedor.

Flujo de eventos.

1. El usuario accede a la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario.
3. El usuario selecciona la opción de menú de gestión de proveedor.
4. El sistema muestra los datos relativos al proveedor, tanto datos personales como de pedidos y productos.
5. El usuario modifica datos del proveedor.
6. El sistema recoge la información y la guarda en la base de datos.

Condiciones de fallo.

1. Si algún campo del formulario no se ha rellenado correctamente el sistema impedirá el alta y mostrará al usuario un mensaje de error indicándolo.
2. En caso de que se produzca un error al tratar de guardar los datos del usuario, el sistema también mostrará un mensaje de error indicándolo.

Interfaz: Paquete funcional administrador.

3.- Caso de uso Gestión Productos:

Actor: Usuario registrado con perfil administrador.

Descripción: Un usuario con perfil administrador puede gestionar los productos del sistema.

Precondiciones: Seleccionar la opción de menú de gestión de productos.

Postcondiciones: Se modifican los datos relativos al producto.

Flujo de eventos.

1. El usuario accede a la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario.
3. El usuario selecciona la opción de menú de gestión de productos.
4. El sistema muestra los datos relativos al producto.
5. El usuario modifica datos del producto.
6. El sistema recoge la información y la guarda en la base de datos.

Condiciones de fallo.

1. Si algún campo del formulario no se ha rellenado correctamente el sistema impedirá el alta y mostrará al usuario un mensaje de error indicándolo.
2. En caso de que se produzca un error al tratar de guardar los datos del usuario, el sistema también mostrará un mensaje de error indicándolo.

Interfaz: Paquete funcional administrador.

4.- Caso de uso Gestión Pedidos:

Actor: Usuario registrado con perfil administrador.

Descripción: Un usuario con perfil administrador puede gestionar los pedidos tanto de clientes como .de proveedores.

Precondiciones: Seleccionar la opción de menú de gestión de pedidos.

Postcondiciones: Se modifican los datos relativos al pedido.

Flujo de eventos.

1. El usuario accede a la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario.
3. El usuario selecciona la opción de menú de gestión de pedidos clientes.
4. El usuario hace una búsqueda de pedidos.
5. El usuario puede consultar los productos y los datos asociados al pedido.
6. El usuario seleccione la opción de menú de gestión de pedidos de proveedor
7. El usuario hace una búsqueda de pedidos.
8. El usuario puede consultar los datos del pedido y los productos asociados al pedido.

Condiciones de fallo.

1. Si algún campo del formulario no se ha rellenado correctamente el sistema impedirá el alta y mostrará al usuario un mensaje de error indicándolo.

Interfaz: Paquete funcional administrador.

5.- Caso de uso Consulta Stock:

Actor: Usuario registrado con perfil administrador.

Descripción: El administrador podrá consultar la cantidad que dispone de cada producto.

Precondiciones: El administrador podrá consultar la cantidad que dispone de cada producto.

Postcondiciones: Se mostrará un listado con el stock de los productos buscados (en función de los filtros de búsqueda introducidos) mostrando para cada uno de ellos su código, nombre y stock.

Flujo de eventos principal.

1. El administrador accede a la opción del menú correspondiente a Consulta de Stock tras logarse en la web (corresponde al caso de uso Identificarse).
2. El sistema accede y muestra el formulario de consulta de stock.
3. El administrador podrá rellenar los filtros de búsqueda que necesite (tienda y/ producto) en función de los datos que desee consultar.
4. El sistema recoge los filtros de búsqueda introducidos por el administrador y devuelve un listado de productos con su código, nombre y número de unidades disponible.

Condiciones de fallo.

1. El sistema mostrará un mensaje de error en caso de que los filtros de búsqueda sean incorrectos.
2. Si los filtros de búsqueda no producen ningún resultado, el sistema mostrará un mensaje de error indicándolo.
3. Si ocurriera algún problema al realizar la búsqueda de la información en la base de datos, el sistema mostrará un mensaje de error advirtiéndolo al usuario.

Interfaz: Paquete funcional administrador.

6.- Caso de uso Gestión Foro:

Actor: Usuario registrado con perfil administrado.

Descripción: El usuario administrador podrá realizar sus funciones como moderador del foro.

Precondiciones: Deben existir mensajes creados por un usuario.

Postcondiciones: Se muestra un mensaje de éxito sobre la acción del usuario.

Flujo de eventos principal.

1. El usuario accede a la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario.
3. El usuario selecciona la opción de menú relativa al foro.
4. El sistema accede, y muestra los mensajes del foro agrupados por temas y subtemas.
5. El usuario puede eliminar temas, subtemas y mensajes además de hacer aportaciones al foro.
6. El sistema guarda la información y actualiza la interfaz de usuario.

Condiciones de fallo.

1. El sistema mostrará un mensaje de error en caso de no haber podido realizar la operación.

Interfaz: Paquete funcional administrador.

7.- Caso de uso Generar Catálogo:

Actor: Usuario registrado con perfil administrador.

Descripción: La aplicación permite al usuario administrador seleccionar los productos que se van a añadir al catálogo de productos disponibles para la venta.

Precondiciones: El producto deberá estar dado de alta correctamente y tener un precio de venta al público.

Postcondiciones: Se chequea el producto como ofertado.

Flujo de eventos principal.

1. El usuario accede a la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario.
3. El usuario selecciona la opción de menú generar catálogo.
4. El sistema muestra el listado de productos.
5. El usuario chequea el producto como ofertado

Condiciones de fallo.

1. El sistema mostrará un mensaje de error cuando por algún motivo no haya podido actualizar el campo en base de datos.

Interfaz: Paquete funcional administrador.

8.- Caso de uso Visualización Gráficos:

Actor: Usuario registrado con perfil administrador.

Descripción: La aplicación permite visualizar mediante un gráfico las ventas de la tienda.

Precondiciones: Si se selecciona ventas por centro tiene que estar dado de alta algún centro previamente, además es obligatorio también el alta de algún producto.

Postcondiciones: Se muestra un gráfico con las ventas en un determinado periodo de tiempo.

Flujo de eventos principal.

1. El usuario accede a la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso

cualquier usuario.

3. El usuario selecciona la opción de menú de visualizar gráficos.
4. El sistema accede, y muestra los parámetros sobre los que generar el gráfico.
5. El usuario rellena los parámetros
6. El sistema recibe la información genera el gráfico.

Condiciones de fallo.

1. El sistema mostrará un mensaje de error en caso de que algún campo del formulario de contacto sea incorrecto o no se haya rellenado la dirección de correo electrónico de la tienda a la cual se enviara el formulario.
2. También se mostrará un mensaje de error cuando por algún motivo (caída del sistema, pérdida de conexión con la Base de Datos etc...) no se haya podido enviar el correo.

Interfaz: Paquete funcional administrador.

4. Análisis de los Casos de Uso

El objetivo de esta actividad es identificar las clases cuyos objetos son necesarios para realizar un caso de uso y describir su comportamiento mediante la interacción dichos objetos. Esta actividad se lleva a cabo para cada uno de los casos de definidos en la anterior actividad.

4.1. Identificación de Clases Asociadas a un Caso de Uso

Por cada caso de uso pasamos a identificar las clases implicadas.

Módulo público

Clases de Entidad

1. Alta usuario – Identificamos la clase cliente.
2. Catalogo – Identificamos la clase producto.
3. Centros – Identificamos la clase centro.
4. Formulario de contacto – Identificamos la clase mail.

Módulo cliente

Clases de Entidad

1. Alta Pedido – Intervienen las clases cliente, producto, banco y pedido.
2. Foro - Intervienen las entidades mensaje, tema y subtema.
3. Borrado/Modificación datos de usuario – Interviene la clase usuario.
4. Mendel – Interviene la clase pájaro.

Módulo Administración

Clases de entidad

1. Gestión Clientes – Interviene la clase cliente.
2. Gestión Proveedor – Intervienen las clases proveedor y productos.
3. Gestión Productos – Interviene la clase productos.
4. Gestión Envíos – Intervienen las clases producto y pedidos.
5. Gestión Pedidos – Intervienen clientes, proveedores, productos.
6. Consulta Stock – Interviene productos.
7. Gestión Foro – Intervienen las entidades mensaje, tema y subtema.
8. Generar Catálogo – Interviene la entidad producto.
9. Visualizar Gráfico – Intervienen las clases producto y centros.

4.2. Descripción de la Iteración de Objetos

El objetivo de esta tarea es describir la cooperación entre los objetos utilizados para la realización de un caso de uso, que ya fueron identificados en la tarea anterior. Para representar esta información, se usan diagramas de secuencia, que representan la iteración de un conjunto de objetos a través del tiempo.

A continuación pasaremos a describir brevemente los tipos de objetos que formarán parte de estos diagramas:

- **Usuario:** objeto que representa la persona o usuario que inicia la secuencia de interacciones (por ejemplo darse de alta en el sistema).
- **Ventana_i:** los objetos que comienzan con la nomenclatura "Ventana" representan los diferentes formularios de navegación a través de los cuales el Usuario interactúa con la aplicación.
- **_Action:** los objetos que finalizan con la palabra "Action" representan nuestra lógica de negocio y la parte en la que se aplica dicha lógica.
- **_BO:** los objetos que finalizan con la palabra "BO" representan los objetos que se encargan de representar nuestro modelo de datos y serán los encargados de persistir la información que se esté manejando.

MÓDULO PÚBLICO

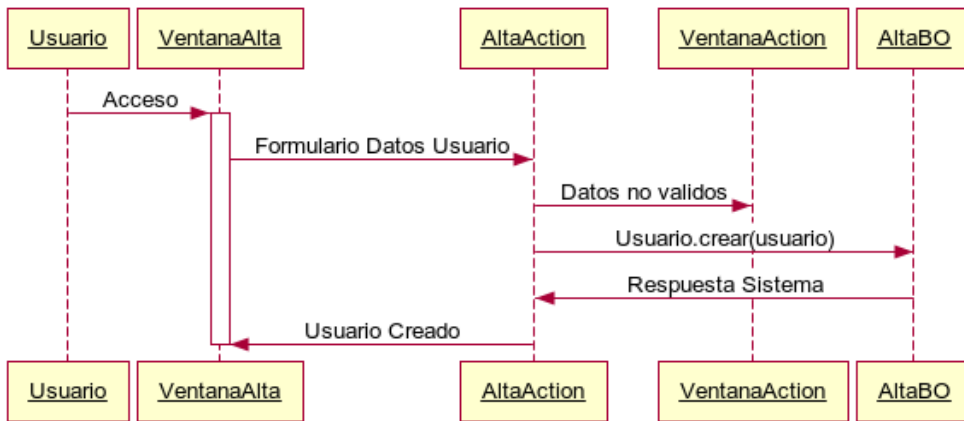


Figura 6. Alta Usuario

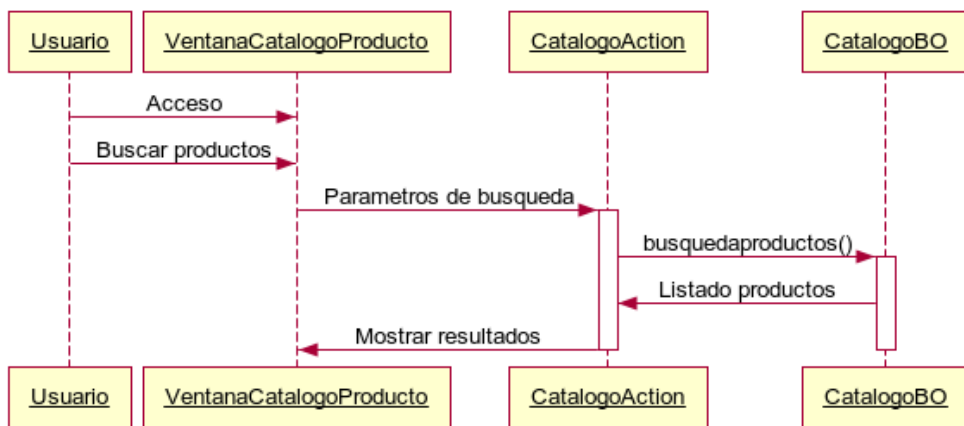


Figura 7. Catálogo Productos

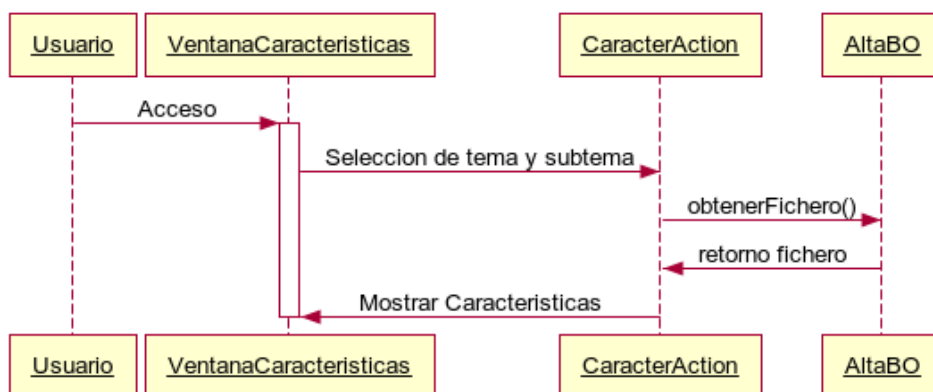


Figura 8. Características

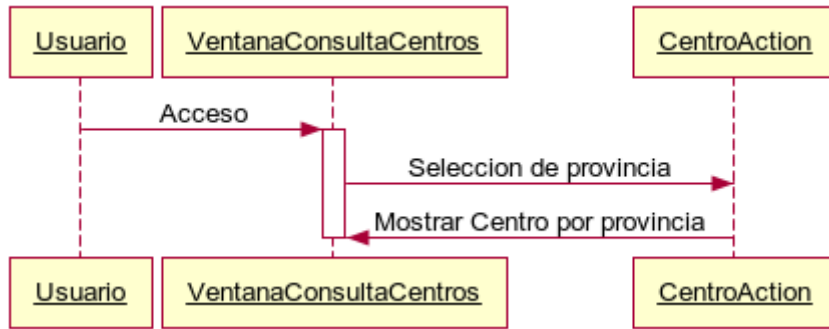


Figura 9. Consulta Centros

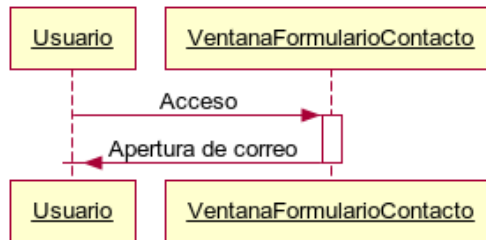


Figura 10. Contacto

MÓDULO CLIENTE

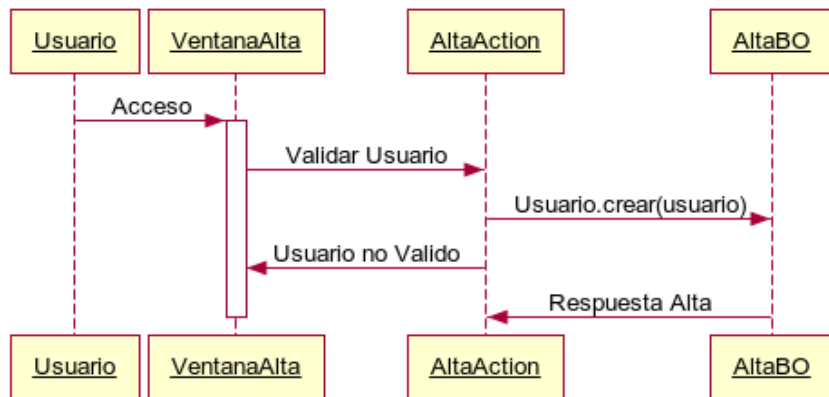


Figura 11. Login

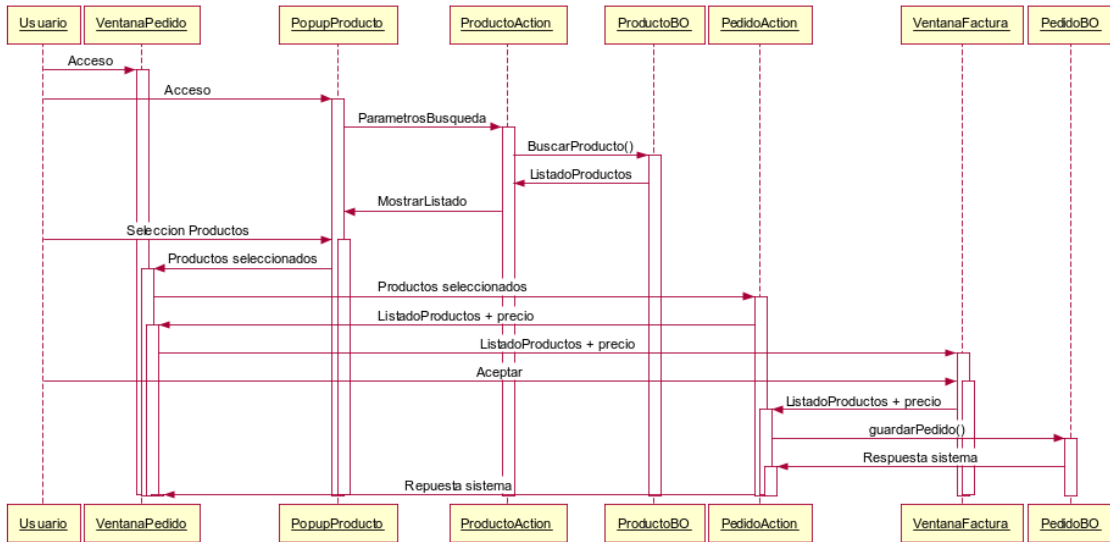


Figura 12. Pedido

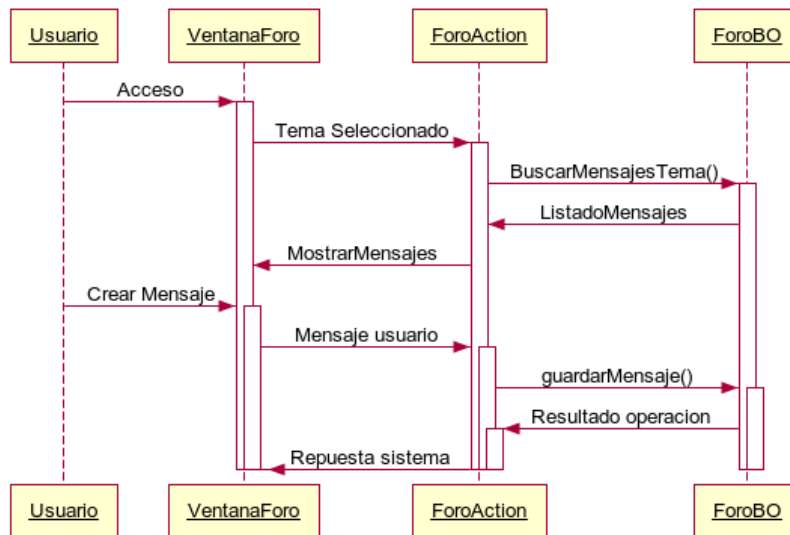


Figura 13. Foro

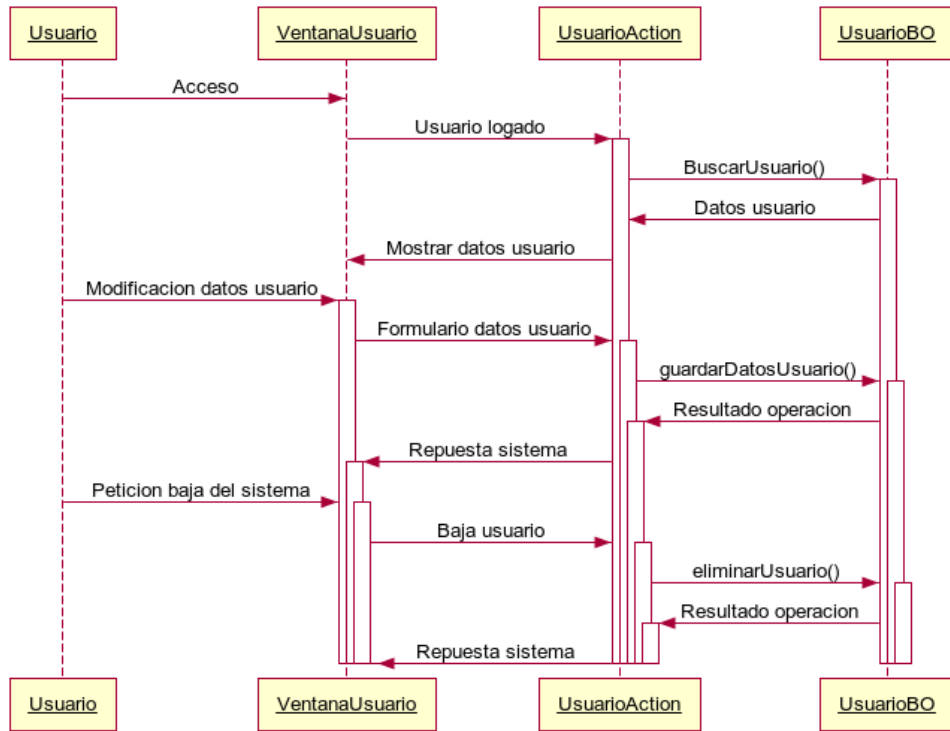


Figura 14. Modificación Usuario

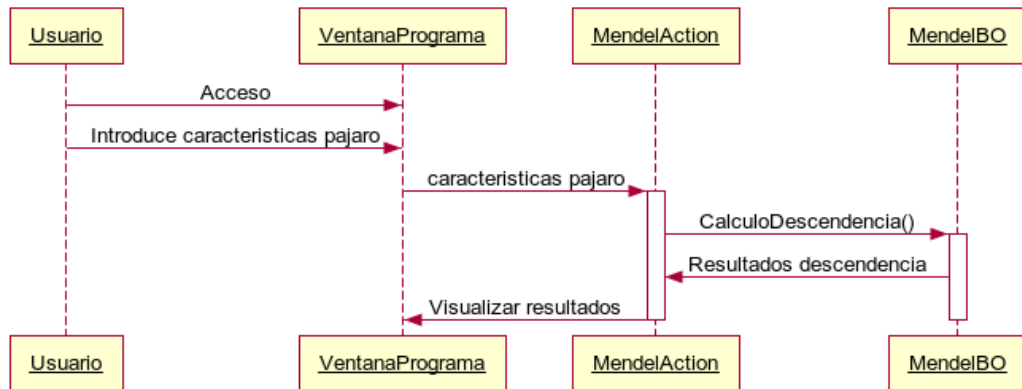


Figura 15. Mendel

MÓDULO ADMINISTRADOR

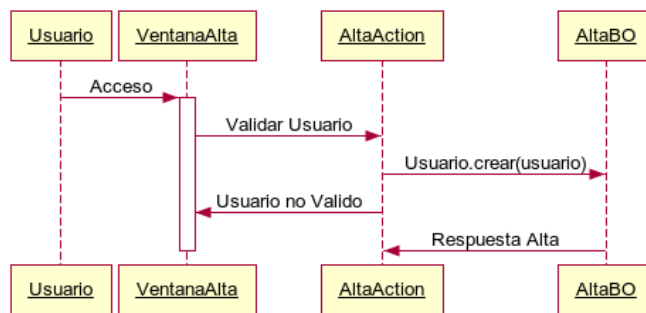


Figura 16. Login

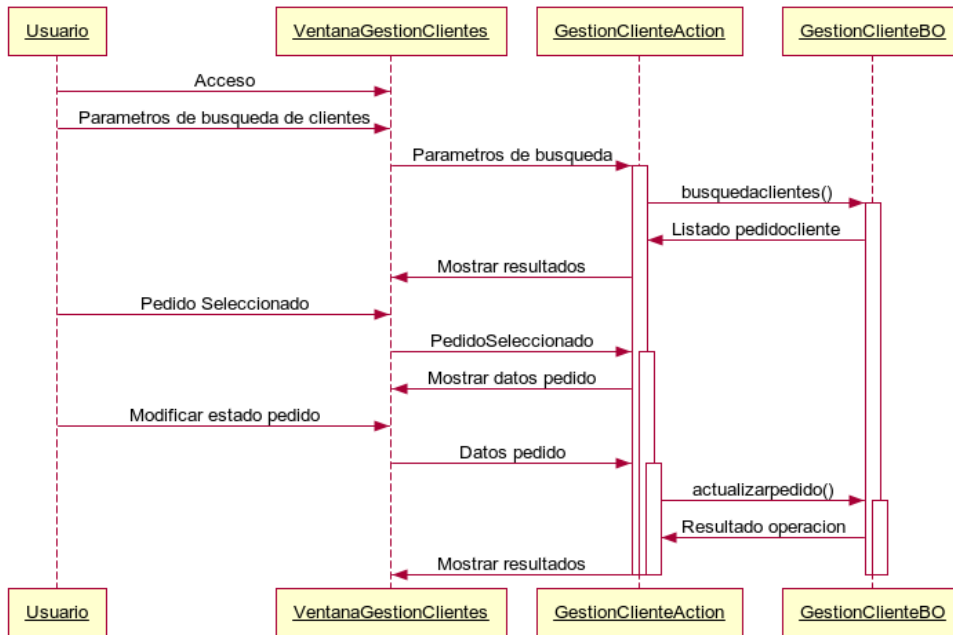


Figura 17. Gestión Clientes

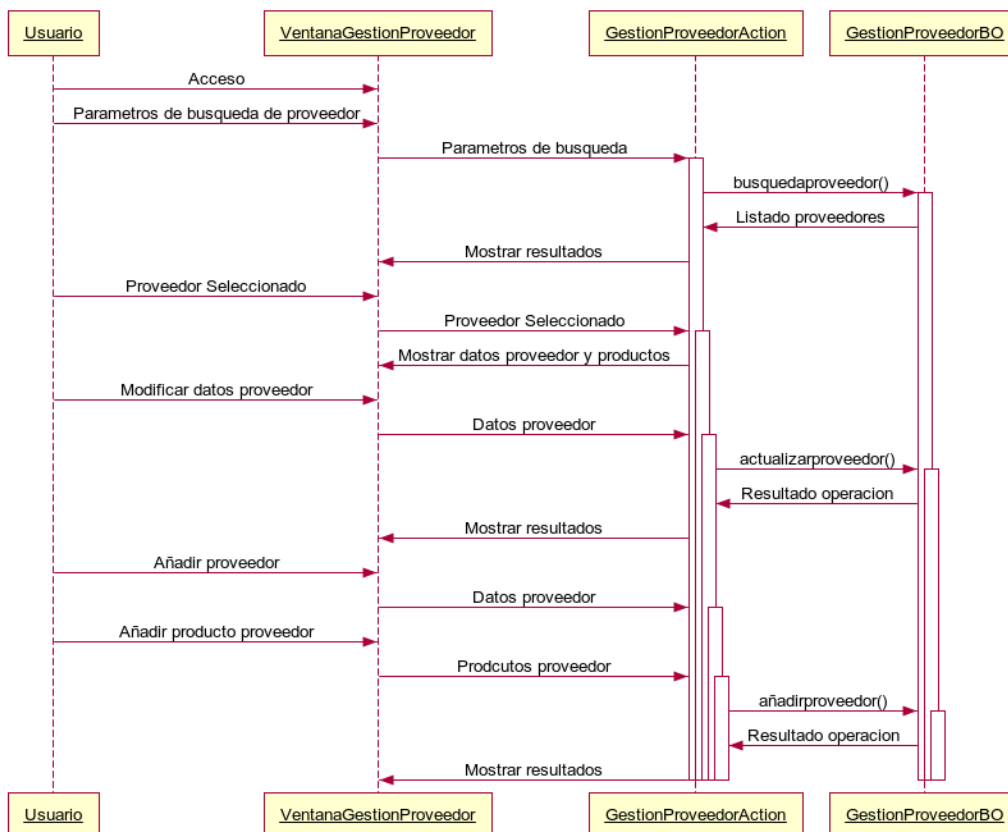


Figura 18. Gestión Proveedor

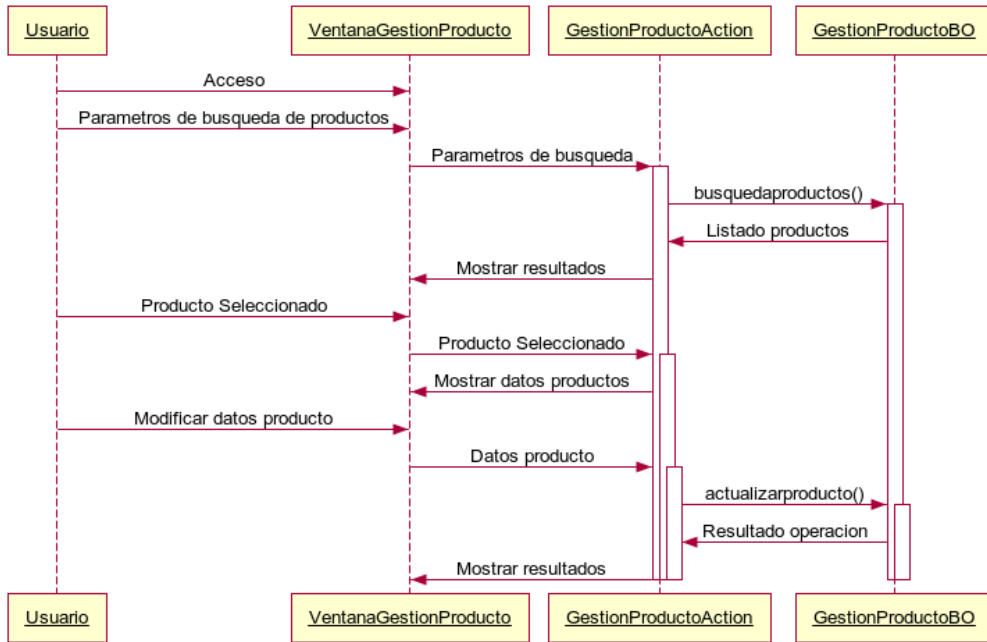


Figura 19. Gestión Productos

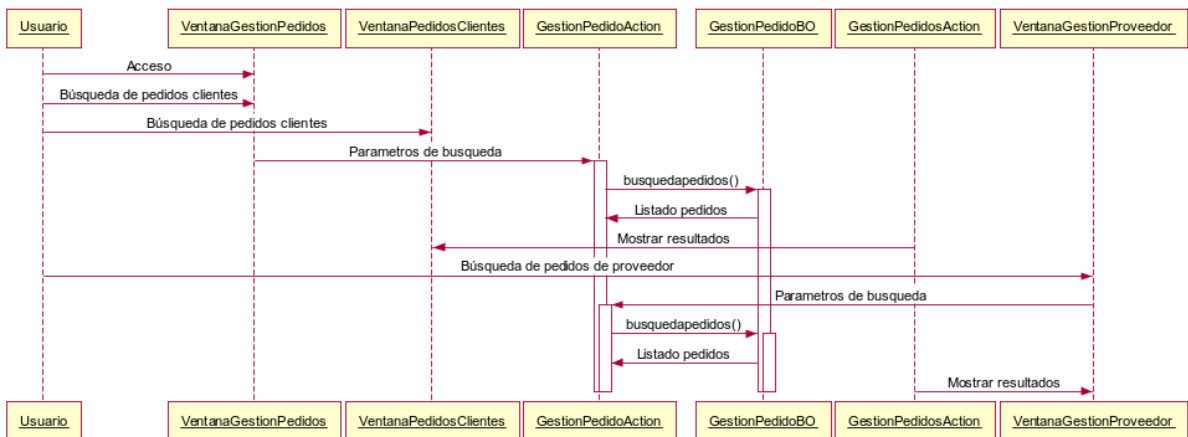


Figura 20. Gestión Pedidos

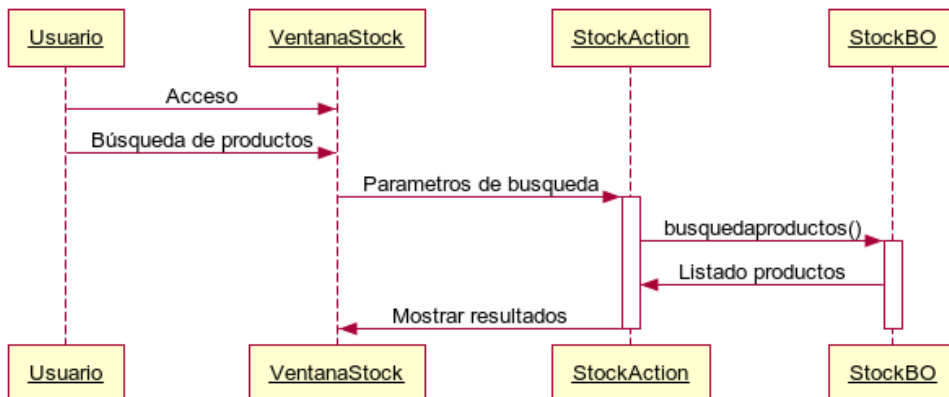


Figura 21. Consulta Stock

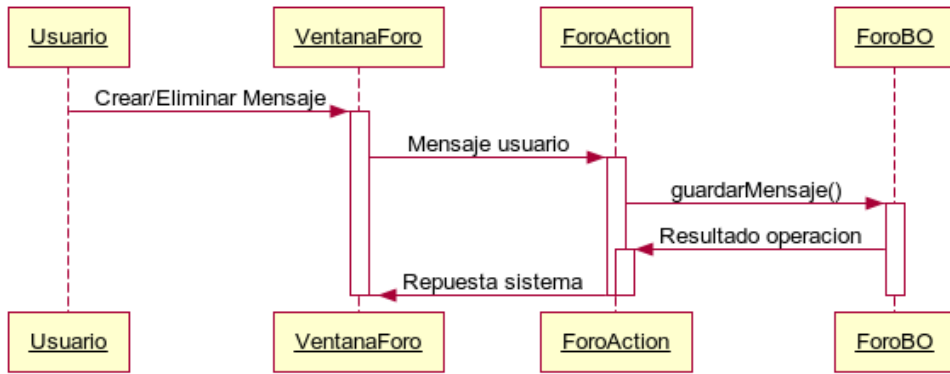


Figura 22. Gestión Foro

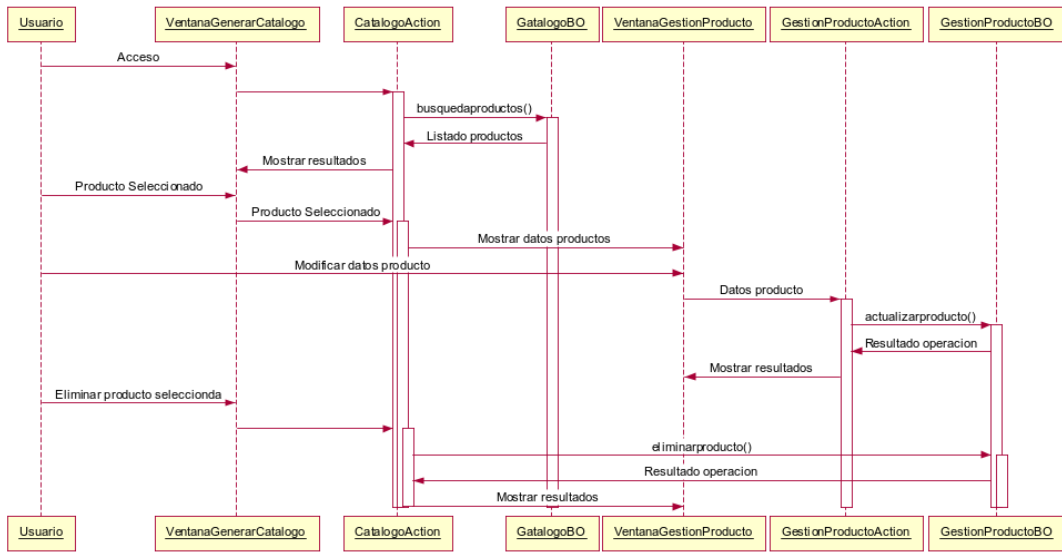


Figura 23. Generar Catálogo

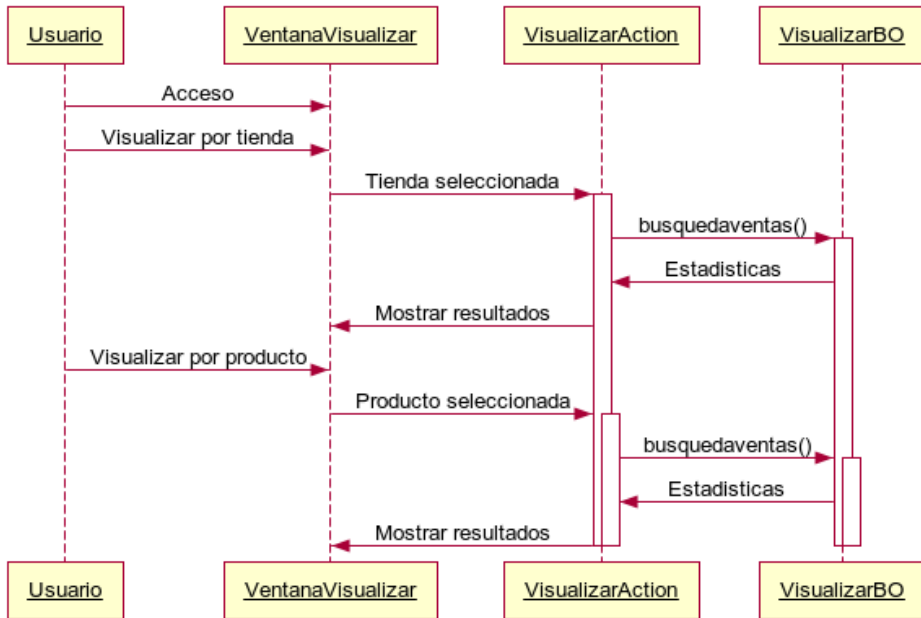


Figura 24. Visualizar Gráfico

5. Análisis de Clases

5.1. Identificación de Responsabilidades y Atributos

En esta tarea se identifican las responsabilidades y atributos relevantes de una clase. Las responsabilidades de una clase definen la funcionalidad de una clase y los atributos especifican las propiedades de una clase.

5.2. Identificación de Asociaciones y Agregaciones

En esta tarea se especifican los mensajes (relaciones) establecidos entre los objetos del diagrama de iteración, estas relaciones se caracterizan por el papel que desempeña en la relación entre los objetos, su dirección y la cardinalidad.

5.3. Identificación de Generalizaciones

El objetivo de esta tarea es definir la herencia y agrupación de las distintas clases. El estudio de los anteriores apartados da por resultado el siguiente Diagrama de Clases.

DIAGRAMA DE CLASES

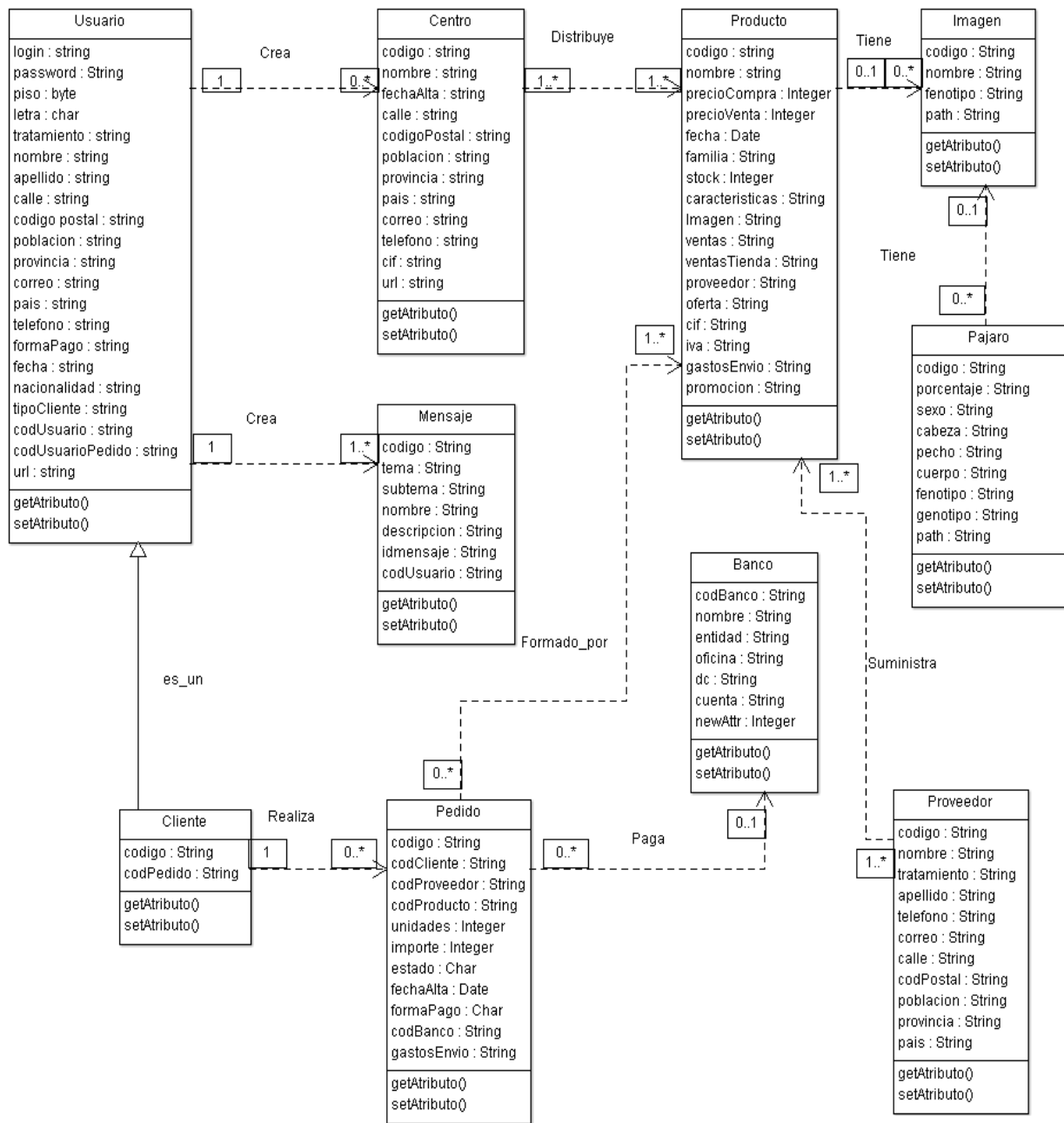


Figura 25. Diagrama de Clases

6. Definición de las Interfaces de Usuario

En esta actividad se especifican las interfaces entre el sistema y el usuario: formatos de pantallas, diálogos, e informes, principalmente. El objetivo es realizar un análisis de los procesos del sistema de información en los que se requiere una interacción del usuario, con el fin de crear una interfaz que satisfaga todos los requisitos establecidos, teniendo en cuenta los diferentes perfiles a quiénes va dirigido.

6.1. Principios generales

Todas las interfaces de entrada/salida del sistema seguirán la misma plantilla definido en la etapa de diseño, esto es, una cabecera con el logo de la empresa en la parte superior izquierda, el componente de logado en la parte superior derecha, que deberá estar siempre disponible, un menú lateral izquierdo diferente por cada perfil de usuario y una sección donde se recrea la navegación de la interfaz de usuario.

6.2. Especificación de formatos Individuales de la Interfaz

Los campos de texto deberán tener un tamaño, en la medida de lo posible, acorde con el tamaño requerido para dicho campo, es decir, si un campo de texto admite sólo 4 caracteres el campo de texto deberá tener un ancho de 4 caracteres.

Los botones deberán estar estandarizados, y por cada pantalla de entrada de datos deberá existir un botón “borrar” que permita resetear los campos introducidos previamente quedando limpio el formulario.

Para evitar posibles errores en la introducción de datos deberán utilizarse en la medida de lo posible componentes de tipo combo y radio button.

Las etiquetas de los componentes de tipo texto deberán alinearse a la derecha.

6.3. Formatos de impresión

Existen cuatro tipos de interfaz de usuario

- Pantallas de consulta. En las pantallas de consulta los elementos mostrados deberán aparecer en modo deshabilitado para impedir la modificación de estos por parte del usuario.
- Pantallas de modificación. En las pantallas de modificación se cargará la información disponible de modo que los campos no editables aparecerán deshabilitados y aquellos que el usuario pueda modificar aparecerán habilitados.
- Pantallas de entrada de datos. En estas pantallas estarán habilitados todos los campos. Se extenderá el uso de componentes de tipo combo y radio button para evitar en la medida de lo posible errores por inserción de usuario.
- Pantallas de búsquedas. En estas pantallas el resultado de la búsqueda debe mostrarse en un componente de tipo tabla, y en él sólo se mostrará la información útil para el usuario.

7. Especificación del Plan de Pruebas

En esta actividad se inicia la definición del plan de pruebas, el cual sirve como guía para la realización de las pruebas, y permite verificar que el sistema de información cumple las necesidades establecidas por el usuario, con las debidas garantías de calidad.

Los niveles de prueba que se realizarán a lo largo de los sucesivos procesos de ciclo de vida de software son los siguientes:

- Pruebas unitarias. Las pruebas unitarias tienen como objetivo verificar la funcionalidad y estructura de cada componente individualmente una vez que ha sido codificado
- Pruebas de integración. El objetivo de las pruebas de integración es verificar el correcto ensamblaje entre los distintos componentes una vez que han sido probados unitariamente con el fin de comprobar que interactúan correctamente a través de sus interfaces, tanto internas como externas, cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales especificados.
- Pruebas del sistema. Prueban a fondo el sistema, comprobando su funcionalidad e integridad globalmente en un entorno lo más parecido al entorno de producción.
- Pruebas de implantación. Comprueba el correcto funcionamiento del sistema dentro del entorno real de producción.
- Pruebas de aceptación. La labor principal de este tipo de pruebas es que el usuario valide el sistema como paso previo a la puesta en marcha de la aplicación. Deberán tenerse en cuenta aspectos como :
 - Procesos críticos del sistema
 - Rendimiento del sistema
 - Seguridad
 - Disponibilidad

8. Aprobación del Análisis del sistema de Información

En esta tarea se aprueba la presentación del Análisis del sistema de Información.

La siguiente tabla muestra a los usuarios participantes en la aprobación del análisis y las tareas desarrolladas en este documento.

Participante	Actividades												
	AS I1	AS I2	AS I3	AS ASI4	AS I5	AS I6	AS I7	AS I8					
Analistas	X	X	X	X	X	X	X	X					
Directores									X				
Usuarios	X												
Usuarios Expertos			X	X				X	X				
Arquitectos	X												
Jefe de Proyecto	X			X				X	X				

Figura 26. Diagrama de perfiles por tareas

ASI1 - Definición del Sistema

ASI2 - Establecimiento de Requisitos

ASI3 - Identificación de Subsistemas de Análisis

ASI4 - Análisis de los Casos de Uso

ASI5 - Análisis de Clases

ASI6 - Definición de las Interfaces de Usuario

ASI7 - Especificación del Plan de Pruebas

ASI8 - Aprobación del Análisis del sistema de Información

Diseño del Sistema

Índice de Contenido

Diseño del Sistema de Información	78
1. Antecedentes.....	78
1.1. Definición del proyecto	78
1.2. Objetivos del sistema	78
1.3. Objetivos del proyecto.....	79
2. Definición de la Arquitectura del Sistema	80
2.1. Definición de Niveles de Arquitectura.....	80
2.2. Identificación de Requisitos de Diseño y Construcción.....	82
2.3. Especificación de Excepciones.....	83
2.3.1 DNS Caído	83
2.3.2 Servidor Web Caído.....	83
2.3.3 ISP no da servicio	83
2.2.4 Servidor de Aplicaciones Caído	83
2.2.5 Servidor de Base de datos Caído.....	83
2.4. Especificación de Estándares y Normas de Diseño y Construcción	84
2.5. Identificación de los Subsistemas de Diseño	85
2.6. Especificación del Entorno Tecnológico.....	86
2.7. Especificación de Requisitos de Operación y Seguridad.....	91
3. Diseño de la Arquitectura de Soporte	92
4. Diseño de Casos de Uso Reales	93
4.1.- Diagrama casos de uso tienda virtual	93
4.2.- Diagrama casos de uso gestión del programa Mendel.....	98
4.3.- Diagrama casos de uso gestión de usuario	100
4.4.- Diagrama casos de uso gestión de productos y proveedores.....	103
4.5.- Diagrama casos de uso compra on-line	109
5. Matriz de Rastreabilidad	114
6. Diseño de Clases.....	119
6.1. Identificación de Clases Adicionales	119
6.2. Diagrama de clases.....	126
7. Diseño de la Arquitectura de Módulos del Sistema	144
7.1. Diseño del Modelo Físico de Datos.....	144
7.1.1. Diseño Conceptual	144
7.1.2. Diseño Lógico	156
7.1.3. Diseño Físico de Base de Datos	163
7.2. Optimización del Modelo Físico de Datos (Esquema Normalizado).....	164
8. Generación de Especificaciones de Construcción	176
8.1. Especificación del Entorno de Construcción	176
8.1.1 Entorno Tecnológico	176
8.1.2 Herramientas de construcción.....	176

8.1.3. Planificación	176
8.1.4. Requisitos de Operación y seguridad	176
9. Especificación Técnica del Plan de Pruebas	178
10. Modelo de Comportamiento del Sistema	179
10.1. Diagramas de Secuencia del Sistema	179
10.1.1. Diagrama de Secuencia Insertar Entidad	179
10.1.2. Diagrama de Secuencia Eliminar Entidad.....	180
10.1.3. Diagrama de Secuencia Modificar Entidad.....	180
10.1.4. Diagrama de Secuencia Buscar Información.....	181
10.1.5. Diagrama de Secuencia Comprar Producto.....	182
10.1.6. Diagrama de Secuencia Cambiar Credenciales Cliente.....	184
10.1.7. Diagrama de Secuencia Consultar Información.....	184
10.1.8. Diagrama de Secuencia Identificarse	185
10.2. Diagramas de Actividad del Sistema	186
10.2.1. Diagrama de Actividad Insertar Entidad	186
10.2.2. Diagrama de Actividad Eliminar Entidad	186
10.2.3. Diagrama de Actividad Modificar Entidad	187
10.2.4. Diagrama de Actividad Buscar Información	187
10.2.5. Diagrama de Actividad Comprar Producto	188
10.2.6. Diagrama de Actividad Cambiar Credenciales Cliente	189
10.2.7. Diagrama de Actividad Consultar Información.....	190
10.2.8. Diagrama de Actividad Identificarse.....	190

El objetivo del Diseño del Sistema de Información es la definición de la Arquitectura del sistema y del entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información.

En esta parte del proyecto obtendremos un análisis detallado de todo el proceso de construcción sistema de información, la definición de la arquitectura técnica del sistema, un plan de pruebas detallado, la definición de todos los requisitos de implantación y el diseño del procedimiento de migración y la carga inicial del proceso.

1. Antecedentes

1.1. Definición del proyecto

PAJARRACO'S PAJARERÍA es una aplicación web que proporciona todas las herramientas necesarias para la gestión online de una pajarería.

Esta aplicación no sólo pretende ser una tienda on-line donde los clientes realicen compras de productos especializados. Además pretende ofrecer un portal donde consultar información útil, así como interactuar con otros clientes a través de un foro gestionado por un usuario administrador para resolver dudas sobre temas específicos. Una de las herramientas que hacen más atractiva esta aplicación para los clientes es el Programa Mendel. Se trata de una aplicación embebida dentro del sistema donde los clientes podrán calcular las probabilidades de descendencia en el cruce de dos especies de Diamantes de Gould. Por último la aplicación provee de todo lo necesario para que un usuario con perfil administrador realice todas las tareas relativas a la gestión de la tienda, desde la gestión de centros, gestión de los pedidos de usuario y gestión de proveedores y productos. Además la herramienta ha de ser capaz de generar estadísticas de las ventas realizadas para poder realizar estudios que permitan mejorar la estrategia comercial.

1.2. Objetivos del sistema

El objetivo es desarrollar una herramienta capaz de poder ofrecer distintas funcionalidades basándose en el perfil del usuario que se registre en la web. En concreto el sistema identificara tres perfiles de usuarios que se corresponderán con tres menús de opciones diferentes.

Estos perfiles son:

- **PERFIL USUARIO:** usuario no registrado.

Podrá navegar por la web pero con acceso restringido (consultas de productos)
Registrarse como cliente.

- **PERFIL CLIENTE:** usuario registrado.

Podrá navegar por toda la web.
Podrá realizar pedidos.
Tendrá acceso al foro.
Podrá modificar sus datos de usuario o darse de baja.
Tendrá acceso al uso del programa "Mendel".
Podrá recibir en su correo los catálogos personalizados o descargarse en la web el catálogo de productos.

- **PERFIL ADMINISTRADOR:** usuario encargado de la gestión de la tienda.

Podrá navegar por la web.
Podrá gestionar altas, bajas y modificación de clientes.
Podrá gestionar altas, bajas y modificación de proveedores.
Podrá realizar pedidos de productos y gestionar los envíos a clientes.
Gestión del stock de las diferentes tiendas.
Podrá gestionar el foro.
Altas, bajas y modificación de productos.
Generar catálogos a usuarios.
Visualización de gráficos.

1.3. Objetivos del proyecto

El objetivo del proyecto no es otro que el Análisis, Desarrollo e Implantación de PAJARRACO'S PAJARERÍA (hardware, software base, software aplicativo y comunicaciones).

Las etapas en el desarrollo del proyecto se enumeran a continuación:

- Planificación del Sistema de Información
- Estudio de la viabilidad del Sistema de Información
- Análisis del Sistema de Información
- Diseño del Sistema de Información
- Construcción del Sistema de Información
- Implantación y Aceptación del Sistema de Información
- Mantenimiento del Sistema de Información

2. Definición de la Arquitectura del Sistema

En esta actividad se define la arquitectura general del sistema de información, especificando las distintas particiones físicas del mismo, la descomposición lógica en subsistemas de diseño y la ubicación de cada subsistema en cada partición. También se detallará la infraestructura tecnológica necesaria para dar soporte al sistema de información.

El particionamiento físico del sistema de información se especifica identificando los nodos y las comunicaciones entre los mismos, con cierta independencia de la infraestructura tecnológica que da soporte a cada nodo.

En cuanto al diseño lógico se especifica la estructura de la aplicación y sus componentes sin tener en cuenta donde se localizará el software, hardware ni la infraestructura.

Con el fin de organizar y facilitar el diseño, se realiza una división del sistema de información en subsistemas de diseño, como partes lógicas coherentes y con interfaces claramente definidas.

Como resultado de esta actividad, se actualizan los catálogos de requisitos y normas, y se generan los siguientes productos:

- Diseño de la Arquitectura del Sistema, como producto que engloba el particionamiento físico del sistema de información y la descripción de subsistemas de diseño.
- Entorno Tecnológico del Sistema, que a su vez comprende la especificación del entorno tecnológico, las restricciones técnicas y la planificación de capacidades.
- Catálogo de Excepciones.
- Procedimientos de Operación y Administración del Sistema.
- Procedimientos de Seguridad y Control de Acceso.

2.1. Definición de Niveles de Arquitectura

La arquitectura adoptada para desarrollar nuestro sistema se basa en la arquitectura cliente-servidor de 3 capas sobre un sistema multiusuario distribuido a través de una red de ordenadores.

En este tipo de arquitectura existen 3 figuras principales, por un lado el **cliente** que es quién interactúa directamente con los usuarios finales a través de la interfaz gráfica de usuario. El cliente para llevar a cabo su actividad necesita hacer peticiones de recursos y servicios, el receptor de estas peticiones es nuestra segunda figura, el **servidor**. Por otro lado separamos la lógica de negocio de la gestión de base de datos, es aquí donde aparece la tercera figura que será el **servidor de base de datos**. El servidor debe ser capaz de atender y controlar múltiples peticiones de clientes, las evalúa y decide si son aceptadas o rechazadas, si una petición es aceptada, devolverá la información solicitada al cliente.

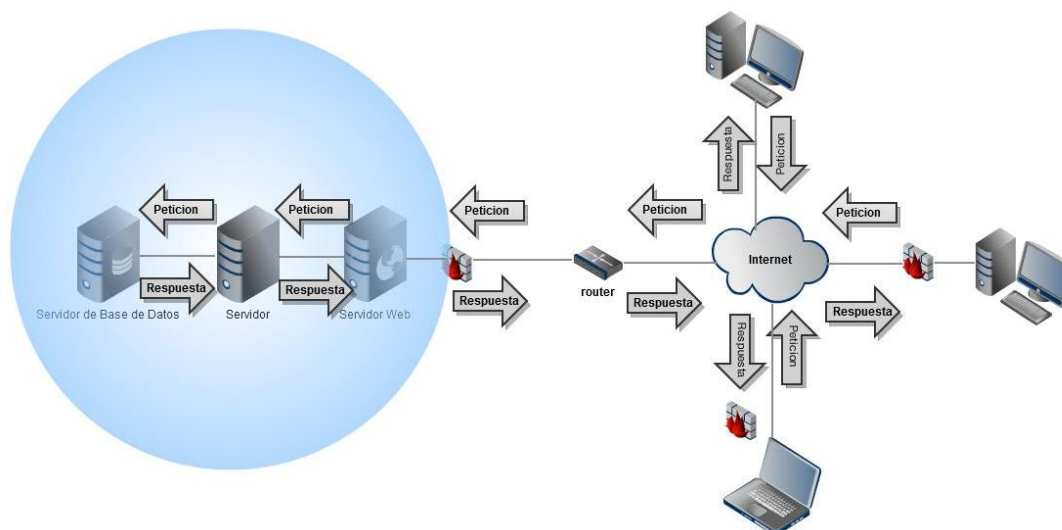


Figura 1. Jerarquía de usuarios

La comunicación entre los servidores clientes y el servidor de aplicaciones se lleva a cabo a través de una red, mediante el envío de unidades de información que se agrupan lógicamente y que se denominan *paquetes de datos*. Estos paquetes

necesitan de un protocolo para definir el formato en el que viajaran los datos entre diferentes dispositivos, para que sean reconocidos por ambos. Dicho protocolo establece una serie de propiedades en la comunicación entre las que se pueden destacar:

- **Sintaxis:** formato de los datos que viajaran a través de la red.
- **Semántica:** coordinación y control en el manejo de errores devueltos tras ejecutar una petición al servidor.

Las características más destacadas de este tipo de arquitectura son:

- Combinación entre un cliente que interactúa con el usuario, y un servidor que interactúa con los recursos compartidos por la aplicación. Como ya hemos dicho el cliente proporciona una interfaz entre el usuario y el resto del sistema. El servidor procesa los datos del cliente y a su vez hace peticiones al servidor de base de datos.
- La relación establecida es de muchos a uno, en la que el servidor puede dar servicio a muchos clientes, regulando el acceso a sus recursos compartidos.
- Los clientes corresponden a procesos activos que hacen peticiones al servidor que tiene un carácter pasivo, ya que espera esas solicitudes.
- La comunicación entre clientes y servidores se establece mediante el intercambio de mensajes entre ambos. El mensaje es el mecanismo para la petición y entrega de solicitudes de servicio.
- Tanto la plataforma hardware como el sistema operativo de las máquinas clientes, servidor y servidor de base de datos no tienen que coincidir.

Si atendemos a las características que deben tener las máquinas clientes y la servidora podemos nombrar las siguientes:

En cuanto al servidor:

- ✚ Comunicación pasiva: esperan a recibir peticiones de clientes.
- ✚ Procesan las peticiones y envían respuesta a los clientes.
- ✚ Procesan la lógica de la aplicación e interactúa con el servidor de base de datos.
- ✚ Son capaces de dar respuesta a múltiples peticiones.

En cuanto al servidor de base de datos:

- ✚ Comunicación pasiva: esperan a recibir peticiones de clientes.
- ✚ Procesan las peticiones y envían respuesta al servidor.
- ✚ Son capaces de dar respuesta a múltiples peticiones.

En cuanto al cliente:

- ✚ Administra la interfaz de usuario e interactúa con él.
- ✚ Comunicación activa: inician la comunicación con el servidor mediante el envío de solicitudes o peticiones.
- ✚ Espera y recibe las respuestas a las peticiones por parte del servidor y formatea los resultados.
- ✚ Es capaz de conectarse a varios servidores simultáneamente.
- ✚ Proporciona la interfaz gráfica que será el vínculo entre los usuarios de las aplicaciones con los servidores.

Además este tipo de arquitectura presenta una serie de ventajas que explican las razones por las cuales es una de las más utilizadas en la actualidad:

- ⇒ Cada vez son más baratas las plataformas hardware lo que ha provocado que cada vez se utilicen más las arquitecturas cliente – servidor.
- ⇒ Otra ventaja fundamental es la posibilidad de integrar sistemas diferentes. No todas las máquinas tienen porque utilizar el mismo sistema operativo.
- ⇒ Este tipo de arquitectura permite el uso de interfaces gráficas más amigables y manejables para los usuarios. De esta manera no siempre es necesario transmitir información gráfica por la red, ya que esta puede residir en el cliente, lo cual permite hacer un uso más óptimo de la red.
- ⇒ El mantenimiento y desarrollo de las aplicaciones que utilizan este tipo de arquitectura es más rápido ya que pueden usarse herramientas existentes como por ejemplo servidores SQL, sockets...
- ⇒ El hecho de estar utilizando los recursos de las máquinas clientes hace que la capacidad del sistema aumente y se puedan atender mayor número de peticiones hacia el servidor de la aplicación. Todo esto se traduce en un que el coste asumible para el funcionamiento y mantenimiento de la plataforma es menor.

2.2. Identificación de Requisitos de Diseño y Construcción

En esta tarea se realiza la especificación de los requisitos que están directamente relacionados con la adopción o diseño de una arquitectura o infraestructura concreta, y que pueden condicionar el diseño o la construcción del sistema de información. Entre estos requisitos pueden estar los relacionados con lenguajes, rendimiento de los distintos elementos de la arquitectura, así como criterios de ubicación de módulos y datos en los distintos nodos.

DNS (Domain name service) - Domain Name System o DNS (en español: **sistema de nombres de dominio**) es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada. Este sistema asocia información variada con nombres de dominios asignados a cada uno de los participantes. Su función más importante, es traducir (resolver) nombres inteligibles para las personas en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente.

El servidor DNS utiliza una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque como base de datos el DNS es capaz de asociar diferentes tipos de información a cada nombre, en nuestro caso lo usaremos para asignar a nuestra IP pública un nombre de dominio

Router – Es un dispositivo que proporciona conectividad a nivel de red. Su función principal consiste en enviar o encaminar paquetes de datos de una red a otra, es decir interconectar subredes. En nuestro caso configuraremos el router como un servidor virtual de modo que los usuario remotos puedan conectarse vía web o ftp a nuestro sitio vía IP pública y las peticiones se redireccionan al servidor web con una dirección IP privada, es decir, el router será el encargado de hacer el nateo de ip pública – ip privada y así redirigir peticiones web a nuestro servidor web.

Firewall – Es una parte del sistema que se encarga de bloquear las peticiones no autorizadas.

Servidor Web o servidor HTTP - es un programa informático que procesa una aplicación del lado de servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. La principal función de un servidor web es almacenar los archivos de un sitio y emitirlos por internet para poder ser visitado por los usuarios. Básicamente, un servidor web es una gran computadora que guarda y transmite datos vía Internet. Cuando un usuario entra en una página de Internet su navegador se comunica con el servidor enviando y recibiendo datos que determinan qué es lo que ve en la pantalla. Por eso decimos que los servidores web están para almacenar y transmitir datos de un sitio según lo que pida el navegador de un visitante.

Para nuestro sistema utilizaremos un servidor web **Apache**.

Servidor de aplicaciones - Es un dispositivo software encargado de ejecutar la aplicación. El servidor de aplicaciones gestiona las funciones lógicas de negocio y el acceso a los datos de la aplicación. Como servidor de aplicaciones utilizaremos **Websphere 6.1** que soporta el estándar J2EE.

Servidor de Base de Datos - es la máquina, física o lógica, sobre la que corre un servicio que ejecuta un Sistema Gestor de Bases de Datos Relacionales (SGBDR). En nuestro caso usaremos como sistema gestor de Base de Datos **MySQL**

2.3 Especificación de Excepciones

El objetivo de esta tarea es la definición de los comportamientos no habituales en el sistema, que reflejan situaciones anómalas o secundarias en el funcionamiento y ejecución del sistema de información. Para ello, se establece previamente el nivel de especificación de las mismas, así como los criterios de catalogación y clasificación.

2.3.1 DNS Caído

Descripción – El servidor DNS está caído o está fuera de servicio.

Elemento Afectado – Acceso a la aplicación.

Respuesta del sistema – El navegador devuelve el tipo de error servidor no encontrado.

2.3.2 Servidor Web Caído

Descripción – El servidor web está caído o está fuera de servicio.

Elemento Afectado – Acceso a la aplicación

Respuesta del sistema – El navegador devuelve el tipo de error servidor no encontrado

2.3.3 ISP no da servicio

Descripción – Nuestro ISP no nos da servicio.

Elemento Afectado – Acceso a la aplicación.

Condiciones Previas – El dns está levantado.

Respuesta del sistema – El navegador devuelve el tipo de error servidor no encontrado (Error 404).

2.2.4 Servidor de Aplicaciones Caído

Descripción – Nuestro Servidor Web está caído.

Elemento Afectado – Acceso a la aplicación.

Condiciones Previas – El dns está levantado y el ISP está dando servicio.

Respuesta del sistema – El navegador devuelve el tipo de error Servidor no encontrado (Error 404).

2.2.5 Servidor de Base de datos Caído

Descripción – Nuestro Servidor Web está caído.

Elemento Afectado – Cualquier consulta a base de datos.

Condiciones Previas – El dns está levantado y el ISP está dando servicio al igual que nuestro servidor web y de aplicaciones.

Respuesta del sistema – El navegador devuelve el tipo de error de aplicación (Error 500).

2.4 Especificación de Estándares y Normas de Diseño y Construcción

En esta tarea se definen los estándares técnicos y de nomenclatura, normas y recomendaciones de codificación para este proyecto que se va a desarrollar dentro de la tecnología J2EE.

Los objetivos de esta tarea es definir estándares de codificación para el desarrollo de una aplicación J2EE desarrollada mediante el patrón MVC e implementada con el framework de Struts.

Definiciones y Acrónimos

Acrónimo	Definición
JEE	Java Enterprise Edition
JSP	Java Servlet Pages
MVC	Model View Controller
UTF-8	8-bit Unicode Transformation Format
CSS	Cascading Style Sheets
URI	Uniform resource Identifier

Los documentos utilizados en la especificación de esta normativa son los siguientes:

Referencia	Nombre
http://www.oracle.com/technetwork/java/codeconv-138413.html	Code Conventions for the Java Programming Language.
http://java.sun.com/blueprints/code/namingconventions.html	Guidelines, Patterns, and code for end-to-end Java applications

Estándares generales

Encoding – Para evitar problemas de compatibilidad todos los ficheros deben ser codificados en UTF-8.

Estructura de directorios – El formato del proyecto debe seguir la estructura de plantilla de un proyecto web.

A continuación se detalla la estructura del proyecto

1º nivel	2º nivel	3º nivel	4º nivel	5º nivel	6º nivel	7º Nivel	Descripción
Pajareria Web							Nombre del proyecto
	JavaSource						Directorio con los fuentes del proyecto
		es	proyecto	common s	utils		Clases de utilidad del proyecto
		es	proyecto	pajareria	action		Clases controladoras
		es	proyecto	pajareria	bo		Clases del modelo de acceso a datos
		es	proyecto	pajareria	vo		Interfaz de las entidades del proyecto
		es	proyecto	pajareria	vo	impl	Clases que implementa las interfaces
		es	proyecto	pajareria	form		Formularios
		es	proyecto	pajareria	exceptio ns		Clases para el control de excepciones
		pajareriaw eb	resources				Directorio con los properties multiidiomas
	WebContent						Directorio de contenido Web

WebContent	css	Hojas de Estilo
	js	Javascript
	jsp	Jsp
	images	Imágenes de la aplicación
	WEB-INF	Directorio de configuración de ficheros de despliegue (web.xml, struts-config.xml.)
	WEB-INF classes	Ruta de los binarios de la aplicación

Nomenclatura – Por convenio aquella nomenclatura que por estandarización pueda escribirse en inglés se mantendrá en inglés (insert, delete, action.) y toda aquella que tenga que ver con lógica de negocio se definirá en castellano (nombres de procedimiento, constantes).

Nombres de clase – Los nombres de clases intercalarán mayúsculas y minúsculas. Los nombres de clases deben ser cortos y descriptivos. Todas las entidades tienen que tener un constructor así como los get y set de todos sus atributos. El número máximo de caracteres por línea es de 80. Cada clase tendrá un máximo de 2000 líneas de código y cada procedimiento de 150 líneas.

Métodos obsoletos – No está permitido el uso de métodos deprecated.

Constantes – Existirá una clase de constantes que contendrá todos los literales de la aplicación para tenerlos centralizados. Los nombres de las constantes se escriben en mayúsculas

Application-context – Para todos los literales de las jsp o componentes visuales existirá un fichero de propiedades (uno por idioma) para permitir su internalización.

Comentarios – Los comentarios de clases deben dar información al desarrollador sobre la implementación de la clase. Para cada clase se deben especificar los parámetros de entrada y de salida.

Variables – Las variables de clase deben estar inicializadas.

Propiedades – El acceso y/o modificación de las propiedades (atributos) de una clase deben realizarse a través de los métodos get() y set().

Código JSP – Sólo está permitido el uso de los taglib de struts. No está permitido el uso de includes salvo el caso de los menús laterales. En lugar del uso de los includes se hará uso de los tags de tiles.

Javascript – Todas las funciones javascript estarán dentro los ficheros js. No está permitido su uso dentro de las jsp.

CSS – Los estilos para los componentes visuales de la aplicación se darán mediante css.

Formularios – El paso de parámetros entre la vista y el controlador se realizará a través de formularios. El nombre para estos elementos tiene que ser identificativo de la entidad a la que representa con el sufijo Form.java.

2.5 Identificación de los Subsistemas de Diseño

En esta tarea se divide de forma lógica el sistema de información en subsistemas de diseño, con el fin de reducir la complejidad y facilitar el mantenimiento.

Hemos agrupado los distintos subsistemas que forman el sistema por funcionalidad de modo que hemos obtenido 5 subsistemas. Otros aspectos que hemos tenido en cuenta a la hora de identificar los subsistemas son que exista el menor acoplamiento posible entre ellos, de modo que permite su desarrollo en paralelo y que el mantenimiento o modificación de cada módulo no afecte al resto.

El siguiente diagrama muestra los subsistemas en los que se ha dividido nuestra aplicación:

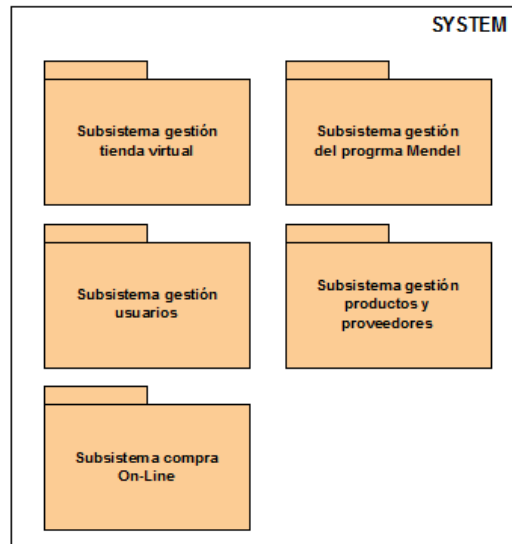


Figura 2. Diagrama de Subsistemas

Subsistema de gestión tienda virtual.

Agrupar todas las operaciones que puede realizar un usuario no logado en el sistema tales como consulta de información, consulta de centros, formularios de contacto, consulta de las características de una especie. También incluye todas las operaciones del foro (tanto a nivel del cliente como del administrador), la gestión de centros del administrador y la explotación de estadísticas también por parte del administrador.

Subsistema de gestión de usuarios.

Agrupar toda la operativa de alta de usuarios y la consulta baja y modificación de usuarios por parte del administrador del sistema.

Subsistema compra On-line.

Agrupar la operativa de consulta del catálogo de productos por parte del usuario logado (cliente), la operativa de compra de productos por parte del cliente y la gestión de pedidos por parte del administrador del sistema.

Subsistema de gestión Programa Mendel.

Agrupar todas las operaciones para el cálculo de probabilidades sobre la posible descendencia de dos diamantes de Gould con las características introducidas por el cliente.

Subsistema de gestión de productos y proveedores.

Agrupar todas las operaciones de gestión de productos y proveedores tales como alta, baja y modificación de productos y proveedores, gestión del stock y generación del catálogo de productos.

2.6. Especificación del Entorno Tecnológico

En esta tarea se definen en detalle los distintos elementos de la infraestructura técnica que dan soporte al sistema de información, determinando la implementación concreta de los nodos y comunicaciones especificados en la tarea Definición de Niveles de Arquitectura.

En la tarea de definición de los niveles de Arquitectura identificamos 3 elementos principales, cliente, servidor y servidor de base de datos. El protocolo utilizado será TCP/IP y dentro de este el protocolo HTTP que es el que se utiliza para acceder a las páginas web. En cuanto a la parte cliente es necesario una computadora con navegador de Internet, en nuestro caso nuestro sistema está optimizado para el navegador Internet Explorer 7.

Estructura Física del Sistema

El particionamiento físico de nuestro sistema viene determinado por la estructura de desarrollo de aplicaciones modelo cliente-servidor, siendo, para este caso, un cliente ligero sobre navegador web que no engloba lógica de negocio.

Así pues, el diseño lógico del sistema será en tres capas:

- Ubicación de la lógica de negocio en la capa servidora que, físicamente, deberá correr sobre el servidor de la aplicación.
- Capa de acceso a datos, ubicados físicamente sobre un esquema de datos relacional en el servidor de Base de Datos.
- Capa cliente. Cliente ligero sobre navegador web. El servidor de aplicación mandará la información al navegador a través de la red, una vez validados los requisitos de acceso al servidor de aplicaciones.

Entre otras, las ventajas de este sistema son:

- Centralización y actualización constante de una única base de datos espacial y alfanumérica.
- Flexibilidad ante cambios de máquinas.
- Facilidad de mantenimiento en cada una de las capas.

La arquitectura de tres capas consiste en la siguiente distribución con Struts:

Estructura Física con Struts

Nuestro sistema se desarrolla bajo el patrón MVC. Hoy en día existen múltiples herramientas que dan soporte para el desarrollo de aplicaciones web bajo el patrón MVC y la plataforma J2EE. Para el desarrollo de nuestro sistema vamos a utilizar el framework Struts.

Struts es un framework *open source* para el desarrollo de aplicaciones MVC construidas sobre Servlets y JSP's.

Las aplicaciones web difieren de las aplicaciones convencionales en que estas pueden crear respuestas dinámicas. Muchos sitios web sólo proporcionan contenido estático mientras que las aplicaciones web interactúan con base de datos y la lógica de negocio para proporcionar una respuesta a una petición de usuario.

Antes, las JavaServer Pages aglomeraban el código de acceso a base de datos, el diseño de la interfaz de usuario y el flujo de negocio dentro del mismo fichero jsp. Al introducirse el modelo vista controlador se separa cada una de estas partes en módulos independientes. El framework de Struts está diseñado para ayudar a los desarrolladores a implementar aplicaciones basadas en este modelo.

Este framework provee de tres componentes clave:

- Un manejador de peticiones que proporciona a la aplicación un modo de mapear peticiones a una URI estándar.
- Un manejador de respuestas que proporciona un control para tratar y transferir las peticiones a otros sistemas hasta formar respuestas completas.
- Unas librerías de tags que ayudan a los desarrolladores a interactuar entre los formularios de usuario y con las páginas del servidor.

Funcionamiento con Struts

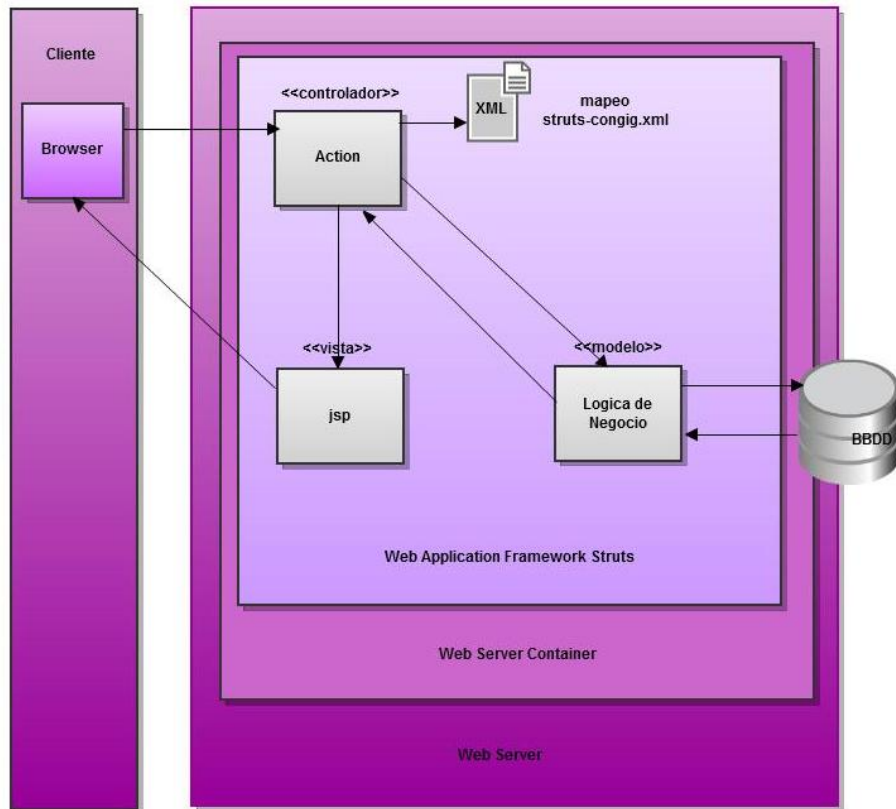


Figura 3. Funcionamiento Struts

- **ActionServlet** – La clase ActionServlet es el corazón del controlador de struts y se configura como servlet en el fichero web.xml. Cada vez que un usuario hace una petición es manejada por el ActionServlet de Struts quien intercepta la url y a través del fichero de configuración struts-config.xml dirige la petición a la clase de acción (DispatchAction) correspondiente. El DispatchAction es la encargada de comunicarse con el modelo. Todas las peticiones atendidas por el framework siguen un patrón en su nomenclatura (*.do). El método más importante de la clase Action es el método execute().

```
public ActionForward execute(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response)
    throws IOException, ServletException;
```

El método execute() lo invoca el controlador cuando se recibe una petición de un cliente. El controlador crea una instancia de la clase Action si no existe ya alguna. Solo se creara una instancia de la clase Action por cada aplicación. Se debe crear la propia implementación del método execute.

- **ActionForm** – Será el objeto que almacena el bean con los datos enviados por el usuario.
- **HttpServletRequest** – request es el objeto que proporciona los datos que se almacena durante el tiempo de vida de la petición. Es el objeto que contendrá la información que enviaremos como respuesta al browser.
- **ActionMapping** – será el objeto que almacena en memoria la entrada para el posterior mapeo en el struts-config.xml
- **Struts-config.xml** – Es el fichero encargado de controlar el flujo de la aplicación. En este fichero se mapean las peticiones (una acción se mapea a un determinada vista o en este caso jsp). En este fichero también se declaran los beans de formulario.
- **DispatchAction** – Esta clase hereda de la clase Action de struts y es la encargada de mediar entre la vista y el modelo.

- **Vista** – Struts facilita y soporta la construcción de la interfaz de la aplicación utilizando un conjunto de tags predefinidos. Busca evitar el uso de scriptlets que son trozos de código java en las jsp entre `<% %>` para ganar mantenibilidad y rendimiento.

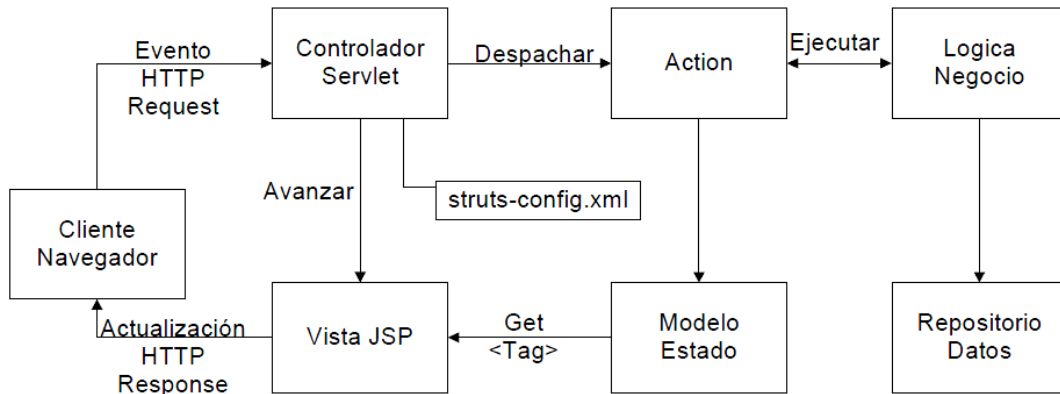


Figura 4. Flujo de una petición con Struts

El navegador genera una petición que es atendida por el controlador. El controlador analiza la solicitud y a través del fichero de configuración `struts-config.xml` llama al `DispatchAction` correspondiente pasándole los parámetros enviados. El paso de parámetros se realiza a través de los `ActionForm`. La clase de acción (`DispatchAction`) instanciará y/o usará los objetos de negocio que necesite. Según el resultado que retorne el controlador devolverá el control de la interfaz a una o más jsp las cuales podrán consultar los objetos del modelo para mostrar la información.

Struts y ámbito

El Frameworks Struts utiliza varias áreas de recursos compartidos para almacenar objetos. Las áreas de recursos compartidos tienen una vida útil y una regla de visibilidad que definen el ámbito del recurso.

- **Objeto `HttpServletRequest`**

Cada vez que un cliente lanza una petición HTTP, el servidor crea un objeto que implementa el interface: `javax.servlet.http.HttpServletRequest`. Este objeto contiene una colección de parejas atributo/valor que pueden ser usados para almacenar objetos durante el tiempo de vida de la petición. La clave para cada par es un `String` y el valor puede ser cualquier tipo de `Object`. Este objeto estará accesible desde el controlador (clase `Action`) para poder operar con él y generar la salida o respuesta del sistema.

Los métodos para almacenar y obtener objetos en un ámbito de petición sesión son:

```

public void setAttribute(String name, Object obj);
public Object getAttribute(String name);
  
```

Una vez el servidor ha atendido la petición y ha enviado la respuesta al cliente, la petición pasa a ser procesada por el `garbage collected` de la máquina Virtual, muriendo.

Struts contiene funcionalidades para almacenar JavaBeans dentro de la petición. Así pueden ser usados por componentes de presentación como páginas JSP. Estos javaBeans se almacenan en los formularios de nuestro sistema y se definen dentro del fichero de configuración `struts-config.xml`. Gracias a estos formularios es mucho más sencillo acceder a los datos de un JavaBeans, sin tener que `ServletContext` preocuparnos por limpiar o remover los objetos más tarde. Esta tarea la hará el contenedor web por nosotros. Los objetos que son almacenados en una petición no son visibles por otra petición de otro cliente.

- **Objeto `Session`**

El siguiente nivel de visibilidad es el nivel de ámbito de sesión. El contenedor web crea un objeto que implementa el interfaz `javax.servlet.http.HttpSession` para identificar a un usuario a lo largo de varias peticiones. La sesión de usuario se va a mantener por un tiempo que se basa en la frecuencia en la que el usuario realiza peticiones (se puede configurar este parámetro de inactividad a través del fichero `web.xml`), aunque también se puede invalidar invocando al método `invalidate()` del objeto de sesión. En la sesión existe una colección de objetos que son almacenados basados en pares de atributo/valor, similar a los objetos de petición. La

única diferencia entre unos objetos y otros es la duración, los objetos que se guardan en sesión permanecen más tiempo vivos que los que se almacenan en la petición. El framework de Struts usa atributos de sesión que extiende. Un ejemplo de un objeto que puede ser almacenado en un atributo de sesión es el objeto *Locale* del usuario que almacena el idioma del usuario. Los objetos almacenados en una sesión de usuario no son visibles para usuarios con una sesión diferente.

- **Objeto `ServletContext`**

En este nivel, la visibilidad y duración de los objetos almacenados es mayor. Aquí hay objetos que puede ser visibles por todos los clientes y threads de la aplicación web actual. Viven hasta que todos los programas desaparecen o cuando la aplicación termina. El servidor crea un objeto que implemente el interface *javax.servlet.ServletContext* por cada y para todas las aplicaciones web que están instaladas en el contenedor. Almacena objetos y los regresa exactamente igual que los objetos que residen en ámbitos de petición y sesión con la diferencia que los objetos son visibles por toda la aplicación y persisten por el tiempo de vida de la aplicación. El framework de Struts usa ámbito de aplicación para almejar JavaBeans que necesitan ser visibles para todos los usuarios. Normalmente, los objetos son almacenados en este ámbito durante el inicio de la aplicación y persisten mientras la aplicación existe.

- **Objeto `PageContext`**

El último nivel tratado es referido a las páginas JSP, y se refiere al ámbito de página. Los objetos con ámbito de página son solo accesibles desde la página JSP que son creados. Una vez la respuesta es enviada al cliente o la pagina redirige a otro recurso, el objeto ya no es accesible. Los objetos son almacenados en el objeto *javax.servlet.jsp.PageContext* por cada página. Todos las paginas JSP tiene un objeto implícito referenciado llamado `pageContext` que puede ser usado para almacenar y devolver objetos a nivel de pagina. Están incluidos los mismos métodos `getAttribute()` y `setAttribute()` que ofrecen los otros niveles de ámbito y su función es la misma.

Paso de parámetros por URL

Los parámetros URL son cadenas que se envían al servidor con la petición del cliente. Los parámetros se insertan en el objeto *HttpServletRequest* desde la cadena de consulta URI y los datos que se envían en un método POST. Los parámetros se formatean como pares clave/valor a los que acceden las aplicaciones como parámetros de petición.

Componentes de vista de Struts

Los componentes utilizados en la vista de una aplicación de Struts son:

- **ActionForm**

Los objetos `ActionForm` de Struts se utilizan para pasar los datos de entrada del cliente entre el usuario y la capa de negocio. El framework recupera automáticamente la entrada de datos de la petición y pasa estos datos a una `Action` utilizando un form bean, que luego se puede pasar a la capa de negocio.

- **JSP**

Las páginas JSP comprenden casi todo lo que se tiene que crear para los componentes de vista de Struts. Combina das con librerías de etiquetas personalizadas y HTML, JSP hace más sencillo el hecho de proporcionar un conjunto de vistas para una aplicación.

Librerías de Struts

Struts proporciona una serie de librerías para la implementación de las jsp's. También se puede ampliar las etiquetas Struts o crear sus propias etiquetas personalizadas. También proporciona un conjunto de clases para realizar la lectura de recursos de mensaje de una clase Java o un archivo de propiedades. Estas clases surgen a partir de `org.apache.struts.util.MessageResources`. Dentro del fichero `struts-config.xml` se define la entrada al fichero de configuración multi-idioma `ApplicationResources.properties` y a partir de aquí ya puede ser utilizada dentro de las jsp a través del tag `bean:message`.

2.7. Especificación de Requisitos de Operación y Seguridad

El objetivo de esta tarea es definir los procedimientos de seguridad y operación necesarios para no comprometer el correcto funcionamiento del sistema y garantizar el cumplimiento de los niveles de servicios que exigirá el sistema en cuanto a la gestión de operaciones. A partir de los requisitos establecidos para el sistema, la arquitectura y el entorno tecnológico definido hemos identificado los siguientes requisitos de operación y seguridad.

Requisitos de seguridad y control de Acceso

Acceso al sistema y a sus recursos: Cada usuario de la aplicación será autenticado mediante la asignación de un login y una contraseña. El acceso a las distintas partes de la aplicación se controla a través de distintos roles de usuario existirá un rol cliente y un rol administrador. La aplicación establecerá un tiempo de seguridad (time-out de sesión) transcurrido el cual, si no se detectó actividad, caducará el objeto en sesión y todos sus datos como el usuario logado.

El acceso a Base de Datos se hará mediante un usuario autenticado con permisos de administrador.

En cuanto a las copias de seguridad de la base de datos, las bases de datos relacionales proveen mecanismos específicos para los resguardos de seguridad y la recuperación ante una eventual necesidad. Estos mecanismos varían desde el backup a nivel sistema de archivo hasta copias replicadas en línea para cambiar el servidor de base de datos y continuar operando sin interrupciones. En nuestro sistema ejecutaremos back up de forma programada diariamente.

Requisitos de Operación

Los trabajos no pueden ser recuperados o reanudados. Simplemente, cada operación puede ejecutarse satisfactoriamente, o no aunque siempre que sea posible se realizarán tareas de roll-back en operaciones compuestas. Por ejemplo, si damos de alta un producto nuevo asociado a un proveedor y falla el alta del proveedor el sistema debe ser capaz de hacer un roll-back de toda la operación y eliminar el producto dado de alta.

3. Diseño de la Arquitectura de Soporte

En esta actividad se lleva a cabo la especificación de la arquitectura de soporte. Comprende el diseño de los subsistemas de soporte identificados en la actividad de Definición de la Arquitectura del Sistema (DSI 1), y la determinación de los mecanismos genéricos de diseño. Estos últimos sirven de guía en la utilización de diferentes estilos de diseño, tanto en el ámbito global del sistema de información, como en el diseño de detalle.

El diseño de los subsistemas de soporte, conceptualmente, es similar al diseño de los subsistemas específicos, aunque debe cumplir con unos objetivos claros de reutilización. De esta manera, se consigue simplificar y abstraer el diseño de los subsistemas especificados en la complejidad del entorno tecnológico, dotando al sistema de información y de una mayor independencia de la infraestructura que le da soporte.

La arquitectura de la aplicación se basa en el modelo vista controlador el cual se implementará montado sobre componentes del framework de struts. A continuación en la siguiente figura se detalla un diagrama de secuencia genérico, donde se muestra cómo interactúan las distintas clases cuando se trabaja con estos componentes de struts:

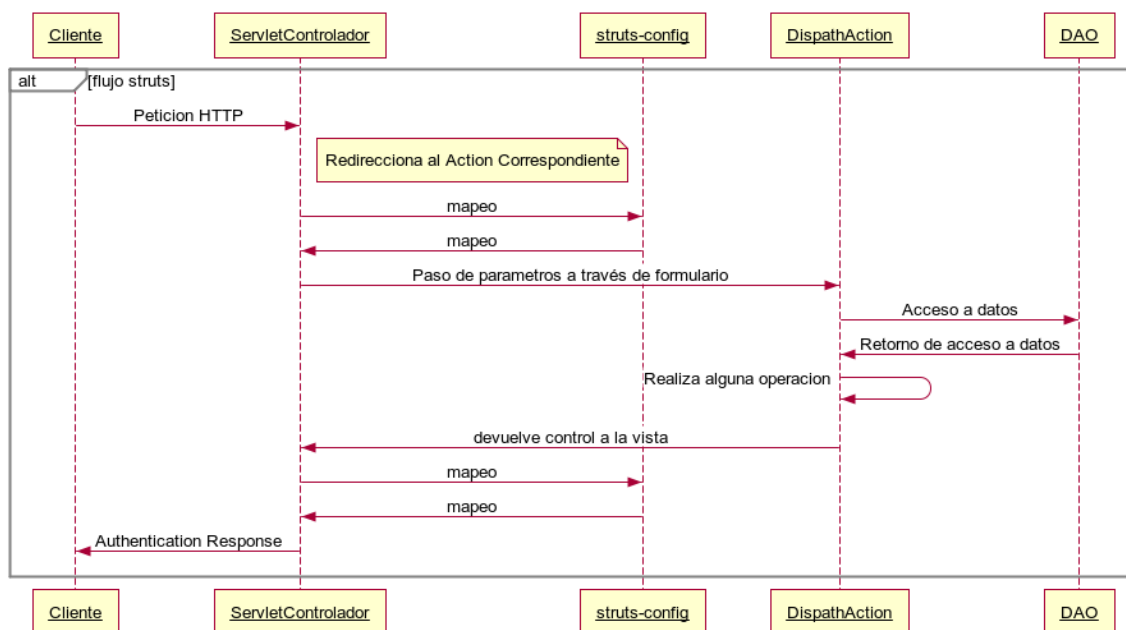


Figura 5. Diagrama de Secuencia Genérico

4. Diseño de Casos de Uso Reales

Esta actividad tiene como propósito especificar el comportamiento del sistema de información para un caso de uso, mediante objetos o subsistemas de diseño que interactúan, y determinar las operaciones de las clases e interfaces de los distintos subsistemas de diseño.

A partir de los subsistemas de Análisis generamos los diagramas de casos de uso.

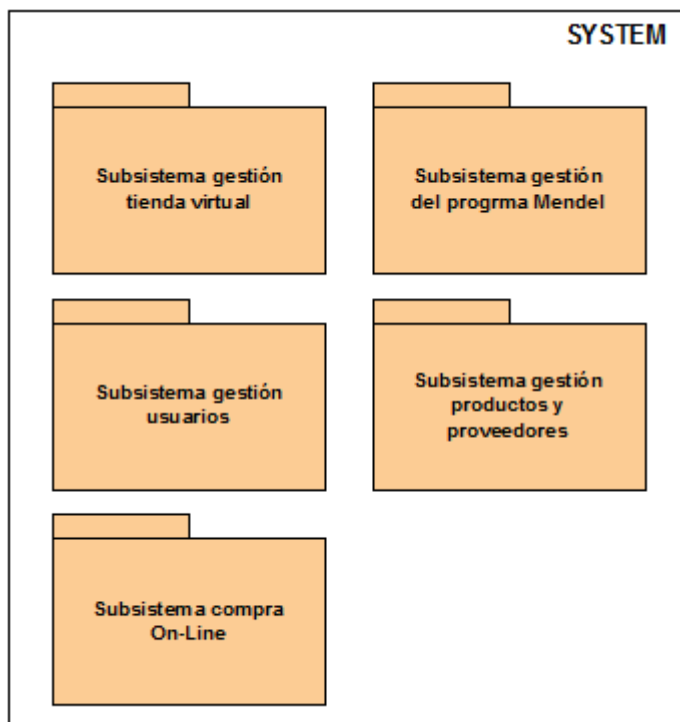


Figura 6. Diagrama de Subsistemas

4.1.- Diagrama casos de uso tienda virtual

CASOS DE USO DEL SUBSISTEMA TIENDA VIRTUAL

1.- Caso de uso Consultar información (RF-01):

Descripción: Muestra información de la empresa.

Objetivo: Permite al usuario conocer aspectos de la empresa como sus inicios, situación actual, información práctica, servicios que se van a poder encontrar en la página, como comprar, como realizar los pagos, dudas y sugerencias.

Actor: Usuario; no necesita ningún privilegio especial para poder acceder a la información general de la empresa.

Precondiciones: Acceder a la pantalla principal de la aplicación.

Postcondiciones: Ninguna.

Flujo de eventos principal:

1. El usuario accede a la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario y en la parte central de la pantalla toda la información general acerca de la empresa.

Escenario: Subsistema tienda virtual.

2.- Caso de uso Mostrar características (RF-02):

Descripción: Contiene información de diferentes aspectos acerca de varias especies de aves.

Objetivo: Permite al usuario conocer información como cuidados, habilidades, enfermedades y curiosidades acerca de diferentes especies de aves. De esta manera el usuario podrá conocer más a fondo el comportamiento y los cuidados necesarios de cada ave antes de realizar una compra. Este caso de uso de uso incluye el caso de uso Selección empresa.

Actor: Usuario; no necesita ningún privilegio especial para poder acceder a la información acerca de las características de distintas especies de aves.

Precondiciones: Elegir un aspecto del cual desea conocer información y la especie acerca de la cual desea informarse.

Postcondiciones: Se muestran por pantalla la información solicitada de la especie elegida.

Flujo de eventos principal:

2. El usuario accede a la web de la aplicación.
3. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario.
4. El usuario solicita el acceso las características de las aves.
5. El sistema accede, y muestra el menú horizontal con las opciones de: cuidados, habilidades, enfermedades y curiosidades. Dentro de cada opción se muestra una lista desplegable con las distintas especies de aves y variedades dentro de cada una de ellas.
6. El usuario elige una opción del menú horizontal y una especie (corresponde al caso de uso Selección especie).
7. El sistema recoge la elección del usuario y muestra la información correspondiente a la opción seleccionada y a la especie y variedad de ave.

Escenario: Subsistema tienda virtual.

3.- Caso de uso Selección especie (RF-03):

Descripción: Menú horizontal que muestra las opciones que puede elegir el usuario. Estas opciones son cuidados, habilidades, enfermedades y curiosidades. Dentro de cada opción del menú, se visualizará un submenú con las especies de las cuales la aplicación muestra información.

Objetivo: Permite al usuario elegir la información que desea conocer.

Actor: Usuario; no necesita ningún privilegio especial.

Precondiciones: Acceder a la opción de menú de características.

Postcondiciones: Se muestran por pantalla la información solicitada de la especie elegida.

Flujo de eventos principal:

1. El usuario elige una opción del menú horizontal y una especie.

Escenario: Subsistema tienda virtual.

4.- Caso de uso Ponerse en contacto (RF-04):

Descripción: La aplicación contiene un formulario a través del cual el usuario puede formular cualquier pregunta a la empresa a través de la web.

Objetivo: Realizar cualquier consulta por medio de un correo electrónico. Este caso de uso incluye el caso de uso Enviar pregunta.

Actor: Usuario; no necesita ningún privilegio especial para poder acceder a la opción que le permite contactar con la empresa.

Precondiciones: Posibilidad de elegir el centro al que se quiere enviar la consulta introduciendo su dirección de correo electrónico.

Postcondiciones: Se muestra un mensaje de éxito si todo ha ido bien y se envía el email al centro elegido

Flujo de eventos principal:

1. El usuario accede a la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario.
3. El usuario solicita el acceso al formulario de contacto con la empresa.
4. El sistema accede, y muestra el formulario de contacto.
5. El usuario rellena el formulario, elige una provincia y posteriormente un centro al que enviará la pregunta y lo envía al sistema (corresponde al caso de uso enviar pregunta).
6. El sistema recibe la información, y abre una ventana de Outlook rellena con los datos del formulario para ser enviado al centro elegido.

Flujo de eventos excepcional:

1. El sistema mostrará un mensaje de error en caso de que algún campo del formulario de contacto sea incorrecto o no se haya relleno la dirección de correo electrónico de la tienda a la cual se enviara el formulario.
2. También se mostrará un mensaje de error cuando por algún motivo (caída del sistema, pérdida de conexión con la Base de Datos etc....) no se haya podido enviar el correo.

Escenario: Subsistema tienda virtual.

5.- Caso de uso Enviar pregunta (RF-05):

Descripción: Enviar una consulta a la empresa.

Objetivo: Enviar el formulario de contacto relleno por el usuario.

Actor: Usuario.

Precondiciones: Tener rellenado todos los campos del formulario.

Postcondiciones: Se envía el email a la tienda elegida.

Flujo de eventos principal:

El sistema recoge la información del formulario y la envía al centro que corresponda.

Flujo de eventos excepcional:

1. El sistema mostrará un mensaje de error en caso de que se haya producido algún fallo o problema al intentar enviar el formulario de contacto.

Escenario: Subsistema tienda virtual.

6.- Caso de uso Consulta de centros (RF-06):

Descripción: La aplicación muestra en un mapa de España la ubicación de todos los centros asociados a la empresa.

Objetivo: El usuario podrá conocer la localización y contacto de los distintos centros pertenecientes a la empresa con un solo vistazo. Este caso de uso de uso incluye el caso de uso Selección provincia.

Actor: Usuario; no necesita ningún privilegio especial para poder acceder a la consulta de centros.

Precondiciones: Realizar una búsqueda de centros. Seleccionar una provincia dentro del mapa.

Postcondiciones: Se muestra la información de contacto (dirección, teléfono de contacto e email) de la tienda elegida por el usuario.

Flujo de eventos principal:

1. El usuario accede a la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario.
3. El usuario solicita el acceso a la consulta de centros.
4. El sistema accede, y muestra un mapa de España con todas sus provincias delimitadas.
5. El usuario elige una provincia y pincha sobre ella (corresponde al caso de uso Selección provincia).
6. El sistema muestra todas las tiendas ubicadas en esa provincia (con su dirección, teléfono y correo electrónico para poder contactar) o un mensaje informativo de que no todavía no existen tiendas en la provincia seleccionada.

Escenario: Subsistema tienda virtual.

7.- Caso de uso Selección provincia (RF-07):

Descripción: El usuario elige una provincia del mapa para ver si la empresa tiene centros abiertos en ella.

Objetivo: Conocer los datos de contacto de las tiendas de una determinada provincia

Actor: Usuario.

Precondiciones: Pasar el cursor por encima de una provincia del mapa y pinchar en ella.

Postcondiciones: Se mostrará un panel de texto con la información de contacto de los centros ubicados en esta provincia, o un mensaje informativo en el caso de no existir tiendas.

Flujo de eventos principal:

1. El usuario selecciona una provincia del mapa.
2. El sistema realiza una búsqueda y muestra al usuario todos los centros que existen en la provincia que ha elegido.
3. El sistema mostrará información de contacto del centro como su provincia, dirección, email y teléfono.

Flujo de eventos excepcional:

1. En caso de que la provincia que seleccione el usuario no disponga de ninguna tienda física, el sistema mostrará al usuario un mensaje de aviso.

Escenario: Subsistema tienda virtual.

8.- Caso de uso Identificarse (Login) (RF-08):

Descripción: El cliente puede logarse en la web de la aplicación introduciendo su login y password.

Objetivo: Permite al usuario identificarse en la aplicación para poder acceder a las opciones que le estén permitidas como cliente.

Actor: Cliente.

Precondiciones: El usuario debe de estar dado de alta en el sistema.

Postcondiciones: El sistema accede a las opciones de menú disponibles para los clientes.

Flujo de eventos principal:

1. El sistema solicita al usuario la introducción de su login y password.
2. El sistema verifica que los datos del cliente sean correctos.
3. El sistema muestra el menú para clientes.

Flujo de eventos alternativo:

1. Si el usuario no está registrado el sistema le permitirá el acceso al formulario de alta de cliente.

Flujo de eventos excepcional:

1. Si el cliente no es válido o ha introducido mal los datos el sistema emitirá un mensaje de error.
Escenario: Subsistema tienda virtual.

9.- Caso de uso Escribir mensaje (RF-09):

Descripción: El cliente puede escribir un mensaje en el foro acerca del tema y especie de ave que desee.

Objetivo: Este mensaje puede abrir un nuevo hilo de conversación en el foro o responder a alguno ya existente. Este caso de uso incluye el caso de uso Identificarse y Visualizar Mensajes.

Actor: Cliente.

Precondiciones: El cliente debe seleccionar una categoría y una especie de ave, para posteriormente solicitar al sistema la creación del mensaje.

A continuación rellenará el formulario con los datos del mensaje que quiere añadir y lo guardará en el sistema.

Postcondiciones: El sistema creará un nuevo mensaje almacenándolo en la base de datos y mostrando la nueva lista de mensajes actualizada.

Flujo de eventos principal:

1. El cliente accede a la opción del menú correspondiente al foro tras logarse en la web (corresponde al caso de uso Identificarse).
2. El cliente elige una categoría y especie del foro.
3. El sistema muestra al cliente el listado de mensajes (hilos de conversación) para esa categoría.
4. El cliente puede iniciar un nuevo hilo de conversación en esta categoría (para ello debe pulsar el botón crear mensaje) o responder a alguno de los ya existentes (debe pinchar sobre el mensaje que desea responder).
5. El sistema muestra al cliente el formulario de creación de mensajes nuevos.
6. El cliente escribe su mensaje y pide al sistema que lo guarde.
7. El sistema guarda el mensaje y muestra al cliente el listado de mensajes actualizado (corresponde al caso de uso Visualizar mensajes).

Flujo de eventos excepcional:

1. Si durante la creación del mensaje ocurriera algún error el sistema emitirá un mensaje avisando al cliente de lo ocurrido y no guardará el mensaje.

Escenario: Subsistema tienda virtual.

10.- Caso de uso Visualizar mensajes (RF-10):

Descripción: El cliente puede visualizar los mensajes existentes en cualquier tema del foro.

Objetivo: El sistema muestra los mensajes del tema que elija el cliente. Este caso de uso incluye el caso de uso Identificarse.

Actor: Cliente.

Precondiciones: El cliente debe seleccionar una categoría y especie.

Postcondiciones: El sistema mostrará la lista de mensajes existentes (en función de la categoría que haya elegido).

Flujo de eventos principal:

2. El cliente accede a la opción del menú correspondiente al foro tras logarse en la web (corresponde al caso de uso Identificarse).
3. El cliente elige una categoría y especie del foro.
4. El sistema muestra al cliente todos los mensajes existentes para esa categoría.

Escenario: Subsistema tienda virtual.

11.- Caso de uso Visualizar mensaje:

Descripción: El cliente puede visualizar el mensaje que desee de cualquier tema del foro.

Objetivo: El sistema visualiza el contenido del mensaje que el cliente haya escogido. Este caso de uso es una extensión del caso de uso Visualizar mensajes.

Actor: Cliente.

Precondiciones: El cliente debe seleccionar una categoría y especie para posteriormente escoger uno de los mensajes del listado.

Postcondiciones: El sistema muestra el contenido del mensaje elegido.

Flujo de eventos principal:

1. El cliente accede a la opción del menú correspondiente al foro tras logarse en la web (corresponde al caso de uso Identificarse).
2. El cliente elige una categoría y especie.
3. El sistema muestra al cliente el listado de mensajes para esa categoría.
4. El cliente elige el mensaje que quiera consultar.
5. El sistema muestra al cliente la información para ese mensaje.

Escenario: Subsistema tienda virtual.

12.- Caso de uso Eliminar mensaje (RF-11):

Descripción: Un administrador del sistema tiene la posibilidad de borrar cualquier mensaje que considere.

Objetivo: Eliminar del foro los mensajes que no se consideren aptos para el hilo del foro en el que se encuentren.

Actor: Administrador.

Precondiciones: El administrador elegirá una categoría y especie para posteriormente escoger los mensajes del listado que vaya a eliminar.

Postcondiciones: El sistema muestra la lista de mensajes actualizada.

Flujo de eventos principal:

1. El administrador accede a la opción del menú correspondiente al foro tras logarse en la web (corresponde al caso de uso Identificarse).
2. El administrador elige una categoría y especie.
3. El sistema muestra el listado de mensajes para esa categoría (corresponde al caso de uso Visualizar mensajes).
4. El administrador elige los mensajes que quiera borrar del foro.
5. El sistema muestra de nuevo el listado de mensajes actualizado.

Flujo de eventos excepcional:

1. El sistema mostrará un mensaje de error en caso de que se haya producido algún error al eliminar el mensaje del foro.

Escenario: Subsistema tienda virtual.

13.- Caso de uso Gestión centros (RF-12):

Descripción: Un administrador realizará todas las labores relacionadas con la gestión y mantenimiento de los centros que la empresa tiene abiertos.

Objetivo: Realizar el mantenimiento de todos los centros.

Actor: Administrador.

Precondiciones: El administrador debe acceder a la pantalla de gestión de centros.

Postcondiciones: El sistema actualizará en base de datos las modificaciones que el administrador realice tanto en los centros ya existentes como la creación de nuevos centros.

Flujo de eventos principal:

1. El administrador accede a la opción del menú correspondiente a la gestión de centros, tras logarse en la web (corresponde al caso de uso Identificarse).
2. El administrador puede realizar las siguientes acciones:
 - Modificación de los datos de contacto de cualquier centro dado de alta en el sistema.
 - Eliminación de un centro. Cuando cualquier tienda se cierre se deberá dar de baja en el sistema.
 - Anadir un nuevo centro, cuando se abra una tienda en cualquier provincia.
3. El sistema actualizará la base de datos con la información requerida por el administrador y mostrará un mensaje de éxito en caso de que el proceso haya sido satisfactorio.

Flujo de eventos excepcional:

1. El sistema mostrará un mensaje de error en caso de que se produzca un error en modificar/borrar o añadir información en la base de datos.

Escenario: Subsistema tienda virtual.

14.- Caso de uso Búsqueda centros (RF-13):

Descripción: El administrador tiene la posibilidad de realizar búsquedas de centros en el sistema.

Objetivo: Acceder de forma rápida a los centros que el administrador quiere gestionar. Este caso de uso extiende el caso de uso Gestión centros.

Actor: Administrador.

Precondiciones: El administrador debe acceder a la pantalla de gestión de centros y rellenar algún filtro de búsqueda como el código de centro y/o provincia en caso de que el usuario lo necesite.

Postcondiciones: El sistema muestra la lista de centros que cumple con los filtros de búsqueda.

Flujo de eventos principal:

1. El administrador rellena los filtros de búsqueda que considere (en caso de que lo desee) y pulsa en el botón de buscar.
2. El sistema muestra el listado de centros que cumplen con los requisitos de búsqueda iniciales.

Flujo de eventos excepcional:

1. Si los parámetros de búsqueda introducidos por el usuario no son correctos el sistema debe de generar un mensaje informando del error al usuario.

Escenario: Subsistema tienda virtual.

15.- Caso de uso Visualizar gráficos (RF-14):

Descripción: Un administrador tiene la posibilidad de obtener graficas de los resultados de las ventas de los productos de la empresa o las ventas realizadas por cada una de las tiendas.

Objetivo: Realizar el mantenimiento de todos los centros.

Actor: Administrador.

Precondiciones: El administrador debe acceder a la pantalla de visualización de gráficos.

Postcondiciones: Ninguna.

Flujo de eventos principal:

1. El administrador accede a la opción del menú correspondiente a la visualización de gráficos, tras logarse en la web (corresponde al caso de uso Identificarse).
2. El sistema muestra el formulario de gráficos para que el administrador rellene los datos de búsqueda que requiera según la información que necesite.

Escenario: Subsistema tienda virtual.

16.- Caso de uso Rellenar filtros búsqueda (RF-15):

Descripción: El administrador puede rellenar filtros de búsqueda.

Objetivo: Mostrar gráficos con la información que el administrador necesite en cada momento. Este caso de uso extiende el caso de uso Visualizar gráficos.

Actor: Administrador.

Precondiciones: El administrador debe acceder a la pantalla de visualización de gráficos e introducir algún filtro de búsqueda como familia de productos, tienda, o fechas.

Postcondiciones: El sistema muestra el gráfico acorde a la información solicitada por el usuario.

Flujo de eventos principal:

1. El administrador rellenará los filtros de búsqueda que considere y pulsa en el botón de visualizar producto o visualizar tienda.
2. El sistema muestra el gráfico con la información.

Flujo de eventos excepcional:

1. Si los parámetros de búsqueda introducidos por el usuario no son correctos el sistema debe de generar un mensaje informando del error al usuario.

Escenario: Subsistema tienda virtual.

4.2.- Diagrama casos de uso gestión del programa Mendel

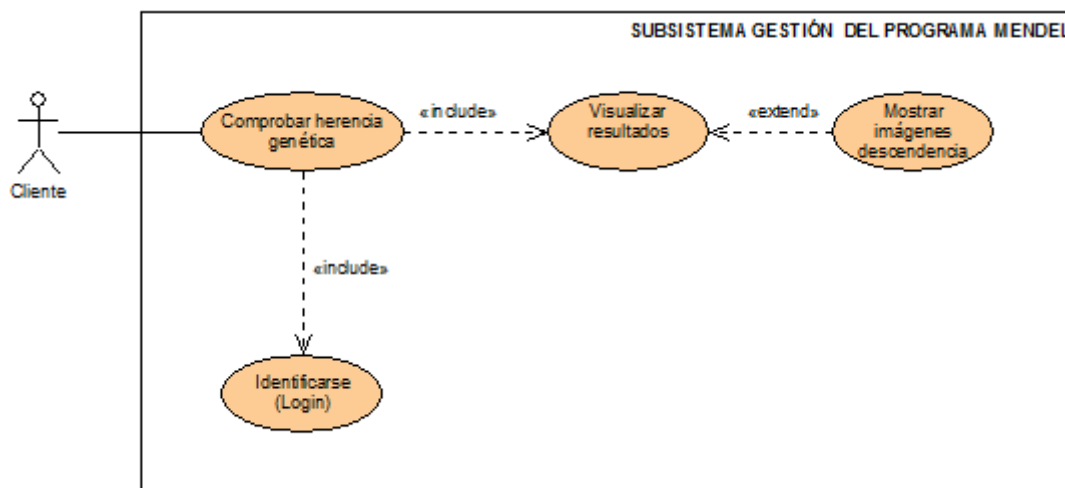


Figura7. Diagrama casos de uso Gestión del Programa Mendel

CASOS DE USO DEL SUBSISTEMA GESTION DEL PROGRAMA MENDEL

1.- Caso de uso Comprobar herencia genética (RF-16):

Descripción: La aplicación muestra un formulario que le permite al cliente introducir los datos de los pájaros macho y hembra de los cuales se quiere comprobar la posible descendencia genética que tendrían.

Objetivo: Este módulo permite al cliente calcular la herencia genética de los diamantes de gould a partir de los

fenotipos que el cliente elija para cada progenitor. Este caso de uso de uso incluye el caso de uso Identificarse y Visualizar resultados.

Actor: Cliente logado en la web de la aplicación.

Precondiciones: El cliente debe ser un usuario válido, rellenar el formulario con los fenotipos de los progenitores y solicitar al sistema que calcule la descendencia con esos datos.

Postcondiciones: El sistema calculará y mostrará la posible descendencia a partir de los datos proporcionados por el cliente.

Flujo de eventos principal:

1. El cliente accede a la opción del menú correspondiente al Programa Mendel tras logarse en la web (corresponde al caso de uso Identificarse).
2. El cliente introduce los fenotipos del macho y la hembra.
3. El sistema validará que los fenotipos introducidos por el usuario son válidos y calculará la posible descendencia.

Flujo de eventos excepcional:

1. Si el sistema detecta algún error durante el proceso de cálculo de la descendencia mostrará un mensaje indicándolo y volverá a la página principal del programa Mendel.

Escenario: Subsistema gestión del programa Mendel.

2.- Caso de uso Visualizar resultados (RF-17):

Descripción: Muestra los fenotipos de las crías.

Objetivo: Permite al cliente visualizar los resultados tras el cálculo de la descendencia.

Actor: Cliente.

Precondiciones: Haber realizado el cálculo genético a partir de los fenotipos de los padres.

Postcondiciones: Ninguna.

Flujo de eventos principal:

1. Tras haber realizado el cálculo de probabilidades de descendencia genética, el sistema visualizará la información en forma de listados para las distintas partes del cuerpo del pájaro (cabeza, pecho, cuerpo), y además se incluirá una tabla con todos los datos calculados.

Escenario: Subsistema gestión del programa Mendel.

3.- Caso de uso Mostrar imágenes descendencia (RF-18):

Descripción: Visualización de las crías mediante imágenes.

Objetivo: Permite al cliente visualizar las imágenes de las posibles crías obtenidas tras el cálculo de la descendencia. Este caso de uso es una extensión del caso de uso Visualizar resultados.

Actor: Cliente.

Precondiciones: El cliente debe elegir la cría que desee visualizar.

Postcondiciones: El sistema abrirá una pantalla nueva con la imagen y los datos de la cría que se ha escogido.

Flujo de eventos principal:

1. El cliente selecciona del listado de resultados la cría que desea visualizar.
2. El sistema busca la imagen y la muestra en una nueva pantalla junto con los datos genéticos y sexo de la cría que el cliente ha escogido.

Flujo de eventos excepcional:

1. Si el sistema detecta algún error al cargar las imágenes, emitirá un mensaje advirtiéndolo al cliente y volverá a la pantalla de resultados.

Escenario: Subsistema gestión del programa Mendel.

4.3.- Diagrama casos de uso gestión de usuario

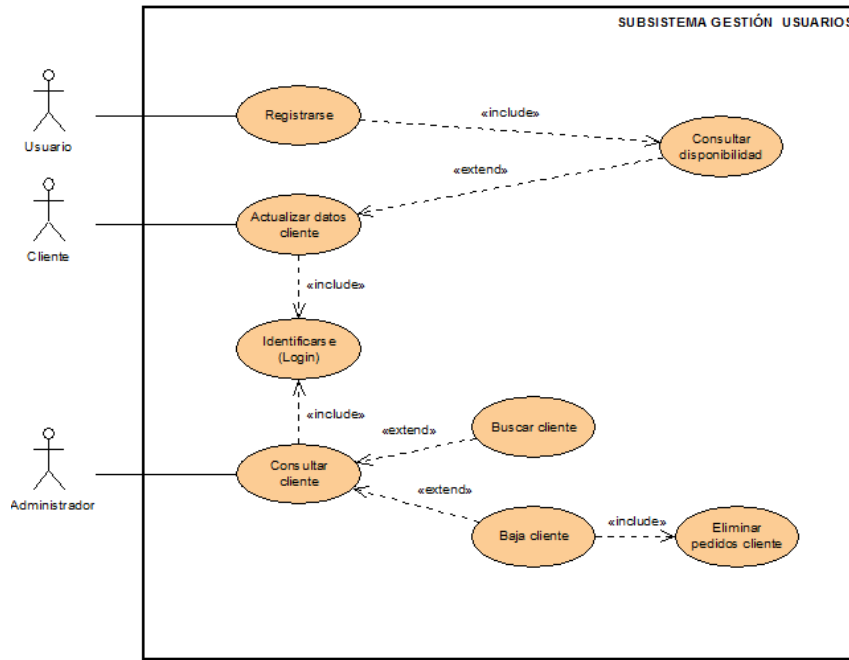


Figura8. Diagrama casos de uso Gestión Usuarios

CASOS DE USO DEL SUBSISTEMA GESTION DE USUARIOS

1.- Caso de uso Registrarse:

Descripción: Cualquier usuario que acceda a la web de la pajarería puede registrarse rellenando un formulario de alta con sus datos personales.

Objetivo: Una vez que el usuario se identifique en la web, podrá acceder a funcionalidades exclusivas de los usuarios registrados en la aplicación. Este caso de uso incluye el caso de uso Consultar disponibilidad, es decir cuando un usuario rellena el formulario de alta, el sistema comprueba que el login que ha escogido el usuario no está asociado a otro cliente ya guardado en el sistema.

Actor: Usuario no registrado.

Precondiciones: Seleccionar el enlace que accede al formulario para registrarse en el sistema.

Postcondiciones: Se da de alta al usuario en la web de la pajarería.

Flujo de eventos principal:

1. El usuario accede a la web de la aplicación.
2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario.
3. El usuario solicita el acceso al registro en la aplicación.
4. El sistema accede, y muestra el formulario que el usuario deberá rellenar para darse de alta en la web.
5. El usuario rellenará el formulario de alta.
6. El sistema recoge el formulario rellenado y guarda la información en la base de datos. A partir de ese momento el usuario pasará a ser un cliente de la aplicación y por tanto podrá acceder a la funcionalidad asociada a este perfil.

Flujo de eventos excepcional:

1. Si algún campo del formulario no se ha rellenado correctamente el sistema impedirá el alta y mostrará al usuario un mensaje de error indicándolo.
2. En caso de que se produzca un error al tratar de guardar los datos del usuario, el sistema también mostrará un mensaje de error indicándolo.

Escenario: Subsistema gestión Usuarios.

2.- Caso de uso Consultar disponibilidad (RF-19):

Descripción: Un usuario puede modificar sus datos de identificación en el sistema para lo cual puede comprobar si el “login” que desea está disponible.

Objetivo: Permite al cliente asegurarse de que el login que desea no está siendo usado por cualquier otro cliente de la web. Este caso de uso extiende el caso de uso Actualizar datos cliente.

Actor: Cliente.

Precondiciones: El cliente debe introducir un login.

Postcondiciones: El sistema verificará que el login no esté siendo usado por otro cliente mostrando un mensaje con el resultado de la consulta.

Flujo de eventos principal:

1. El cliente accede a la opción del menú correspondiente a la Gestión de datos de Usuario tras logarse en la web (corresponde al caso de uso Identificarse).
2. El sistema muestra el formulario de modificación de los datos de usuario.
3. El cliente introduce un nuevo login y consulta su disponibilidad en el sistema al pichar el botón de comprobar disponibilidad.
4. El sistema comprueba que ese nuevo login introducido por el cliente no esté siendo utilizado por otro usuario del sistema. En caso de estar disponible, mostrará un mensaje al cliente advirtiéndole de esto.

Flujo de eventos excepcional:

1. Si el nuevo login elegido por el usuario no está disponible, el sistema emitirá un mensaje alertando al cliente de esto y pidiéndole la introducción de uno nuevo.

Escenario: Subsistema gestión Usuarios.

3.- Caso de uso Actualizar datos cliente (RF-20):

Descripción: Un usuario puede modificar sus datos de identificación en el sistema para lo cual puede comprobar si el "login" que desea está disponible.

Objetivo: Permite al cliente modificar sus datos personales. Este caso de uso incluye el caso de uso Identificarse.

Actor: Cliente.

Precondiciones: El cliente debe modificar los datos del formulario que desee y solicitar al sistema que los guarde.

Postcondiciones: El sistema actualiza la información en base de datos mostrando un mensaje de éxito de la operación.

Flujo de eventos principal:

1. El cliente accede a la opción del menú correspondiente a la Gestión de datos de Usuario tras logarse en la web (corresponde al caso de uso Identificarse).
2. El sistema muestra el formulario de modificación de los datos de usuario.
3. El cliente puede consultar, modificar o borrar (darse de baja) sus datos de usuario en el sistema.
4. En caso de que el cliente solicite al sistema guardar modificaciones sobre sus datos de usuario, éste validará que los nuevos datos introducidos por el cliente son correctos y en caso de serlo los guardará en el sistema.
5. En caso de que lo que quiera el cliente sea darse de baja, el sistema pide confirmación por parte del cliente para continuar con el proceso de baja. Si el cliente confirma su decisión, el sistema eliminará sus datos de alta y toda la información referente a él. A continuación la aplicación mostrará al cliente el menú de usuario ya que no está ya dado de alta en el sistema.

Flujo de eventos excepcional:

1. Si el cliente tiene pedidos pendientes, el sistema emitirá un mensaje advirtiéndole de esto e impedirá que el cliente pueda darse de baja en la web.
2. En caso de que se produzca algún fallo al actualizar la información en la base de datos se mostrará un mensaje de error y no se guardarán las modificaciones.

Escenario: Subsistema gestión Usuarios.

4.- Caso de uso Consultar cliente (RF-21):

Descripción: Cualquier usuario con privilegios de administrador puede ver los clientes que han realizado algún pedido a través de la web de la pajarería.

Objetivo: Identificar a los clientes actuales (entendiendo como cliente aquel usuario identificado en el sistema que ha realizado como mínimo un pedido). Este caso de uso incluye el caso de uso Identificarse.

Actor: Administrador.

Precondición: El administrador debe realizar una búsqueda y seleccionar un cliente con uno de sus pedidos del listado mostrado tras realizar dicha búsqueda.

Postcondición: El sistema muestra una pantalla con un formulario con los datos de alta del usuario y el estado del pedido que se ha seleccionado.

Flujo de eventos principal:

1. El cliente accede a la opción del menú correspondiente a la Gestión de Clientes tras logarse en la web (corresponde al caso de uso Identificarse).
2. El sistema muestra el formulario de búsqueda de clientes.
3. Tras realizar una búsqueda, el administrador elige uno de los clientes con un pedido que ha realizado.
4. El sistema muestra un formulario con los datos de contacto del usuario y el estado del pedido seleccionado.
5. El administrador tiene la posibilidad de modificar el estado de este pedido.

Flujo de eventos excepcional:

1. Si ocurriera algún error durante la actualización del estado del pedido del cliente se emitirá un mensaje advirtiéndolo al administrador

Escenario: Subsistema gestión Usuarios.

5.- Caso de uso Buscar cliente (RF-22):

Descripción: Poder encontrar clientes de la pajarería de manera fácil y rápida, mostrando únicamente aquellos que le interesan al administrador según sus necesidades.

Objetivo: Permite al administrador realizar búsquedas de clientes a partir de los parámetros de búsqueda introducidos. Este caso de uso es una extensión del caso de uso Consultar cliente.

Actor: Administrador.

Precondición: Introducir los filtros de búsqueda en caso de que los requiera el administrador.

Postcondición: El sistema muestra un listado con los clientes y sus pedidos, que se ajustan a los parámetros de búsqueda introducidos.

Flujo de eventos principal:

1. El administrador rellena los filtros de búsqueda que considere (en caso de que lo desee) y pulsa en el botón de buscar.
2. El sistema muestra el listado con los clientes que cumplen con los requisitos de búsqueda iniciales.

Flujo de eventos excepcional:

1. Si ocurriera algún error durante el proceso de búsqueda se emitirá un mensaje advirtiéndolo al administrador.
2. Si la búsqueda no produce resultados se emitirá un mensaje advirtiéndolo.
3. Si los parámetros de búsqueda introducidos por el usuario no son correctos el sistema debe de generar un mensaje informando del error al usuario.

Escenario: Subsistema gestión Usuarios.

6.- Caso de uso Baja cliente (RF-23):

Descripción: Un administrador puede eliminar los clientes que quiera del sistema.

Objetivo: Eliminar los clientes que el administrador considere. Este caso de uso es una extensión del caso de uso Consultar cliente. Este caso de uso incluye el caso de uso Eliminar pedidos cliente.

Actor: Administrador.

Precondición: Tras realizar una búsqueda de clientes, el administrador debe seleccionar aquel que desea dar de baja.

Postcondición: El sistema eliminará al cliente de la base de datos.

Flujo de eventos principal:

1. El administrador elegirá un cliente de la lista obtenida tras realizar una búsqueda de clientes (corresponde al caso de uso buscar cliente).
2. El sistema muestra los datos asociados a ese cliente.
3. El administrador pincha el botón dar de baja a ese cliente.
4. El sistema pide confirmación por parte del administrador
5. El administrador confirma la operación.
6. El sistema elimina al cliente de la base de datos y toda la información relacionada con él

Flujo de eventos alternativo:

1. Si el administrador cancela la operación de baja de un cliente el sistema no lo eliminará.

Flujo de eventos excepcional:

1. Si ocurriera algún error durante el proceso de baja de clientes se emitirá un mensaje advirtiéndolo al administrador.

Escenario: Subsistema gestión Usuarios.

7.- Caso de uso Eliminar pedidos cliente (RF-24):

Descripción: Cuando se elimine un cliente del sistema también se borrará toda la información asociada a él, como por ejemplo los pedidos que ha realizado.

Objetivo: Eliminar por completo el rastro del cliente que se ha eliminado.

Actor: Administrador.

Precondición: El administrador debe borrar un cliente.

Postcondición: El sistema eliminará de la base de datos los pedidos que haya realizado el cliente que se esté borrando.

Flujo de eventos principal:

1. Tras seleccionar un cliente y pulsar el botón dar de baja, se borrarán todos los pedidos que ese cliente ha realizado.

Flujo de eventos excepcional:

1. Si ocurriera algún error durante la eliminación de la información del sistema se emitirá un mensaje de error y no se borrará la información.

Escenario: Subsistema gestión Usuarios.

4.4.- Diagrama casos de uso gestión de productos y proveedores

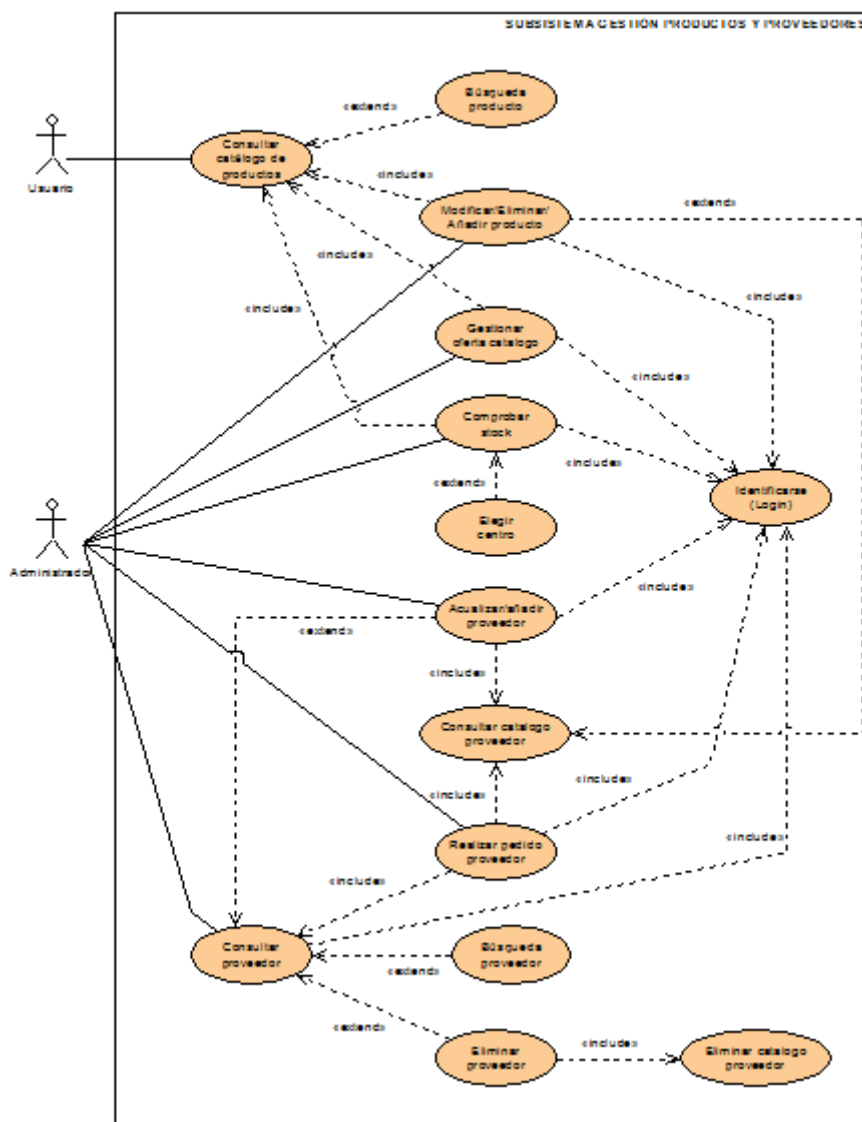


Figura9. Diagrama casos de uso Gestión Productos y Proveedores

CASOS DE USO DEL SUBSISTEMA GESTION DE PRODUCTOS Y PROVEEDORES

1.- Caso de uso Consultar catalogo de productos (RF-25):

Descripción: Cualquier usuario de la aplicación puede consultar el catálogo de productos que se venden en la empresa.

Objetivo: Permite al usuario conocer los productos que se venden. Así podrá saber si el producto que busca está disponible en ese momento o no queda stock. De cada producto se mostrará su código, nombre, precio unitario y stock.

Actor: Usuario; no necesita ningún privilegio especial para poder acceder al catálogo de productos.

Precondiciones: Realizar una búsqueda de productos.

Postcondiciones: Se muestra el listado para la consulta de productos de la tienda con su código, nombre, precio unitario y stock.

Flujo de eventos principal:

1. El usuario accede a la web de la aplicación.

2. El sistema accede y muestra un menú lateral que contendrá todos los servicios a los que tiene acceso cualquier usuario.
3. El usuario solicita el acceso al catálogo de productos.
4. El sistema accede, y muestra el formulario de búsqueda.
5. El usuario realiza la búsqueda.
6. El sistema recoge la información y muestra un listado de productos de acuerdo con los parámetros de búsqueda que ha introducido el usuario.

Escenario: Subsistema gestión Productos y Proveedores.

2.- Caso de uso Búsqueda producto (RF-26):

Descripción: El usuario puede buscar un producto concreto mediante su código (en caso de que lo conozca), o bien pulsando la lupa que abrirá un “popup” donde podrá seleccionar un producto del listado.

Objetivo: Encontrar de manera rápida el producto que se quiere consultar. Este caso de uso es una extensión del caso de uso Consultar catálogo de productos.

Actor: Usuario.

Precondiciones: Pulsar la lupa de producto para realizar la búsqueda o introducir un código de producto que exista en base de datos.

Postcondiciones: El producto con sus características se cargará en el listado del catálogo de productos.

Flujo de eventos principal:

1. El administrador rellena los filtros de búsqueda que considere (en caso de que lo desee) y pulsa en el botón de buscar. El sistema permite al usuario buscar un producto por código o familia o nombre (en la lupa).
2. Una vez realizada la búsqueda por parte del usuario el sistema visualizará el listado de productos que se ajusten a los filtros de búsqueda.

Flujo de eventos excepcional:

1. Si los parámetros de búsqueda suministrados por el usuario no son correctos el sistema debe de generar un mensaje informando del error al usuario.
2. Si ocurriera algún error durante el proceso de búsqueda se emitirá un mensaje advirtiéndolo al administrador.
3. Si la búsqueda no produce resultados se emitirá un mensaje advirtiéndolo.

Escenario: Subsistema gestión Productos y Proveedores.

3.- Caso de uso Modificar/Eliminar/Añadir producto (RF-27):

Descripción: El administrador puede modificar los datos de cualquier producto del catálogo y ampliar o reducir dicho catálogo dando de alta o baja los productos que quiera.

Objetivo: Modificar el catálogo de productos de la empresa para adecuarlo a sus necesidades en cada momento. Este caso de uso incluye el caso de uso Consultar catálogo de productos e Identificarse.

Actor: Administrador.

Precondición: El administrador debe acceder a la opción de gestión de productos.

Postcondición: El sistema actualizará la información que solicite al administrador en base de datos.

Flujo de eventos principal:

1. El administrador accede a la opción del menú correspondiente a la Gestión de Productos tras logarse en la web (corresponde al caso de uso Identificarse).
2. El sistema muestra el formulario de gestión de productos.
3. El administrador puede solicitar al sistema añadir nuevos productos (realizar pedido a un proveedor), eliminar alguno de los existentes o modificar los datos de los productos del catálogo.
4. En caso de que el administrador solicite el alta de un producto, debe introducir los datos del proveedor al cual va a realizar el pedido del producto y el producto que desea solicitar. El sistema guardará los datos del nuevo producto en el catálogo.
5. Cuando el administrador desee borrar o modificar algún producto, antes debe realizar una búsqueda de productos. Del listado obtenido el administrador puede seleccionar los productos que desee a la vez y eliminarlos del sistema. Si lo que quiere es modificar las características de alguno de los productos bastará con que pinche sobre su nombre. Entonces la aplicación mostrará un formulario con los datos del producto para que el administrador modifique el que quiera. A continuación el administrador solicitará al sistema que guarde los cambios realizados.

Flujo de eventos excepcional:

1. Si ocurriera algún problema durante el proceso de actualización de la información en la base de datos, el sistema mostrará un mensaje de error advirtiéndolo al usuario.
2. Si el administrador intentara hacer un pedido de un producto que no suministrara el proveedor que ha escogido, el sistema mostrará un mensaje de aviso e impedirá continuar con el alta del producto.

Escenario: Subsistema gestión Productos y Proveedores.

4.- Caso de uso Gestionar oferta catálogo (RF-28):

Descripción: La empresa dispondrá de una serie de productos ofertados durante un determinado período de tiempo que servirán de reclamo publicitario.

Objetivo: El objetivo del catalogo de ofertas es generar un folleto virtual en el que se muestren una serie de productos a precios más bajos de los habituales en un periodo de tiempo determinado. El administrador puede gestionar el catalogo de ofertas realizando para ello operaciones como alta de nuevas ofertas, eliminar ofertas antiguas o descatalogadas o por ultimo modificar el precio de ofertas ya existentes en el catalogo. Una oferta podrá localizarse por el nombre del producto ofertado. Este caso de uso incluye el caso de uso Consultar catálogo de productos e Identificarse.

Actor: Administrador.

Precondición: El administrador debe acceder a la opción de generar catálogo. En caso de que se quiera modificar o eliminar ofertas, el catálogo debe disponer de al menos una oferta para que el administrador pueda seleccionarla.

Postcondición: El sistema actualizará la información el catalogo de ofertas.

Flujo de eventos principal:

1. El administrador accede a la opción del menú correspondiente a Generar catálogo tras logarse en la web (corresponde al caso de uso Identificarse).
2. El sistema muestra un listado con todos los productos ofertados por la empresa en ese momento.
3. El administrador puede solicitar al sistema el alta, modificación o borrado de ofertas del catálogo.
4. En caso de que lo que se solicite añadir una nueva oferta el administrador debe seleccionar el botón de añadir oferta. A continuación el sistema mostrará el formulario de alta de una oferta en el que el administrador deberá escoger el producto que desea ofertar y el precio al cual se va a vender. Posteriormente el sistema guardará la oferta en la base de datos y mostrará el listado con todas las ofertas actualizado.
5. Cuando el administrador quiera dar de baja ofertas del catálogo, debe seleccionar aquellas ofertas que vaya a dar de baja. Entonces el sistema borrará las ofertas de la base de datos y mostrará de nuevo el listado de ofertas actualizado.
6. Por último en caso de que el administrador quiera modificar una oferta, deberá seleccionar una del listado y solicita al sistema su modificación. El sistema mostrará un formulario relleno con los datos de la oferta. El administrador podrá modificar el precio del producto. Entonces el sistema actualizará el precio mostrando de nuevo el listado de ofertas actualizado.

Flujo de eventos excepcional:

1. En caso de que se está añadiendo la oferta de un producto ya existente en el catalogo de ofertas, la aplicación mostrará un mensaje de advertencia e impedirá que se guarde en el sistema la oferta.
2. En caso de que algún campo del formulario de ofertas (tanto el de alta como el de modificación) sea incorrecto, el sistema también generará un mensaje de error.
3. Si ocurriera algún problema durante el proceso de actualización de la información en la base de datos, el sistema mostrará un mensaje de error advirtiéndolo al usuario.

Escenario: Subsistema gestión Productos y Proveedores.

5.- Caso de uso Comprobar stock (RF-29):

Descripción: El administrador podrá consultar la cantidad que dispone de cada producto.

Objetivo: El administrador puede consultar en cualquier momento la cantidad de cada producto que hay disponible en el almacén o en tiendas concretas. También puede realizar búsquedas de productos para conocer únicamente las unidades de un producto determinado en una tienda o en el almacén. Este caso de uso incluye el caso de uso Consultar catálogo de productos e Identificarse.

Actor: Administrador.

Precondición: Realizar búsquedas de productos por productos o tiendas correctos y que produzcan resultados.

Postcondición: Se mostrará un listado con el stock de los productos buscados (en función de los filtros de búsqueda introducidos) mostrando para cada uno de ellos su código, nombre y stock.

Flujo de eventos principal:

1. El administrador accede a la opción del menú correspondiente a Consulta de Stock tras logarse en la web (corresponde al caso de uso Identificarse).
2. El sistema accede y muestra el formulario de consulta de stock.
3. El administrador podrá rellenar los filtros de búsqueda que necesite (tienda y/ producto) en función de los datos que desee consultar.
4. El sistema recoge los filtros de búsqueda introducidos por el administrador y devuelve un listado de productos con su código, nombre y número de unidades disponible.

Flujo de eventos excepcional:

1. El sistema mostrará un mensaje de error en caso de que los filtros de búsqueda sean incorrectos.
2. Si los filtros de búsqueda no producen ningún resultado, el sistema mostrará un mensaje de error indicándolo.

3. Si ocurriera algún problema al realizar la búsqueda de la información en la base de datos, el sistema mostrará un mensaje de error advirtiendo de ello al usuario.

Escenario: Subsistema gestión Productos y Proveedores.

6.- Caso de uso Elegir centro (RF-30):

Descripción: El administrador puede consultar el stock de los productos de cualquier tienda asociada a la empresa.

Objetivo: Filtrar las búsquedas de stock de productos en función de una tienda. Este caso de uso es una extensión del caso de uso Comprobar stock.

Actor: Administrador.

Precondición: Introducir una tienda de la cual se desea conocer el stock de productos que dispone.

Postcondición: Se mostrará un listado con el stock de los productos existentes en el centro solicitado (en función de los filtros de búsqueda introducidos) mostrando para cada uno de ellos su código, nombre y stock.

Flujo de eventos principal:

1. El administrador rellena el filtro de búsqueda correspondiente a la tienda (en caso de que lo desee) y pulsa en el botón de buscar.
2. El sistema muestra el listado de productos que se venden en el centro solicitado con el stock disponible de cada uno de ellos.

Flujo de eventos excepcional.

1. El sistema mostrará un mensaje de error en caso de que el centro introducido no exista en el sistema.
2. Si los filtros de búsqueda no producen ningún resultado, el sistema mostrará un mensaje de error indicándolo.
3. Si ocurriera algún problema al realizar la búsqueda de la información en la base de datos, el sistema mostrará un mensaje de error advirtiendo de ello al usuario.

Escenario: Subsistema gestión Productos y Proveedores.

7.- Caso de uso Actualizar/añadir proveedor (RF-31):

Descripción: Una de las labores principales de un administrador del sistema es la gestión de los proveedores que suministran los productos a las tiendas asociadas a la empresa.

Objetivo: Permite al administrador dar de alta a un nuevo proveedor o la modificación de cualquiera de los existentes. La gestión de cualquier proveedor incluye también la modificación del catálogo de productos que ofrece cada uno de los proveedores. Este caso de uso incluye el caso de uso Identificarse y Consultar catálogo proveedor.

Actor: Administrador.

Precondición: El administrador debe completar el formulario del proveedor correctamente tanto en el caso de altas nuevas como el de las modificaciones. En caso de que lo que se esté realizando es la modificación de un proveedor deberá elegir el proveedor que desea modificar.

Postcondición: El sistema guardará la información solicitada por el administrador en base de datos.

Flujo de eventos principal:

1. El administrador accede a la opción del menú correspondiente a la Gestión de Proveedores tras logarse en la web (corresponde al caso de uso Identificarse).
2. El sistema accede y muestra el formulario de búsqueda de proveedores.
3. En caso de que el administrador solicite dar de alta un nuevo proveedor, deberá pinchar el botón de añadir. El sistema mostrará al administrador el formulario de alta de los datos del proveedor. Una vez completado dicho formulario el sistema permite al administrador asociar productos al proveedor que está dando de alta. Esos productos serán aquellos de los que el proveedor disponga para suministrar a nuestra empresa. Una vez que el administrador ha añadido todos los productos de los que dispone ese proveedor (puede añadir o eliminar los que considere), el administrador solicitará al sistema que guarde el nuevo proveedor.
4. Si el administrador quiere modificar un proveedor existente (o bien sus datos personales o modificar su catálogo de productos), primero debe realizar una búsqueda de proveedores. El sistema muestra el listado de proveedores que se ajustan a los filtros de búsqueda. El administrador elige el proveedor que va a modificar. A continuación el sistema muestra el formulario con los datos de contacto del proveedor para que el administrador modifique los que considere. Una vez realizados los cambios se pulsa el botón continuar. Entonces se mostrará una pantalla con el listado del catálogo de productos asociado al proveedor que estamos modificando. Entonces el administrador tiene la posibilidad de dar de alta o eliminar los productos del catálogo del proveedor. Entonces el sistema guardará los cambios en el sistema.

Flujo de eventos excepcional:

1. Si ocurriera algún error durante el proceso de actualización del sistema, se emitirá un mensaje advirtiendo de ello al administrador.

2. El sistema mostrará un mensaje de error en caso de que el proveedor que se esté buscando no exista en el sistema.
3. Si los filtros de búsqueda no producen ningún resultado, el sistema mostrará un mensaje de error indicándolo.
4. Si ocurriera algún problema al realizar la búsqueda de la información en la base de datos, el sistema mostrará un mensaje de error advirtiéndolo al usuario.

Escenario: Subsistema gestión Productos y Proveedores.

8.- Caso de uso Consultar catálogo proveedor (RF-32):

Descripción: Cada uno de los proveedores de la empresa tendrá su propio catálogo de productos. Estos catálogos pueden ser consultados en cualquier momento por un administrador del sistema.

Objetivo: Conocer los productos que vende cada proveedor. Este caso de uso es una extensión del caso de uso Modificar/Añadir/Eliminar producto.

Actor: Administrador.

Precondición: El administrador debe realizar una búsqueda del proveedor o proveedores de los que desea visualizar el catálogo.

Postcondición: El sistema muestra el catálogo de productos del proveedor seleccionado.

Flujo de eventos principal:

1. El administrador accede a la opción del menú correspondiente a la Gestión de Proveedores tras logarse en la web (corresponde al caso de uso Identificarse).
2. El sistema accede y muestra el formulario de búsqueda de proveedores.
3. El administrador debe realizar una búsqueda de proveedor (corresponde al caso de uso Búsqueda proveedor).
4. El sistema muestra un listado con los proveedores que se ajustan a los parámetros de búsqueda solicitados.
5. El administrador debe seleccionar el enlace de productos del proveedor que quiera.
6. El sistema abrirá una nueva ventana con el listado de productos que vende el proveedor.

Escenario: Subsistema gestión Productos y Proveedores.

9.- Caso de uso Realizar pedido proveedor (RF-33):

Descripción: En cualquier momento se pueden realizar pedidos de los productos que se necesiten en la empresa a los proveedores que los suministran.

Objetivo: Reponer el stock de los productos que se hayan agotado o queden pocas unidades. Este caso de uso incluye el caso de uso Consultar catálogo proveedor, Consultar proveedor e Identificarse.

Actor: Administrador.

Precondición: Los datos introducidos por el usuario a la hora de realizar un pedido (el producto y proveedor elegido) deben existir en base de datos y el número de unidades elegido por el administrador debe ser mayor de 0) deben ser válidos. Además el administrador debe aceptar la factura generada por el sistema para poder finalizar el pedido de productos a proveedores.

Postcondición: El sistema actualizará el stock del almacén con los productos adquiridos en el pedido.

Flujo de eventos principal:

1. El administrador accede a la opción del menú correspondiente a la Gestión de Productos tras logarse en la web (corresponde al caso de uso Identificarse).
2. El administrador debe pulsar el botón de realizar pedido.
3. En ese momento la aplicación mostrará el formulario de pedido.
4. El administrador debe elegir el proveedor al que realizará el pedido y los productos que desea pedir y la cantidad de cada uno de ellos para ir añadiéndolos a la lista del pedido.
5. El administrador podrá en cualquier momento eliminar los productos de esta lista antes de continuar con el proceso de compra. Una vez añadidos todos los productos necesarios, el administrador pulsará el botón Generar factura de su pedido.
6. El sistema calcula la factura asociada al producto y visualizará el reporte final del pedido y en ese punto el administrador debe aceptar la factura para que sea efectivo.
7. Finalmente el sistema actualiza el stock de los productos pedidos en el almacén.
8. El administrador tiene la posibilidad de imprimir los datos de la factura.

Flujo de eventos excepcional:

1. Si ocurriera algún error al realizar el pedido se emitirá un mensaje advirtiéndolo al administrador.
2. Si el administrador no acepta la factura, el pedido no se habrá realizado con éxito y no será guardado en base de datos.

Escenario: Subsistema gestión Productos y Proveedores.

10.- Caso de uso Consultar proveedor (RF-34):

Descripción: Cada administrador del sistema puede visualizar los datos de los proveedores que están asociados a la empresa.

Objetivo: Consultar los datos de un proveedor y su catálogo de productos. Este caso de uso incluye el caso de uso Identificarse.

Actor: Administrador.

Precondición: El administrador debe realizar una búsqueda de proveedores y seleccionar aquel que desea consultar.

Postcondición: El sistema muestra los datos de alta del proveedor y su catálogo de productos si el administrador pincha en los productos de dicho proveedor.

Flujo de eventos principal:

1. El administrador accede a la opción del menú correspondiente a la Gestión de Proveedores tras logarse en la web (corresponde al caso de uso Identificarse).
2. El administrador debe realizar una búsqueda de proveedores (corresponde al caso de uso Búsqueda proveedor).
3. El administrador elige el proveedor que quiera consultar, de la lista de proveedores obtenida tras el proceso de búsqueda y solicita al sistema que muestre la información de este.
4. El sistema muestra al administrador los datos del proveedor.
5. El administrador podrá también consultar la lista de productos asociada al proveedor para lo cual debe pinchar el enlace de productos asociado al proveedor que quiera. Entonces la aplicación mostrará un popup con el listado de productos ofertados por ese proveedor.

Escenario: Subsistema gestión Productos y Proveedores.

11.- Caso de uso Búsqueda proveedor (RF-35):

Descripción: Un proveedor puede localizarse tras realizar una búsqueda en el sistema.

Objetivo: Permite al administrador buscar proveedores a partir de los parámetros de búsqueda introducidos. Este caso de uso es una extensión del caso de uso Consultar proveedor.

Actor: Administrador.

Precondición: Los filtros de búsqueda introducidos deben ser válidos.

Postcondición: El sistema mostrará un listado con los proveedores que se ajustan a los parámetros de búsqueda introducidos en el formulario.

Flujo de eventos principal:

1. El administrador rellena los filtros de búsqueda que quiera y solicita al sistema que realice la búsqueda a partir de los parámetros introducidos en la etapa anterior.
2. El sistema realiza la búsqueda y muestra los resultados en forma de listado por pantalla.

Flujo de eventos excepcional:

1. Si ocurriera algún error durante el proceso de búsqueda se emitirá un mensaje advirtiendo de ello al administrador.
2. Si la búsqueda no produce resultados se emitirá un mensaje advirtiendo de ello al administrador.

Escenario: Subsistema gestión Productos y Proveedores.

12.- Caso de uso Eliminar proveedor (RF-36):

Descripción: Proceso por el cual un administrador tiene la posibilidad de dar de baja a un proveedor cuando ya no se solicitan sus productos.

Objetivo: Permite al administrador dar de baja a un proveedor. Este caso de uso es una extensión del caso de uso Consultar proveedor. Además este caso de uso incluye el caso de uso Eliminar catálogo proveedor.

Actor: Administrador.

Precondición: El administrador debe seleccionar el proveedor que desea dar de baja. El administrador deberá confirmar la operación.

Postcondición: El sistema eliminará al proveedor y su catálogo de productos de la base de datos.

Flujo de eventos principal:

1. El administrador accede a la opción del menú correspondiente a la Gestión de Proveedores tras logarse en la web (corresponde al caso de uso Identificarse).
2. El administrador debe realizar una búsqueda de proveedores (corresponde al caso de uso Búsqueda proveedor).
3. El administrador elige el proveedor que quiera eliminar del sistema.
4. El sistema pide confirmación por parte del administrador antes de darlo de baja.
5. El administrador confirma la acción.
6. El sistema ejecuta la acción solicitada por el usuario eliminado toda la información referente al proveedor seleccionado (datos de contacto y catálogo).

Flujo de eventos excepcional:

1. Si ocurriera algún error durante el proceso de baja de proveedores se emitirá un mensaje advirtiendo de ello al administrador.

Escenario: Subsistema gestión Productos y Proveedores.

13.- Caso de uso Eliminar catálogo proveedor (RF-37):

Descripción: Tras eliminar un proveedor automáticamente se borra también el catálogo de productos asociado al él.

Objetivo: Eliminar información innecesaria en el sistema, ya que no tiene utilidad tener un catálogo de productos de un proveedor que ya no tiene relación con la empresa.

Actor: Administrador.

Precondición: El administrador debe seleccionar el proveedor a dar de baja.

Postcondición: El sistema eliminará el catálogo del proveedor en base de datos.

Flujo de eventos principal:

1. Cuando un proveedor se da de baja (corresponde al caso de uso Eliminar proveedor), en el sistema automáticamente se dará de baja el catálogo de sus productos.

Escenario: Subsistema gestión Productos y Proveedores.

4.5.- Diagrama casos de uso compra on-line

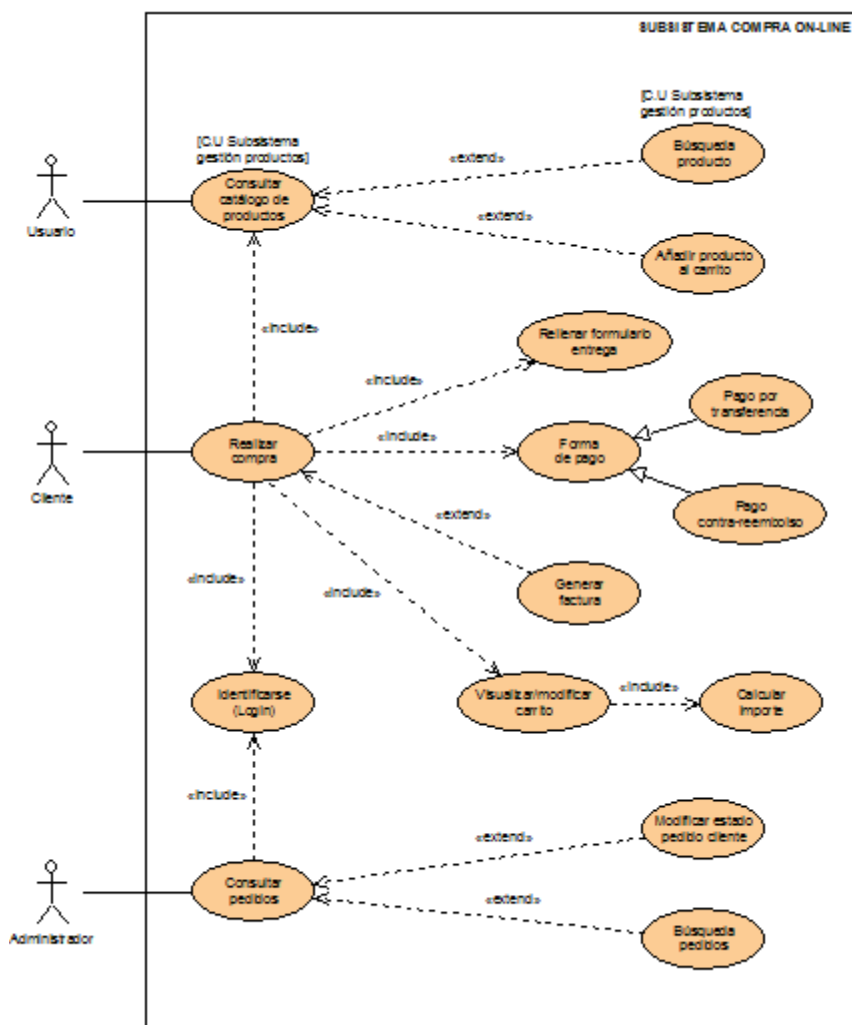


Figura9. Diagrama casos de uso Compra On-Line

CASOS DE USO DEL SUBSISTEMA COMPRA ON-LINE

1.- Caso de uso Consultar catálogo de productos: vease caso de uso Consultar catálogo de productos del subsistema Gestión de productos y proveedores.

Escenario: Subsistema Compra on-line.

2.- Caso de uso Búsqueda producto: vease caso de uso Búsqueda producto del subsistema Gestión de productos y proveedores.

Escenario: Subsistema Compra on-line.

3.- Caso de uso Añadir producto al carrito (RF-38):

Descripción: Cuando un usuario quiere realizar un pedido debe elegir los productos que desea adquirir.

Objetivo: El usuario podrá añadir los productos que quiera a su carrito de la compra. Este caso de uso es una extensión del caso de uso Consultar catálogo de productos.

Actor: Cliente.

Precondición: El usuario debe elegir los productos que quiere comprar y la cantidad que desea adquirir de cada uno de ellos.

Postcondición: El sistema carga los productos seleccionados por el cliente en su carrito de la compra y accede a la pantalla del desglose de la compra para continuar con el proceso de compra.

Flujo de eventos principal:

1. El usuario elegirá los productos del catálogo (para ello podrá realizar búsquedas de productos) que quiera junto con la cantidad que desea comprar de cada uno y los irá añadiendo uno a uno a su listado de productos (carrito de la compra).
2. Una vez que ha terminado de añadir productos, deberá aceptar su carrito para continuar con el proceso de compra.

Flujo de eventos excepcional:

1. Si el usuario trata de comprar más cantidad de un producto del stock que dispone la empresa, la aplicación no añadirá el producto a su lista.

Escenario: Subsistema Compra on-line.

4.- Caso de uso Rellenar formulario de entrega (RF-39):

Descripción: Para que un cliente pueda completar su compra tiene que validar su formulario de entrega.

Objetivo: El usuario debe verificar la dirección de entrega en la que se servirá el pedido y comprobar que todos sus datos de contacto son correctos.

Actor: Cliente.

Precondición: El usuario debe haber añadido algún producto a su carrito de la compra. Si el cliente ha eliminado todos los productos de la lista, el sistema impide al cliente continuar con el proceso de compra.

Postcondición: El cliente debe elegir el método de pago del pedido.

Flujo de eventos principal:

1. El sistema muestra el formulario con los datos del cliente. Estos datos son con los que se dio de alta en la web de la aplicación.
2. El cliente puede modificar los datos del formulario que considere.

Flujo de eventos excepcional:

1. Si alguno de los datos del formulario del cliente no se ha rellenado correctamente, el sistema mostrará un mensaje de error impidiendo continuar con el proceso de compra.

Escenario: Subsistema Compra on-line.

4.- Caso de uso Forma de pago (RF-40):

Descripción: El cliente tiene la posibilidad de elegir la forma en la que va a abonar el pedido de entre las posibilidades que le ofrece la empresa. Dentro de los métodos de pago que puede seleccionar el cliente, están el pago mediante transferencia bancaria a una de las cuentas de la empresa (el cliente deberá elegir la cuenta en la que se abonará el pedido y el sistema mostrará al cliente el listado del pedido con los datos de la factura, listado de los datos de envío y un listado con la información bancaria que el cliente necesita para hacer la transferencia) o pago contra reembolso tras recibir el pedido en su casa (en este caso el sistema mostrará los mismos listados que para el pago por transferencia bancaria salvo el listado de información bancaria, ya que para este caso no aplica).

Objetivo: Dar la opción de escoger la forma de pago que le resulte más cómoda al cliente.

Actor: Cliente.

Precondición: Los datos de contacto del cliente deben estar correctamente rellenados.

Postcondición: El proceso de compra continúa. El sistema calcula la factura asociada al pedido mostrando los datos de esta y los datos del formulario introducido por el cliente.

Flujo de eventos principal:

1. El sistema solicita al cliente que elija la forma de pago que desee de entre las posibles.
2. El cliente elige la forma de pago o bien transferencia bancaria o pago contra reembolso.
3. Dependiendo de la forma de pago elegida por el cliente el sistema visualizará una determinada información u otra.
4. Una vez aceptado el formulario el sistema generará la factura asociada al pedido y el estado del pedido pasa a ser pendiente de cobro.

Flujo de eventos excepcional:

1. Hasta que el cliente no elija la forma de pago no se habilitará el botón para poder continuar con el proceso de compra.

Escenario: Subsistema Compra on-line.

5.- Caso de uso Generar factura (RF-41):

Descripción: Este caso de uso sólo es válido cuando el pedido lo está realizando la empresa a un proveedor.

Objetivo: La empresa tendrá un justificante del pedido que ha realizado al proveedor por medio de la factura que le proporcionará dicho proveedor. Esta factura puede ser imprimida. Este caso de uso es una extensión del caso de uso Realizar compra.

Actor: Administrador.

Precondición: El usuario debe haber realizado un pedido.

Postcondición: El pedido genera una factura.

Flujo de eventos principal:

1. El sistema calculará la factura asociada al pedido realizado (mostrará el desglose de precios de los productos adquiridos).
2. El usuario tiene la posibilidad de imprimir la factura de su pedido.

Escenario: Subsistema Compra on-line.

6.- Caso de uso Visualizar/modificar carrito (RF-42):

Descripción: El carrito de la compra podrá modificarse por el usuario que está realizando la compra.

Objetivo: El usuario podrá visualizar el carrito con todos los productos que va a comprar y podrá quitar o modificar la cantidad de los que ya tiene en el carrito o añadir aquellos que quiera. Este caso de uso incluye el caso de uso Calcular importe.

Actor: Cliente.

Precondición: El carrito de la compra debe haberse rellenado con algún producto.

Postcondición: El proceso de compra continúa.

Flujo de eventos principal:

1. A medida que el cliente selecciona nuevos productos, estos se irán añadiendo a la lista de pedidos. La información proporcionada por el sistema de cada uno de ellos será el código del producto, nombre, número de unidades, stock y precio unitario.
2. Una vez que el cliente ha terminado de elegir productos, el sistema muestra al cliente su carrito con todos los productos añadidos en la etapa anterior.
3. El cliente puede eliminar productos del carrito. Una vez acepte el carrito continuará su proceso de compra y pasará a la siguiente etapa del proceso de compra.

Flujo de eventos excepcional:

1. Si el cliente ha eliminado todos los productos de la lista, el sistema impide al cliente continuar con el proceso de compra mostrando un mensaje de aviso.

Escenario: Subsistema Compra on-line.

7.- Caso de uso Calcular importe (RF-43):

Descripción: Una vez que el cliente ha terminado de completar su carrito el sistema calculará el importe final de su compra.

Objetivo: Conocer el coste que se abonará por el pedido que se esté realizando. El sistema calculará la factura asociada al pedido y mostrará al cliente el resultado.

Precondición: El usuario debe haber añadido productos a su carrito.

Postcondición: El sistema calculará el importe asociado al pedido mostrando el desglose del pedido.

Flujo de eventos principal:

1. Tras completar el carrito el sistema muestra un listado con el desglose del importe del pedido. En este listado se muestra el importe unitario y total de cada producto del carrito.
2. Al finalizar el proceso de compra se mostrará el listado anterior pero además se mostrarán los importes

totales de cada producto y el coste total del pedido incluyendo además los gastos de envío.

Flujo de eventos excepcional:

1. Si el sistema detecta algún problema durante el cálculo del importe del pedido, mostrará un mensaje de error y anulará el pedido.

Escenario: Subsistema Compra on-line.

8.- Caso de uso Realizar compra (RF-44):

Descripción: Un cliente o administrador puede realizar un pedido a través de la web de la pajarería.

Objetivo: En caso de que el pedido lo realice un cliente su finalidad es la de adquirir un producto que ofrezca la empresa. Si por el contrario el pedido lo realiza un administrador su objetivo es abastecer a la empresa de los productos que necesite (en este caso el pedido se realiza a través de la web pero a proveedores asociados a la empresa). Este caso de uso incluye los casos de uso Identificarse, Consultar catálogo de productos, Rellenar formulario de entrega, Forma de pago y Visualizar/modificar carrito.

Actor: Cliente/Administrador.

Precondición: El usuario debe haber seleccionado la opción que le permitirá realizar un pedido a través de web.

Postcondición: El sistema almacena la información asociada al pedido realizado por el usuario.

Flujo de eventos principal:

1. El administrador accede a la opción del menú correspondiente a Pedidos (si el pedido lo está realizando un cliente a la empresa) o Gestión de productos (si el pedido lo está realizando un administrador a un proveedor), tras logarse en la web (corresponde al caso de uso Identificarse).
2. El sistema inicia el proceso de gestión de pedidos creando un pedido vacío y con estado iniciado.
3. Una vez finalizado todo el proceso de compra correctamente (rellenar el carrito de la compra, completar formulario de entrega y forma de pago), el sistema muestra un resumen del pedido con los datos de la factura, del envío y un mensaje del éxito de la operación. En ese momento el estado del pedido pasa a pendiente de cobro.

Flujo de eventos excepcional:

1. Si el sistema detecta alguna anomalía en el proceso mostrará un mensaje de error y anulará el pedido.

Escenario: Subsistema Compra on-line.

9.- Caso de uso Consultar pedidos (RF-45):

Descripción: Esta opción está disponible únicamente para administradores del sistema, los cuales podrán conocer el estado de un pedido realizado por un cliente y los productos que se han adquirido en cualquier pedido tanto lo realizados a la empresa como los que la empresa ha realizado a cualquier proveedor.

Objetivo: Conocer en detalle información de cualquier pedido como puede ser la fecha en la que se ha realizado, el proveedor al cual se le ha realizado el pedido (en caso de que el pedido se haya realizado a un proveedor) o cliente que ha realizado el pedido (en caso de que sea el cliente el que esté realizando un pedido a través de la web), importe, y productos adquiridos. Este caso de uso incluye el caso de uso Identificarse.

Actor: Administrador.

Precondición: Deben realizarse búsquedas de pedidos que produzcan resultados.

Postcondición: El sistema muestra el detalle del pedido que quiera el administrador.

Flujo de eventos principal:

1. El administrador accede a la opción del menú correspondiente a la Gestión de Pedidos tras logarse en la web (corresponde al caso de uso Identificarse).
2. El sistema muestra dos opciones para que el administrador elija el tipo de pedidos que quiera consultar o pedidos a clientes o pedidos a proveedores.
3. En cualquiera de los dos casos el administrador deberá realizar una búsqueda de pedidos (corresponde al caso de uso Búsqueda pedidos) para lo cual puede rellenar algún filtro de búsqueda que quiera. En caso de búsquedas de pedidos a proveedores los filtros que puede rellenar son el código de pedido, proveedor, tipo de producto, fecha de inicio y fecha de fin del pedido. Si se trata de búsquedas de pedidos a clientes los filtros de búsqueda a rellenar son código del pedido, estado, cliente, tipo de producto, fecha de inicio y fecha de fin.
4. Tras realizar la búsqueda el sistema mostrará el listado de pedidos que se ajuste a los filtros de búsqueda.
5. En caso de que la búsqueda sea de pedidos realizados por clientes, el administrador podrá elegir el pedido que desee. Entonces el sistema muestra una pantalla con toda la información asociada al pedido como son datos del pedido, datos de envío y datos bancarios.

Flujo de eventos excepcional:

1. Si el sistema detecta alguna anomalía durante el proceso de búsqueda, mostrará un mensaje de error y no devolverá registros.

Escenario: Subsistema Compra on-line.

10.- Caso de uso Modificar estado pedido cliente (RF-46):

Descripción: Un administrador de la web debe actualizar el estado de los pedidos realizados por los clientes.

Objetivo: Actualizar el estado del pedido al momento que corresponda. Los estados posibles son pendiente de envío (estado inicial al finalizar un pedido), pagado, entregado y realizado (cuando se ha completado todo el ciclo del pedido).

Actor: Administrador.

Precondición: El usuario debe haber elegido un pedido.

Postcondición: El sistema actualizará el estado del pedido que ha modificado.

Flujo de eventos principal:

1. El administrador accede a la opción del menú correspondiente a Gestión de clientes, tras logarse en la web (corresponde al caso de uso Identificarse).
2. A continuación el administrador deberá realizar una búsqueda de pedidos de clientes (corresponde al caso de uso Búsqueda pedidos) para lo cual puede rellenar algún filtro de búsqueda que quiera.
3. El sistema muestra los pedidos que se ajusten a los filtros de búsqueda. El administrador elige un pedido del listado.
4. El sistema muestra los datos del cliente que ha realizado ese pedido y el estado del pedido.
5. El administrador actualizará el estado del pedido que ha escogido para adecuarlo a su nueva situación.

Flujo de eventos excepcional:

1. Si se produce algún fallo durante el proceso de actualización del estado del pedido en el sistema la aplicación arrojará un mensaje de error.

Escenario: Subsistema Compra on-line.

11.- Caso de uso Búsqueda pedidos (RF-47):

Descripción: El sistema permite a cualquier administrador realizar búsquedas de pedidos.

Objetivo: Localizar de manera rápida los pedidos que necesita consultar o gestionar.

Actor: Administrador.

Precondiciones: Rellenar el formulario de búsqueda en caso de que el administrador lo necesite (los filtros de búsqueda introducidos deben ser válidos) y pinchar el botón de buscar.

Postcondición: El sistema mostrará un listado con los pedidos que se ajustan a los parámetros de búsqueda introducidos en el formulario.

Flujo de eventos principal:

1. El administrador rellena los filtros de búsqueda que quiera y solicita al sistema que realice la búsqueda a partir de los parámetros introducidos en la etapa anterior.
2. El sistema realiza la búsqueda y muestra los resultados en forma de listado por pantalla.

Flujo de eventos excepcional:

1. El sistema mostrará un mensaje de error en caso de que los filtros de búsqueda sean incorrectos.
2. Si los filtros de búsqueda no producen ningún resultado, el sistema indicará al administrador que la búsqueda no ha producido ningún resultado.

Escenario: Subsistema Compra on-line.

5. Matriz de Rastreabilidad

La matriz de rastreabilidad es una herramienta que se usa para saber qué requisitos de una aplicación han sido probados y cubiertos en cada test de pruebas diseñado. Es decir, relaciona los requisitos del sistema con objetivos de prueba. No puede haber objetivos que queden sin reflejar en los casos de uso.

En el caso de nuestra aplicación web, se definieron anteriormente todos los requisitos que hemos identificado en el sistema que recopilamos a continuación:

REQUISITO DE INFORMACION	
Datos de Usuario	RI-01
Datos de Producto	RI-02
Datos de Pedido	RI-03
Datos de Venta/Factura	RI-04
Datos de Proveedor	RI-05
Datos del foro	RI-06
REQUISITOS NO FUNCIONALES	
Interfaz de Usuario	RNF-01
Interfaz Hardware y Rendimiento	RNF-02
Interfaz Software	RNF-03
Seguridad	RNF-04
REQUISITOS FUNCIONALES	
Consultar información	RF-01
Mostrar características	RF-02
Selección especie	RF-03
Ponerse en contacto	RF-04
Enviar pregunta	RF-05
Consulta de centros	RF-06
Selección provincia	RF-07
Identificarse (Login)	RF-08
Escribir mensaje	RF-09
Visualizar mensaje/s	RF-10
Eliminar mensaje	RF-11
Gestión centros	RF-12
Búsqueda centros	RF-13
Visualizar gráficos	RF-14

Rellenar filtros búsqueda	RF-15
Comprobar herencia genética	RF-16
Visualizar resultados	RF-17
Mostrar imágenes descendencia	RF-18
Consultar disponibilidad	RF-19
Actualizar datos cliente	RF-20
Consultar cliente	RF-21
Buscar cliente	RF-22
Baja cliente	RF-23
Eliminar pedidos cliente	RF-24
Consultar catálogo de productos	RF-25
Búsqueda producto	RF-26
Modificar/Eliminar/Añadir producto	RF-27
Gestionar oferta catálogo	RF-28
Comprobar stock	RF-29
Elegir centro	RF-30
Actualizar/añadir proveedor	RF-31
Consultar catálogo proveedor	RF-32
Realizar pedido proveedor	RF-33
Consultar proveedor	RF-34
Búsqueda proveedor	RF-35
Eliminar proveedor	RF-36
Eliminar catálogo proveedor	RF-37
Añadir producto al carrito	RF-38
Rellenar formulario de entrega	RF-39
Forma de pago	RF-40
Generar factura	RF-41
Visualizar/modificar carrito	RF-42
Calcular importe	RF-43
Realizar compra	RF-44
Consultar pedidos	RF-45
Modificar estado pedido cliente	RF-46
Búsqueda pedidos	RF-47

Figura 10. Requisitos del sistema

Además definiremos nuestros objetivos o casos de prueba que deberá pasar nuestro sistema. Estos son:

OBJETIVOS	
Gestión de usuarios	OBJ-01
Gestión de centros	OBJ-02
Gestión del foro	OBJ-03
Gestión de productos	OBJ-04
Gestión de proveedores	OBJ-05
Gestión de ventas	OBJ-06
Gestión de pedidos	OBJ-07
Gestión de información de negocio	OBJ-08
Gestión de la capacidad del sistema	OBJ-09
Gestión de la información	OBJ-10

Figura 11.
Objetivos del sistema

Por tanto nuestra matriz de rastreabilidad quedará definida de la siguiente manera:

	OBJ-01	OBJ-02	OBJ-03	OBJ-04	OBJ-05	OBJ-06	OBJ-07	OBJ-08	OBJ-09	OBJ-10
RI-01	X								X	X
RI-02				X				X	X	X
RI-03							X		X	X
RI-04						X		X	X	X
RI-05					X				X	X
RI-06			X						X	X
RNF-01	X	X	X	X	X	X	X	X		
RNF-02									X	
RNF-03									X	
RNF-04	X									X
RF-01								X		
RF-02				X						
RF-03				X						
RF-04								X		X
RF-05								X		X

RF-06					X	
RF-07					X	
RF-08	X					X
RF-09			X			
RF-10			X			
RF-11			X			
RF-12		X				
RF-13		X				
RF-14				X	X	
RF-15				X	X	
RF-16					X	
RF-17					X	
RF-18					X	
RF-19	X					X
RF-20	X					X
RF-21	X					
RF-22	X					
RF-23	X				X	
RF-24	X				X	
RF-25			X			
RF-26			X			
RF-27			X			X
RF-28			X			X
RF-29			X			
RF-30			X			
RF-31				X		X
RF-32			X	X		
RF-33				X	X	
RF-34				X		
RF-35				X		

RF-36			X	
RF-37	X		X	X
RF-38				X
RF-39	X			X
RF-40	X			X
RF-41				X
RF-42				X
RF-43				X
RF-44			X	X
RF-45			X	X
RF-46			X	X
RF-47			X	X

Figura 12. Matriz de rastreabilidad

6. Diseño de Clases

El propósito de esta actividad, que se realiza sólo en el caso de diseño orientado a objetos, es transformar el modelo de clases lógico, que proviene del análisis, en un modelo de clases de diseño. Dicho modelo recoge la especificación detallada de cada una de las clases, es decir, sus atributos, operaciones, métodos, y el diseño preciso de las relaciones establecidas entre ellas, bien sean de agregación, asociación o jerarquía. Para llevar a cabo todos estos puntos, se tienen en cuenta las decisiones tomadas sobre el entorno tecnológico y el entorno de desarrollo elegido para la implementación. Se identifican las clases de diseño, que denominamos **clases adicionales**, en función del estudio de los escenarios de los casos de uso. Entre ellas se encuentran clases abstractas, que integran características comunes con el objetivo de especializarlas en clases derivadas. Se diseñan las clases de interfaz de usuario, que provienen del análisis. Hay que considerar que pueden aparecer nuevas clases por el diseño de **asociaciones y agregaciones** o desaparecer clases junto con sus **atributos** y **métodos** por motivos de optimización. En esta actividad también se identifican las **jerarquías** de clases y los atributos y métodos de cada clase gracias al análisis de los casos de uso de la actividad anterior. Además, en los casos en que sea necesaria una **migración de datos** de otros sistemas o una **carga inicial** de información, se realizará su especificación a partir del modelo de clases y las estructuras de datos de los sistemas origen.

6.1. Identificación de Clases Adicionales

En este apartado pasamos a definir cada una de las clases identificadas a partir del análisis del diagrama de subsistemas de aplicación. Todas las clases correspondientes al controlador de la aplicación heredarán de la clase `org.apache.struts.actions.DispatchAction`.

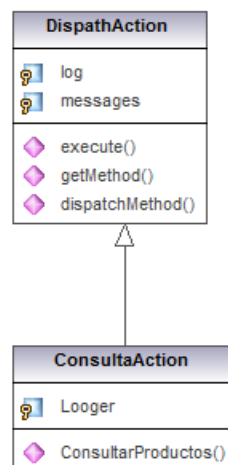


Figura 13. Clase DispatchAction

Esta clase está dentro del paquete de struts y provee de la funcionalidad necesaria para gestionar la parte del controlador del sistema.

Atributos	
protected java.lang.Class	Instancia de la clase DispatchAction class.
Protected static org.apache.commons.logging.Log	log Commons Logging instance.
protected static MessageResources	messages The message resources for this package.
protected java.util.HashMap	methods The set of Method objects we have introspected for this class, keyed by method name.

protected java.lang.Class[]	Types The set of argument type classes for the reflected method call.
-----------------------------	--

Constructor	
DispatchAction()	

Métodos	
protected ActionForward	cancelled (ActionMapping mapping, ActionForm form, javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) Método que despacha la petición de usuario cuando es cancelada a través del botón cancel.
protected ActionForward	dispatchMethod (ActionMapping mapping, ActionForm form, javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response, java.lang.String name) Método que redirecciona la petición al método correspondiente.
ActionForward	execute (ActionMapping mapping, ActionForm form, javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) Proceso especificado por la petición http y que se encarga de crear la correspondiente respuesta del sistema.
protected java.lang.reflect.Method	getMethod (java.lang.String name) Introspect the current class to identify a method of the specified name that accepts the same parameter types as the execute method does.
Protected java.lang.String	getMethodName (ActionMapping mapping, ActionForm form, javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response, java.lang.String parameter) Devuelve el nombre del método y sus parámetros.
protected java.lang.String	getParameter (ActionMapping mapping, ActionForm form, javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) Devuelve el valor del parámetro.
protected ActionForward	unspecified (ActionMapping mapping, ActionForm form, javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) Method which is dispatched to when there is no value for specified request parameter included in the request.

Clase Controlador

En las clases controlador será donde se desarrolle toda la lógica de negocio, en nuestro sistema vamos a llamarlas clases de acción. La nomenclatura de estas clases será de tipo NombreProcesoNegocioAction y heredarán de la clase `DispatchAction`. Por cada clase de acción tendremos un formulario asociado y los métodos a los que se redirecciona desde la interfaz de usuario a través de las jsp.

La implementación de cada método tiene que cumplir la siguiente secuencia:

- Obtener el formulario con los datos de usuario
- Operar con los datos de usuario
- Llamada a las interfaces de acceso a base de datos
- Preparar la respuesta del sistema almacenándola en los distintos áreas que provee el sistema (objeto request, session...)
- Mapear la redirección a la siguiente navegación

A continuación se detallan atributos y métodos que como mínimo tiene que tener una clase de acción cualquiera de nuestro sistema.

Atributos	
private es.proyecto.pajareria.bo.ClaseBO	ClaseBO Clase de acceso a base de datos.

Métodos	
public ActionForward	busqueda (ActionMapping mapping, ActionForm form, javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) Método que devuelve un resultado de búsqueda y redirecciona la petición a la interfaz de usuario correspondiente.
public ActionForward	alta (ActionMapping mapping, ActionForm form, javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) Método que hace el alta en el sistema y redirecciona la petición a la interfaz de usuario correspondiente.
public ActionForward	baja (ActionMapping mapping, ActionForm form, javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) Método que la baja en el sistema y redirecciona la petición a la interfaz de usuario correspondiente.
public ActionForward	modificacion (ActionMapping mapping, ActionForm form, javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response) Método que modifica los datos y redirecciona la petición a la interfaz de usuario correspondiente.

El siguiente esquema muestra una implementación de una clase de acción cualquiera de nuestro sistema.

```

public ActionForward search(
    final ActionMapping aMapping,
    final ActionForm aForm,
    final HttpServletRequest aRequest,
    final HttpServletResponse aResponse) {

    CentrosForm tmpForm = (CentrosForm) aForm;
    ActionForward tmpForward = new ActionForward();
    CentroVO tmpCentro = new CentroVOImpl();
    CentroBO tmpCentroBO = new CentroBO();
    inicializarFormulario(tmpForm);
    try {
        if (tmpForm.getBoton() != null && tmpForm.getBoton().equals("1")){
            Collection tmpListaCentros =
                tmpCentroBO.getListadoCentros (tmpForm.getCodigo(), tmpForm.getNombre());
            aRequest.setAttribute(Constants.LISTADO_CENTROS, tmpListaCentros);
            aRequest.getSession().setAttribute(Constants.LISTADO_CENTROS,
                tmpListaCentros);
            tmpForm.setListCentros(tmpListaCentros);
            aRequest.setAttribute("listaCentros", tmpListaCentros);
        }
    } catch (Exception e) {
        e.printStackTrace();
        String mensaje = "error al obtener el listado bbdd";
        tmpForm.setError(mensaje);
    }
    tmpForward = aMapping.findForward("success");
    return tmpForward;
}

```

Clase de Acceso a Base de Datos

Son las clases que implementan toda la lógica de acceso a Base de Datos. La nomenclatura de estas clases será del tipo NombreProcesoNegocioBO.

La implementación de los métodos debe seguir la siguiente secuencia :

- Creamos el objeto `connection`.
- Creamos la sentencia para ejecutar la query
- Ejecuta la sentencia.
- Devuelve el resultado de la operación.

A continuación se detallan atributos y métodos que como mínimo tiene que tener una clase de de acceso a base de datos cualquiera de nuestro sistema.

Atributos	
Public static final String	URL String con la url de conexión a base de datos.
Métodos	
public java.util.Collection	busqueda (Object objeto) Método que realiza la búsqueda en base de datos que se ajusta a los parámetros de búsqueda introducidos por cabecera. El resultado lo devuelve en un objeto Collection.
public	alta (Object objeto) Método que realiza el alta en base de datos de un objeto que se ajusta a los parámetros introducidos por cabecera. En caso de error se pasa el error al controlador.
public	baja (Object objeto) Método que realiza la baja en base de datos de un objeto que se ajusta a los parámetros introducidos por cabecera. En caso de error se pasa el error al controlador.
public	modificacion (Object objeto) Método que realiza la modificación en base de datos de un objeto que se ajusta a los parámetros introducidos por cabecera. En caso de error se pasa el error al controlador..

Método que realiza el alta en base de datos de un objeto que se ajusta a los parámetros introducidos por cabecera. En caso de error se pasa el error al controlador.

```

public String getCodigo(){
    String codigo = "";
    try {
        Class.forName("org.gjt.mm.mysql.Driver");
        Connection con = DriverManager.getConnection(urlMysql, "root", "admin");
        Statement stmt = con.createStatement();
        String consulta = "select max(id_centro) as id_centro from centro where id_centro like 'CN%'";
        ResultSet centro = stmt.executeQuery(consulta);
        while (centro.next()) {
            codigo = centro.getString("id_centro");
        }
        con.close();
    } catch (SQLException e) {
        e.printStackTrace();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
    return codigo;
}

```

Clases de Excepciones

Estas clases son las que se usan para el control de las excepciones del sistema. A parte de las que provee la arquitectura J2EE desarrollaremos algunas propias.

Estas clases heredan de otra clase de tipo `Exception`. La nomenclatura para estas clases sigue el estándar `nombreException`

A continuación se detallan atributos y métodos que como mínimo tiene que tener una clase de acceso a base de datos cualquiera de nuestro sistema.

Constructor	
<code>DispatchAction()</code>	

Todas las clases formulario de nuestro sistema heredan de la clase `org.apache.struts.action.ActionForm` que proporciona struts.

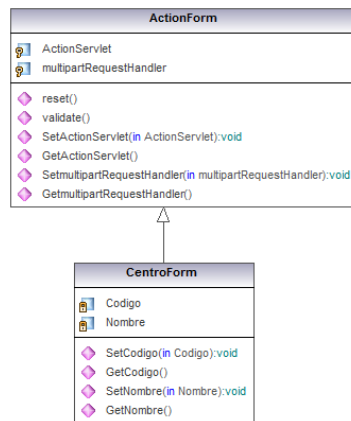


Figura 14. Clase ActionForm

Clase ActionForm

Esta clase esta dentro del paquete de struts y provee de la funcionalidad necesaria para gestionar la parte del controlador del sistema.

Atributos	
protected MultipartRequestHandler	multipartRequestHandler El MultipartRequestHandler para el controlador del formulario, puede ser nulo.
protected ActionServlet	servlet Servlet que va a ser instanciado.

Constructor	
<code>ActionForm()</code>	Constructor

Métodos	
MultipartRequestHandler	getMultipartRequestHandler() Devuelve el MultipartRequestHandler del formulario..
protected ActionServlet	getServlet() Devuelve la instancia del servlet.
ActionServletWrapper	getServletWrapper() Devuelve la instancia del servlet controlador.
Void	reset(ActionMapping mapping, javax.servlet.http.HttpServletRequest request) Inicializa las propiedades del bean.
Void	reset(ActionMapping mapping, javax.servlet.ServletRequest request) Inicializa las propiedades del bean al valor por defecto.
Void	setMultipartRequestHandler(MultipartRequestHandler multipartRequestHandler) Proporciona un manejador para los ficheros de carga.
Void	setServlet(ActionServlet servlet) Obtiene la instancia del servlet en caso de que no sea nulo.
ActionErrors	validate(ActionMapping mapping, javax.servlet.http.HttpServletRequest request) Valida las propiedades pasadas a través de la petición HTTP y devuelve un objeto ActionErrors que encapsula cualquier error que pueda ser encontrado.

Formularios

Estas clases serán las encargadas de almacenar los parámetros recogidos de la interfaz de usuario, almacenarlos en la request y pasárselos a las clases de acción.

Estas clases heredan de la clase `ActionForm` que proporciona struts. La nomenclatura para estas clases sigue el estándar `nombreForm`.

A continuación se detallan atributos y métodos que como mínimo tiene que tener una clase de este tipo cualquiera de nuestro sistema.

Atributos	
private Object	Parámetro String/Boolean/Integer... parámetro del formulario de la jsp.

Métodos	
public Object	Get() Método que retorna el valor de un parámetro del formulario
Public void	Set(Object objeto) Método que da el valor a un parámetro del formulario al introducido por cabecera

Interfaces VO

Estas interfaces representan las entidades del sistema. La nomenclatura para estas clases sigue el estándar `nombreVO`.

A continuación se detallan atributos y métodos que como mínimo tiene que tener una clase de este tipo cualquiera de nuestro sistema.

Métodos	
public Object	Get() Método que retorna el valor de un parámetro del formulario
Public void	Set(Object objeto) Método que da el valor a un parámetro del formulario al introducido por cabecera

Clases VOImpl

Son las clases encargadas de implementar las interfaces anteriores. La nomenclatura para estas clases sigue el estándar nombreVOImpl y existirá una por cada interfaz.

A continuación se detallan atributos y métodos que como mínimo tiene que tener una clase de este tipo cualquiera de nuestro sistema.

Atributos	
private Object	parámetro String/Boolean/Integer... parámetro del formulario de la jsp.

Métodos	
public Object	Get() Método que retorna el valor de un parámetro del formulario
Public void	Set(Object objeto) Método que da el valor a un parámetro del formulario al introducido por cabecera

6.2. Diagrama de clases

A continuación se especifican los diagramas de clases. Dada la complejidad de la aplicación agrupamos los diagramas de clases a partir de cada controlador identificado.

AltaUsuarioAction

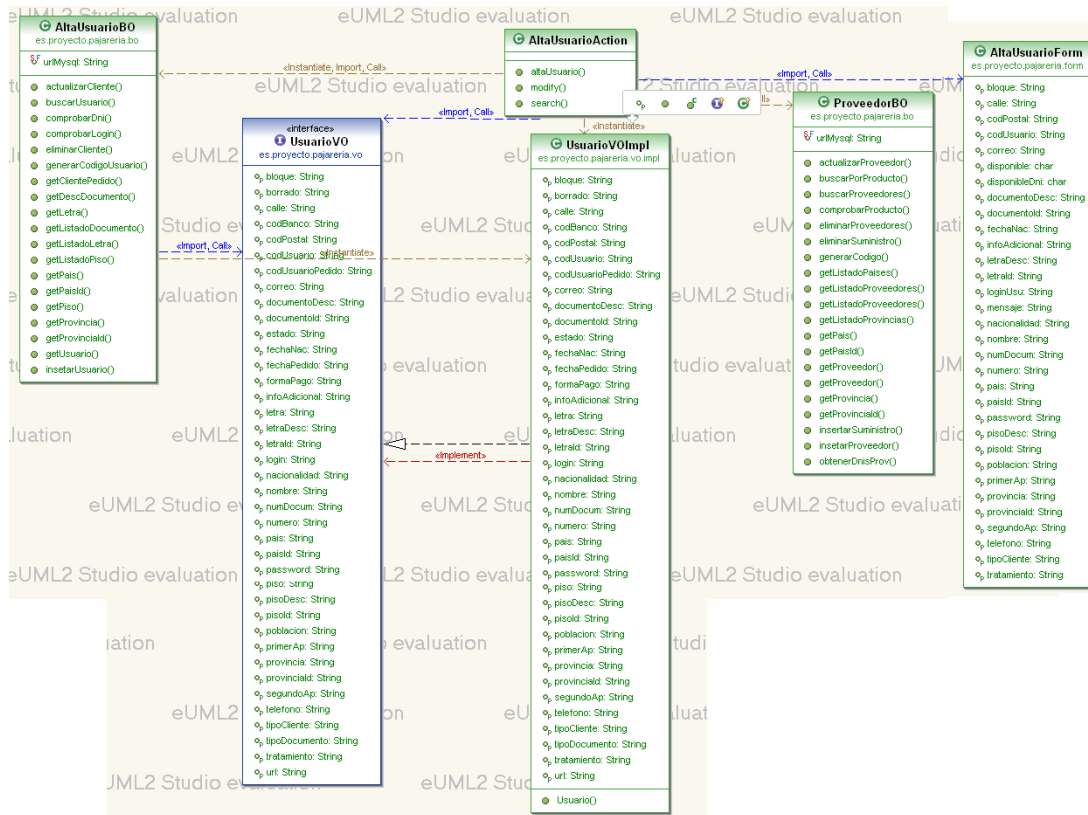


Figura 15. Alta Usuario

CaracterAction

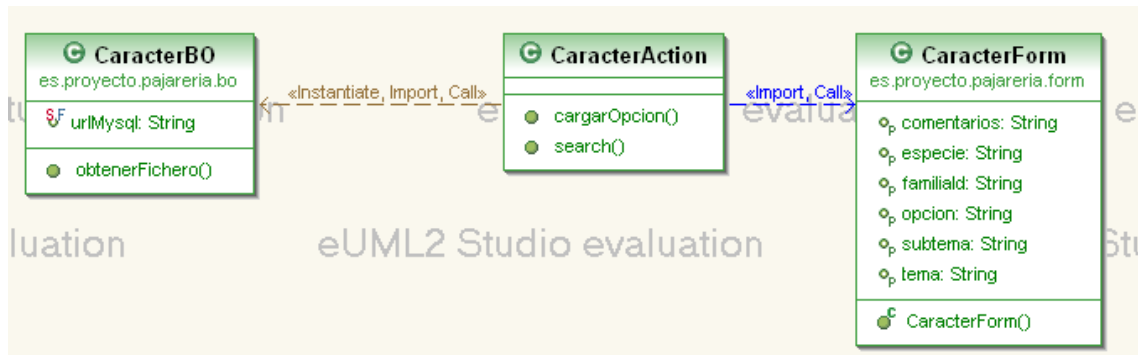


Figura 16. CaracterBO

CatalogoProductosAction

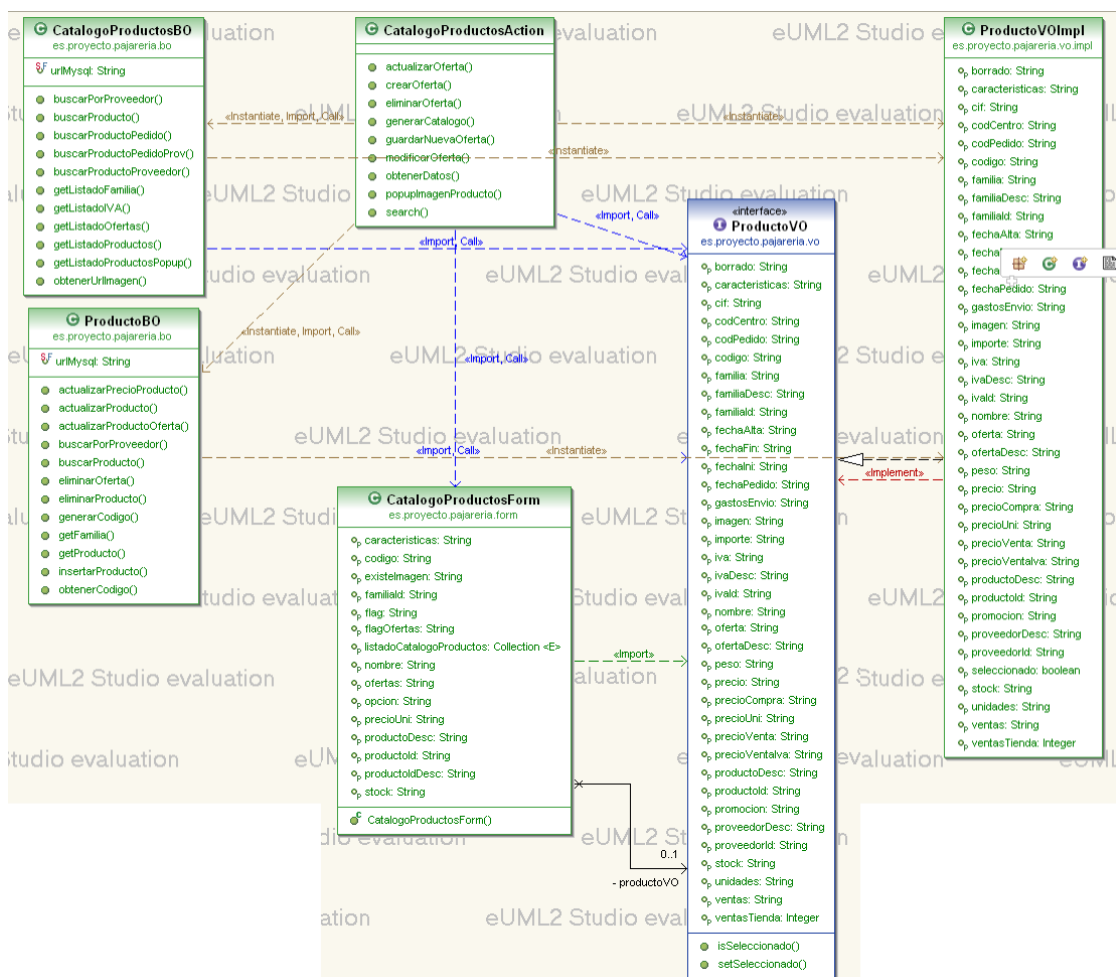


Figura 17. Catalogo Productos

CentroAction

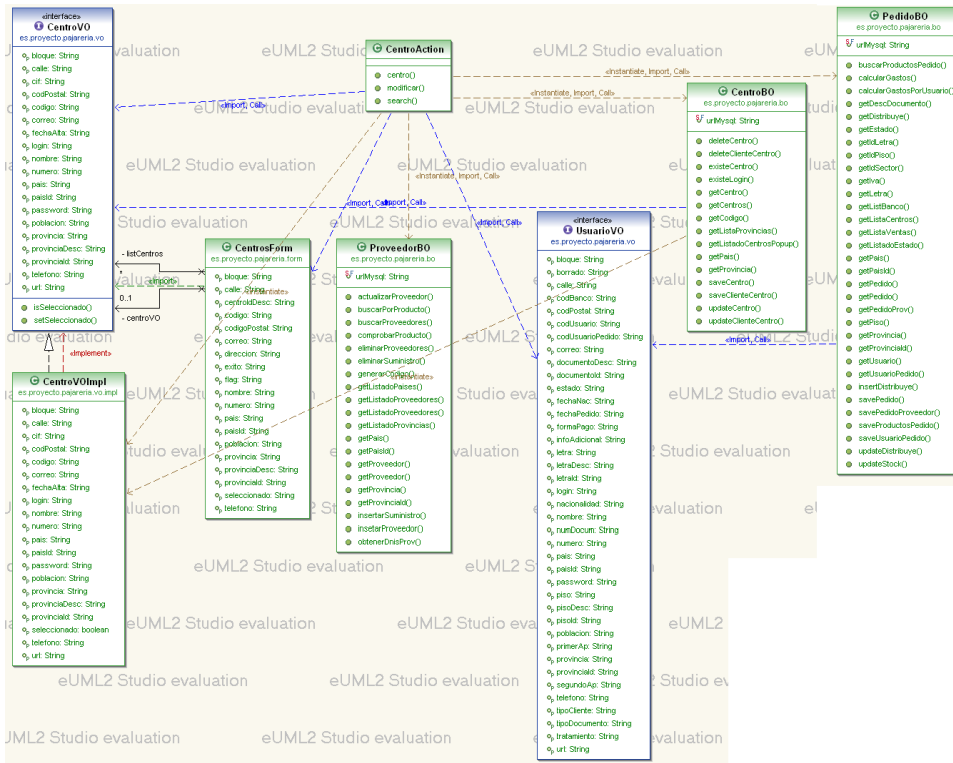


Figura 18. Centro

CentrosAction

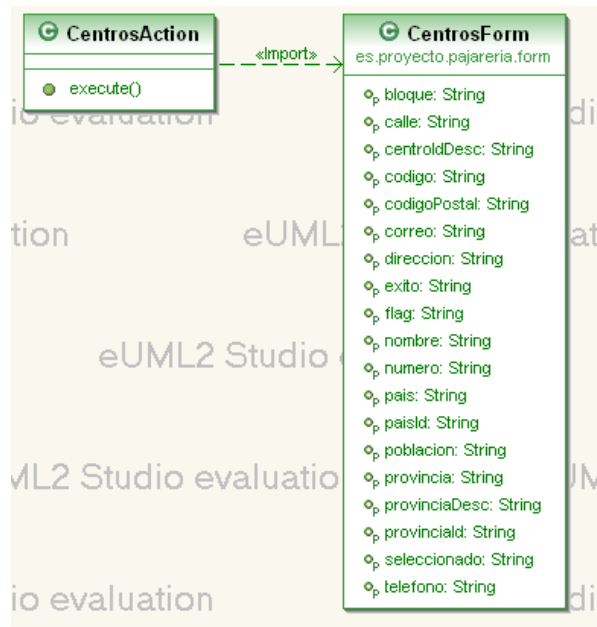


Figura 19. Centro

CharAction

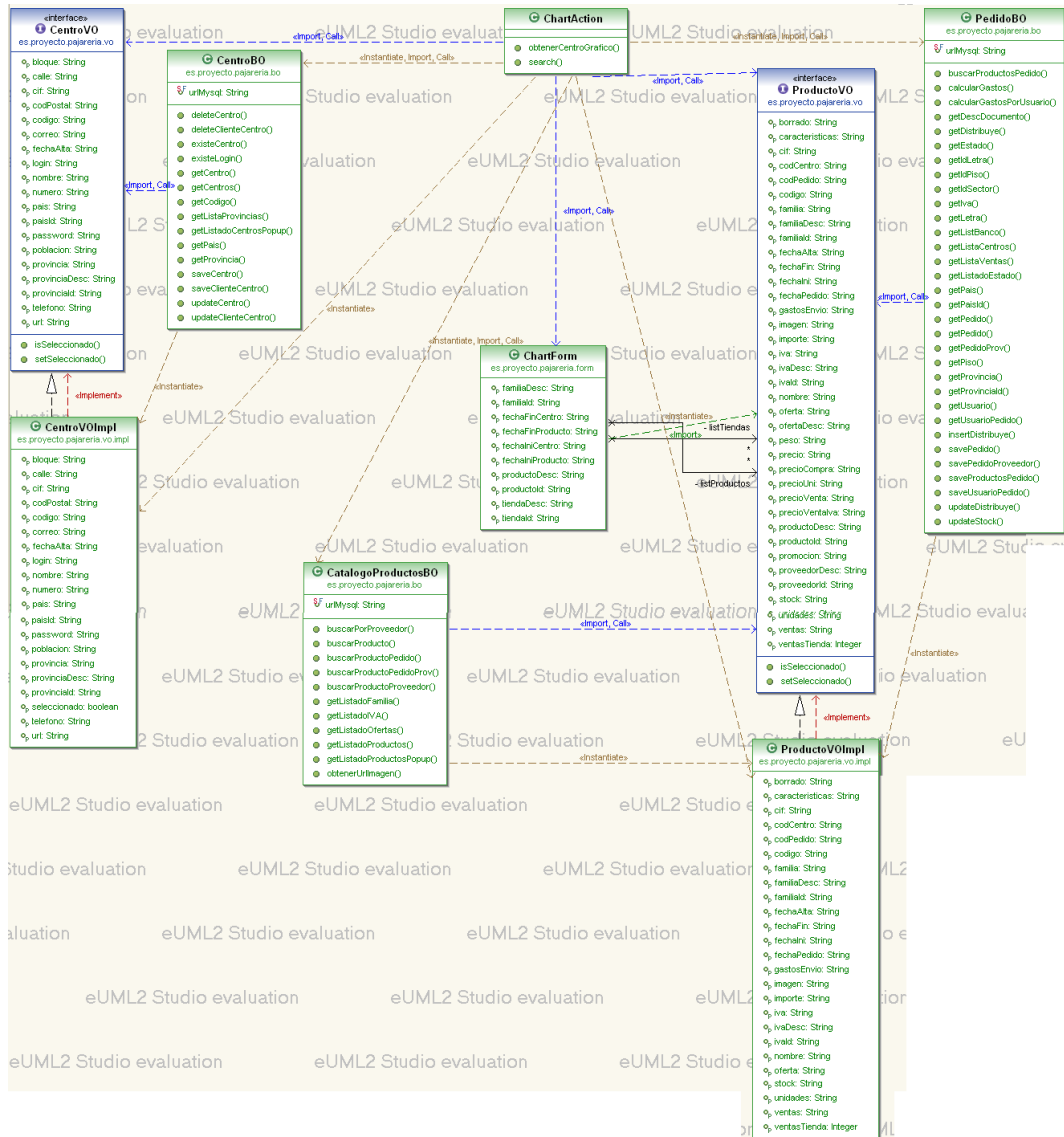


Figura 20. Char

ConsultaStockAction

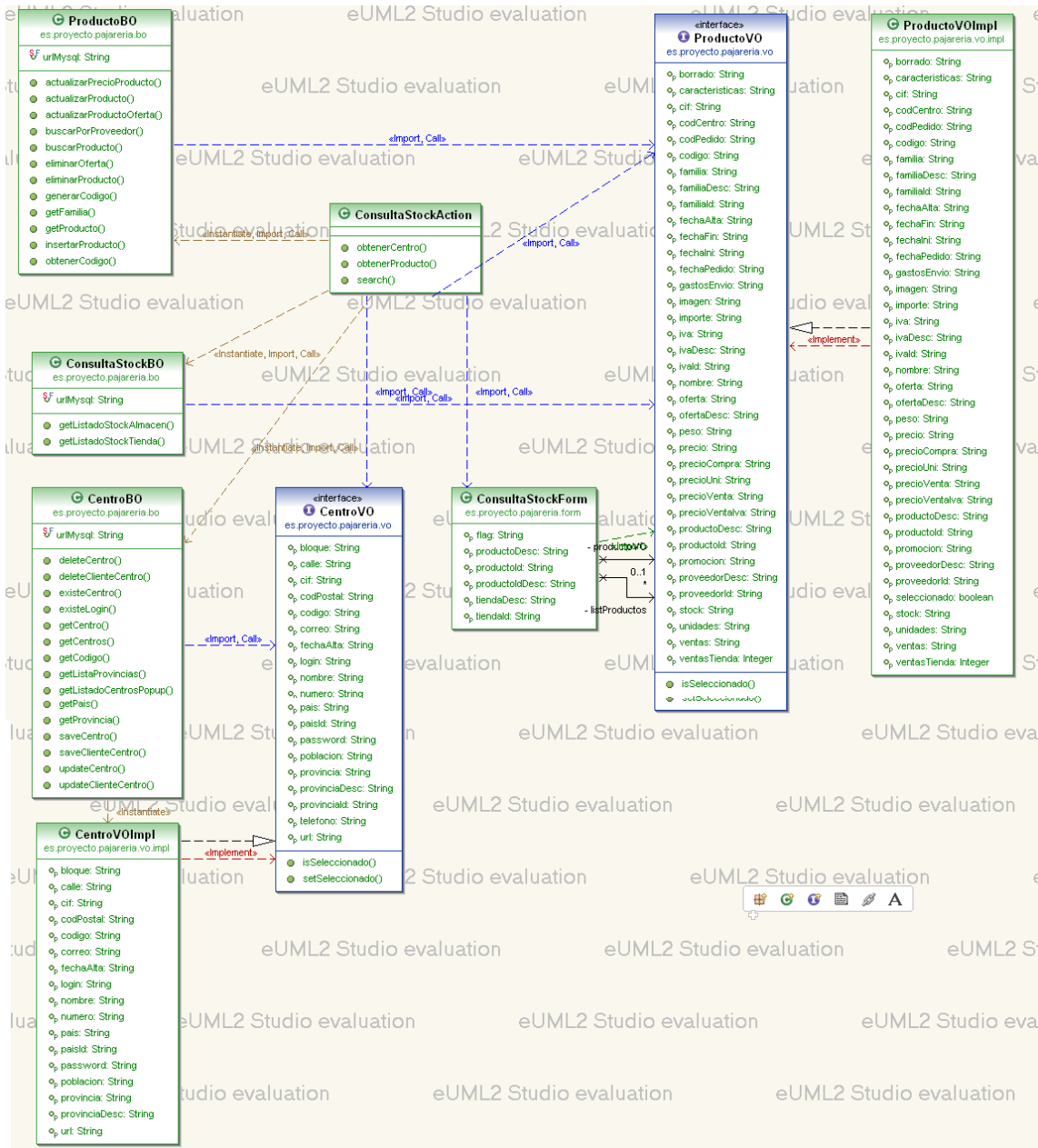


Figura 22. Consulta Stock

ContactoAction

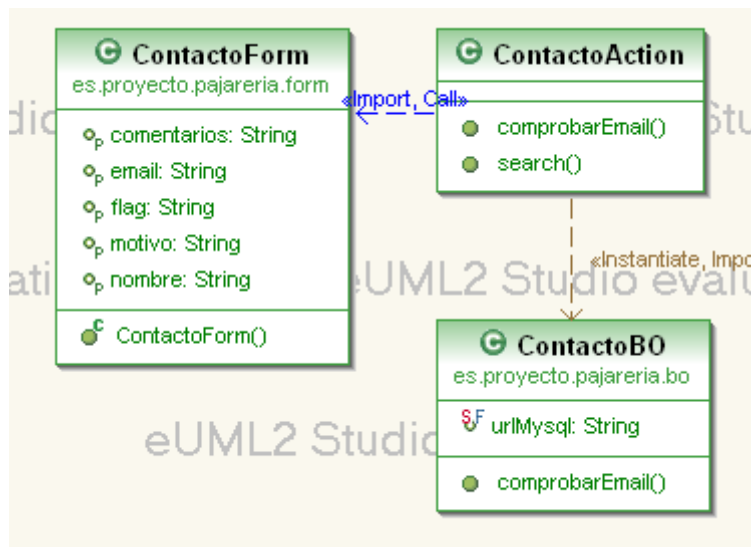


Figura 23. Contacto

EnviosAction

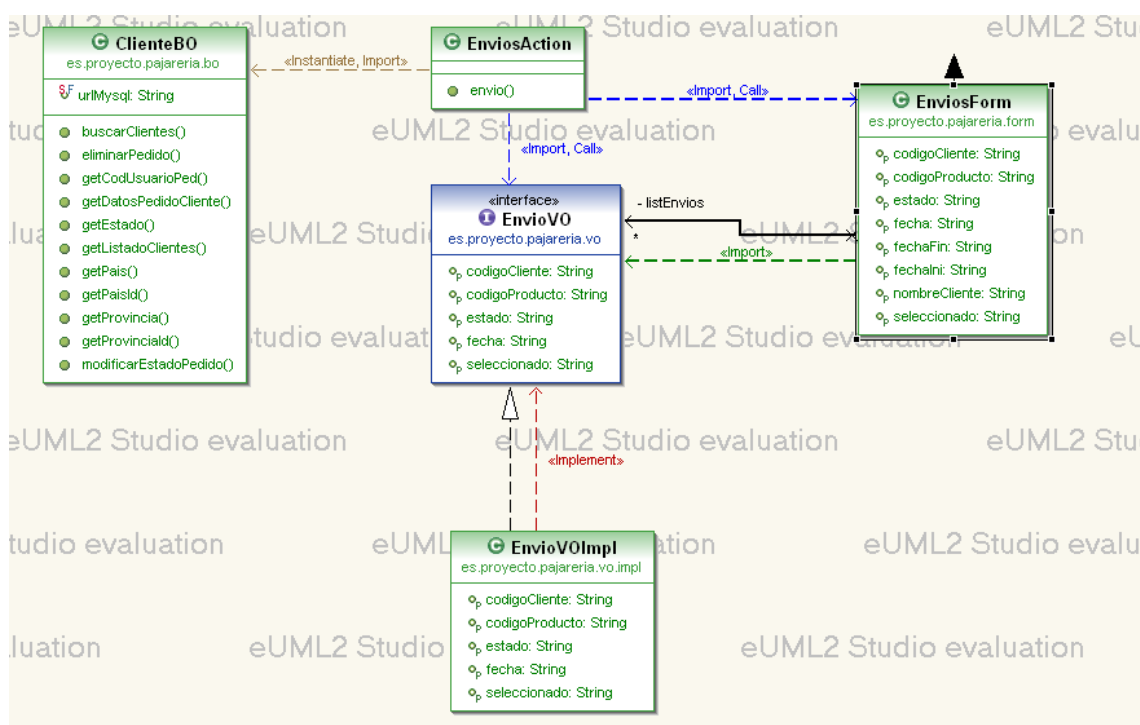


Figura 24. Envíos

ForoAction

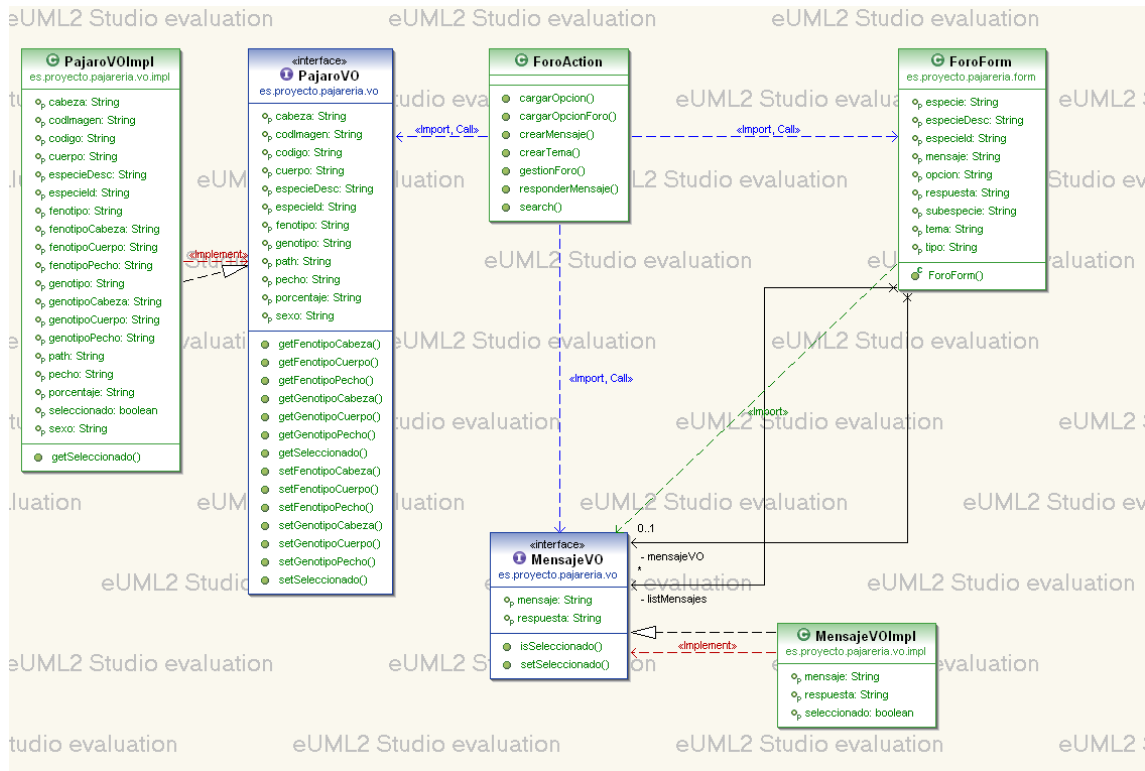


Figura 25. Foro

GestionClienteAction

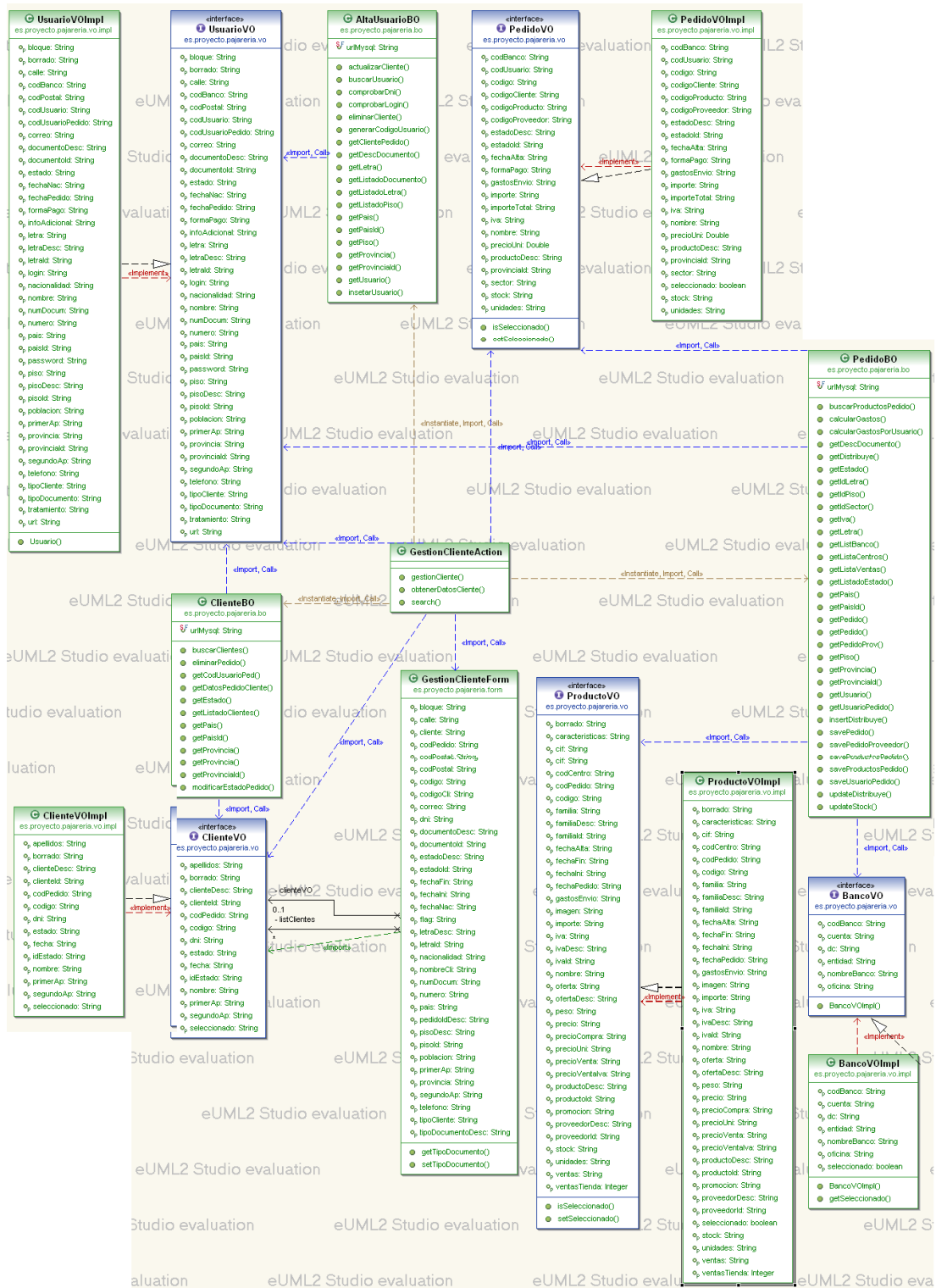


Figura 26. Gestión Cliente

GestionPedidosAction

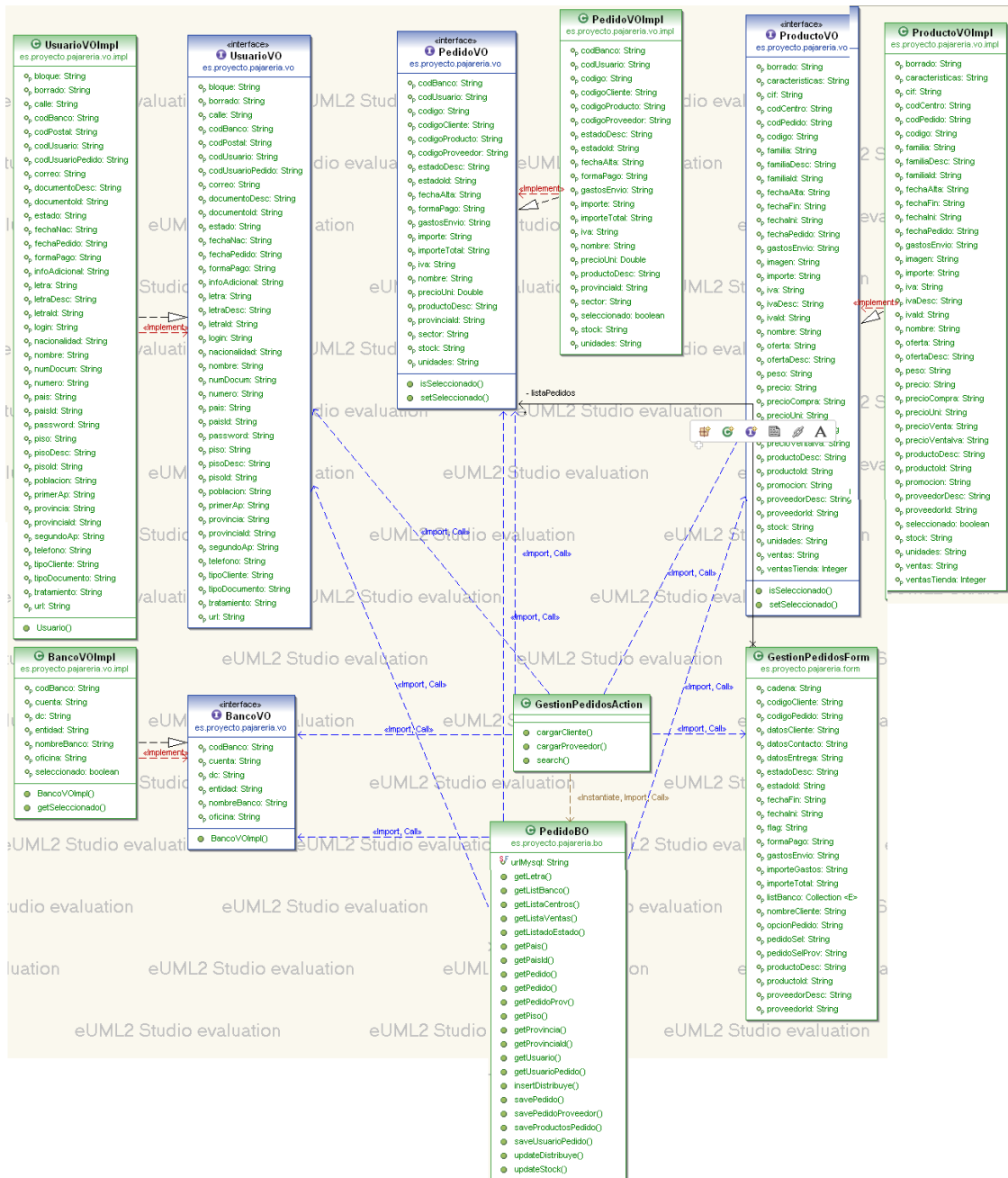


Figura 27. Gestión Pedidos

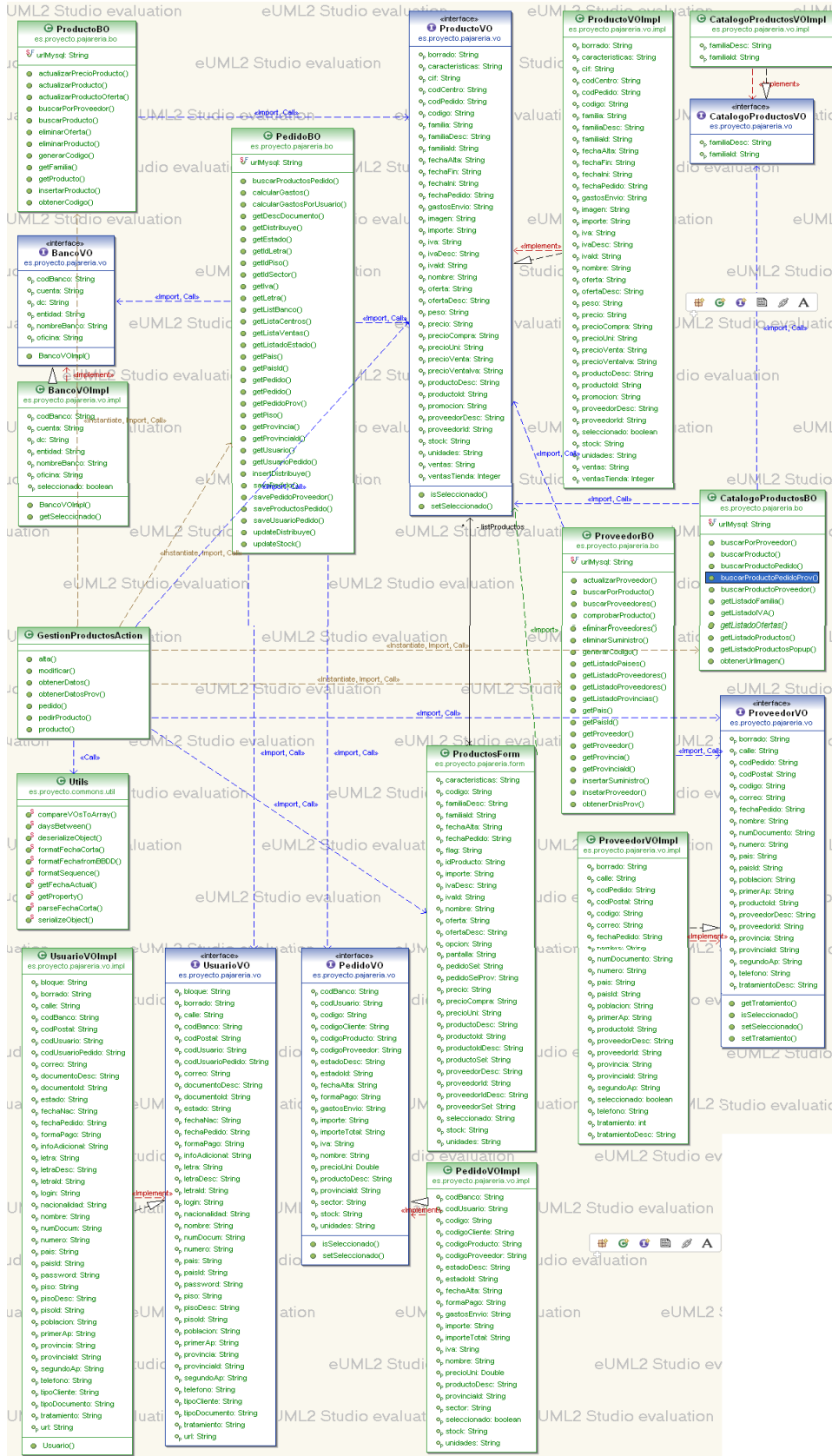


Figura 28. Gestión Productos

GestionProveedorAction

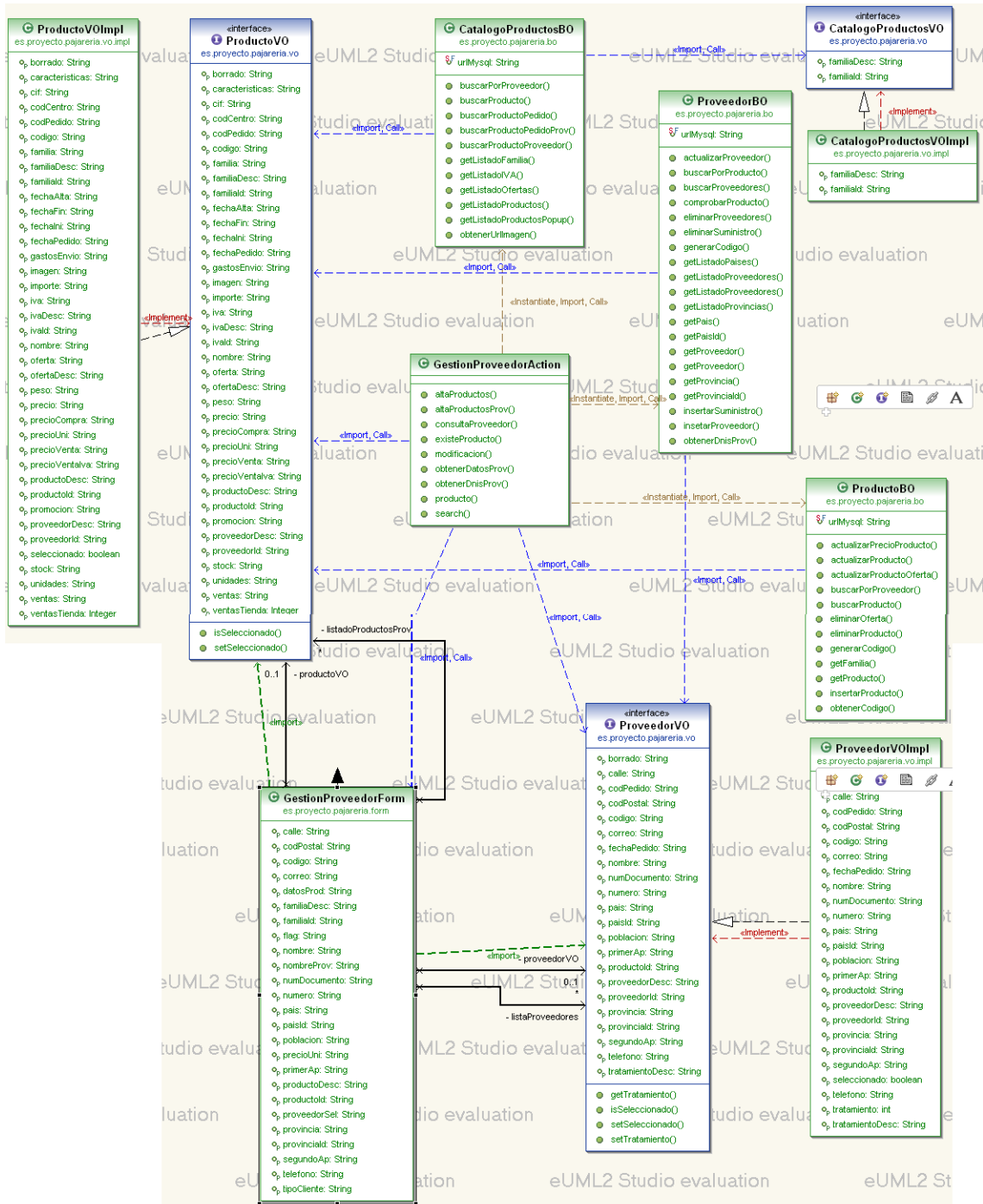


Figura 29. Gestión Proveedor

HerenciaAction

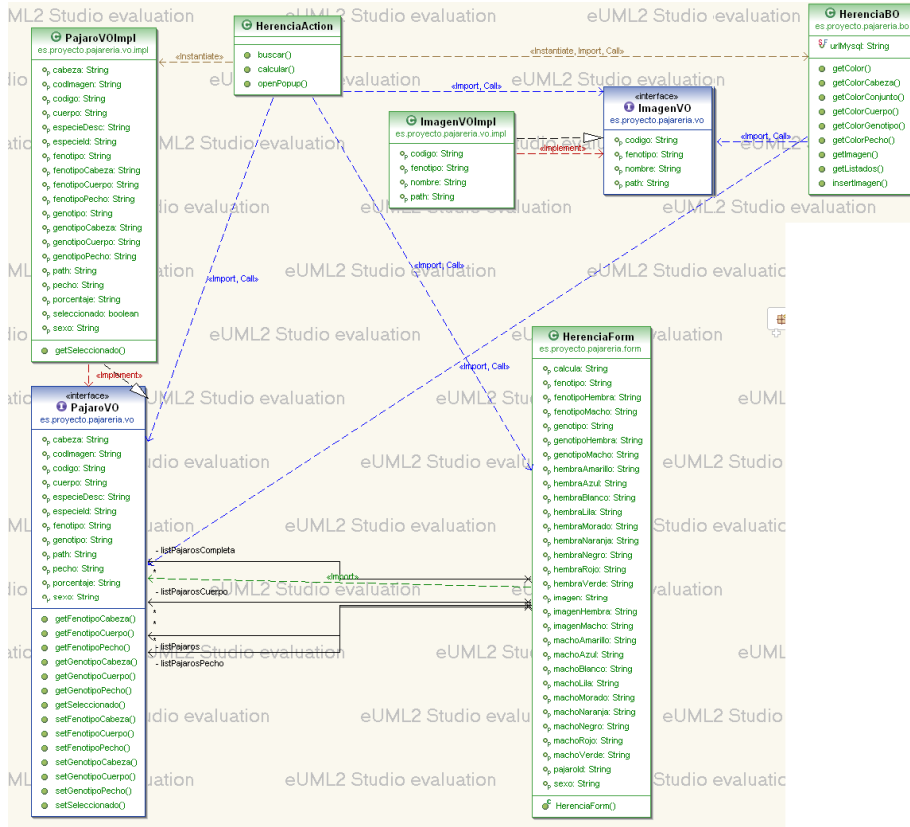


Figura 30. Herencia

LoginAction

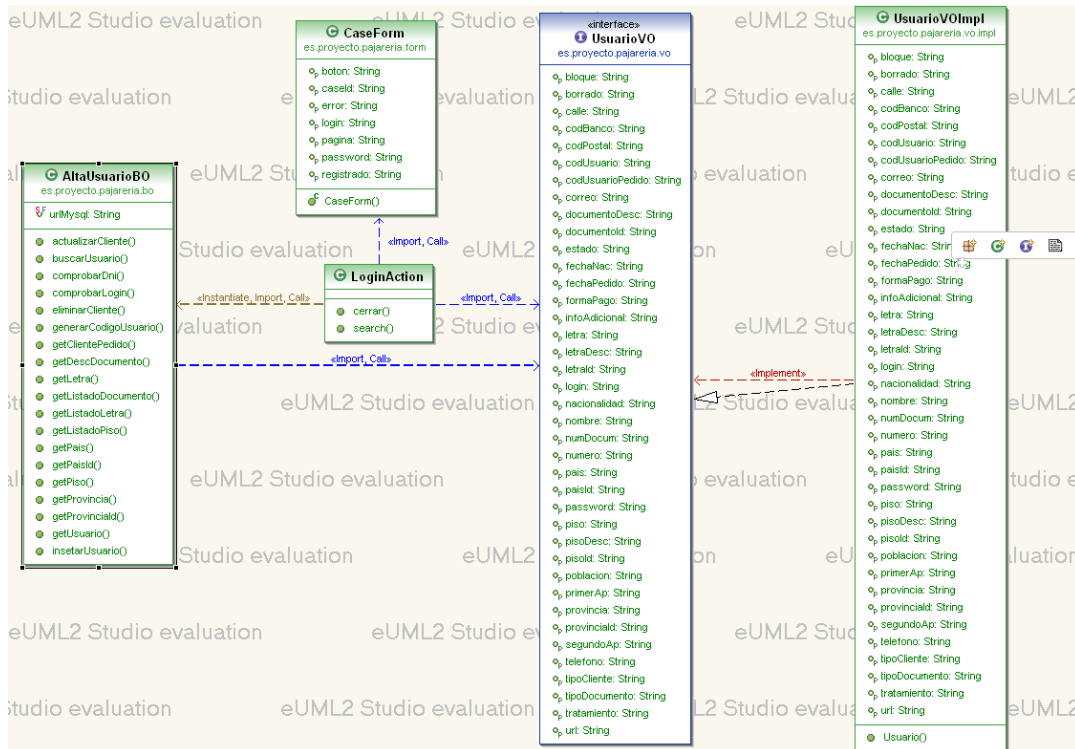


Figura 31. Login

PedidoAction

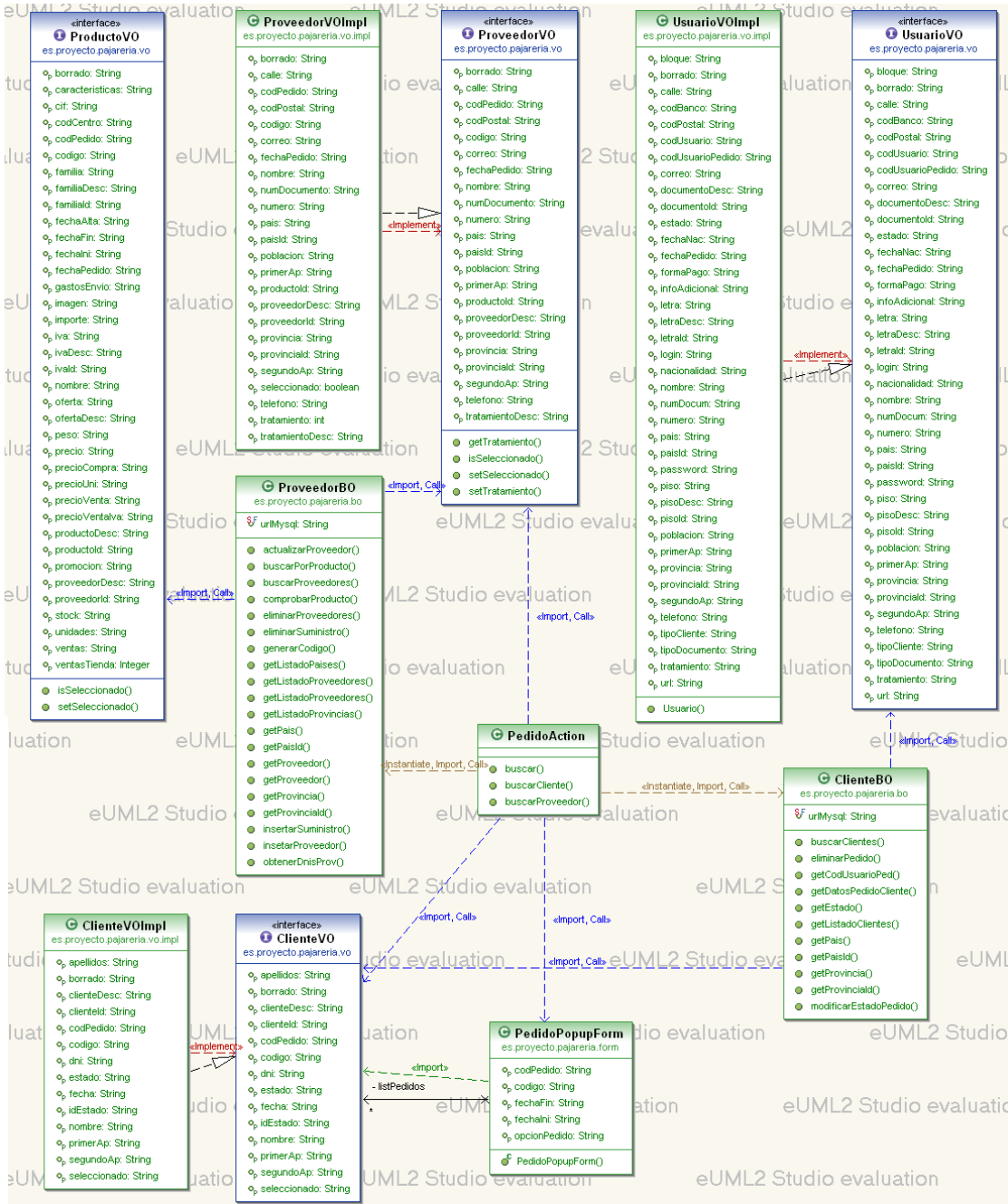


Figura 32. Pedido

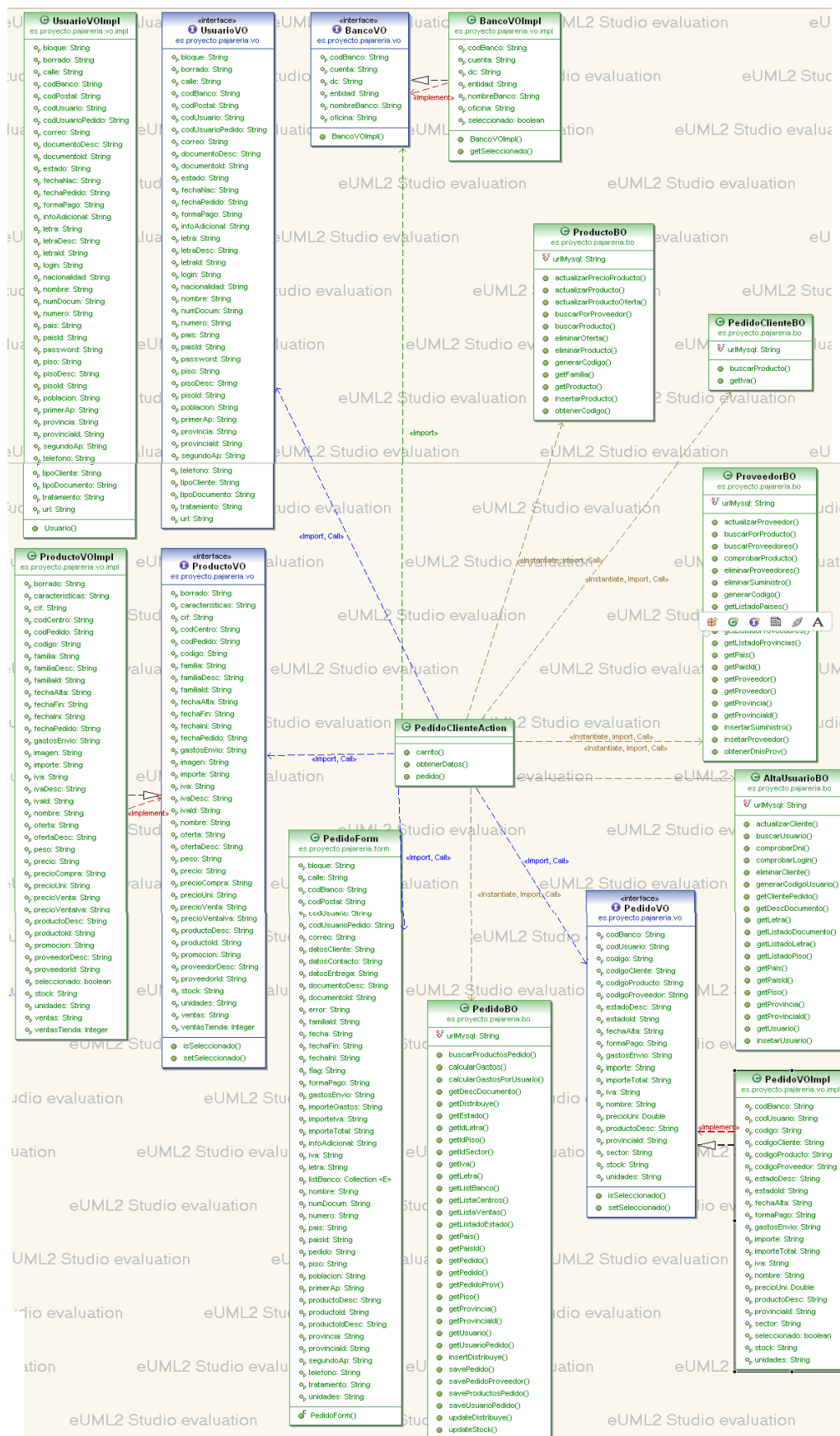


Figura 33. Pedido Cliente

PopUpProductoAction

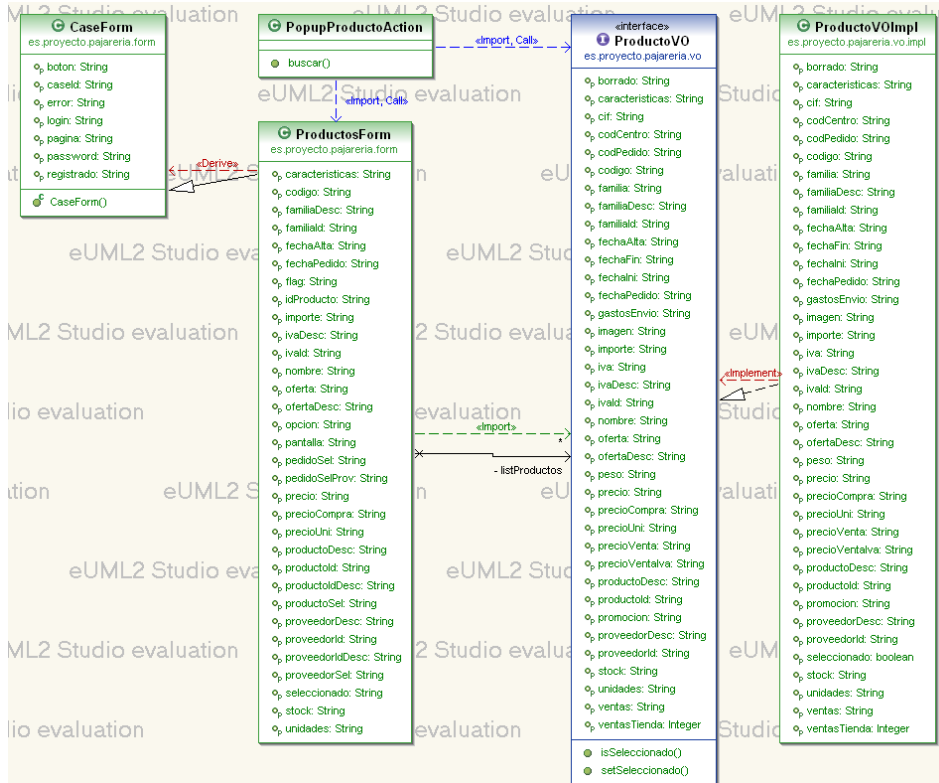


Figura 34. PopUp Producto

ProductosAction

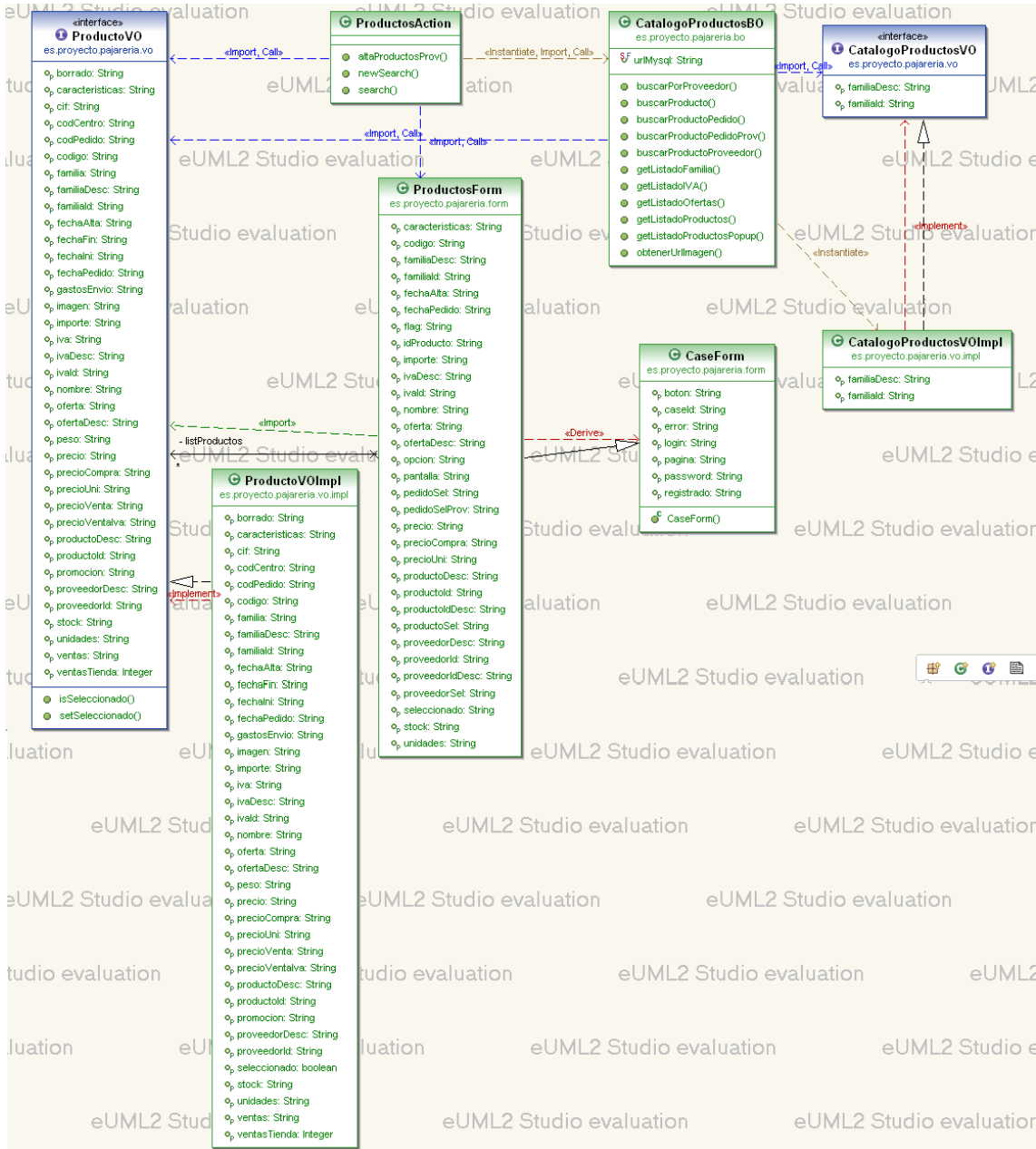


Figura 35. Productos

7. Diseño de la Arquitectura de Módulos del Sistema

Cuando abordamos el desarrollo de una aplicación Java, uno de los primeros requerimientos que se debe tener en cuenta es la integración de la aplicación con una base de datos para guardar, actualizar y recuperar la información que utilizará dicha aplicación (tanto los datos producidos durante la interacción de los usuarios con la aplicación como datos que son mostrados a los usuarios). Por ello necesitamos que nuestra aplicación sea persistente, es decir que sea capaz de guardar y recuperarse desde un medio de almacenamiento (base de datos). La manera de conseguir implementar esto es gracias a la API de Java JDBC que nos permitirá conectar nuestra aplicación a su base de datos. En este apartado se definirán todos los aspectos relativos a la parte de diseño de la aplicación como son el modelo de datos, las clases que conformarán nuestra aplicación así como las relaciones y dependencias entre ellas. Para ello nos basaremos en el modelo estático del sistema.

7.1. Diseño del Modelo Físico de Datos

En este apartado expondremos el diseño físico de nuestro modelo de datos.

Diseño de la Base de Datos

Aquí se definirá la forma o diseño que tendrá nuestra base de datos. Nuestro sistema de base de datos es relacional y estará formado por un conjunto de registros, datos y relaciones que reflejan las necesidades de información de nuestra aplicación. Todos los elementos que conforman nuestras bases de datos están interrelacionados. Las bases de datos relacionales se basan precisamente, en generar un conjunto de esquemas de relaciones, que permitan almacenar la información con un mínimo de redundancia, pero que a la vez faciliten la recuperación de la información. Estas relaciones que agrupan información se conocen como “*tuplas*”.

Entre las cualidades básicas que debe tener nuestro sistema de base de datos se encuentran:

- Evitar la información redundante.
- Mantener la integridad entre los datos almacenados.
- Proporcionar un acceso seguro y fiable a los datos almacenados.
- Ha de permitir captar las relaciones del mundo real.

Por tanto para abordar el diseño de una base de datos se deben seguir las siguientes fases de diseño:

- **Diseño conceptual:** descripción global y genérica de la información sin tener en cuenta las consideraciones físicas. Contiene descripciones de los tipos de datos, relaciones entre ellos y restricciones.
 - Nosotras utilizaremos para el diseño de nuestro esquema conceptual el modelo E-R (entidad relación), que describe los datos como entidades, vínculos (relaciones) y atributos.
- **Diseño lógico:** descripción de la información usando un modelo de datos específico. El siguiente paso en el proceso de diseño consiste en implementar la base de datos con un S.G.B.D. (sistema gestor de base de datos), transformando el modelo conceptual.
- **Diseño físico:** implementación física de la base de datos.

7.1.1. Diseño Conceptual

7.1.1.1. Modelo de Entidad Relación

El diagrama de entidad relación es una herramienta de modelado de datos, a través de la cual estableceremos todas las entidades, relaciones y atributos relevantes en nuestro sistema, y que proporcionará la estructura de nuestra base de datos.

Para ello se basan en los siguientes componentes:

- **Entidad:** cualquier objeto del mundo real que tiene existencia por sí mismo y del que se recoge información de interés que se almacenará en la base de datos. Pueden ser fuertes (no dependen de ninguna otra ya y se representan con un rectángulo de trazo simple), o débiles (su existencia está ligada a otra entidad y se representan con rectángulos de doble trazo).

- **Atributo:** propiedades de una entidad o relación común para todas las ocurrencias de una misma entidad. Están basados en un dominio (conjunto de valores posibles que pueden tomar). Su valor puede ser único o multivaluado.
- **Clave primaria:** atributo que permite identificar a las distintas instancias de una entidad.
- **Relación o interrelación:** expresa una asociación entre dos o más ocurrencias de una entidad. Pueden tener atributos propios. Pueden tener las siguientes cardinalidades (número de ocurrencias de una entidad que pueden asociarse con otra entidad):
 - **1:1:** cada ocurrencia de una entidad A le corresponde como máximo una ocurrencia de la otra entidad B.
 - **1:N:** cada ocurrencia de una entidad A le pueden corresponder varias ocurrencias de la otra entidad B.
 - **N:M:** cada ocurrencia de una entidad A le pueden corresponder varias ocurrencias de la otra entidad B y viceversa.

A continuación se muestra nuestro modelo entidad-relación:

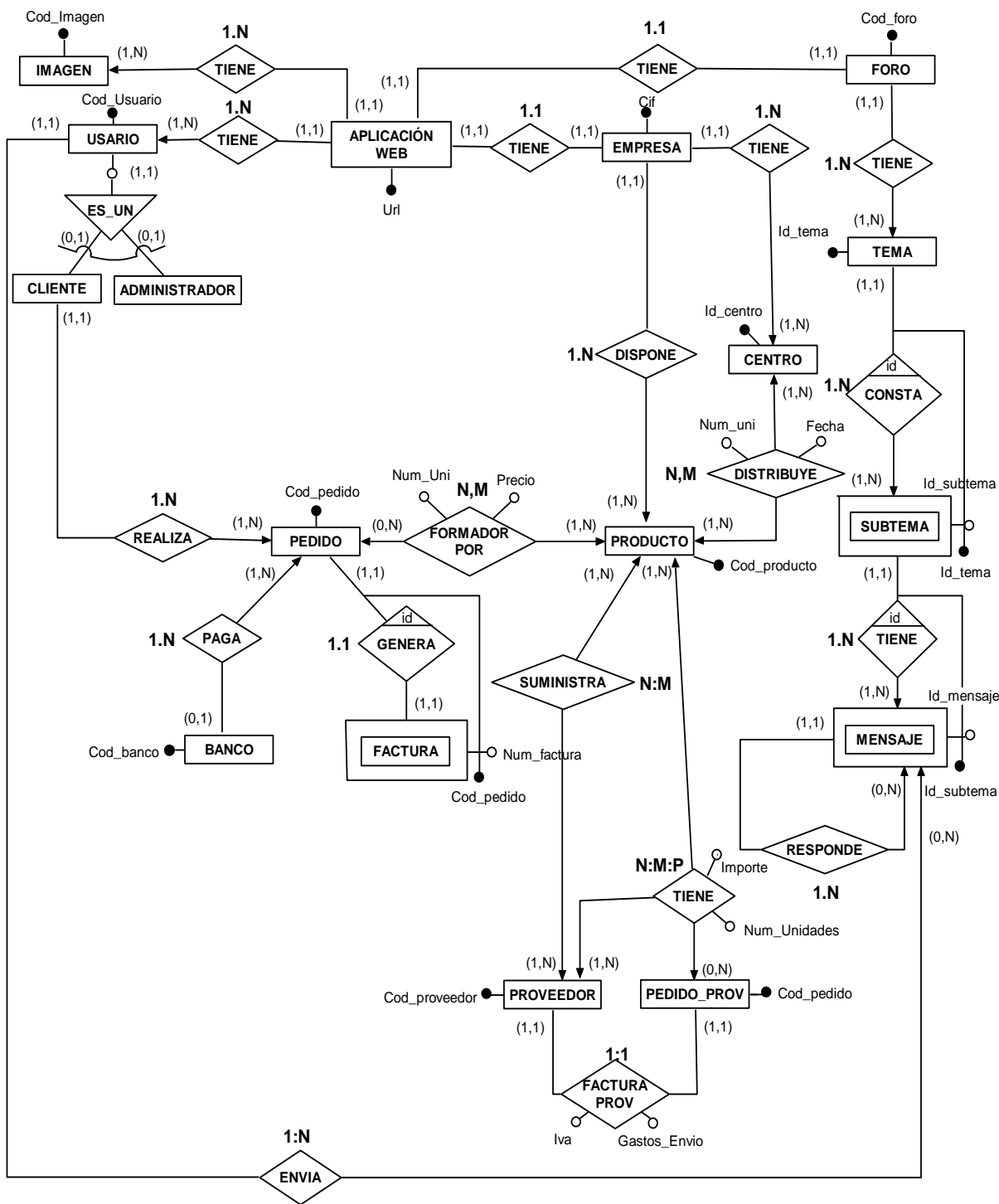


Figura 6.- Diagrama Entidad Relación

DICCIONARIO DE DATOS DEL MODELO ENTIDAD RELACIÓN

ENTIDADES

Entidad Imagen: representa cada foto de las crías que pueden salir de un posible análisis de descendencia genética.

Atributos:

- Cod_Imagen: identificador de la entidad. Tipo: int.(10), clave principal.
- Nombre: nombre con el que se identifica a la imagen. Tipo varchar(40), no nulo.
- Fenotipo: combinación genética necesaria para visualizar esa imagen. Tipo varchar(50), no nulo.
- Path: ruta donde se encuentra ubicada físicamente la imagen. Tipo varchar(90), no nulo.
- Url: representa la url de la aplicación a la que pertenecen las imágenes. Tipo varchar(50), no nulo.

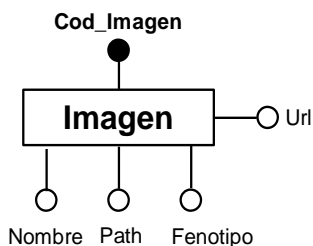


Figura 38. Entidad Imagen

Entidad Aplicacion_Web: representa los datos de la aplicación web.

Atributos:

- Url: identificador de la entidad que contiene la dirección web de la aplicación. Tipo varchar (50), clave principal.
- Nombre: representa el nombre que se asocia a la web de la pajarería. Tipo varchar(50), no nulo.

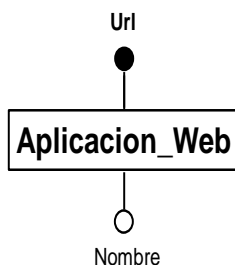


Figura 39. Entidad Aplicacion_Web

Entidad Usuario: representa a los usuarios de la aplicación web

Atributos:

- Cod_Usuario: identificador de la entidad que representa a un usuario que accede a la web. Tipo varchar (10), clave principal.
- Login: usuario de acceso a la web. Tipo varchar(10), no nulo.
- Password: contraseña de acceso del usuario a la web. Tipo varchar(10), no nulo.
- Dni: documento de identificación del usuario. Tipo varchar(9), nulo por defecto.
- Nombre: nombre del usuario. Tipo varchar(20), nulo por defecto.
- Apellido: primer apellido del usuario. Tipo varchar(20), nulo por defecto.
- Apellido2: segundo apellido del usuario. Tipo varchar(20), nulo por defecto.
- Tratamiento: tratamiento que solicita el usuario en la web. Tipo varchar(1), nulo por defecto.
- Telefono: teléfono de contacto. Tipo varchar(9), nulo por defecto.
- Id_Provincia: código de la provincia donde reside el usuario. Tipo int(10), nulo por defecto.

- Ciudad: ciudad donde reside el usuario. Tipo varchar(40) , nulo por defecto.
- Calle: calle donde reside el usuario. Tipo varchar(50) , nulo por defecto.
- Id_Pais: código del país de origen del usuario. Tipo int(2) , nulo por defecto.
- Cp: código postal del lugar donde reside el usuario. Tipo varchar(5) , nulo por defecto.
- Correo: dirección de correo electrónico del usuario. Tipo varchar(40) , nulo por defecto.
- Nacionalidad: nacionalidad del usuario. Tipo varchar(25) , nulo por defecto.
- F_Nac: fecha de nacimiento del usuario. Tipo date , nulo por defecto.
- Tipo: clase de usuario que se está dando de alta (particular o empresa). Tipo char , nulo por defecto.
- Tipo_Documento: tipo de documento que se escoge para identificarse. Tipo varchar(2) , nulo por defecto.
- Numero: numero de la dirección de contacto. Tipo varchar(3), no nulo.
- Bloque: bloque de la dirección de contacto. Tipo varchar(2) , nulo por defecto.
- Info_Adicional: comentarios adicionales que desee añadir el usuario. Tipo varchar(50) , nulo por defecto.
- Id_Piso: número del piso de contacto del usuario. Tipo varchar(2) , nulo por defecto.
- Id_Letra: letra del piso de contacto del usuario. Tipo varchar(2) , nulo por defecto.
- Url: dirección web de la pajarería donde se están dando de alta. Tipo varchar(50) , nulo por defecto.
- Borrado: indica si un usuario ha sido eliminado o se ha dado de baja del sistema.

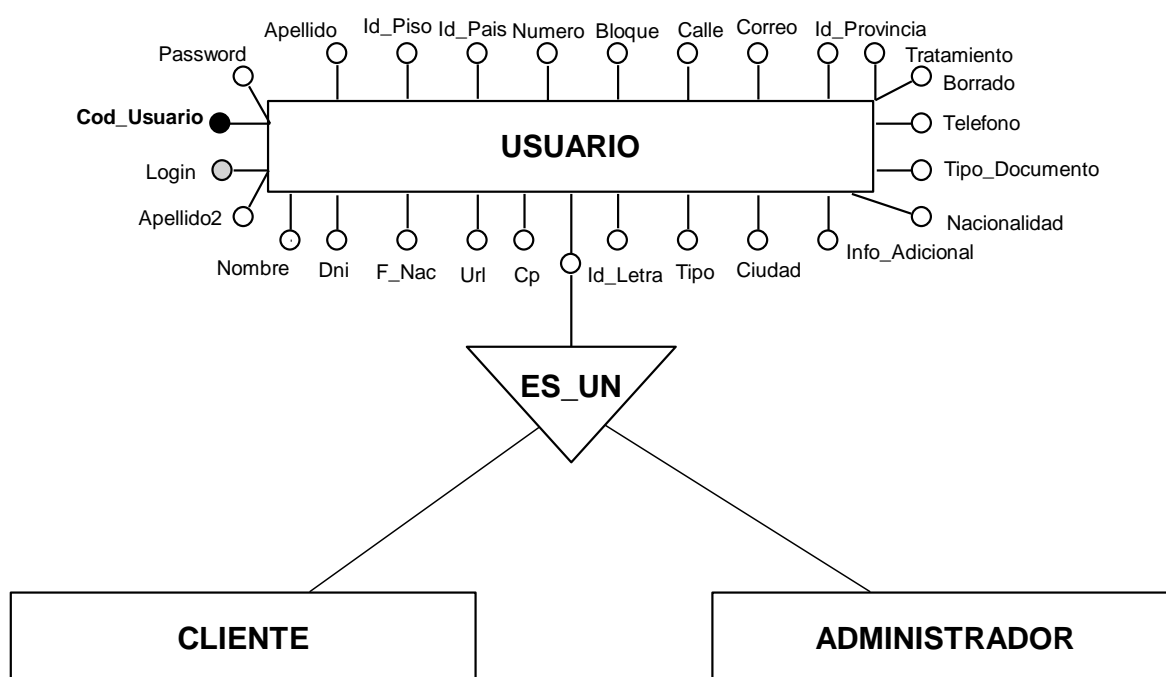


Figura 40. Entidad Usuario

Entidad Empresa: representa la empresa asociada a la web de la pajarería.

Atributos:

- Cif: representa el código de identificación fiscal de la empresa. Tipo varchar (9), clave principal.
- Nombre: nombre de la empresa. Tipo varchar(20), no nulo.
- Id_Provincia: código de la provincia donde se ubica la empresa. Tipo int(10).
- Ciudad: ciudad donde se ubica la empresa. Tipo varchar(40), no nulo.
- Calle: calle donde se sitúa la empresa. Tipo varchar(50), no nulo.
- Id_Pais: código del país donde se ubica la empresa. Tipo int(2).
- Cp: código postal correspondiente a la dirección donde se ubica la empresa. Tipo varchar(5), no nulo.
- Correo: correo electrónico de contacto con la empresa. Tipo varchar(40), no nulo.
- Url: web de la pajarería a la que está asociada la empresa. Tipo varchar(50), no nulo.

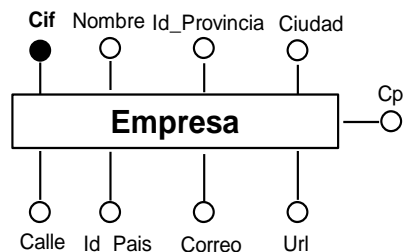


Figura 41. Entidad Empresa

Entidad Foro: representa al foro de la aplicación donde los usuarios intercambian información.

Atributos:

- Cod_Foro: representa el código de identificación del foro de la aplicación. Tipo varchar (2), clave principal.
- Nombre_Foro: representa el nombre del foro de la empresa. Tipo varchar(10), no nulo.
- Url: web de la pajarería a la que pertenece el foro. Tipo varchar(50), no nulo.

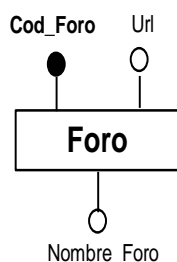


Figura 42. Entidad Foro

Entidad Tema: representa cada tema de conversación que existe dentro del foro.

Atributos:

- Id_Tema: representa el código de identificación de cada uno de los temas del foro. Tipo varchar (2), clave principal.
- Nombre_Tema: representa el nombre del foro de la empresa. Tipo varchar(20), no nulo.
- Cod_Foro: código de identificación del foro de la aplicación. Tipo varchar(2).

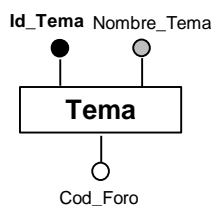


Figura 43. Entidad Tema

Entidad Subtema: representa al foro de la aplicación donde los usuarios intercambian información.

Atributos:

- Id_Subtema: representa el código de identificación del subtema del foro de la aplicación. Tipo varchar (2), clave principal.
- Id_Tema: representa el código de identificación del tema al que pertenece el. Tipo varchar(2), clave principal.
- Nombre_Subtema: nombre del subtema del foro de la aplicación. Tipo varchar(20), no nulo.
- Características: muestra la ruta donde se encuentra el fichero que se cargará para el subtema seleccionado. Tipo varchar(50), no nulo.

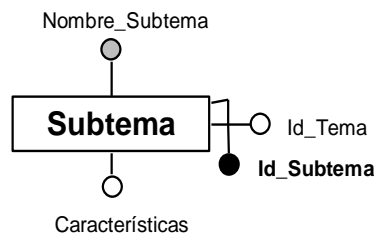


Figura 44. Entidad Subtema

Entidad Mensaje: representa al foro de la aplicación donde los usuarios intercambian información.

Atributos:

- Id_Mensaje: representa el código de identificación del mensaje que el usuario está creando en el foro de la aplicación. Tipo varchar (2), clave principal.
- Id_Subtema: representa el código de identificación del subtema en el cual se está escribiendo el mensaje. Tipo varchar (2), clave principal.
- Id_Tema: representa el código de identificación del tema en el cual se está escribiendo el mensaje. Tipo varchar(2), clave principal.
- Nombre_Mensaje: nombre del mensaje del foro de la aplicación. Tipo varchar(20), no nulo.
- Descripción: muestra el contenido del mensaje que se ha generado en el foro. Tipo varchar(100), no nulo.
- Id_Mensaje_Res: muestra el identificador del mensaje al que está asociado un mensaje. Tipo varchar(2), no nulo.
- Cod_Usuario: indica el código del usuario que ha creado un mensaje. Tipo varchar(10) , nulo por defecto.

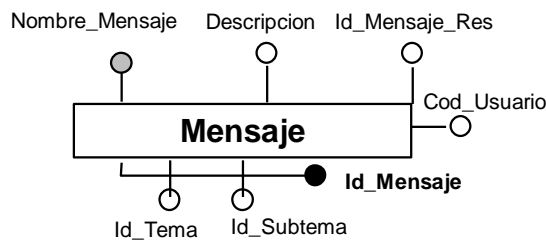


Figura 45. Entidad Mensaje

Entidad Centro: representa un centro o tienda de la empresa.

Atributos:

- Id_Centro: código que identifica a cada centro asociado a la empresa. Tipo int(2), clave principal.
- Nombre_Centro: nombre de la tienda. Tipo varchar(20), no nulo.
- Id_Provincia: provincia en la cual se encuentra la tienda. Tipo int (2), no nulo.
- Ciudad: ciudad donde se sitúa el centro. Tipo varchar(40), no nulo.
- Calle: calle donde se encuentra la tienda. Tipo varchar(50), no nulo.
- Numero: número donde se ubica el centro. Tipo varchar(3), no nulo.
- Bloque: muestra el identificador del bloque donde está ubicada la tienda. Tipo varchar(2), nulo por defecto.
- Id_Pais: país en el que se encuentra la tienda. Tipo int(2) , nulo por defecto.
- Cp: código postal correspondiente a la ubicación donde está la tienda. Tipo varchar(5), no nulo.
- E-Mail: dirección de correo electrónico de la tienda. Tipo varchar(40), no nulo.
- Telefono: teléfono de contacto del centro. Tipo varchar(9), no nulo.
- Cif: documento de identificación fiscal de la empresa a la que pertenece la tienda. Tipo varchar(9), no nulo.

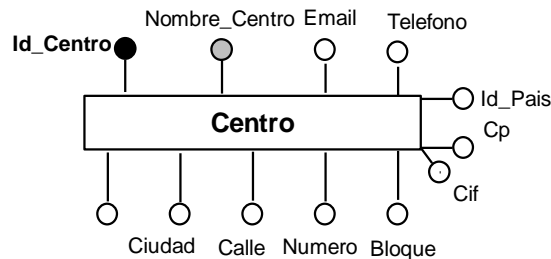


Figura 46. Entidad Centro

Entidad Producto: artículo de venta en la web de la pajarería.

Atributos:

- Cod_Producto: código que identifica a cada producto. Tipo varchar (6), clave principal.
- Nombre_Producto: nombre del producto. Tipo varchar(30), no nulo.
- Fecha: fecha en la que se ha dado de alta el producto en el sistema. Tipo date, no nulo.
- Precio_Compra: precio a l cual la empresa ha comprado el producto. Tipo decimal(6,2), no nulo.
- Precio_Venta: precio de venta al público del producto . Tipo decimal(6,2), nulo por defecto.
- Id_Familia: muestra el identificador de la familia a la que pertenece el producto. Tipo varchar(2), no nulo.
- Características: descripción de las características del producto. Tipo varchar(20) , nulo por defecto.
- Oferta: indica si el producto está en oferta o no. Tipo varchar(3), no nulo.
- Stock: cantidad de unidades disponibles del producto. Tipo varchar(9), no nulo.
- Id_Iva: identificador del iva que se aplicará al producto. Tipo char, nulo por defecto.
- Cif: identificador fiscal de la empresa donde se vende el producto. Tipo varchar(9), no nulo.
- Promoción: identifica si el producto está en promoción. Tipo char(1), no nulo.
- Cod_Imagen: identificador de la imagen asociada al producto. Tipo int(10), nulo por defecto.
- Peso: peso del producto. Tipo decimal decimal(5,2).
- Borrado: identificador de un producto cuando se ha dado de baja en el catálogo de la empresa. Tipo char no nulo siendo su valor por defecto 'N'.

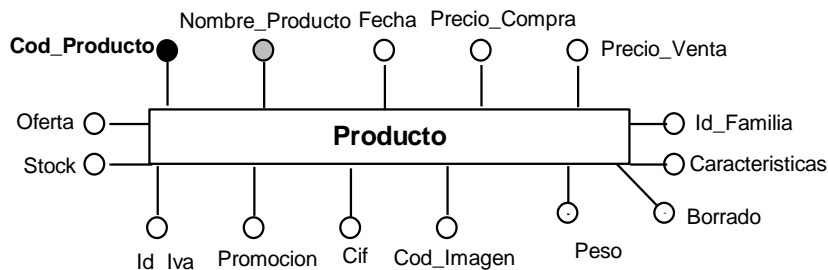


Figura 47. Entidad Producto

Entidad Pedido_Prov: representa el pedido que una tienda hace a un proveedor.

Atributos:

- Cod_Pedido: código que identifica cada pedido que se hace a un proveedor. Tipo int (6), clave principal. Se autoincrementa su valor.
- Fecha: fecha en la que se realiza el pedido. Tipo date, no nulo.

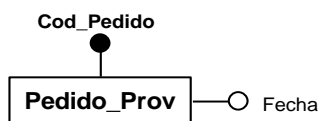


Figura 48. Entidad Pedido_Prov

Entidad Proveedor: representa a un proveedor asociado a la empresa.

Atributos:

- Cod_proveedor: código que identifica a un proveedor. Tipo varchar (6), clave principal.
- Dni: representa al código de identificación del proveedor. Tipo varchar(9), no nulo.
- Nombre: nombre del proveedor. Tipo varchar(20), no nulo.
- Apellido: primer apellido del proveedor. Tipo varchar(20), nulo por defecto.
- Apellido2: segundo apellido. Tipo varchar(20), nulo por defecto.
- Correo: dirección de correo electrónico del proveedor. Tipo varchar(40), nulo por defecto.
- Telefono: número de teléfono de contacto. Tipo varchar(9), no nulo.
- Id_Provincia: código de la provincia del proveedor. Tipo int(2), no nulo.
- Ciudad: ciudad donde se ubica el proveedor. Tipo varchar(40), no nulo.
- Calle: dirección de localización del proveedor. Tipo varchar(50), no nulo.
- Id_Pais: código del país del proveedor. Tipo int(2), nulo por defecto.
- Cp: código postal de la dirección de contacto del proveedor. Tipo varchar(5), no nulo.
- Numero: numero del centro del proveedor. Tipo varchar(3), no nulo.
- Tipo: representa el tipo de proveedor. Tipo char(1), no nulo.
- Borrado: identificador de un proveedor cuando se ha dado de baja como proveedor de la empresa. Tipo char no nulo siendo su valor por defecto 'N'.

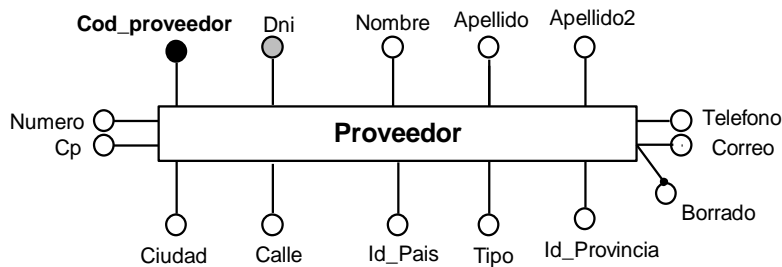


Figura 49. Entidad Proveedor

Entidad Pedido: representa un pedido que realiza un cliente a través de la web de la pajarería.

Atributos:

- Cod_Pedido: código que identifica a un pedido que realiza un cliente a través de la web. Tipo int (6), clave principal. Se autoincrementa.
- Fecha: representa la fecha en la que se ha realizado el pedido. Tipo date, no nulo.
- Id_Estado: estado en el que se encuentra un pedido. Tipo varchar(2), no nulo.
- Forma_Pago: método de pago elegido por el cliente. Tipo char, no nulo.
- Cod_Banco: código que identifica una cuenta asociada al usuario. Tipo varchar(4), nulo por defecto.
- Cod_Usuario: código del usuario que realiza el pedido. Tipo varchar(10), no nulo.
- Cod_Usuario_Ped: código asignado a un usuario que realiza un pedido. Tipo int(10), no nulo.
- Id_Gastos: representa el código que generará los gastos de envío del pedido. Tipo varchar(2) nulo por defecto.
- Importe: importe total del pedido. Tipo decimal(7,2), no nulo.

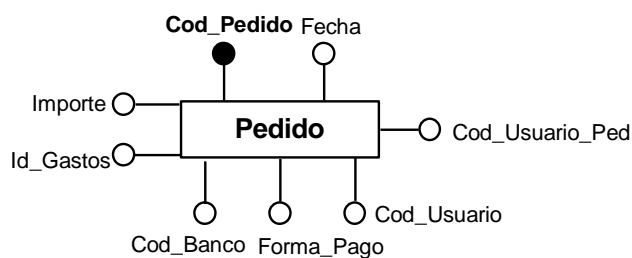


Figura 50. Entidad Pedido

Entidad Factura: representa una factura de un pedido.

Atributos:

- Num_Factura: código que identifica a cada factura. Tipo int (4), clave principal. Se autoincrementa.
- Cod_Pedido: código que identifica un pedido realizado a través de la web. Tipo int(6), clave principal.
- Id_Sector: id de los gastos de envío que se aplicarán a un pedido. Tipo varchar(2), nulo por defecto.
- Importe: importe total del pedido. Tipo decimal (7,2), no nulo.

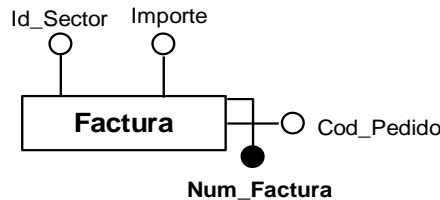


Figura 51. Entidad Factura

Entidad Banco: representa los datos bancarios de las cuentas asociadas a la pajarería donde los clientes podrán realizar transferencias bancarias de los pedidos que realicen.

Atributos:

- Cod_Banco: código que identifica cada cuenta bancaria. Tipo varchar(4), clave principal. Se autoincrementa.
- Nombre: nombre del banco al que pertenece la cuenta. Tipo varchar(20), no nulo.
- Entidad: entidad donde se abrió la cuenta bancaria. Tipo varchar(4), no nulo.
- Oficina: oficina donde se abrió la cuenta bancaria. Tipo varchar(4), no nulo.
- Dc: dígito de control de la cuenta. Tipo varchar(2), no nulo.
- Cuenta: número de cuenta. Tipo varchar(10), no nulo.

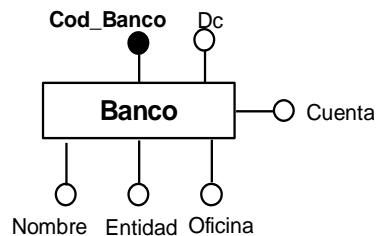


Figura 52. Entidad Banco

RELACIONES

Relación Tiene (Imagen-Applicación web):

- Descripción: Relación entre la entidad Imagen y Aplicación web. Una aplicación web contendrá una serie de imágenes necesarias para la visualización de los diamantes de Gould del programa Mendel, así como de los productos que se venden a través de la web.
- Restricciones: A una aplicación web le pertenecen varias imágenes o una como mínimo y una imagen pertenece a una sola aplicación. Las cardinalidades entre las entidades imagen y aplicación web serán (1,N) y (1,1) con una interrelación Tiene de tipo 1:N.
- Atributos: Ninguno.

Relación Tiene (Usuario-Aplicación web):

- Descripción: Los usuarios de la aplicación podrán identificarse en la web por medio del login y password con el que se dieron de alta.
- Restricciones: Dado que la aplicación web puede tener como mínimo un usuario o múltiples, las cardinalidades entre las entidades Usuario y Aplicación web serán (1,N) y un usuario concreto pertenece a una Aplicación web (1,1). Por tanto la interrelación Tiene que existe entre ambas entidades es de tipo 1:N.
- Atributos: Ninguno.

Relación Es_un (Usuario-Cliente-Administrador):

- Descripción: Respecto a la entidad Usuario podemos decir que existen dos tipos específicos de usuarios (aquellos que necesitan identificarse en la web con un login de usuario y password), que son cliente o administrador.
- Restricciones: La entidad usuario se modula como una jerarquía total y exclusiva ya que cada usuario sólo puede ser o un cliente o un administrador de la aplicación pero no ambos a la vez. Las cardinalidades entre el supertipo (Usuario) y el subtipo (Cliente o Administrador) serán de tipo (1,1) y (0,1) respectivamente.
- Atributos: Ninguno.

Relación Tiene (Aplicación-Empresa):

- Descripción: La aplicación web está asociada a una única empresa y a una empresa le pertenece una aplicación web.
- Restricciones: Las cardinalidades entre las entidades Empresa y Aplicación web serán (1,1) y (1,1) con una interrelación Tiene de tipo 1:1.
- Atributos: Ninguno.

Relación Tiene (Empresa-Centros):

- Descripción: Una empresa puede tener más de una tienda ubicadas en diferentes ciudades, como mínimo tendrá una. Sin embargo cada tienda pertenece a una única empresa.
- Restricciones: Por tanto las cardinalidades entre las entidades Centro y Empresa serán (1,N) y (1,1) respectivamente, con una interrelación Tiene de tipo 1:N.
- Atributos: Ninguno.

Relación Distribuye (Centro-Producto):

- Descripción: Cada centro o tienda distribuyen una serie de productos disponibles en la empresa. Cada centro puede distribuir varios productos y un producto puede ser distribuido como mínimo en un centro o por varios centros.
- Restricciones: Las cardinalidades entre las entidades Centro y Producto serán (1,N) y (1,N) con una interrelación Distribuye de tipo N:M donde además debemos almacenar el número de unidades distribuidas a cada centro y la fecha.
- Atributos: Num_uni, Fecha.

Relación Dispone (Empresa-Producto):

- Descripción: Una empresa puede distribuir muchos productos a través de sus centros. Cada producto sólo puede venderse en una única empresa.
- Restricciones: Por otro lado las cardinalidades entre las entidades Empresa y Producto serán (1,N) y (1,1) con una interrelación Dispone de tipo 1:N.
- Atributos: Ninguno.

Relación Tiene (Aplicación web-Foro):

- Descripción: Los clientes de la aplicación pueden intercambiar información a través de un foro que contiene la aplicación, o consultar entre otras cosas las características especiales de varias especies de aves. Una aplicación web tiene como máximo un foro mientras que cada foro es utilizado en exclusiva por una única aplicación.
- Restricciones: Las cardinalidades entre las entidades Foro y Aplicación web serán (1,1) y (1,1) con una interrelación Tiene de tipo 1:1.
- Atributos: Ninguno.

Relación Tiene (Foro-Tema):

- Descripción: El foro está dividido en varios temas que son habilidades, enfermedades y curiosidades. Cada tema está asociado a un único foro y en cada foro puede haber muchos temas, como mínimo uno.
- Restricciones: Por tanto las cardinalidades entre las entidades Foro y Tema serán (1,1) y (1,N) con una interrelación Tiene de tipo 1:N.
- Atributos: Ninguno.

Relación Consta (Tema-Subtema):

- Descripción: Por cada tema del foro vamos a tener varias especies de aves (acerca de las cuales se puede consultar la información requerida), a las que llamaremos subtemas del foro. La relación entre las entidades Tema y Subtema va a ser de entidad fuerte entidad débil dado que para identificar cada subtema necesitamos el identificador del tema al que está asociado. Un tema puede contener muchos subtemas (uno como mínimo), mientras que un subtema está asociado a un único tema del foro.
- Restricciones: Las cardinalidades son (1,1) y (1,N) con una interrelación Consta de tipo 1:N.
- Atributos: Ninguno.

Relación Tiene (Subtema-Mensaje):

- Descripción: Por cada subtema del foro puede haber mensajes asociados aunque también puede no tener ningún mensaje. Sin embargo un mensaje pertenece a un único subtema. Dado que un mensaje necesita el código del subtema al que está asociado para identificarse la relación entre las entidades será de entidad fuerte entidad débil. Cada mensaje vendrá identificado por el código del subtema al cual pertenece, el código del tema y un código del mensaje.
- Restricciones: Las cardinalidades entre las entidades Subtema y Mensaje mediante la relación Tiene serán (1,1) y (0,N) respectivamente y la relación Tiene será (1:N).
- Atributos: Ninguno.

Relación Envía (Usuario- Mensaje):

- Descripción: Un usuario de la aplicación podrá enviar varios mensajes en cualquiera de los temas del foro, cada uno de los cuales es creado por un único usuario.
- Restricciones: Por tanto las cardinalidades entre las entidades Usuario y Mensaje serán (1,1) y (0,N), y la relación Envía será de tipo 1:N.
- Atributos: Ninguno.

Relación Realiza (Usuario-Pedido):

- Descripción: Los clientes pueden realizar pedidos a través de la web de la aplicación. Cada cliente dado de alta en la web puede no haber realizado ningún pedido todavía o varios pedidos pero cada pedido sólo puede ser realizado por un cliente.
- Restricciones: Por tanto las cardinalidades entre las entidades Usuario y Pedido serán (1,1) y (0,N) respectivamente y la relación Realiza será de tipo 1:N.
- Atributos: Ninguno.

Relación Formado_por (Pedido-Producto):

- Descripción: Un pedido está formado por los productos que el cliente ha comprado, como mínimo un producto, pero cada tipo de producto puede formar parte de muchos pedidos o no haber sido adquirido por ningún cliente.
- Restricciones: Las cardinalidades entre las entidades Pedido y Producto son (0,N) y (1,N) respectivamente y la cardinalidad de la relación Formado_por es de tipo (N:M), donde además habrá que añadir los atributos precio al cuál se vendió el producto y número de unidades que se compró de cada producto.
- Atributos: Precio, Num_Uni.

Relación Genera (Pedido-Factura):

- Descripción: Cada pedido genera una factura y a su vez cada factura está asociada a un único pedido. La entidad Factura será una entidad débil respecto a la entidad pedido y necesita del código del pedido para identificarse además de su número de pedido.
- Restricciones: Las cardinalidades entre las entidades Pedido y Factura serán (1,1) y (1,1) respectivamente y la cardinalidad de la relación Genera será (1:1).

- Atributos: Ninguno.

Relación Paga (Pedido-Banco):

- Descripción: El cliente puede realizar el pago de un pedido mediante una transferencia bancaria a cualquiera de las cuentas asociadas a la empresa. Sin embargo el pago mediante transferencia bancaria no es la única forma de pago de un pedido. Por tanto podemos decir que un pedido puede ser pagado o no por transferencia bancaria y una transferencia bancaria pertenece al pago de uno o incluso varios pedidos.
- Restricciones: Las cardinalidades entre las entidades Pedido y Banco mediante la relación Paga serán (1,N) y (0,1) respectivamente y la relación Paga será (1:N).
- Atributos: Ninguno.

Relación Suministra (Proveedor-Producto):

- Descripción: Los proveedores suministran productos al almacén. Cada proveedor suministrará una serie de productos los cuales pueden a la vez ser suministrados por varios proveedores. Entonces podemos decir que un proveedor suministrará como mínimo un único producto o varios a la empresa mientras que un producto que se venda en la web puede ser proporcionado por un proveedor o varios.
- Restricciones: Por tanto las cardinalidades entre las entidades Proveedor y Producto mediante la relación Suministra serán (1,N) y (1,N) respectivamente y la relación Suministra será (N:M).
- Atributos: Ninguno.

Relación Tiene (Proveedor-Producto-Pedido_prov):

- Descripción: El administrador puede realizar un pedido de productos a distintos proveedores, pero a uno como mínimo de un producto. Un pedido estará formado por varios productos que pueden haber sido suministrados por diferentes proveedores. Sin embargo no todos los productos formaran parte de un mismo pedido, por lo que habrá productos que no estén en un pedido concreto. Se establece una relación ternaria entre las entidades Producto, Proveedor y Pedido_prov.
- Restricciones: Por tanto las cardinalidades entre las entidades Pedido_prov, Producto y Proveedor serán (0,N) y (1,N) (1,N) respectivamente y la relación ternaria Tiene será de tipo N:M:P.
- Atributos: La relación Tiene tendrá atributos el Importe del pedido y el Numero de unidades de cada producto que pertenece a un pedido hecho a un proveedor.

Relación Factura_prov (Proveedor- Pedido_prov):

- Descripción: El proveedor generará una factura asociada a cada pedido que la empresa le realiza. Una factura se corresponderá con un único pedido hecho a un proveedor y un pedido corresponderá a un único proveedor.
- Restricciones: Por tanto Las cardinalidades entre las entidades Pedido_prov y Proveedor serán (1,1) y (1,1) respectivamente con una relación Factura_prov que será de tipo 1:1.
- Atributos: la relación tendrá unos atributos Gastos de envío del pedido e Iva asociado a éste.

DOMINIOS:

- varchar: representa el tipo de dato cadena de caracteres.
- int: representa el tipo de datos entero.
- Decimal: representa el tipo de dato entero que admite decimales.
- Date: representa el tipo de dato que se utiliza para almacenar fechas.

RESTRICCIONES GLOBALES

- Si se elimina un cliente también se borrarán todos los pedidos que ha realizado a través de la web.
- Si se elimina un país también se borrarán todas las provincias pertenecientes a él.
- Al eliminar una cuota de gasto de envío se eliminarán los sectores donde se aplica ese gasto de envío.
- Si se elimina el un tema del foro se borrarán automáticamente todos los subtemas y mensajes englobados en esa categoría. De igual manera si lo que se elimina es un subtema del foro, los mensajes que existan abiertos en él, serán borrados de nuestra base de datos.
- Cuando se da de baja un proveedor, se borrará el catálogo de productos que tiene asociado con la tienda.
- Cuando se cierra una tienda o centro se la da de baja en el sistema y automáticamente también se elimina su catálogo de productos.

7.1.2. Diseño Lógico

En esta etapa, se transformará el esquema conceptual en un esquema lógico, que permite que se puedan representar de manera eficiente posibles recursos, para estructurar datos y modelar las restricciones disponibles en el modelo lógico. Conforme se va desarrollando el esquema lógico, éste se debe ir probando y validando que cumple con los requisitos definidos para la aplicación. Posteriormente se realizará sobre el diseño lógico de nuestra base de datos, un proceso de *normalización*. Este proceso se usa para comprobar la validez de los esquemas lógicos basados en el modelo relacional, ya que asegura que las relaciones (tablas) obtenidas no tienen datos redundantes.

El esquema lógico condicionará el diseño físico de nuestra base de datos.

MODELO RELACIONAL

Este modelo permite representar la información del mundo real de manera intuitiva. Además mantiene información sobre las características de las base de datos (metadatos). Establece una estructura sobre los datos, aunque estos puedan ser posteriormente almacenados de muchas maneras para aprovechar las características físicas de la máquina donde vaya a ubicarse la base de datos.

Las bases de datos relacionales se definen por un conjunto de tablas formadas por distintos campos, que se relacionan entre sí por un campo común denominado identificador o clave.

Los elementos empleados para representar este modelo son:

- **Tablas:** también denominadas relaciones y que tienen que tener un nombre único.
- **Atributos:** cada columna de una tabla y que deben tener un nombre único dentro de la misma tabla. Todos los datos almacenados en la misma columna serán del mismo tipo o pertenecerán al mismo dominio.
- **Dominio:** valores que puede tomar un atributo.
- **Tupla:** fila de una tabla. No pueden existir dos filas iguales dentro de la misma tabla.

Para manejar estas bases de datos se utiliza un software que se conoce como SGBD (sistema gestor de base de datos: conjunto de programas que permite a los usuarios de la Base de Datos definir, crear, mantener y controlar el acceso a la Base de Datos).

Las bases de datos relacionales pasan por un proceso que se le conoce como normalización de una base de datos, que consiste en optimizar el uso que se hace de la base de datos.

Las ventajas del uso de este modelo son:

- Garantiza herramientas para evitar la duplicidad de registros, a través de campos claves o llaves.
- Garantiza la integridad referencial: esto implica que al eliminar un registro se eliminarán automáticamente todos los registros relacionados dependientes.
- Favorece la normalización por ser más comprensible y aplicable.

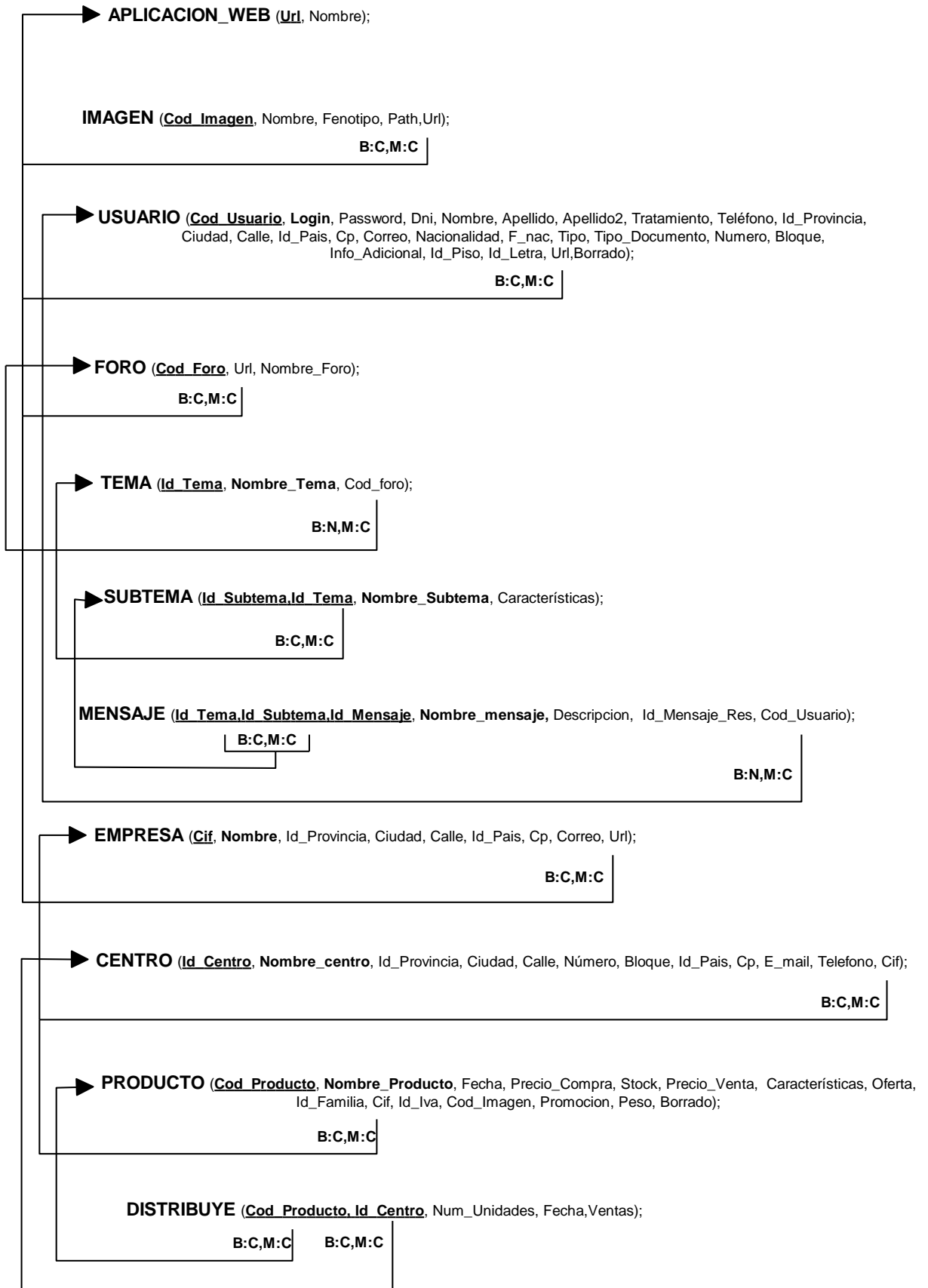
GRAFO RELACIONAL

Denominamos grafo relacional a un esquema o dibujo compuesto por un conjunto de nodos donde cada uno representa una tabla de nuestra base de datos (relación).

Las claves primarias se representan subrayadas, las alternativas en negrita sin estar subrayadas y por último las claves ajenas se representan con arcos. También se deberá definir cómo actuará nuestro sistema de base de datos cuando se elimina una clave primaria que es clave ajena de otra tabla (borrado en cascada, restringido....).

En el grafo se han representado las relaciones principales mientras que las relaciones que no proporcionan ningún valor o información adicional no han sido representadas, como por ejemplo la tabla que contiene las provincias.

A continuación mostramos los grafos relacionales correspondientes a la base de datos nuestra aplicación.



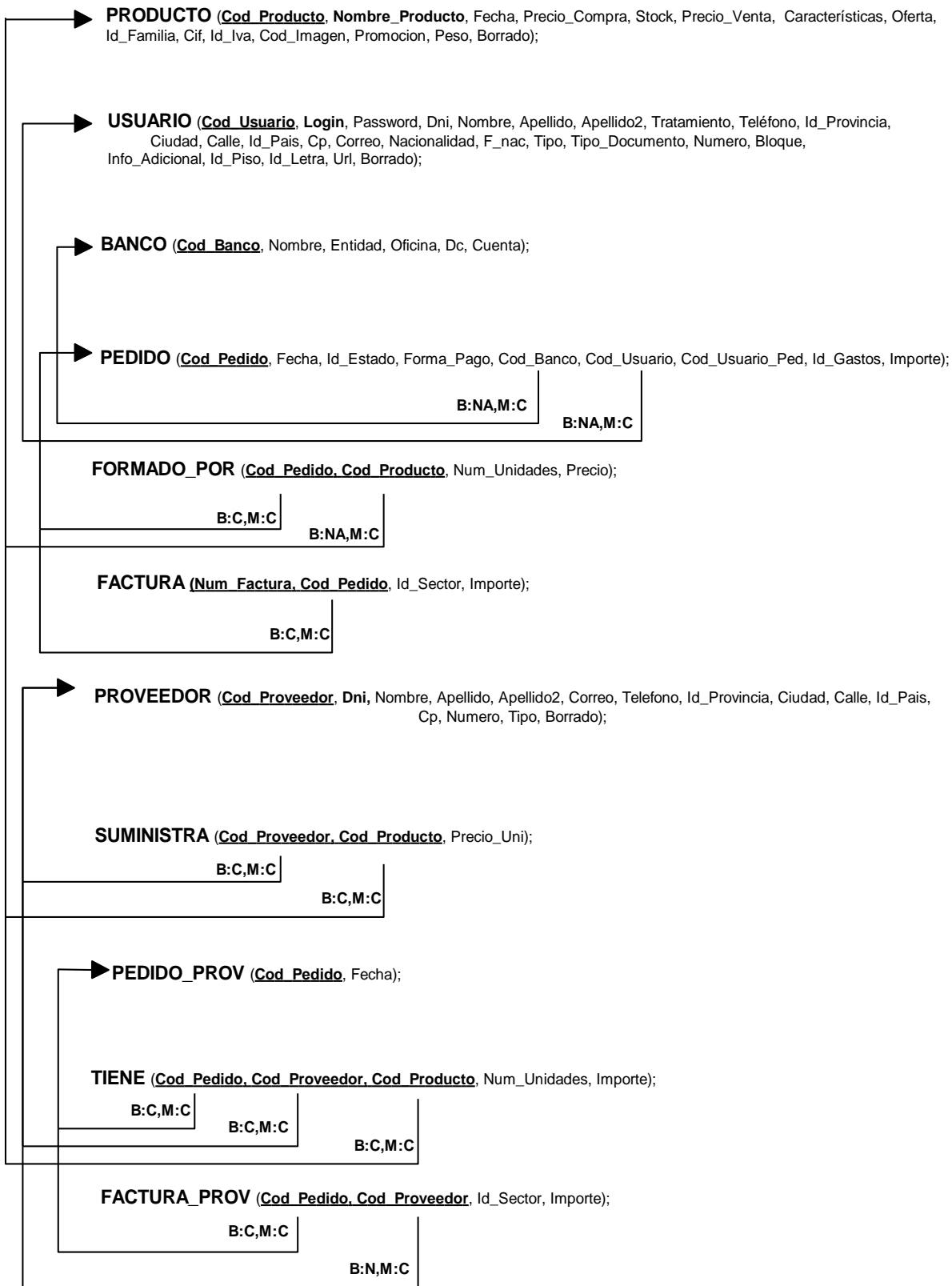


Figura 53. Grafo Relacional

A continuación pasamos a analizar los borrados y modificaciones que se producen entre las relaciones.

Relación IMAGEN atributos Url:

La entidad IMAGEN se relaciona con la entidad APLICACIÓN WEB mediante la interrelación TIENE. Es un tipo de interrelación 1:N, siendo url clave ajena de la relación imagen respecto a la relación aplicación web.

- **Borrado:** Si se borra una aplicación web no tiene sentido conservarlo en la relación imagen ya que cuando borramos una aplicación también borraremos toda la información relacionada con ella por tanto el borrado es en cascada.
- **Modificación:** la modificación es en cascada ya que si se modifica la url de la aplicación web deberá actualizarse también en la relación imagen.

Relación USUARIO atributos Url:

La entidad USUARIO se relaciona con la entidad APLICACIÓN WEB mediante la interrelación TIENE. Es un tipo de interrelación 1:N, así que se transforma mediante una propagación de clave, siendo url clave ajena de la relación usuario respecto a la relación aplicación web.

- **Borrado:** Si se borra una aplicación web no tiene sentido conservarlo en la relación usuario ya que cuando borramos una aplicación también borraremos toda la información relacionada con ella por tanto el borrado es en cascada.
- **Modificación:** la modificación es en cascada ya que si se modifica la url de la aplicación web deberá actualizarse también en la relación usuario.

Relación FORO atributos Url:

La entidad FORO se relaciona con la entidad APLICACIÓN WEB mediante la interrelación TIENE. Es un tipo de interrelación 1:N así que se transforma mediante una propagación de clave siendo url clave ajena de la relación foro respecto a la relación aplicación web.

- **Borrado:** Si se borra una aplicación web no tiene sentido conservarlo en la relación foro ya que cuando borramos una aplicación también borraremos toda la información relacionada con ella , por tanto el borrado es en cascada.
- **Modificación:** la modificación es en cascada ya que si se modifica la url de la aplicación web deberá actualizarse también en la relación foro.

Relación TEMA atributos Cod_Foro:

La entidad FORO se relaciona con la entidad TEMA mediante la interrelación TIENE. Es un tipo de interrelación 1:N, así que se transforma mediante una propagación de clave, siendo Cod_Foro clave ajena de la relación tema respecto a la relación foro.

- **Borrado:** Si se borra un foro debemos conservar los temas en la base de datos ya que todavía tienen información útil que podría ser utilizada en otras partes de la aplicación a pesar de que ya no exista el foro. Por tanto el borrado será con puesta a nulo.
- **Modificación:** la modificación es en cascada ya que si se modifica el Cod_Foro del foro de la aplicación web deberá actualizarse también en la relación tema.

Relación SUBTEMA atributos Id_Tema:

La entidad SUBTEMA se relaciona con la entidad TEMA mediante la interrelación CONSTA. Es un tipo de interrelación 1:N, así que se transforma mediante una propagación de clave, siendo Id_Tema clave ajena de la relación subtema respecto a la relación tema.

- **Borrado:** Subtema es una entidad débil respecto a tema con una dependencia en identificación por tanto si borramos un tema borraremos toda la información relacionada con él incluidos los subtemas, por tanto el borrado es en cascada.
- **Modificación:** la modificación es en cascada ya que si se modifica el Id_Tema del tema deberá actualizarse también en la relación subtema.

Relación MENSAJE atributos Id_Tema, Id_Subtema:

La entidad MENSAJE se relaciona con la entidad SUBTEMA mediante la interrelación TIENE. Es un tipo de interrelación 1:N, así que se transforma mediante una propagación de clave, siendo Id_Subtema e Id_Tema clave ajena de la relación mensaje respecto a la entidad subtema. La interrelación es identificación ya que ya que el mensaje necesita de la clave principal de la entidad subtema (Id_Subtema, Id_Tema) para identificarse por eso pasa a ser parte de la clave principal de la relación mensaje.

- **Borrado:** Si se borra un subtema o el tema al que esté asociado el mensaje no tiene sentido conservarlo en la relación mensaje ya que cuando borramos un subtema o un tema también borraremos toda la información relacionada con él por tanto en ambos casos, el borrado es en cascada.
- **Modificación:** la modificación para ambos casos es en cascada ya que si se modifica un código del tema o del subtema deberá actualizarse también en la relación.

Relación MENSAJE atributos Cod_Usuario:

La entidad MENSAJE se relaciona con la entidad USUARIO mediante la interrelación ENVIA. Es un tipo de interrelación 1:N, así que se transforma mediante propagación de clave, siendo Cod_Usuario clave ajena de la relación mensaje respecto de la entidad usuario.

- **Borrado:** Si borramos un usuario o se da de baja en el sistema, no tendría sentido borrar los mensajes (ya que forman parte de hilos de conversación del foro), ni perder la información de quien los creo por tanto el borrado será restringido.
- **Modificación:** la modificación es en cascada ya que si se modifica un código de usuario deberá actualizarse también en la relación.

Relación MENSAJE atributos Id_Mensaje:

La entidad MENSAJE se relaciona con la entidad MENSAJE mediante la interrelación reflexiva RESPONDE. Es un tipo de interrelación 1:N, así que se transforma mediante propagación de clave, siendo Id_Mensaje clave ajena de la relación mensaje respecto de la entidad mensaje.

- **Borrado:** Si borramos un mensaje asociado a otro mensaje (en este caso sería una respuesta asociada a un mensaje) no tendría sentido conservar las “respuestas” asociadas a ese mensaje, por tanto el borrado es en cascada.
- **Modificación:** la modificación es en cascada ya que si se modifica un id_Mensaje deberá actualizarse también en la relación.

Relación EMPRESA atributos Url:

La entidad EMPRESA se relaciona con la entidad APLICACIÓN WEB mediante la interrelación TIENE. Es un tipo de interrelación 1:N, así que se transforma mediante una propagación de clave, siendo url clave ajena de la relación empresa respecto a la relación aplicación web.

- **Borrado:** Si se borra una aplicación web no tiene sentido conservar las empresas, ya que cuando borramos una aplicación también borraremos toda la información relacionada con ella por tanto el borrado es en cascada.
- **Modificación:** la modificación es en cascada ya que si se modifica la url de la aplicación web deberá actualizarse también en la relación empresa.

Relación CENTRO atributos Cif:

La entidad CENTRO se relaciona con la entidad EMPRESA mediante la interrelación TIENE. Es un tipo de interrelación 1:N, así que se transforma mediante una propagación de clave, siendo CIF clave ajena de la relación centro respecto a la relación empresa.

- **Borrado:** Si se borra una empresa no tiene sentido conservarlo en la relación centro ya que cuando borramos una empresa también borraremos toda la información relacionada con ella por tanto el borrado es en cascada.
- **Modificación:** la modificación es en cascada ya que si se modifica la CIF de la empresa deberá actualizarse también en la relación centro.

Relación PRODUCTO atributos Cif:

La entidad PRODUCTO se relaciona con la entidad EMPRESA mediante la interrelación DISPONE. Es un tipo de interrelación 1:N, así que se transforma mediante una propagación de clave, siendo CIF clave ajena de la relación producto

respecto a la relación empresa.

- **Borrado:** Si se borra una empresa no tiene sentido conservarlo en la relación producto ya que cuando borramos una empresa también borraremos toda la información relacionada con ella. Por tanto el borrado es en cascada.
- **Modificación:** la modificación es en cascada ya que si se modifica la CIF de la empresa deberá actualizarse también en la relación producto.

Relación DISTRIBUYE atributos Cod Producto y Id Centro:

La entidad PRODUCTO se relaciona con la entidad CENTRO mediante la interrelación DISTRIBUYE. Es un tipo de interrelación N:M, así que se transforma en una relación, siendo Cod_Producto clave ajena respecto a la relación Producto e Id_Centro clave ajena respecto a la relación Centro.

- **Borrado:** Si se borra un centro no tiene sentido conservarlo en la relación tiene, ya que cuando borramos un centro también borraremos toda la información relacionada con él. Por tanto el borrado es en cascada. Para la clave ajena Cod_Producto el borrado será a también en cascada ya que si borramos un producto relacionado con un centro no tendría sentido conservar esta información de la relación.
- **Modificación:** la modificación para ambos casos es en cascada ya que si se modifica un código del producto o del centro deberá actualizarse también en cada relación.

Relación PEDIDO atributos Cod Banco y Cod Usuario:

La entidad CLIENTE se relaciona con la entidad PEDIDO mediante la interrelación REALIZA. Es un tipo de interrelación 1:N así que se transforma mediante propagación de clave siendo Cod_Usuario clave ajena respecto de la entidad cliente. La entidad BANCO se relaciona con la entidad PEDIDO mediante la interrelación PAGA. Es un tipo de interrelación 1:1 así que se transforma mediante propagación de clave siendo Cod_Banco clave ajena respecto de la entidad BANCO. Este atributo puede ser nulo dado que un determinado pedido se puede pagar contra reembolso y no necesita estar asociado a ningún banco.

- **Borrado:** Si se borra un cliente no tiene sentido perder los pedidos que ha realizado por tanto el borrado es restringido. Para la clave ajena Cod_Banco el borrado será restringido ya que si borramos un banco de la base de datos no tendría sentido perder el resto de la información del pedido incluido el banco elegido por el cliente para realizar el pago.
- **Modificación:** la modificación para ambos casos es en cascada ya que si se modifica un código de banco o del cliente deberá actualizarse también en cada relación.

Relación FORMADO POR atributos Cod Pedido y Cod Producto:

La entidad PEDIDO se relaciona con la entidad PRODUCTO mediante la interrelación FORMADO_POR. Es un tipo de interrelación N:M así que se transforma en una nueva relación donde la clave primaria la formaran los atributos Cod_Pedido y Cod_Producto y además contendrá los atributos Precio y Num_Unidades.

- **Borrado:** Si se borra un pedido no tiene sentido conservarlo en la relación formado_por ya que, cuando borramos un pedido también borraremos toda la información relacionada con él, incluida los productos que contiene, por tanto el borrado es en cascada. Para el caso de Cod_Producto el borrado será a nulo porque si borramos un producto no tiene sentido perder la información de los pedidos en los que se haya adquirido ese producto.
- **Modificación:** la modificación será en cascada ya que si se modifica un código de pedido o el código del producto deberá actualizarse también en la relación.

Relación FACTURA atributo Cod Pedido:

La entidad PEDIDO se relaciona con la entidad FACTURA mediante la interrelación GENERA. Es un tipo de interrelación 1:1, así que se transforma mediante propagación de clave, siendo Cod_Pedido clave ajena de la relación factura respecto de la relación pedido. La entidad factura es una entidad débil y su relación con la entidad pedido es una interrelación en identificación puesto que factura necesita del código del pedido para identificarse

- **Borrado:** Si se borra una pedido no tiene sentido conservar su factura. Por tanto al borrar un pedido eliminaremos toda la información asociada a él incluida la factura, por tanto el borrado es en cascada.
- **Modificación:** la modificación es en cascada ya que si se modifica el código del pedido deberá actualizarse también en cada relación.

Relación SUMINISTRA atributos Cod Proveedor, Cod Producto:

La entidad PRODUCTO se relaciona con la entidad PROVEEDOR mediante la interrelación SUMINISTRA que es de tipo N:M por tanto se transforma en una nueva relación siendo Cod_Producto clave ajena respecto a la relación producto y Cod_Proveedor clave ajena respecto de la relación proveedor.

- **Borrado:** Para el atributo Cod_Proveedor el borrado será en cascada ya que si eliminamos un proveedor eliminaremos toda la información asociada a él incluidos los productos que suministra. Como ese proveedor ya no existe no nos podrá suministrar sus productos con lo que ya nos es necesaria esa información. Para el caso del atributo Cod_Producto el borrado será también en cascada porque una vez que deja de existir un producto da igual quién lo hubiera estado suministrando, por tanto no tiene sentido conservar esa información.
- **Modificación:** la modificación es en cascada para ambos casos ya que si se modifica un código del producto o del proveedor deberá actualizarse también en la relación.

Relación TIENE atributos Cod Pedido, Cod Proveedor, Cod Producto:

Las entidades PEDIDO_PROV, PROVEEDOR y PRODUCTO se relacionan entre sí mediante la interrelación ternaria TIENE. Esta interrelación es de tipo N:M:P, por tanto se transforma en una nueva relación cuya clave principal estará formada por las claves principales de las entidades que une (Cod_Pedido, Cod_Proveedor, Cod_Producto), además a esta nueva relación hay que incluir los atributos de la interrelación importe y num_unidades.

- **Borrado:** Para el atributo Cod_Pedido el borrado será en cascada ya que si eliminamos un pedido a proveedores eliminaremos toda la información asociada a él incluidos los productos pedidos y los proveedores que suministraran esos productos. Para el atributo Cod_Proveedor el borrado será restringido ya que si eliminamos un proveedor no tiene sentido perder los pedidos a ese proveedor. Por último para el atributo Cod_Producto el borrado será también restringido ya que a pesar de que se elimine un determinado producto que forma parte de un pedido deberemos seguir conservando todos los productos que formen el pedido a pesar de que estos ya no existan.
- **Modificación:** la modificación es en cascada para todos los casos ya que si se modifica un código del producto, del proveedor o del pedido deberá actualizarse también en la relación.

Relación FACTURA PROV atributos Cod Pedido, Cod Proveedor:

Las entidades PEDIDO, PROVEEDOR se relacionan entre sí mediante la interrelación FACTURA_PROV. Esta interrelación es de tipo 1:1, por tanto se transforma en una nueva relación ya que esta interrelación tiene dos atributos, Id_Sector e Importe que pasarán a formar parte de la relación.

- **Borrado:** Para el atributo Cod_Proveedor el borrado será restringido ya que si eliminamos un proveedor no tiene sentido perder las facturas asociadas a ese proveedor. Para el atributo Cod_Pedido el borrado será en cascada ya que si borramos un pedido de un proveedor borraremos toda la información asociada a él, incluida la factura asociada a él.
- **Modificación:** la modificación es en cascada para ambos casos ya que si se modifica un código del proveedor o del pedido deberá actualizarse también en la relación.

7.1.3. Diseño Físico de Base de Datos

A partir del modelo relacional anterior obtenemos el siguiente diagrama de base de datos:

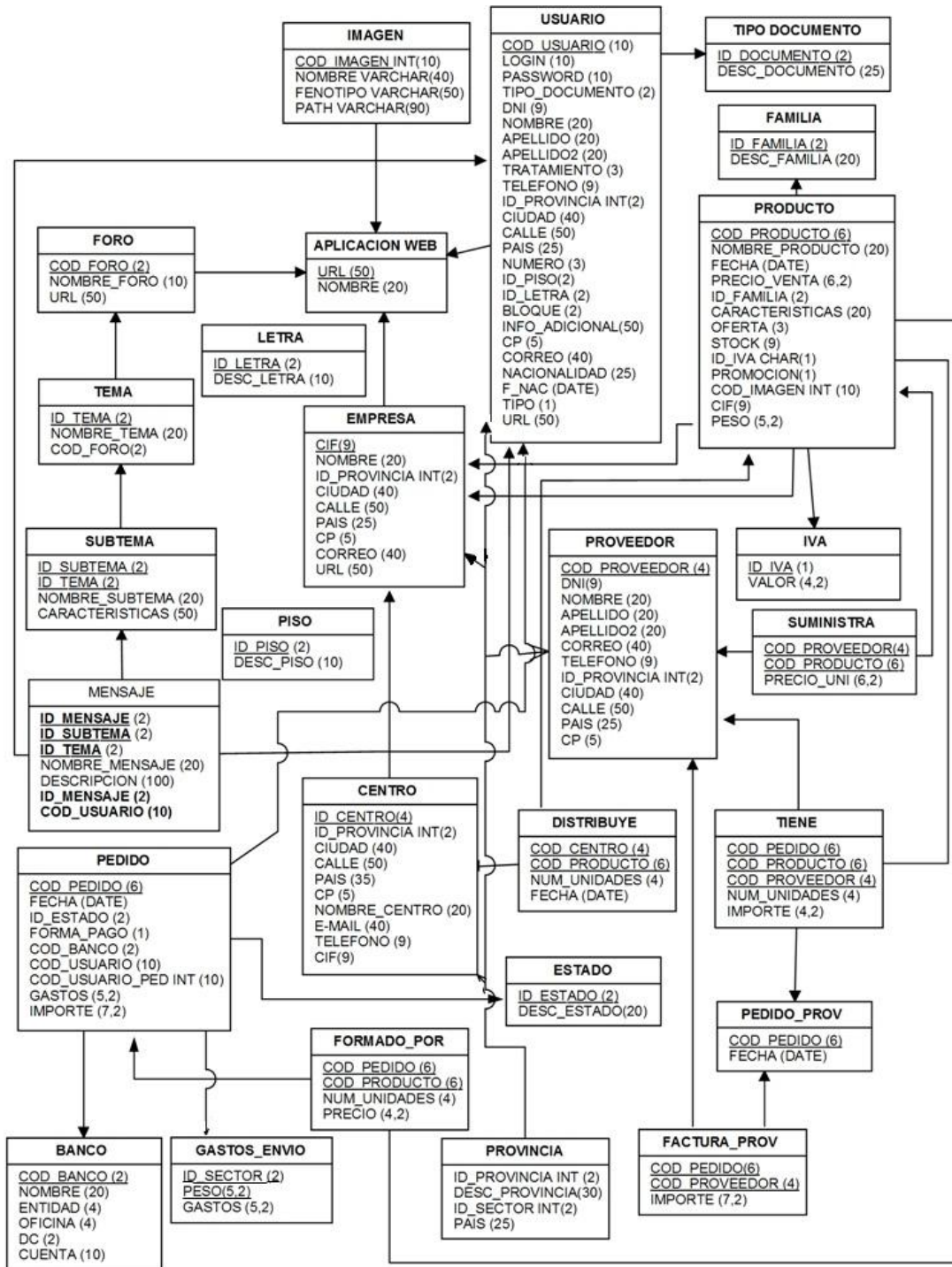


Figura 54. Diagrama de tablas de base de datos

7.2. Optimización del Modelo Físico de Datos (Esquema Normalizado)

La teoría de la normalización tiene por objetivo la eliminación de dependencias entre atributos que originen anomalías en la actualización de los datos, y proporcionar una estructura más regular para la representación de las tablas, constituyendo el soporte para el diseño de bases de datos relacionales. Como resultado de la aplicación de esta técnica se obtiene un modelo lógico de datos normalizado. A continuación se detallan las distintas tablas de la base de datos aplicando sobre ellas las teorías de normalización hasta alcanzar la forma normal de Boyce/Codd.

1. APLICACIÓN WEB

APLICACION WEB
URL (50)
NOMBRE (20)

La entidad aplicación web cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaludados y todas sus tuplas serán diferentes. Dado que la entidad está formada por sólo dos atributos la única dependencia funcional va a ser del atributo nombre con su clave principal y esta dependencia va a ser completa (cumple la segunda forma normal), además no da lugar a dependencias transitivas (cumple la tercera forma normal) y su atributo sólo facilita información sobre la clave principal (cumple forma normal de Boyce/Codd).

2. BANCO

BANCO
COD_BANCO (4)
NOMBRE (20)
ENTIDAD (4)
OFICINA (4)
DC (2)
CUENTA (10)

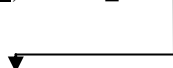
La entidad banco cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaludados y todas sus tuplas serán diferentes.

Todos sus atributos aportan información sobre la clave candidata (cod_banco) y tienen una dependencia funcional completa por tanto cumple la segunda forma normal. Para cada aparición de un determinado nombre de un banco el número de entidad siempre es el mismo, es decir existe redundancia de datos y por tanto peligro de integridad. Debido a esto la entidad no cumple la forma normal de Boyce/Codd ya que existe un atributo “entidad” que depende de otro atributo distinto de la clave.

COD_BANCO	NOMBRE	ENTIDAD	OFICINA	DC	CUENTA
1	Bankinter	1111	1111	11	1111111111
2	Bankinter	1111	1112	11	2222222222
3	Santander	1234	1234	11	3333333333
4	Bankinter	1111	1115	11	4444444444
5	Santander	1234	1255	11	5555555555

La solución al problema consiste en crear una nueva entidad “Nombre” y eliminarlos de la relación quedando de la forma:

BANCO(cod_banco, nombre_banco, oficina, dc, cuenta)



NOMBRE(nombre_banco, entidad)

Debido a que no queremos complicar demasiado la base de datos y los registros (tuplas) de esta tabla van a ser pocos dejaremos la entidad completa.

3. CENTRO

CENTRO
ID_CENTRO(4)
ID_PROVINCIA INT(2)
CIUDAD (40)
CALLE (50)
PAIS (35)
CP (5)
NOMBRE_CENTRO (20)
E-MAIL (40)
TELEFONO (9)
CIF(9)

La entidad centro no cumple la primera forma normal ya que el atributo calle es un atributo compuesto por calle número y bloque. Para solucionarlo crearemos dos nuevos atributos (número y bloque) en la relación.

CENTRO (id_centro, provincia, ciudad, calle, número, bloque, país, cp, nombre_centro, e-mail, cif).

En cuanto al atributo teléfono cada centro sólo podrá tener un número de teléfono asociado por tanto no es un atributo multivaluado.

Dado que todos los atributos aportan información sobre la clave candidata (id_centro) y tienen una dependencia funcional completa, la entidad cumple la segunda forma normal.

Id_centro	Ciudad	Provincia	País
CN0001	Aranda de Duero	Burgos	España
CN0002	Burgos	Burgos	España
CN0004	Lerma	Burgos	España
CN0005	Miranda de Ebro	Burgos	España
CN0006	Guadalajara	Guadalajara	España
CN0007	Guadalajara	Jalisco	Mexico

En el ejemplo podemos ver qué país tiene una dependencia funcional transitiva con respecto a id_centro a través de provincia.

Id_centro → Provincia → País → Id_centro

Por tanto la relación no cumple la tercera forma normal.

Para solucionarlo vamos a crear con los atributos provincia y país eliminando país de la entidad

CENTRO (id_centro, provincia, ciudad, calle, número, bloque, cp, nombre_centro, e-mail, cif)

PROVINCIA (id_provincia, desc_provincia, id_sector, país)

Aprovechando la entidad provincia que sirve para mostrar las provincias incluiremos las distintas provincias incluiremos el atributo país

PROVINCIA (id_provincia, desc_provincia, país)

4. DISTRIBUYE

DISTRIBUYE
COD_CENTRO (4)
COD_PRODUCTO (6)
NUM_UNIDADES (4)
FECHA (DATE)

La entidad distribuye cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaluados y todas sus tuplas serán diferentes.

Los atributos num_unidades y fecha tiene una dependencia funcional completa con respecto a la clave cod_centro y

cod_producto, como todos los atributos de la relación aportan información sobre la clave completa se cumple la regla de la segunda forma normal.

Tampoco hay dependencias transitivas por lo que cumple la tercera forma normal y los atributos no principales sólo proporcionan información sobre las claves, por tanto también cumple la forma normal de Boyce/Codd.

5. EMPRESA

EMPRESA
<u>CIF(9)</u>
NOMBRE (20)
PROVINCIA (20)
CIUDAD (40)
CALLE (50)
PAIS (25)
CP (5)
CORREO (40)
URL (50)

La entidad empresa no cumple la primera forma normal ya que el atributo calle es un atributo compuesto por la calle, el número y el bloque por tanto añadiremos estos dos nuevos atributos a la relación.

EMPRESA (cif, nombre id_provincia, ciudad, calle, numero, bloque, país, cp, correo, url)

En cuanto a la segunda forma normal ya que todos los atributos aportan información sobre la clave principal y las dependencias funcionales de estos atributos con respecto a la clave serán completas la entidad cumple la segunda forma normal.

En la relación nos encontramos con el mismo problema que en la entidad centro ya que existe una dependencia transitiva de país respecto al cif a través de id_provincia. Para solucionarlo utilizaremos la misma fórmula que en centros, es decir, eliminaremos el atributo país de la relación dado que ya tenemos reflejada esa información a través de la relación provincia.

Una vez eliminado el atributo provincia podremos decir que los atributos de la relación sólo proporcionan información a la clave candidata y no al resto de los atributos de la relación por lo que podemos decir que cumple la forma normal de Boyce/Codd.

6. ESTADO

ESTADO
<u>ID_ESTADO (2)</u>
DESC_ESTADO(20)

La entidad estado cumple la primera forma normal ya que todos sus atributos son atómicos.

Dado que la entidad está formada por sólo dos atributos la única dependencia funcional va a ser del atributo desc_estado con la clave candidata id_estado y esta dependencia va a ser completa (cumple la segunda forma normal), además no da lugar a dependencias transitivas (cumple la tercera forma normal) y dado que la relación está formada únicamente por dos atributos también cumplirá la forma normal de Boyce/Codd.

7. FACTURA

FACTURA
<u>NUM_FACTURA (4)</u>
<u>COD_PEDIDO (6)</u>
ID_IVA CHAR(1)
GASTOS_ENVIO (3,2)

La entidad factura cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaludados y todas sus tuplas serán diferentes.

Los atributos id_iva y gastos de envío aportan información al pedido pero no al num_factura, por tanto no cumple la

segunda forma normal.

La relación factura tiene dependencia en identificación respecto a la entidad pedido por eso la clave principal está formada por los atributos num_factura y cod_pedido pero dado que siempre que exista una factura existirá un pedido y viceversa el cod_pedido y num_factura que son incrementales coincidirán generando redundancia (*ver ejemplo) la solución que utilizaremos será añadir el atributo gastos de envío a la entidad pedido y eliminar la entidad factura ya que esta estaba pensada para llevar un seguimiento de la facturación de pedidos y podemos prescindir de ella ya que con la nueva solución seguiremos teniendo la información necesaria para obtener los datos de la factura.

Si pudiera existir un pedido que no tuviera asociada una factura tendría sentido dejar la entidad factura pero la aplicación no permite almacenar los datos de un pedido para terminarlo y crear la factura en otro momento.

Num_factura	Cod_pedido	Gastos_envio	Importe
1	1	20.00	402.80
2	2	20.00	228.00
3	3	6.60	426.98
4	4	18.6	262.32

De este modo la entidad pedido quedará de la forma :

PEDIDO (Cod_pedido, fecha, id_estado, forma_pago, cod_banco, cod_usuario, cod_usuario_ped, gastos_envio, importe).

8. FACTURA_PROV

FACTURA_PROV
COD_PEDIDO(6)
COD_PROVEEDOR (4)
IMPORTE (7,2)

La entidad factura_prov cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaluados y todas sus tuplas serán diferentes.

El atributo importe aporta información a la clave principal formada por cod_pedido y cod_proveedor y tendrá una dependencia funcional completa respecto a la toda la clave por tanto cumple la segunda forma normal.

Tampoco existe dependencias funcionales transitivas y los atributos no principales sólo aportan información sobre las claves candidatas no sobre los atributos por tanto también cumple la tercera forma normal y la de Boyce/Codd.

9. FAMILIA

FAMILIA
ID_FAMILIA (2)
DESC_FAMILIA (20)

La entidad familia cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaluados y todas sus tuplas serán diferentes.

Dado que la entidad está formada por sólo dos atributos la única dependencia funcional va a ser del atributo desc_familia con la clave candidata id_familia y esta dependencia va a ser completa (cumple la segunda forma normal), además no da lugar a dependencias transitivas (cumple la tercera forma normal) y dado que la relación está formada únicamente por dos atributos también cumplirá la forma normal de Boyce/Codd.

10. FORMADO_POR

FORMADO_POR
COD_PEDIDO (6)
COD_PRODUCTO (6)
NUM_UNIDADES (4)
PRECIO (4,2)

La entidad formado_por cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaluados y todas sus tuplas serán diferentes.

La clave candidata será la formada por `cod_pedido` y `cod_producto` y el resto de atributos (`num_unidades` y `precio`) aportan información a la clave y tiene una dependencia funcional completa respecto a la clave, por tanto cumple la segunda forma normal.

Esta relación no tiene dependencias funcionales transitivas (cumple la tercera forma normal) y los atributos sólo aportan información a la clave principal y no sobre el resto de atributos (cumple forma normal de Boyce/Codd).

11. FORO

FORO
<u>COD_FORO</u> (2)
NOMBRE_FORO (10)
URL (50)

La entidad foro cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaludados y todas sus tuplas serán diferentes.

Los atributos `nombre_foro` y `url` aportan información a la clave candidata `cod_foro` y tienen una dependencia funcional completa respecto a la clave, por tanto cumple la segunda forma normal.

Esta relación no tiene dependencias funcionales transitivas (cumple la tercera forma normal) y los atributos sólo aportan información a la clave principal y no sobre el resto de atributos (cumple forma normal de Boyce/Codd).

12. GASTOS_ENVIO

GASTOS_ENVIO
<u>ID_SECTOR</u> (2)
<u>PESO</u> (5,2)
GASTOS (5,2)

La entidad `gastos_envio` cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaludados y todas sus tuplas serán diferentes.

La clave candidata está formada por `id_sector` y `peso`. El atributo `gastos` tiene una dependencia funcional completa respecto a la clave, por tanto cumple la segunda forma normal.

Esta relación no tiene dependencias funcionales transitivas (cumple la tercera forma normal) y los atributos sólo aportan información a la clave principal y no sobre el resto de atributos (cumple forma normal de Boyce/Codd).

Para simplificar la base de datos y dado que, como se demuestra más adelante la entidad `sector` desaparece, el cálculo de los gastos de envío se llevará a la parte de negocio de la aplicación y el atributo `GASTOS` pasará a la entidad `PEDIDO`.

13. IMAGEN

IMAGEN
<u>COD_IMAGEN</u> INT(10)
NOMBRE VARCHAR(40)
FENOTIPO VARCHAR(50)
PATH VARCHAR(90)
URL(50)

La entidad `imagen` cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaludados y todas sus tuplas serán diferentes.

Los atributos `nombre`, `fenotipo`, `path` y `url` aportan información a la clave candidata `cod_imagen` y tienen una dependencia funcional completa respecto a la clave, por tanto cumple la segunda forma normal.

Esta relación no tiene dependencias funcionales transitivas (cumple la tercera forma normal) y los atributos sólo aportan información a la clave principal y no sobre el resto de atributos (cumple forma normal de Boyce/Codd).

14. IVA

IVA
ID_IVA (1)
VALOR (4,2)

La entidad IVA cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaludados y todas sus tuplas serán diferentes.

Dado que la entidad está formada por sólo dos atributos la única dependencia funcional va a ser del atributo valor con la clave candidata id_iva y esta dependencia va a ser completa (cumple la segunda forma normal). Además no da lugar a dependencias transitivas (cumple la tercera forma normal) y dado que la relación está formada únicamente por dos atributos también cumplirá la forma normal de Boyce/Codd.

15. LETRA

LETRA
ID_LETRA (2)
DESC_LETRA (10)

La entidad letra cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaludados y todas sus tuplas serán diferentes.

Dado que la entidad está formada por sólo dos atributos la única dependencia funcional va a ser del atributo id_letra con la clave candidata desc_letra y esta dependencia va a ser completa (cumple la segunda forma normal), además no da lugar a dependencias transitivas (cumple la tercera forma normal) y dado que la relación está formada únicamente por dos atributos también cumplirá la forma normal de Boyce/Codd.

16. MENSAJE

MENSAJE
ID_MENSAJE (2)
ID_SUBTEMA (2)
ID_TEMA (2)
NOMBRE_MENSAJE (20)
DESCRIPCION (100)
ID_MENSAJE (2)
COD_USUARIO (10)

La entidad mensaje cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaludados y todas sus tuplas serán diferentes.

La clave candidata será la formada por id_mensaje, id_subtema e id_tema y el resto de atributos proporcionan información sobre la clave completa además tienen una dependencia funcional completa respecto a la clave candidata completa por tanto cumple la segunda forma normal.

No existen dependencias funcionales transitivas por lo que cumplirá la tercera forma normal y los atributos no principales de la relación sólo aportan información sobre la clave principal y no sobre el resto de atributos por lo que cumple la forma normal de Boyce/Codd.

17. PEDIDO

PEDIDO
COD_PEDIDO (6)
FECHA (DATE)
ID_ESTADO (2)
FORMA_PAGO (1)
COD_BANCO (2)
COD_USUARIO (10)
COD_USUARIO_PED INT (10)
GASTOS (5,2)
IMPORTE (7,2)

La entidad pedido cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaludados y todas sus tuplas serán diferentes.

La clave candidata será cod_pedido y el resto de atributos proporcionan información sobre la clave, además tienen una

dependencia funcional completa respecto a la clave por tanto cumple la segunda forma normal.

No existen dependencias funcionales transitivas por lo que cumplirá la tercera forma normal y los atributos no principales de la relación sólo aportan información sobre la clave principal y no sobre el resto de atributos por lo que cumple la forma normal de Boyce/Codd.

Dado que desaparece la entidad GASTOS ENVIO, añadimos el atributo GASTOS_ENVIO (5,2) a la entidad PEDIDO.

18. PEDIDO_PROV

PEDIDO_PROV
COD_PEDIDO (6)
FECHA (DATE)

La entidad pedido_prov cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaluados y todas sus tuplas serán diferentes.

Dado que la entidad está formada por sólo dos atributos la única dependencia funcional va a ser del atributo fecha con la clave candidata cod_pedido y esta dependencia va a ser completa (cumple la segunda forma normal), además no da lugar a dependencias transitivas (cumple la tercera forma normal) y dado que la relación está formada únicamente por dos atributos también cumplirá la forma normal de Boyce/Codd.

19. PISO

PISO
ID_PISO (2)
DESC_PISO (10)

La entidad piso cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaluados y todas sus tuplas serán diferentes.

Dado que la entidad está formada por sólo dos atributos la única dependencia funcional va a ser del atributo desc_piso con la clave candidata id_piso y esta dependencia va a ser completa (cumple la segunda forma normal), además no da lugar a dependencias transitivas (cumple la tercera forma normal) y dado que la relación está formada únicamente por dos atributos también cumplirá la forma normal de Boyce/Codd.

20. PRODUCTO

PRODUCTO
COD_PRODUCTO (6)
NOMBRE_PRODUCTO (20)
FECHA (DATE)
PRECIO_COMPRA (6,2)
PRECIO_VENTA (6,2)
ID_FAMILIA (2)
CARACTERISTICAS (20)
OFERTA (3)
STOCK (9)
ID_IVA CHAR(1)
PROMOCION(1)
COD_IMAGEN INT (10)
CIF(9)
PESO (5,2)

Los atributos de la entidad producto son atómicos y todas sus tuplas serán diferentes pero tiene un atributo, precio compra, que es un atributo multivaluado ya que varios proveedores pueden suministrar un mismo producto a precios diferentes por tanto no cumple la primera forma normal. Para solucionarlo debemos extraer el atributo precio compra de la relación y ponerlo en una nueva relación, donde la clave principal sería la formada por el cod_producto y el precio_compra, pero como tenemos la relación suministra con el cod_producto y el precio de compra a un proveedor nos servirá para solucionar el problema de este atributo.

Por tanto la relación quedaría de la forma :

PRODUCTO (cod_producto, nombre_producto, fecha, precio_venta, id_familia, características, oferta, stock, id_iva, promoción, cod_imagen, cif, peso).

SUMINISTRA (cod_proveedor, cod_producto, precio_uni)

La clave candidata va a ser el cod_producto, y el resto de atributos si aporta información sobre la clave candidata y van a tener una dependencia funcional completa respecto a la clave por tanto cumple la segunda forma normal.

No existen dependencias transitivas por tanto cumple la tercera forma normal, y sus atributos sólo facilitan información sobre la clave principal y no sobre el resto de atributos (cumple forma normal de Boyce/Codd).

21 PROVEEDOR

PROVEEDOR
<u>COD_PROVEEDOR</u> (4)
DNI(9)
NOMBRE (20)
APELLIDO (20)
APELLIDO2 (20)
CORREO (40)
TELEFONO (9)
ID_PROVINCIA INT(2)
CIUDAD (40)
CALLE (50)
PAIS (25)
CP (5)

La entidad proveedor cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaluados y todas sus tuplas serán diferentes.

La clave candidata será cod_proveedor y el resto de atributos proporcionan información sobre la clave por tanto cumple la segunda forma normal.

Para la tercera forma normal con respecto a los atributos cod_proveedor, provincia, país ocurre el mismo problema que en centros, ya que país tiene una dependencia funcional transitiva con respecto al cod_proveedor a través de provincia. Esto se solucionaría con la nueva relación provincia y eliminando el atributo país de la relación proveedor.

El resto de atributos que no forman parte de la clave candidata no aporta información entre ellos por tanto cumple la forma normal de Boyce/Codd.

22. SUBTEMA

SUBTEMA
<u>ID_SUBTEMA</u> (2)
<u>ID_TEMA</u> (2)
NOMBRE_SUBTEMA (20)
CARACTERISTICAS (50)

La entidad subtema cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaluados y todas sus tuplas serán diferentes.

La clave candidata será la formada por id_subtema e id_tema y el resto de atributos proporcionan información sobre la clave además tendrán una dependencia funcional completa respecto a toda la clave por tanto cumple la segunda forma normal.

No existen dependencias funcionales transitivas por lo que cumplirá la tercera forma normal y los atributos no principales de la relación sólo aportan información sobre la clave principal y no sobre el resto de atributos por lo que cumple la forma normal de Boyce/Codd.

23. SUMINISTRA

SUMINISTRA
<u>COD_PROVEEDOR</u> (4)
<u>COD_PRODUCTO</u> (6)
PRECIO_UNI (6,2)

La entidad suministra cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaluados y todas sus tuplas serán diferentes.

La clave candidata será la formada por cod_proveedor y cod_producto y el atributo precio_uni proporciona información sobre la clave y tienen una dependencia funcional completa respecto a la clave por tanto cumple la segunda forma normal.

No existen dependencias funcionales transitivas por lo que cumplirá la tercera forma normal y los atributos no principales de la relación sólo aportan información sobre la clave principal y no sobre el resto de atributos (sólo hay uno) por lo que cumple la forma normal de Boyce/Codd.

24. TEMA

TEMA
ID_TEMA (2)
NOMBRE_TEMA (20)
COD_FORO(2)

La entidad tema cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaludados y todas sus tuplas serán diferentes.

La clave candidata será id_tema y el resto de atributos proporciona información sobre la clave además tendrán una dependencia funcional completa respecto a la clave por tanto cumple la segunda forma normal.

No existen dependencias funcionales transitivas por lo que cumplirá la tercera forma normal y los atributos no principales de la relación sólo aportan información sobre la clave principal y no sobre el resto de atributos por lo que cumple la forma normal de Boyce/Codd.

25. TIENE

TIENE
COD_PEDIDO (6)
COD_PRODUCTO (6)
COD_PROVEEDOR (4)
NUM_UNIDADES (4)
IMPORTE (4,2)

La entidad tiene cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaludados y todas sus tuplas serán diferentes.

La clave candidata será la formada por cod_pedido, cod_producto y cod_proveedor y el resto de atributos (num_unidades y importe) aportan información a la clave además tendrán una dependencia funcional completa respecto a toda la clave, por tanto cumple la segunda forma normal.

Esta relación no tiene relaciones funcionales transitivas (cumple la tercera forma normal) y los atributos sólo aportan información a la clave principal y no sobre el resto de atributos (cumple forma normal de Boyce/Codd).

26. TIPO DOCUMENTO

TIPO DOCUMENTO
ID_DOCUMENTO (2)
DESC_DOCUMENTO (25)

La entidad tipo_documento cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaludados y todas sus tuplas serán diferentes.

Dado que la entidad está formada por sólo dos atributos la única dependencia funcional va a ser del atributo desc_documento con la clave candidata id_documento y esta dependencia va a ser completa (cumple la segunda forma normal), además no da lugar a dependencias transitivas (cumple la tercera forma normal) y dado que la relación está formada únicamente por dos atributos también cumplirá la forma normal de Boyce/Codd.

27. USUARIO

USUARIO
COD_USUARIO (10)
LOGIN (10)
PASSWORD (10)
TIPO_DOCUMENTO (2)
DNI (9)
NOMBRE (20)
APELLIDO (20)
APELLIDO2 (20)
TRATAMIENTO (3)
TELEFONO (9)
ID_PROVINCIA INT(2)
CIUDAD (40)
CALLE (50)
PAIS (25)
INFO_ADICIONAL(50)
CP (5)
CORREO (40)
NACIONALIDAD (25)
F_NAC (DATE)
TIPO (1)
URL (50)

La entidad usuario no cumple la primera forma normal ya que el atributo calle es un atributo compuesto por calle numero, piso, la letra y bloque. Para solucionarlo crearemos cuatro nuevos atributos (número, piso, letra y bloque) en la relación. El resto de restricciones para la primero forma normal las cumple.

En cuando a la segunda forma normal todos los atributos que no son clave candidata aportan información sobre la clave además tienen una dependencia funcional completa respecto a la clave por lo que cumple la segunda forma normal.

Cod_usuario	Ciudad	Provincia	País	Nacionalidad
1	Aranda de Duero	Burgos	España	Española
2	Burgos	Burgos	España	Española
3	Lerma	Burgos	España	Española
4	Miranda de Ebro	Burgos	España	Española
5	Guadalajara	Guadalajara	España	Francesa
6	Guadalajara	Jalisco	Mexico	Mejicana

En el ejemplo podemos ver qué país tiene una dependencia funcional transitiva con respecto a cod_usuario a través de provincia.

Cod_usuario → Provincia → País → Provincia → cod_usuario

Esta dependencia transitiva se soluciona de la misma forma que en la relación centros, es decir, eliminando país de la relación.

Con respecto al atributo nacionalidad se refiere a la nacionalidad del cliente por lo que a partir del país no podremos averiguar el país.

USUARIO (cod_usuario, login, password, tipo_documento, dni, nombre, apellido, apellido2, tratamiento, teléfono, id_provincia, ciudad, numero, letra, piso, bloque, info_adicional, cp, correo, nacionalidad, f_nac, tipo, url).

PROVINCIA (id_provincia, desc_provincia, id_sector, país)

El resto de atributos que no forman parte de la clave candidata aportan información sobre la clave principal y no entre el resto de atributos por tanto cumple la forma normal de Boycce/Codd.

28. USUARIO_PEDIDO

USUARIO_PEDIDO
COD_USUARIO_PED (10)
LOGIN (10)
PASSWORD (10)
DNI (9)
NOMBRE (20)
APELLIDO (20)
APELLIDO2 (20)
TRATAMIENTO (3)
TELEFONO (9)
ID_PROVINCIA INT(2)
CIUDAD (40)
CALLE (50)
CP (5)
CORREO (40)
TIPO_DOCUMENTO (2)
INFO_ADICIONAL(50)
FORMA_PAGO(1)
COD_USUARIO (10)

La entidad usuario_pedido no cumple la primera forma normal ya que el atributo calle es un atributo compuesto por calle numero, piso, la letra y bloque. Para solucionarlo crearemos cuatro nuevos atributos (número, piso, letra y bloque) en la relación. El resto de restricciones para la primera forma normal las cumple.

En cuando a la segunda forma normal todos los atributos que no son clave candidata aportan información sobre la clave y además tienen una dependencia funcional completa respecto a la clave, por lo que cumple la segunda forma normal.

No existen dependencias transitivas por lo que cumple la tercera forma normal y los atributos no principales aportan información sobre la clave principal y no sobre otros atributos por lo que cumple la forma normal de Boyce/Codd.

La entidad usuario_pedido quedará de la siguiente forma :

USUARIO (cod_usuario, logn, password, dni, nombre, apellido, apellido2, tratamiento, teléfono, id_provincia, ciudad, numero, letra, piso, bloque, info_adicional, cp, correo, tipo_documento, forma_pago, cod_usuario).

29. PROVINCIA

PROVINCIA
ID_PROVINCIA INT (2)
DESC_PROVINCIA(30)
ID_SECTOR INT(2)
PAIS (25)

La relación provincia cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaludados y todas sus tuplas serán diferentes.

La clave candidata será id_provincia y el resto de atributos proporcionan información a la clave, dado que sólo hay una clave la dependencia es completa por lo que podemos decir que se cumple la segunda forma normal.

Id_provincia	Desc_provincia	Id_sector	País
1	Badajoz	1	gEspaña
2	Cáceres	1	España
3	Ciudad Real	1	España
4	Las Palmas	2	España
5	Hendaya	2	Francia

En el ejemplo podemos ver que para cada aparición de un determinado sector el país no es siempre el mismo por tanto no existen dependencias transitivas por lo que cumple la tercera forma normal y los atributos no principales aportan información sobre la clave principal y no sobre otros atributos por lo que cumple la forma normal de Boyce/Codd

30. PAÍS

PAIS
ID_PAIS INT (2) PAIS(25)

La entidad país cumple la primera forma normal ya que todos sus atributos son atómicos, no tiene atributos multivaluados y todas sus tuplas serán diferentes.

Dado que la entidad está formada por sólo dos atributos la única dependencia funcional va a ser del atributo país con su clave principal id_país y esta dependencia va a ser completa (cumple la segunda forma normal), además no da lugar a dependencias transitivas (cumple la tercera forma normal) y dado que la relación está formada únicamente por dos atributos también cumplirá la forma normal de Boyce/Codd.

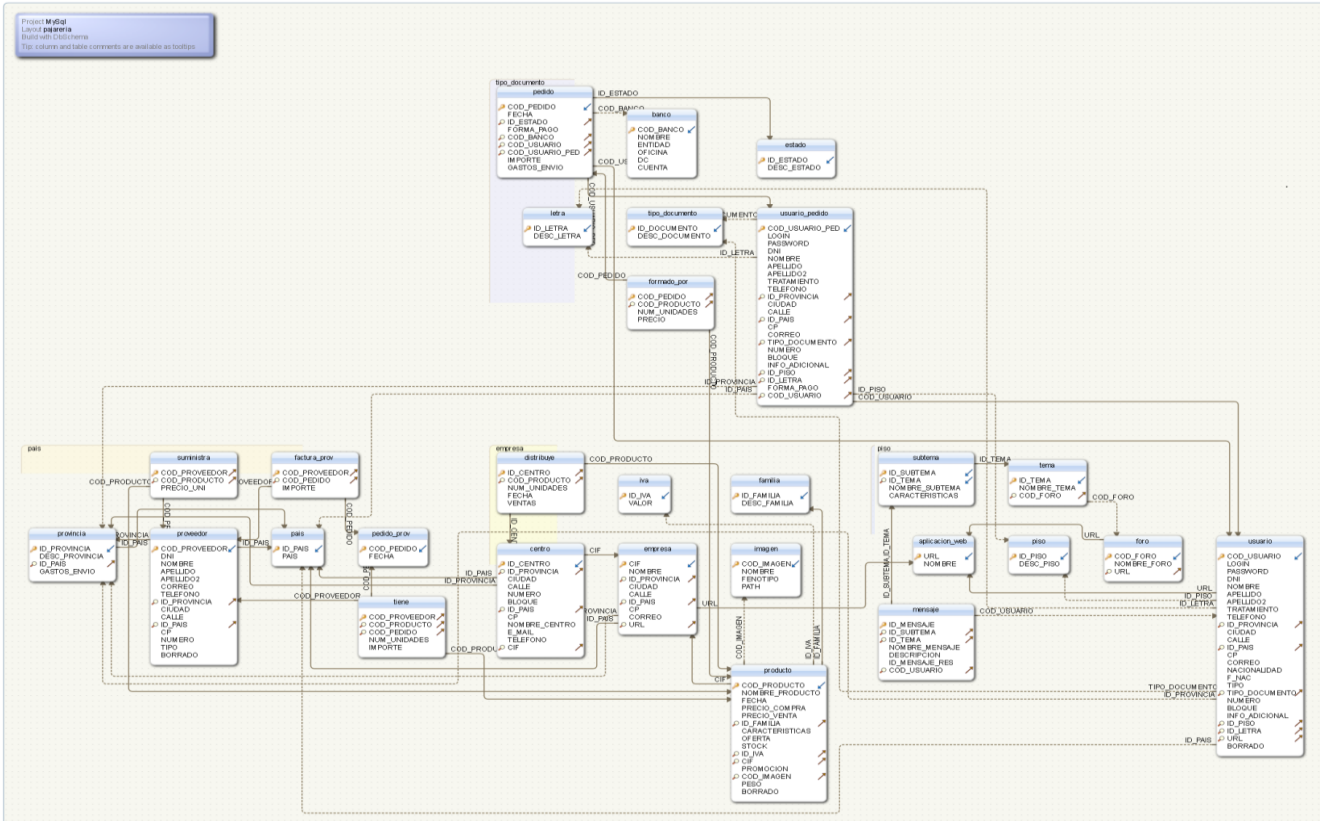


Figura 55. Esquema normalizado

8. Generación de Especificaciones de Construcción

En esta actividad se generan las especificaciones para la construcción del sistema de información.

8.1. Especificación del Entorno de Construcción

El objetivo de esta tarea es la definición detallada y completa del entorno necesario para la construcción de los componentes del sistema de información.

La especificación del entorno se realiza en base a los siguientes conceptos:

8.1.1 Entorno Tecnológico

<i>Procesador</i>	Intel® Core™ 2 Duo CPU P8700 @2.53GHz
<i>Memoria RAM</i>	4,00 GB
<i>Tipo sistema</i>	Windows 7 de 64bits
<i>Disco Duro</i>	50GB

8.1.2 Herramientas de construcción

<i>Herramientas de desarrollo</i>	Websphere Application Server Toolkit
<i>Herramientas gestor base de datos</i>	MySQL Server 5.5
<i>Servidor de aplicaciones</i>	Websphere 6.1
<i>Servidor web</i>	Apache 2.0
<i>Lenguaje de programación</i>	Java J2EE 6.0 JRE 1.5 y javascript

8.1.3. Planificación

La planificación que seguiremos para la implantación del entorno tecnológico seguirá los siguientes pasos:

- Implantación de la base de datos
- Carga Inicial de Datos
- Creación del servidor de aplicaciones
- Despliegue de la aplicación
- Configuración del servidor web
- Mapeo de DNS

8.1.4. Requisitos de Operación y seguridad

La seguridad se debe configurar en los siguientes niveles:

Acceso al servidor: A nivel de sistema operativo, el servidor de base de datos y web, sólo debe ser administrado por un usuario con rol administrador.

Usuario administrador de la base de datos: La base de datos debe ser administrada por un usuario con perfil de súper usuario o administrador, y protegido por clave.

Administración de usuarios que accederán a la aplicación: El acceso al sistema será diferente dependiendo del rol del usuario (administrador, cliente), el acceso será protegido por clave.

Firewalls: El servidor web que está conectado a Internet deberá estar protegido por un firewall o cortafuegos, a efectos de proteger esta información.

Backups: Se realizarán copias de seguridad de todo el código fuente y de las bases de datos de forma diaria, a fin de forma de proteger los desarrollos realizados.

9. Especificación Técnica del Plan de Pruebas

En esta actividad se realiza la especificación del plan de pruebas del sistema de información para cada uno de los niveles de prueba establecidos.

Los niveles de prueba son los siguientes:

- **Pruebas unitarias.** Las pruebas unitarias tienen como objetivo verificar la funcionalidad y estructura de cada componente individualmente una vez que ha sido codificado.
- **Pruebas de integración.** El objetivo de las pruebas de integración es verificar el correcto ensamblaje entre los distintos componentes una vez que han sido probados unitariamente con el fin de comprobar que interactúan correctamente a través de sus interfaces, tanto internas como externas y cubren la funcionalidad establecida además de ajustarse a los requisitos no funcionales especificados en las verificaciones correspondientes.

La definición de las pruebas y su resultado se especifican el siguiente bloque (Construcción del Sistema de Información).

10. Modelo de Comportamiento del Sistema

10.1. Diagramas de Secuencia del Sistema

Los diagramas de secuencia muestran las interacciones que se establecen entre los objetos de la aplicación a través del tiempo. En estos diagramas se indicarán los módulos o clases que forman parte del programa y las llamadas que se hacen en cada uno de ellos para realizar una tarea determinada.

Los mensajes se muestran como flechas entre líneas de vida. Un diagrama de secuencia puede mostrar un escenario, es decir, una historia individual de transacción. Un uso de un diagrama de secuencia es mostrar la secuencia del comportamiento de un caso de uso.

Un diagrama de secuencia posee dos dimensiones: la vertical representa el tiempo, la horizontal representa los objetos que participan en la interacción. En general, el tiempo avanza hacia abajo dentro de la página (se pueden invertir los ejes si se desea). Con frecuencia sólo son importantes las secuencias de mensajes pero en aplicaciones de tiempo real el eje temporal puede ser una métrica. La ordenación horizontal de los objetos no tiene ningún significado.

10.1.1. Diagrama de Secuencia Insertar Entidad

Este diagrama representa la secuencia de acciones que sigue nuestro sistema para realizar la inserción de una entidad en la base de datos. En nuestro caso este diagrama es el mismo para representar las inserciones de entidades como usuarios, productos, proveedores y centros.

Una vez haya escogido el usuario cualquier opción que implique la inserción de una entidad, el sistema le presentará un formulario donde podrá rellenar la información correspondiente al objeto que se quiere dar de alta. A continuación el sistema realiza una validación previa de los datos introducidos y comprueba si ya existe una entidad similar en el sistema. En caso de no existir se insertará la nueva entidad en la base de datos, sino se le mostrará un mensaje de error al usuario.

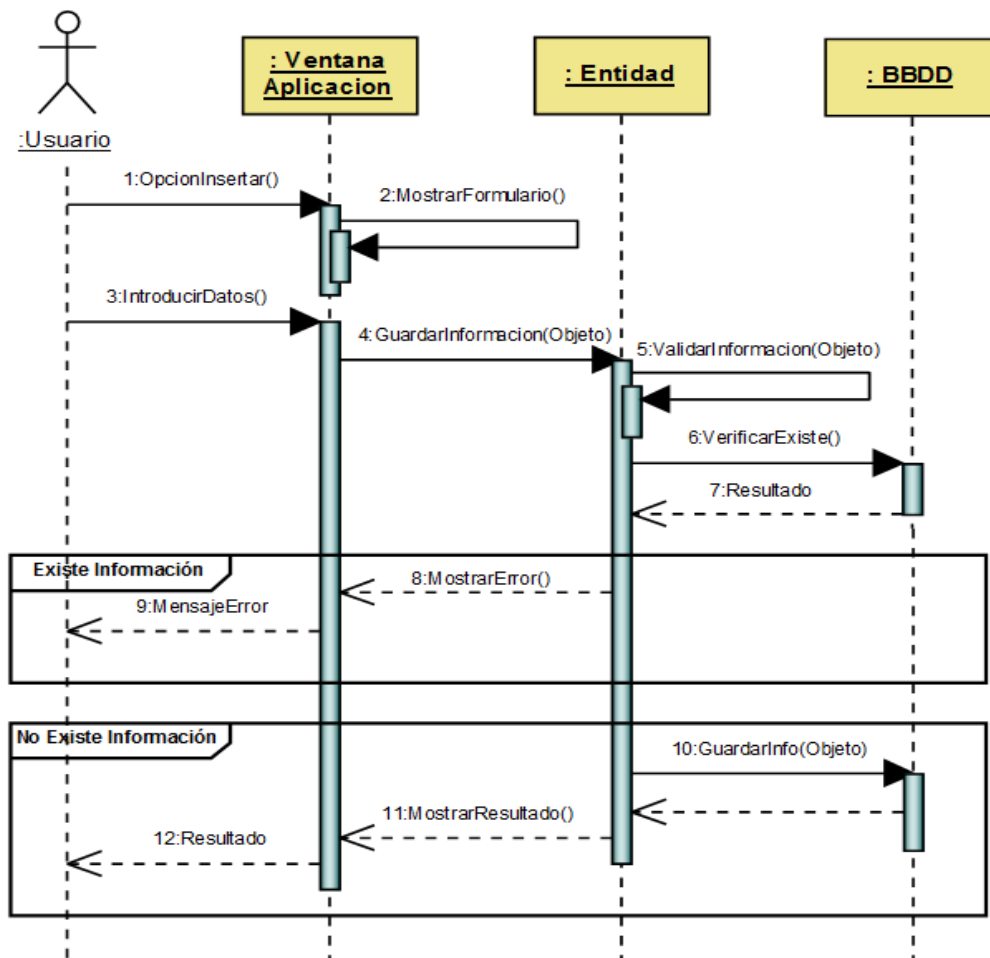


Figura 56. Diagrama de secuencia insertar entidad

10.1.2. Diagrama de Secuencia Eliminar Entidad

En este diagrama se define la secuencia de acciones que dan como resultado la eliminación de una entidad del sistema. En nuestro caso este diagrama es el mismo para representar las eliminaciones de entidades como usuarios, productos, proveedores y centros.

Una vez que el usuario haya escogido cualquier opción que implique la eliminación de una entidad, la aplicación presentará un formulario donde podrá buscar la entidad que desea borrar. El sistema recuperará de base de datos la información solicitada y el usuario confirmará su eliminación. En caso de que el elemento solicitado no exista en base de datos se mostrará un mensaje de error.

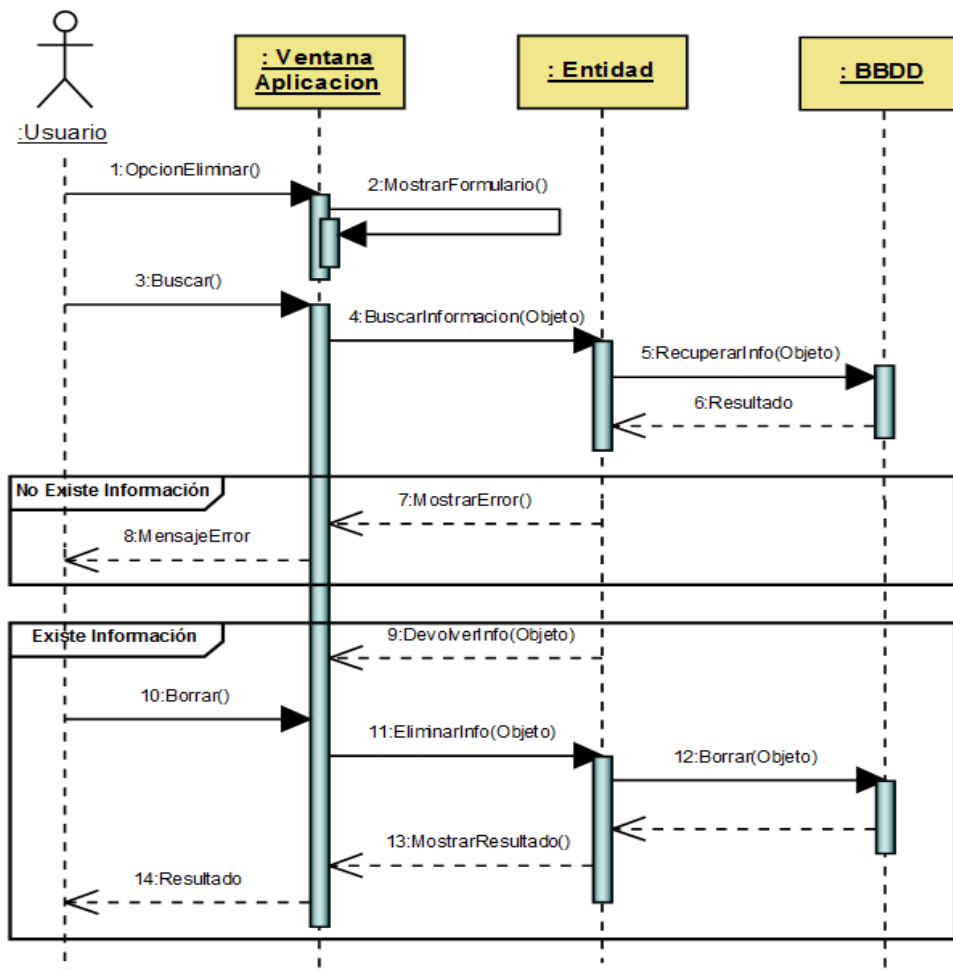


Figura 57. Diagrama de secuencia eliminar entidad

10.1.3. Diagrama de Secuencia Modificar Entidad

En este diagrama se define la secuencia de acciones que dan como resultado la modificación de una entidad del sistema. En nuestro caso este diagrama es el mismo para representar las modificaciones de entidades como usuarios, productos, proveedores y centros.

Una vez que el usuario haya escogido cualquier opción que implique la eliminación de una entidad, la aplicación presentará un formulario donde podrá buscar la entidad que desea borrar. El sistema recuperará de base de datos la información solicitada y el usuario confirmará su eliminación. En caso de que el elemento solicitado no exista en base de datos se mostrará un mensaje de error.

mostrará un mensaje de error.

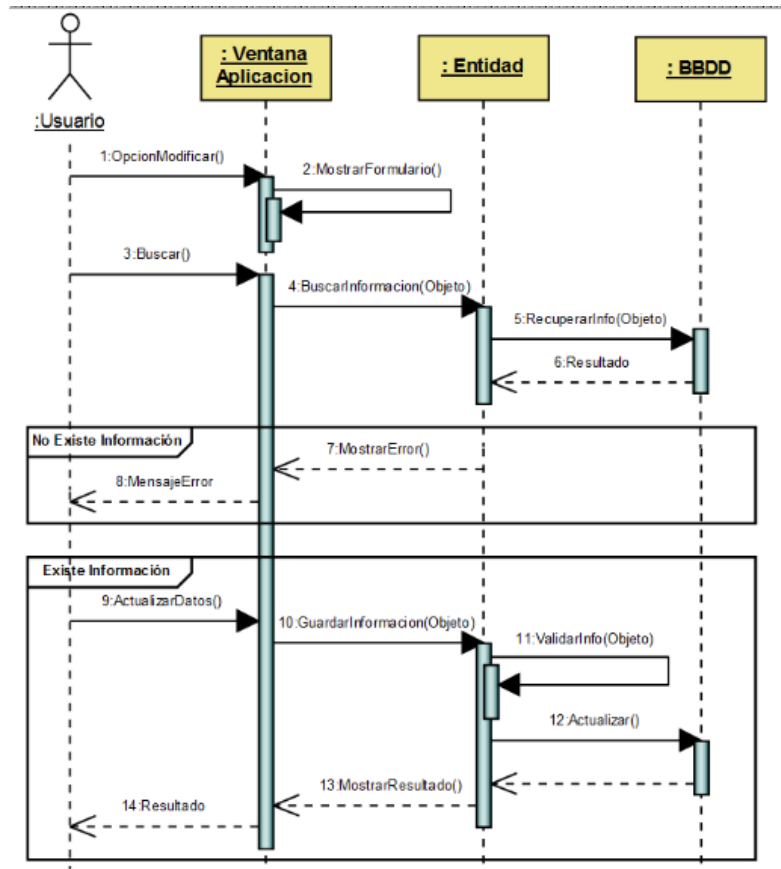


Figura 58. Diagrama de secuencia modificar entidad

10.1.4. Diagrama de Secuencia Buscar Información

En este diagrama se define la secuencia de acciones que dan como resultado la búsqueda de una entidad del sistema. Este diagrama será común para la búsqueda de cualquier entidad del sistema como puede ser la búsqueda de un producto, proveedor, cliente, centro o pedido.

En primer lugar el usuario elegirá una opción de menú en la cual se presente un formulario con algún campo de búsqueda de información. A continuación introducirá el filtro de búsqueda con la información que desea recuperar. La aplicación recuperará la información que cumpla con los criterios de búsqueda que ha introducido el usuario. En caso de no existir ninguna entidad que se ajuste a los parámetros de búsqueda se mostrará un mensaje de error.

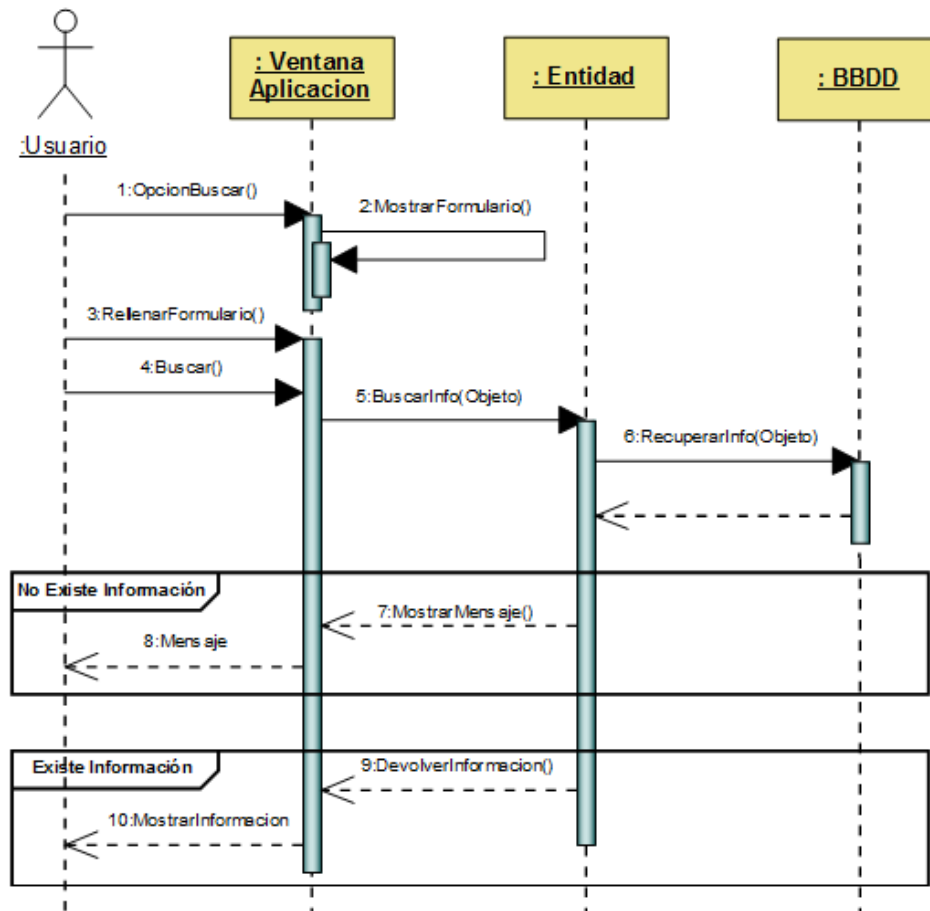


Figura 59. Diagrama de secuencia buscar entidad

10.1.5. Diagrama de Secuencia Comprar Producto

En este diagrama se explica la secuencia de acciones que cualquier usuario de la web debe seguir para realizar un pedido de productos online.

En primer lugar el usuario elegirá la opción del menú correspondiente a la realización de pedidos a través de la web. La aplicación presentará un primer formulario a través del cual el cliente podrá elegir los productos que desea adquirir y la cantidad de cada uno de ellos. Por cada producto que se añada al carrito, el sistema validará que está disponible en el catálogo y que su stock es mayor que el número de unidades que se quiere comprar. En caso de no cumplirse alguna de las validaciones la web mostrará un mensaje de error al usuario. En cualquier momento el usuario podrá modificar el pedido que está realizando eliminando productos del carrito, añadiendo nuevos o modificando la cantidad de alguno de ellos.

Una vez que el usuario está conforme con el carrito continuará el proceso de compra y a continuación la aplicación le mostrará otro formulario con el desglose del importe por productos y el importe total de todo el pedido. En caso de que el cliente continúe con el proceso de compra la aplicación le mostrará un formulario con todos los datos que el usuario proporcionó al darse de alta en el sistema. En este formulario también se le solicitará al usuario que escoja uno de los métodos de pago disponibles, en este caso son pago contra reembolso o mediante transferencia bancaria. El sistema validará que todos los datos del formulario son correctos y en caso de ser así insertará en la base de datos de la aplicación toda la información correspondiente al pedido que se estaba tramitando. Una vez se ha guardado el pedido se mostrará al cliente un resumen del pedido.

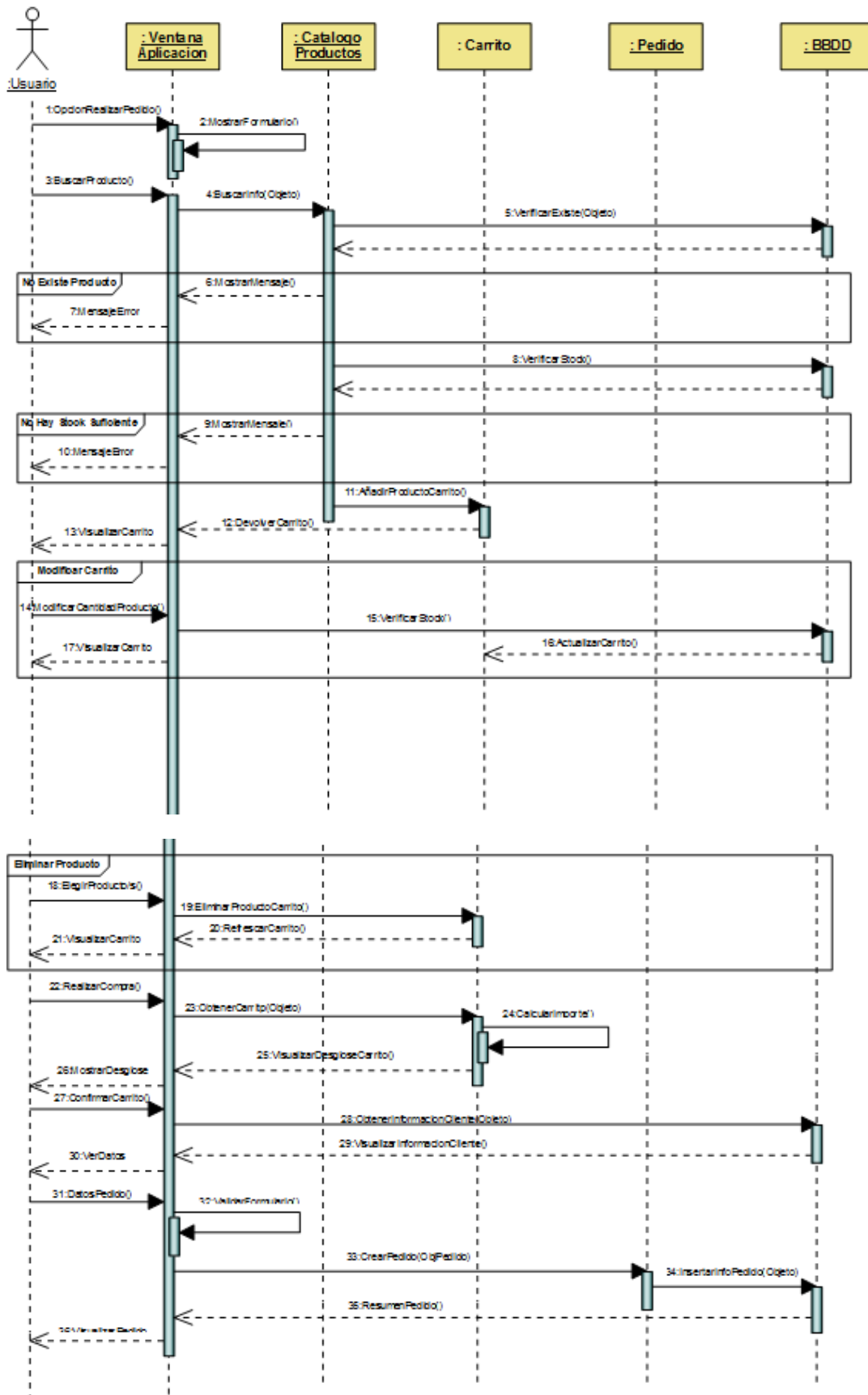


Figura 60. Diagrama de secuencia comprar producto

10.1.6. Diagrama de Secuencia Cambiar Credenciales Cliente

Este diagrama expone la secuencia de acciones que cualquier usuario de la web seguirá para modificar sus credenciales.

Una vez que el usuario ha escogido la opción del menú que le permitirá modificar sus datos de alta, la aplicación recuperará de base de datos toda la información relativa al dicho cliente. El usuario podrá modificar su login de acceso a la web. Una vez modificado, antes de guardar el nuevo login el sistema verificará que no esté siendo utilizado por ningún otro cliente. Si cumple las validaciones se actualizará el nuevo login del cliente. En caso de que no pueda actualizarse el login el sistema mostrará un mensaje de error por pantalla indicándolo.

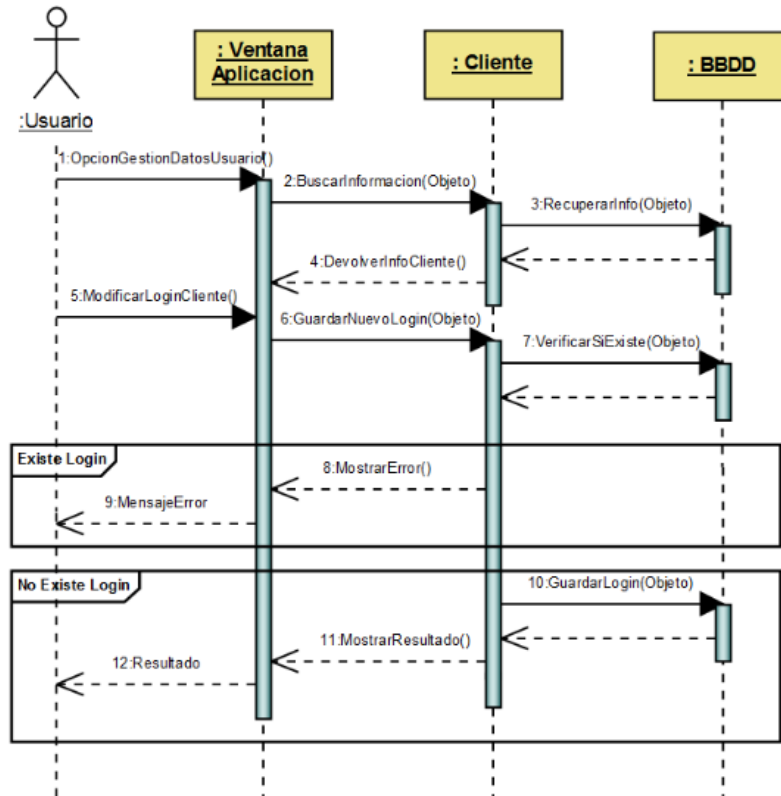


Figura 61. Diagrama de secuencia cambiar credenciales cliente

10.1.7. Diagrama de Secuencia Consultar Información

En este diagrama se define la secuencia de acciones que dan como resultado la consulta de una entidad del sistema.

El sistema mostrará un formulario a través del cual el usuario podrá introducir la entidad que desea consultar. A continuación la aplicación consultará la base de datos para recuperar la información asociada al elemento de búsqueda. En caso de que exista se mostrará la información disponible y en caso contrario se mostrará un mensaje de error por pantalla para indicarlo.

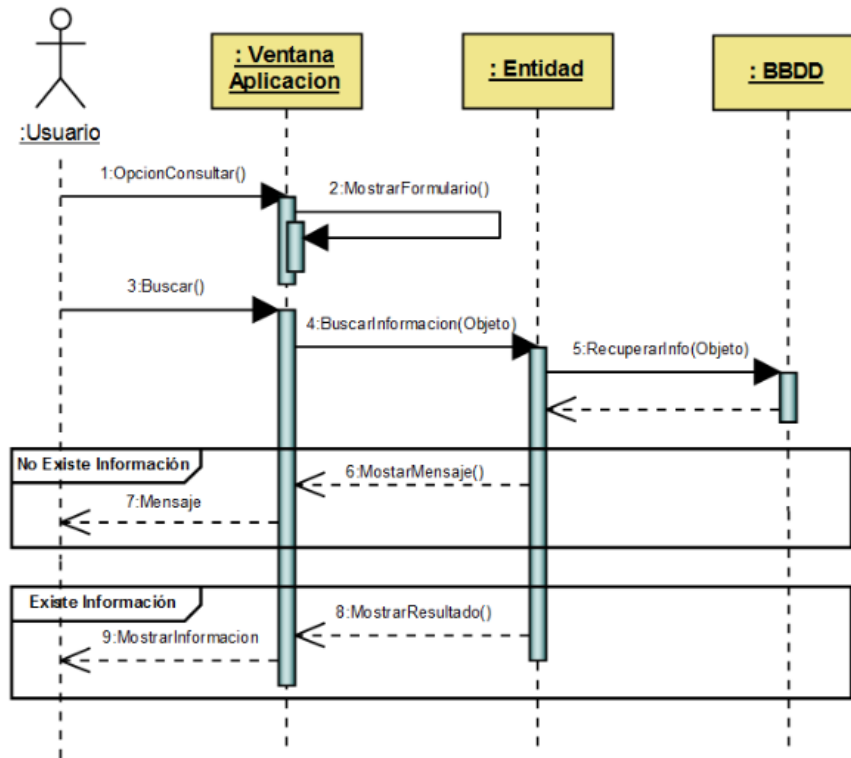


Figura 62. Diagrama de secuencia consultar información

10.1.8. Diagrama de Secuencia Identificarse

Este diagrama muestra la secuencia de acciones que cualquier usuario de la web seguirá para logarse en la web.

Para poder logarse en la web el usuario deberá introducir su login y password en el formulario habilitado para ello. A continuación el sistema verificará que en base de datos existe un usuario con dichas credenciales. En caso de que exista se logará en el sistema mostrándole el menú de cliente o administrador en función del rol del cliente que se esté logando. Si no existe ningún usuario con estas credenciales, la web mostrará un mensaje de error para indicarlo.

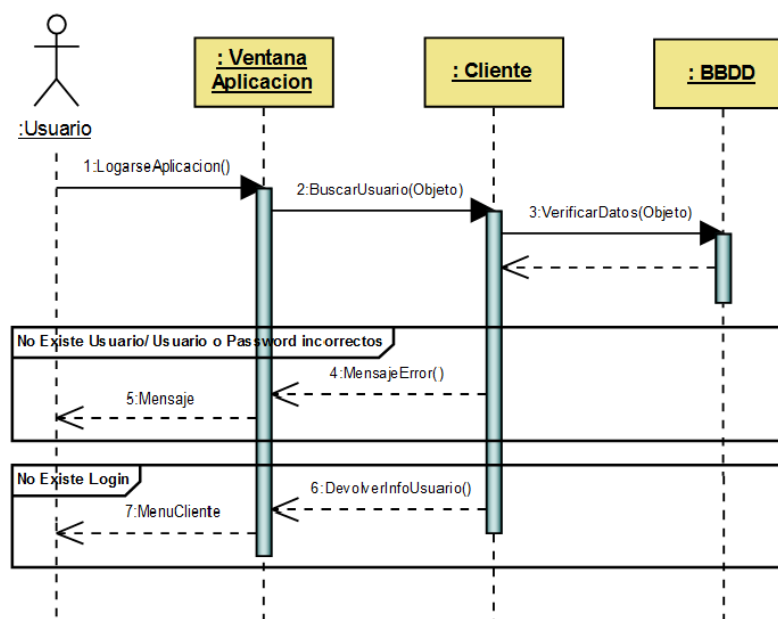


Figura 63. Diagrama de secuencia identificarse

10.2. Diagramas de Actividad del Sistema

Los diagramas de actividad permiten describir como un sistema implementa su funcionalidad. Modelan el comportamiento dinámico de un sistema resaltando la secuencia de actividades que se llevan a cabo y las condiciones que guardan o disparan esas actividades. La ventaja de este tipo de diagrama es que es uno de los elementos de modelado que mejor se comprenden por todos, ya que son herederos directos de los diagramas de flujo pero más expresivos. Heredan características de los diagramas de estado y los diagramas de flujo de datos.

En este tipo de diagramas podemos destacar los siguientes elementos.

- Tienen un único punto de inicio y fin.
- Está formado por un conjunto de actividades: conjunto de acciones que modelan un proceso.
- Está formado por un conjunto de acciones: una acción es un paso de un proceso.
- Decisiones: representan alternativas de flujo de control en un diagrama que se llevan a cabo en función de una condición.

10.2.1. Diagrama de Actividad Insertar Entidad

Este diagrama representa la secuencia de actividades que sigue nuestro sistema para realizar la inserción de una entidad en la base de datos. Al igual que ocurría con el diagrama de secuencia de insertar entidad, este diagrama es el mismo para representar las inserciones de entidades como usuarios, productos, proveedores y centros.

Como se puede ver en el diagrama la primera actividad que tenemos es la inserción de los datos de la nueva entidad por parte del usuario. El sistema validará que la información introducida es correcta y en caso de serlo guardará la nueva entidad en el sistema.

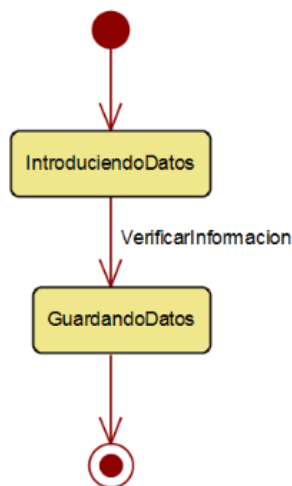


Figura 64. Diagrama de actividad Insertar Entidad

10.2.2. Diagrama de Actividad Eliminar Entidad

El diagrama de actividad de eliminar una entidad es común para representar las eliminaciones de entidades como usuarios, productos, proveedores y centros.

En primer lugar el usuario realizará la búsqueda de la entidad que quiere eliminar. El sistema recuperará la información del sistema y por último realizará la eliminación de la entidad solicitada.

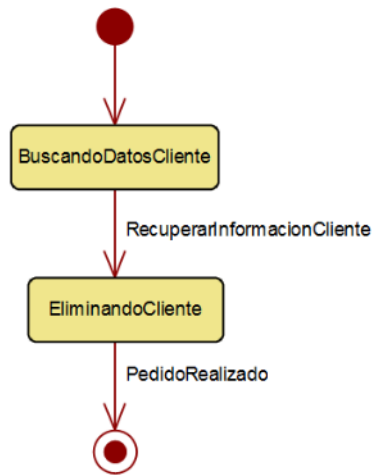


Figura 65. Diagrama de actividad Eliminar Entidad

10.2.3. Diagrama de Actividad Modificar Entidad

En este diagrama se define la secuencia de acciones que dan como resultado la modificación de una entidad del sistema. En nuestro caso este diagrama es el mismo para representar las modificaciones de entidades como usuarios, productos, proveedores y centros.

En el diagrama se puede ver como la primera actividad que se realiza es la búsqueda de los datos que desea modificar el cliente. El sistema recuperará la información de la base de datos y a continuación el usuario realiza las modificaciones que desee en la entidad que se acaba de recuperar. Antes de actualizar la entidad el sistema validará que las modificaciones que ha realizado el usuario son correctas.

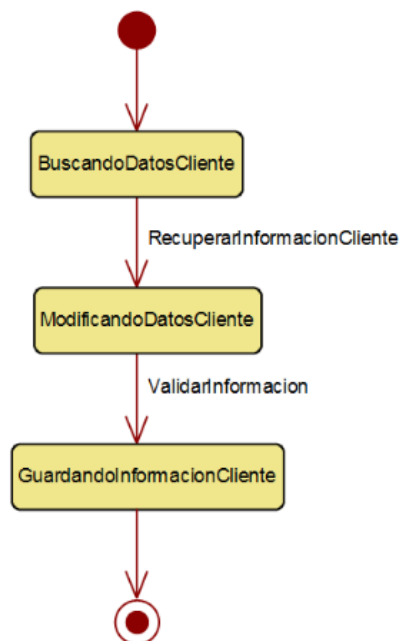


Figura 66. Diagrama de actividad Modificar Entidad

10.2.4. Diagrama de Actividad Buscar Información

En este diagrama tiene como objetivo explicar la secuencia de acciones que dan como resultado la búsqueda de información en el sistema.

La primera actividad que nos encontramos es el relleno de los filtros de búsqueda por parte del usuario. El sistema recuperará la información y la visualizará por pantalla.

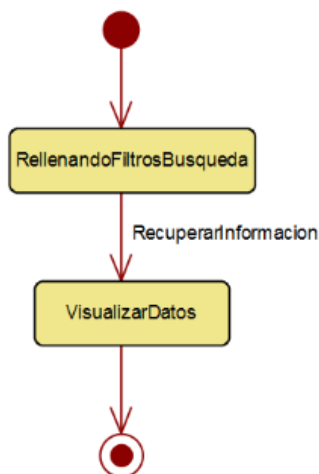


Figura 67. Diagrama de actividad Buscar Información

10.2.5. Diagrama de Actividad Comprar Producto

A continuación expondremos el diagrama de actividad correspondiente a la acción de realizar una compra desde la web.

En primer lugar nos encontramos con la actividad de búsqueda de un producto del catálogo por parte del cliente. Esta actividad se puede repetir tantas veces como productos desee buscar el cliente. A continuación el cliente seleccionará el producto que desea adquirir y la cantidad de cada uno de ellos. El sistema verificará que los datos introducidos por el cliente son correctos. En caso de serlo se añade el producto al carrito. Una vez completada la actividad de rellenar el carrito, el usuario deberá elegir la forma en la que realizará el pago del pedido. Si la opción que se elige es el pago por transferencia bancaria deberá rellenar la información de facturación y el sistema validará si es correcta para poder completar el pedido. En caso de que elija pago por contra reembolso se completará el pedido directamente sin que el sistema realice ninguna validación más.

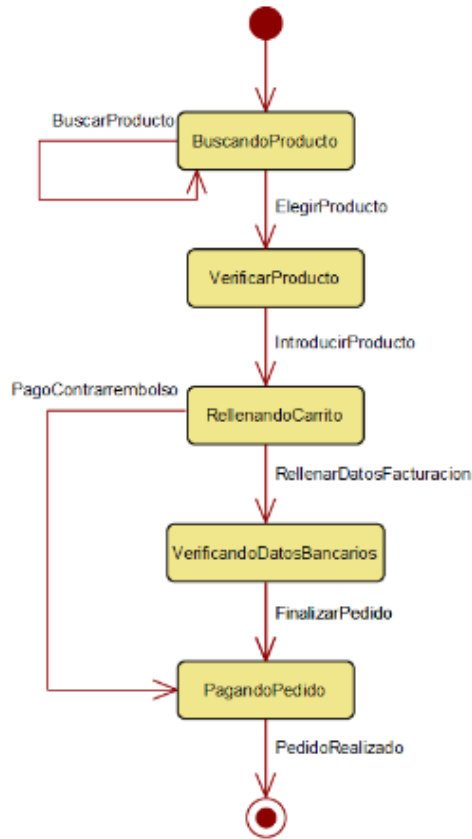


Figura 68. Diagrama de actividad Comprar Producto

10.2.6. Diagrama de Actividad Cambiar Credenciales Cliente

Este diagrama expone las actividades que cualquier usuario de la web seguirá para modificar sus credenciales.

Para que un usuario pueda realizar la acción de modificar sus credenciales, previamente el sistema deberá realizar la actividad de recuperarlos de base de datos. Es en ese momento cuando el cliente ya podrá modificar sus credenciales. Por último el sistema verificará que los nuevos credenciales son válidos y en ese caso actualizará los datos del usuario.

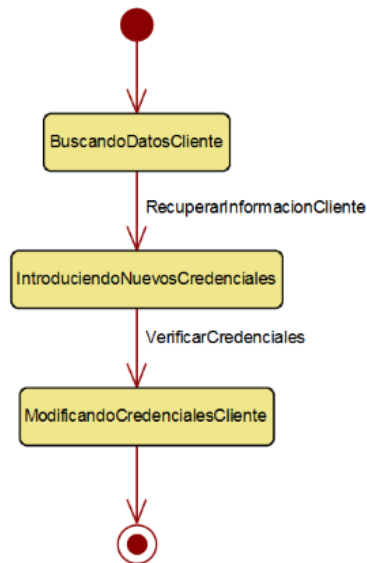


Figura 69. Diagrama de actividad Cambiar Credenciales Cliente

10.2.7. Diagrama de Actividad Consultar Información

En este diagrama se define la secuencia de actividades que dan como resultado la consulta de una entidad del sistema.

El usuario realizará la actividad de la búsqueda de la información que desea recuperar. El sistema recuperará los datos mostrándosela al usuario.

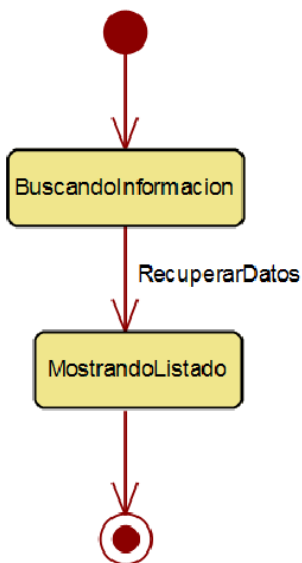


Figura 70. Diagrama de actividad Consultar Información

10.2.8. Diagrama de Actividad Identificarse

A continuación mostramos el diagrama que representa las actividades que representan la acción de logarse en la web.

Para poder logarse en la web el usuario realizará la acción de introducir sus credenciales. El sistema buscará un usuario en base de datos que responda a los credenciales introducidos. En caso de que sí encuentre el usuario la aplicación realizará el proceso de login mostrándole menú correspondiente al rol que tenga dicho usuario.

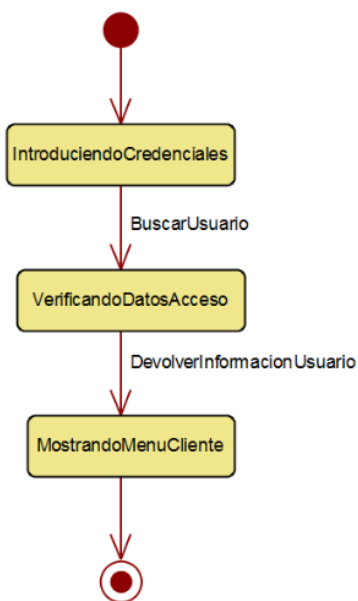


Figura 71. Diagrama de actividad Identificarse

Implementación y Pruebas del Sistema

Índice de Contenido

Construcción del Sistema de Información

1. Preparación del Entorno de Generación y Construcción.....	194
1.1. Implantación de la Base de Datos Física o de Ficheros	194
2. Ejecución de las pruebas unitarias	197
2.1. Página principal de usuario	197
2.2. Alta y modificación de usuario	197
2.3. Catálogo de productos.....	199
2.4. Contacto.....	199
2.5. Pedidos	199
2.6. Formulario de pagos	200
2.7. Programa Mendel.....	201
2.8. Gestión de clientes	205
2.9 Gestión de cliente	206
2.10. Gestión de proveedor	206
2.11. Alta y modificación de proveedor	207
2.12. Gestión de productos.....	207
2.13. Modificación de productos	208
2.14. Pedidos	209
2.15. Gestión de pedido	209
2.16. Gestión de pedidos a clientes	209
2.17. Gestión de pedidos a proveedores.....	210
2.18. Gestión de centros	211
2.19. Alta y modificación de centros	212
2.20. Consulta de stock	213
2.21. Nueva oferta	213
2.22. Modificación oferta	214
2.23. Visualización de gráficos.....	214
3. Ejecución de las pruebas de integración	216
3.1 Diseño pruebas de integración.....	217
3.2. Definición de las pruebas	217
3.2.1. Alta usuario.....	217
3.2.2. Modificación datos usuario	218
3.2.3. Programa Mendel.....	218
3.2.4. Gestión de proveedores.....	223
3.2.5. Gestión de centros	225
3.2.6. Gestión de productos.....	225
3.2.7. Pedidos a proveedores	226
3.2.8. Pedidos de clientes	226
3.2.9. Consulta de stock	227

3.2.10. Generar catálogo	227
3.2.11. Gestión clientes	228
3.2.12. Visualización de gráficos	228

1. Preparación del Entorno de Generación y Construcción

En esta actividad se asegura la disponibilidad de todos los medios para que se pueda llevar a cabo la construcción del sistema de información. Entre estos están incluidos la preparación de puestos, creación y carga inicial de la base de datos, herramientas de generación del código entre otros.

1.1. Implantación de la Base de Datos Física o de Ficheros

Una vez normalizada la base de datos, hemos generado un script para poder crear las tablas de nuestra base de datos automáticamente. Este script junto con los scripts de carga de datos en las tablas (las que necesitan tener datos inicialmente), se encuentran adjuntos junto con toda la información del proyecto.

Las tablas que necesitan de la carga inicial de datos para el correcto funcionamiento de la web se especifican a continuación:

Tabla	Especificación
Aplicación Web	Tabla que almacena el nombre y la url de la compañía
Banco	Tabla que almacena los distintas cuentas donde el cliente puede hacer el pago del pedido
Tema	Tabla que almacena los distintos tipos de temas sobre los que puede consultar el usuario
Subtema	Tabla que almacena los distintos tipos de subtema sobre los que puede consultar el usuario
Empresa	Tabla que almacena las empresas asociadas
Imagen	Tabla que almacena las imagenes asociadas para los distintos tipos de pájaros
Estado	Tabla que almacena los estados por los que pasa cada tipo de pedido
Familia	Tabla que almacena los distintos tipos de productos
Iva	Tabla que almacena los distintos valores de iva
Letra	Tabla que almacena la letra del domicilio del cliente
Piso	Tabla que almacena el piso del domicilio del cliente
Tipo Documento	Tabla que almacena los distintos tipos de documentos del cliente
Usuario Administrador	Los usuarios administradores no se dan de alta a través del sistema sino directamente en base de datos

A continuación se detalla la carga inicial de cada una de las tablas especificadas en la tabla anterior.

Aplicación Web

nombre – pajarracos
url - <http://www.pajarracos-noip.org/>

Banco (de momento con valores ficticios)

cod_banco - 1
nombre – Santander
entidad – 1234
oficina – 1111
dc - 10
nº cuenta – 1234567890

Tema

id_tema – incremental (1..N)
nombre_tema – *

*nombre tema: cuidados, habilidades, enfermedades, curiosidades

Subtema

id_tema – incremental (1..N)
id_subtema – incremental (1..M)
nombre_subtema - *
caracteristicas - *

Por cada fila del tema y para cada especie se especifica la url de acceso al fichero con las características de la especie.

Empresa (de momento con valores ficticios)

cif – A2345678C
nombre - Pajareria Juan
cod_provincia – SG
ciudad – Segovia
calle – CORONEL REXACH
id_pais – España
cp – 4004
correo – <http://www.pajarracos-sg.com>
url - <http://www.pajarracos-noip.org/SG>

Imagen

cod_imagen - incremental
nombre – *
fenotipo – *
path - *

Para cada combinación de los distintos fenotipos obtenemos el tipo de pájaro con el nombre, fenotipo y url del fichero físico donde se almacena la imagen del pájaro.

Estado

id_estado – incremental (1..N)
desc_estado - *

*desc_estado : realizado, pagado, entregado, pendiente de envio

Familia

id_familia – incremental (1..N)
desc_familia - *

*desc_familia : pendiente de especificar

IVA

id_iva – incremental (1..N)
valor - *

*valor : 1.21, 1.01, 1.04

Letra

id_letra – incremental (1..N)
desc_letra - *

*desc_letra : A-Z

Piso

id_piso : incremental (1..N)
desc_piso : 1-50

TipoDocumento

id_documento – incremental (1..N)
desc_documental - *

*desc_documental : DNI; Targeta de residencia, Pasaporte

Usuario (Administrador)

cod_usuario – AD0001
login – admin
password – admin
nombre – Ruben
apellido – Ortega
apellido2 – sanchez
tratamiento – 1
telefono – 656656656
id_provincia – MA
ciudad – Madrid

calle – Bravo Murillo
id_pais – ES
cp – 28080
correo – admin@gmail.com
nacionalidad – española
f_nac – 12-12-1988
tipo – 0
tipo_documento – 1
numero – 1
bloque –
info_adicional – null
id_piso – 1
id_letra – A
url – <http://www.pajarracos-noip.org>
borrado – N

2. Ejecución de las pruebas unitarias

Las pruebas unitarias tienen como objetivo verificar la funcionalidad y estructura de cada componente individualmente una vez que ha sido codificado.

Existen dos enfoques principales para el diseño de casos de prueba:

- **Enfoque estructural o de caja blanca.** Se verifica la estructura interna del componente con independencia de la funcionalidad establecida para el mismo. Por tanto, no se comprueba la corrección de los resultados si éstos se producen.
- **Enfoque funcional o de caja negra.** Se comprueba el correcto funcionamiento de los componentes del sistema de información, analizando las entradas y salidas y verificando que el resultado es el esperado. Se consideran exclusivamente las entradas y salidas del sistema sin preocuparse por la estructura interna del mismo.

Las pruebas unitarias que se detallan a continuación tienen como objetivo verificar la funcionalidad y estructura de cada componente. Esto se realizara mediante pruebas de caja blanca complementándolos con pruebas de caja negra. El plan de pruebas será la ejecución del funcional desarrollado, ejecutando cada pantalla y todas las modificaciones sufridas durante el prototipado y el desarrollo del mismo.

Para los caracteres numéricos el rango se especifica como rango de valores y en los alfabéticos el número de caracteres.

2.1. Página principal de usuario

Página presentación de la tienda. Desde esta pantalla el usuario podrá logarse o darse de alta como cliente. Contendrá también un menú lateral con todas las opciones a las que tendrán acceso todos los usuarios.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Login	1. Campo de texto alfanumérico. 2. $1 \leq \text{login} \leq 10$	3. Caracteres extraños. 4. Campo vacío 5. $\text{login} < 1$ 6. $\text{login} > 10$
Password	7. Campo de texto alfanumérico. 8. $1 \leq \text{login} \leq 10$	9. Caracteres extraños. 10. Campo vacío 11. $\text{login} < 1$ 12. $\text{login} > 10$

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Login y Password	El usuario no existe en la base de datos.	El sistema vuelve a mostrar el formulario para logarse.
	El usuario existe en la base de datos. Dos casos	
	El usuario es un cliente	El sistema muestra la pantalla de cliente
	El usuario es un centro	El sistema muestra la pantalla de cliente
	El usuario es un administrador	El sistema muestra la pantalla de Administrador

Para todas las combinaciones de las clases de equivalencia no validas el sistema advierte de ello al usuario mostrando los mensajes de error.

2.2. Alta y modificación de usuario

Pantalla en la que el usuario deberá introducir sus datos personales para poder darse de alta como cliente en el sistema.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Tratamiento	1. Elige un valor (campo chequeado)	2. No selecciona ningún valor (campo no chequeado)
Nombre	3. Campo de texto alfabético. 4. $1 \leq \text{Nombre} \leq 20$	5. Caracteres extraños o numéricos. 6. $1 > \text{Nombre}$ 7. $20 < \text{Nombre}$
Primer Apellido	8. Campo de texto alfabético. 9. $1 \leq \text{apellido} \leq 20$	10. Caracteres extraños o numéricos. 11. $1 > \text{apellido}$

Segundo Apellido	13. Campo de texto alfabético. 14. 1<= apellido <=20	12. 20< apellido e 15. Caracteres extraños o numéricos. 16. 1> apellido 17. 20< apellido
Tipo Documento	18. Selecciona una opción	19. Campo vacío
DNI	19. Campo de texto alfanumérico 20. Tamaño DNI = 9 21. Formato xxxxxxxxA	22. Caracteres extraños 23. Tamaño DNI != 9 24. No cumple el formato
Fecha nacimiento	25. Los caracteres permitidos son dígitos y "p". 26. Tamaño fecha = 10 27. Fecha válida. 28. Formato dd/mm/aaaa 29. El usuario debe ser mayor de 14 años. Por ejemplo fecha introducida es posterior a dd/mm/1993	30. Caracteres extraños o alfabéticos. 31. Tamaño fecha != 10 32. Fecha incorrecta. Ejemplo 31/02/2008 33. Formato incorrecto 34. El usuario es menor de 14 años Ejemplo 31/02/1998
Nacionalidad	35. Campo de texto con caracteres alfabéticos. 36. 1<= Nacionalidad <=25	36. El campo contiene caracteres extraños o dígitos. 37. 1 > Nacionalidad 38. 25 < Nacionalidad
Correo electrónico	39. 1<= correo <=40 40 Formato nombre@dominio	41. 1> correo 42. 40< correo 43. Formato incorrecto
Teléfono	44. Tamaño teléfono = 9 44. Formato correcto (empieza por 6) 45. Caracteres numéricos	46. Tamaño distinto a 9 46. Formato incorrecto 47. Caracteres alfabéticos o extraños
Tipo de cliente	48. Elige un valor (campo chequeado)	49. No selecciona ningún valor (campo no chequeado)
Calle	50. Caracteres alfabéticos 51. 1<= calle <=50	52. Caracteres numéricos o extraños. 53. 1> calle 54. 50 < calle
Numero	55. Caracteres numéricos 54. 1<= numero <= 999	56. Caracteres extraños o alfabéticos 57. 1 > numero 58. 999 < numero
Piso	59. Selecciona una opción	60. Campo vacío
Letra	61. Selecciona una opción	62. Campo vacío
Bloque	63. Caracteres alfanumérico 64. 1<= bloque <= 2	65. Caracteres extraños o dígitos 66. 1> bloque 67. 2 < bloque
Población	68. Caracteres alfabéticos 69. 1<= población <= 40	70. Caracteres numéricos o extraños. 71. 1 > población 72. 40 < población
Código Postal	73. Caracteres numéricos 74. 00001 <= código <=99999	75. Caracteres extraños o alfabéticos. 76. 00001 > código postal 77. 99999 < código postal
País	78. Selecciona una opción	79. Campo vacío
Provincia	80. Selecciona una opción	81. Campo vacío
Información adicional	82. Caracteres alfanuméricos 83. 1 <= info <= 50	84. Caracteres extraños 85. 1 > info 86. 50 < info
Login	87. Campo de texto alfanumérico. 88. 1<=login<=10	89. Caracteres extraños. 90. Campo vacío 91. login < 1 92. login > 10
Password	93. Campo de texto alfanumérico. 94. 1<=login<=10	95. Caracteres extraños. 96. Campo vacío 97. login < 1 98. login > 10

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Datos de usuario	El usuario no existe en la base de datos y el formulario cumple las clases de equivalencia válidas.	El sistema da de alta al usuario y muestra el módulo de cliente.
	El login introducido es usado por otro cliente.	El sistema muestra el error y pide la entrada de uno nuevo.

Para todos los casos en que no se verifiquen las clases de equivalencia especificadas, el sistema muestra un mensaje de error para cada caso.

Salvo los casos de información adicional y bloque el resto de campos no pueden estar vacíos, en caso de que alguno no cumpla esto el sistema advierte del error al usuario y no continúa con el proceso de alta.

En cuanto a los rangos de tamaño especificados el sistema impide la introducción de más caracteres de los especificados.

2.3. Catálogo de productos

Pantalla para la búsqueda de productos disponibles para los clientes

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Familia	1. Selecciona una opción 2. Campo Vacio	
Producto (Código)	3. Campo de texto numérico. 4. $0 \leq \text{producto} \leq 6$	5. Caracteres extraños o alfabéticos. 6. $\text{producto} < 0$ 7. $\text{producto} > 6$
Visualizar Ofertas	8. Elige un valor (campo chequeado)	9. No selecciona ningún valor (campo no chequeado)

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Familia, Código Producto, Visualizar Ofertas	El producto no existe.	El sistema informa al cliente.
	El/los producto/s existe en la base de datos.	El sistema muestra el listado de producto/s correspondientes a los criterios de búsqueda introducidos por el cliente
	El usuario no introduce los datos	El sistema muestra el listado de productos disponibles para el cliente

Para todos los casos en que no se verifiquen las clases de equivalencia especificadas el sistema muestra un mensaje de error para cada caso. Todas las entradas pueden ser vacías. En cuanto al rango de tamaño especificado para el código del producto el sistema impide la introducción de más caracteres de los especificados.

2.4. Contacto

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Nombre	1. Campo de texto alfabético 2. $1 \leq \text{nombre} \leq 20$	3. Caracteres extraños o numéricos. 4. $1 > \text{nombre}$ 5. $20 < \text{nombre}$
e-mail	6. Campo de texto alfanumérico. 7. $1 \leq \text{email} \leq 40$	8. Caracteres. 9. $1 > \text{e-mail}$ 10. $20 < \text{e-mail}$
Motivo de contacto	11. Campo de texto alfanumérico. 12. $1 \leq \text{motivo} \leq 40$	13. Caracteres. 14. $1 > \text{motivo}$ 15. $20 < \text{motivo}$
Comentarios	16. $1 \leq \text{comentarios} \leq 300$	17. $1 > \text{comentarios}$ 18. $300 < \text{comentarios}$

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Nombre, e-mail, motivo de contacto, Comentarios	El cliente rellena el formulario de contacto con todos los campos y selección "enviar".	El sistema abre Outlook con la información introducida por el cliente.
	El e-mail del centro introducido por el usuario no se corresponde con ningún centro de los existentes	El sistema muestra un mensaje advirtiendo de ello al cliente

Para todos los casos en que no se verifiquen las clases de equivalencia especificadas, el sistema muestra un mensaje de error. Ninguna entrada puede ser vacía. En cuanto al rango de tamaño especificado para las entradas, el sistema impide introducir más caracteres de los especificados.

2.5. Pedidos

Pantalla para la introducción de los productos pedidos por un determinado cliente.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Producto (Código)	1. Campo de texto numérico. 2. $1 \leq \text{producto} \leq 999999$	4. Caracteres extraños o alfabéticos. 5. $\text{producto} < 1$ 6. $\text{producto} > 999999$
Unidades	7. Campo de texto numérico. 8. $1 \leq \text{unidades} \leq 999$	9. Caracteres extraños o alfabéticos. 10. $\text{unidades} < 1$ 11. $\text{unidades} > 999$

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Código Producto, Unidades	El producto no existe.	El sistema informa al cliente.
	El producto existe en la base de datos.	El sistema cargara el nombre del producto en el campo adyacente

	El usuario no introduce los datos	El sistema muestra un mensaje solicitando la introducción de datos.
	El usuario introduce un producto existente en base de datos y elige el número de unidades.	El sistema comprueba que el número de unidades solicitadas por el usuario sea inferior al stock disponible para ese producto
	Unidades =< stock	El sistema añade el producto al listado de productos del carrito
	Unidades > stock	El sistema no añade el producto al carrito y advierte al usuario del error.

Para todos los casos en que no se verifiquen las clases de equivalencia especificadas el sistema muestra un mensaje de error. Todas las entradas pueden ser vacías. En cuanto al rango de tamaño especificado para el código del producto y las unidades, el sistema impide la introducción de valores mayores a los especificados.

2.6. Formulario de pagos

Pantalla en la que el usuario deberá introducir sus datos de contacto para poder continuar con el proceso de envío.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Tratamiento	1. Elige un valor (campo chequeado) 2. No selecciona ningún valor (centros)	
Nombre	3. Campo de texto alfabético. 4. 1<=Nombre <=20	5. Caracteres extraños o numéricos. 6. 1>Nombre 7. 20<Nombre
Primer Apellido	8. Campo de texto alfabético. 9. 0<=apellido <=20	10. Caracteres extraños o numéricos. 12. 20< apellido e
Segundo Apellido	13. Campo de texto alfabético. 14. 0<= apellido <=20	15. Caracteres extraños o numéricos. 16. 20< apellido
Tipo Documento	17. Selecciona una opción	18. Campo vacío
DNI	19. Campo de texto alfanumérico 20. Tamaño DNI = 9 21. Formato xxxxxxxxA	22. Caracteres extraños 23. Tamaño DNI != 9 24. No cumple el formato
Calle	22. Caracteres alfabéticos 23. 1<= calle <=50	24. Caracteres numéricos o extraños. 25. 1> calle 26. 50 < calle
Numero	27. Caracteres numéricos 28. 1<= numero <= 999	29. Caracteres extraños o alfabéticos 30. 1 > numero 31. 999 < numero
Piso / Letra	32. Campo alfanumérico 33. 0 <= piso/letra <= 4	34. Caracteres extraños 35 4 < piso/letra
Bloque	37. Caracteres alfanumérico 38. 1<= bloque <= 2	39. Caracteres extraños o dígitos 40. 1> bloque 41. 2 < bloque
Población	42. Caracteres alfabéticos 43. 1<= población <= 40	44. Caracteres numéricos o extraños. 45. 1 > población 46. 40 < población
Código Postal	47. Caracteres numéricos 48. 00001 <= código <=99999	49. Caracteres extraños o alfabéticos. 50. 00001 > código postal 51. 99999 < código postal
País	52. Selecciona una opción	53. Campo vacío
Provincia	54. Selecciona una opción	55. Campo vacío
Información adicional	56. Caracteres alfanuméricos 57. 1 <= info <= 50	58. Caracteres extraños 59. 1 > info 60. 50 < info
Correo electrónico	61. 1<= correo <=40 62 Formato nombre@dominio	63. 1> correo 64. 40< correo 65. Formato incorrecto
Teléfono	66. Tamaño teléfono = 9 67. Formato correcto (empieza por 6) 68. Caracteres numéricos	69. Tamaño distinto a 9 70. Formato incorrecto 71. Caracteres alfabéticos o extraños
Forma Pago	72. Elige un valor (campo chequeado)	73. No selecciona ningún valor (campo no chequeado)

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Datos de usuario	El usuario no selecciona la forma de pago.	El sistema impide continuar con el proceso de compra.
	Los campos obligatorios del formulario están vacíos.	El sistema impide continuar con el proceso de compra.
	El usuario rellena todos los campos obligatorios del formulario seleccionando pago contra reembolso como forma de pago y continúa con el proceso de compra.	El sistema termina el proceso de compra mostrando todos los datos del pedido.

El usuario rellena todos los campos obligatorios del formulario seleccionando pago por transferencia como forma de pago y continúa con el proceso de compra.

El sistema continúa con el proceso de compra mostrando todos los datos del pedido y solicitando la selección de una oficina para el pago del pedido. Una vez que el usuario selecciona un banco y confirma el pedido el sistema termina el proceso de compra.

Para todos los casos en que no se verifiquen las clases de equivalencia especificadas el sistema muestra un mensaje de error. Salvo los casos de información adicional y bloque, el resto de campos no pueden estar vacíos. En caso de que alguno no cumpla esto, el sistema advierte del error al usuario y no continúa con el proceso de alta. Esto no es aplicado en caso de que el usuario sea un centro en cuyo caso los campos obligatorios serán nombre, calle, número, población, código postal, país, provincia, correo electrónico, teléfono de contacto y forma de pago. En cuanto a los rangos de tamaño especificados el sistema impide la introducción de más caracteres de los especificados.

2.7. Programa Mendel

Aplicación desde la cual el cliente podrá hacer consultas sobre la posible descendencia de los diamantes de gould partiendo los progenitores.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Cabeza Roja Macho	1. Elige un valor (campo chequeado)	2. No selecciona ningún valor (campo no chequeado)
Cabeza Negra Macho	3. Elige un valor (campo chequeado)	4. No selecciona ningún valor (campo no chequeado)
Cabeza Naranja Macho	5. Elige un valor (campo chequeado)	6. No selecciona ningún valor (campo no chequeado)
Pecho Morado Macho	7. Elige un valor (campo chequeado)	8. No selecciona ningún valor (campo no chequeado)
Pecho Lila Macho	9. Elige un valor (campo chequeado)	10. No selecciona ningún valor (campo no chequeado)
Pecho Blanco Macho	11. Elige un valor (campo chequeado)	12. No selecciona ningún valor (campo no chequeado)
Cuerpo Verde Macho	13. Elige un valor (campo chequeado)	14. No selecciona ningún valor (campo no chequeado)
Cuerpo Amarillo Macho	15. Elige un valor (campo chequeado)	16. No selecciona ningún valor (campo no chequeado)
Cuerpo Azul Macho	17. Elige un valor (campo chequeado)	18. No selecciona ningún valor (campo no chequeado)
Cabeza Roja Hembra	19. Elige un valor (campo chequeado)	20. No selecciona ningún valor (campo no chequeado)
Cabeza Negra Hembra	21. Elige un valor (campo chequeado)	22. No selecciona ningún valor (campo no chequeado)
Cabeza Naranja Hembra	23. Elige un valor (campo chequeado)	24. No selecciona ningún valor (campo no chequeado)
Pecho Morado Hembra	25. Elige un valor (campo chequeado)	26. No selecciona ningún valor (campo no chequeado)
Pecho Lila Hembra	27. Elige un valor (campo chequeado)	28. No selecciona ningún valor (campo no chequeado)
Pecho Blanco Hembra	29. Elige un valor (campo chequeado)	30. No selecciona ningún valor (campo no chequeado)
Cuerpo Verde Hembra	31. Elige un valor (campo chequeado)	32. No selecciona ningún valor (campo no chequeado)
Cuerpo Amarillo Macho	33. Elige un valor (campo chequeado)	34. No selecciona ningún valor (campo no chequeado)
Cuerpo Azul Hembra	35. Elige un valor (campo chequeado)	36. No selecciona ningún valor (campo no chequeado)

Leyenda: S – Chequeado Si
 N – Chequeado No
 P – Chequeado Portador

Orden: Rojo Negro Naranja
 Morado Lila Blanco
 Verde Amarillo Azul

ENTRADAS MACHO	CASOS DE PRUEBA	SALIDA ESPERADA
Cabeza Macho	El usuario no selecciona alguno de los caracteres.	El sistema impide continuar y advierte del error al cliente.
S, S, S	El usuario selecciona rojo(si), negro (si) y naranja (si) a la vez	Combinación inválida de caracteres
S, S, N	El usuario selecciona rojo (si), negro (si) y naranja (no)	Combinación inválida de caracteres

S, S, P	El usuario selecciona rojo (si), negro (si) y naranja (portador)	Combinación invalida de caracteres
S, N, S	El usuario selecciona rojo (si), negro (no) y naranja (si)	Combinación valida
S, N, N	El usuario selecciona rojo (si), negro (no) y naranja (no)	Combinación valida
S, N, P	El usuario selecciona rojo (si), negro (no) y naranja (portador)	Combinación valida
S, P, S	El usuario selecciona rojo (si), negro (portador) y naranja (si)	Combinación valida
S, P, N	El usuario selecciona rojo (si), negro (portador) y naranja (no)	Combinación valida
S, P, P	El usuario selecciona rojo (si), negro (portador) y naranja (portador)	Combinación valida
N, S, S	El usuario selecciona rojo (no), negro (si) y naranja (si)	Combinación valida
N, S, N	El usuario selecciona rojo (no), negro (si) y naranja (no)	Combinación valida
N, S, P	El usuario selecciona rojo (no), negro (si) y naranja (portador)	Combinación valida
N, N, S	El usuario selecciona rojo (no), negro (no) y naranja (si)	Combinación valida (Caso especial se correspondería a sí, no, si)
N, N, N	El usuario selecciona rojo (no), negro (no) y naranja (no)	Combinación invalida de caracteres
N, N, P	El usuario selecciona rojo (no), negro (no) y naranja (portador)	Combinación invalida de caracteres
N, P, S	El usuario selecciona rojo (no), negro (portador) y naranja (si)	Combinación valida (Caso especial se correspondería a si, portador, si)
N, P, N	El usuario selecciona rojo (no), negro (portador) y naranja (no)	Combinación invalida de caracteres
N, P, P	El usuario selecciona rojo (no), negro (portador) y naranja (portador)	Combinación invalida de caracteres

ENTRADAS HEMBRA	CASOS DE PRUEBA	SALIDA ESPERADA
Cabeza Hembra	El usuario no selecciona alguno de los caracteres.	El sistema impide continuar y advierte del error al cliente.
S, S, S	El usuario selecciona rojo(si), negro (si) y naranja (si) a la vez	Combinación invalida de caracteres
S, S, N	El usuario selecciona rojo (si), negro (si) y naranja (no)	Combinación invalida de caracteres
S, S, P	El usuario selecciona rojo (si), negro (si) y naranja (portador)	Combinación invalida de caracteres
S, N, S	El usuario selecciona rojo (si), negro (no) y naranja (si)	Combinación valida
S, N, N	El usuario selecciona rojo (si), negro (no) y naranja (no)	Combinación valida
S, N, P	El usuario selecciona rojo (si), negro (no) y naranja (portador)	Combinación valida
N, S, S	El usuario selecciona rojo (no), negro (si) y naranja (si)	Combinación valida
N, S, N	El usuario selecciona rojo (no), negro (si) y naranja (no)	Combinación valida
N, S, P	El usuario selecciona rojo (no), negro (si) y naranja (portador)	Combinación valida
N, N, S	El usuario selecciona rojo (no), negro (no) y naranja (si)	Combinación valida
N, N, N	El usuario selecciona rojo (no), negro (no) y naranja (no)	Combinación invalida de caracteres
N, N, P	El usuario selecciona rojo (no), negro (no) y naranja (portador)	Combinación invalida de caracteres

REGLAS:

Para el carácter de la cabeza sólo se pueden chequear si para el carácter rojo o negro.

Se debe chequear al menos un carácter como si para el negro o rojo.

Existe el caso especial en que se seleccione rojo y negro no o negro portador pero se seleccione si para el carácter naranja en cuyo caso sería equivalente a chequear si para el carácter rojo.

La hembra solo puede ser portadora del gen de cabeza naranja para el resto de caracteres el carácter será puro.

PECHO

Leyenda: S – Chequeado Si

Orden: Rojo Negro Naranja

N – Chequeado No
P – Chequeado Portador

Morado Lila Blanco
Verde Amarillo Azul

ENTRADAS UNISEX	CASOS DE PRUEBA	SALIDA ESPERADA
Cabeza	El usuario no selecciona alguno de los caracteres.	El sistema impide continuar y advierte del error al cliente.
S, S, S	El usuario selecciona morado(si), lila (si) y blanco (si) a la vez	Combinación invalida de caracteres
S, S, N	El usuario selecciona morado (si), lila (si) y blanco (no)	Combinación invalida de caracteres
S, S, P	El usuario selecciona morado (si), lila (si) y blanco (portador)	Combinación invalida de caracteres
S, N, S	El usuario selecciona morado (si), lila (no) y blanco (si)	Combinación valida
S, N, N	El usuario selecciona morado (si), lila (no) y blanco (no)	Combinación valida
S, N, P	El usuario selecciona morado (si), lila (no) y blanco (portador)	Combinación valida
S, P, S	El usuario selecciona morado (si), lila (portador) y blanco (si)	Combinación invalida de caracteres
S, P, N	El usuario selecciona morado (si), lila (portador) y blanco (no)	Combinación valida
S, P, P	El usuario selecciona morado (si), lila (portador) y blanco (portador)	Combinación invalida de caracteres
N, S, S	El usuario selecciona morado (no), lila (si) y blanco (si)	Combinación invalida de caracteres
N, S, N	El usuario selecciona morado (no), lila (si) y blanco (no)	Combinación valida
N, S, P	El usuario selecciona morado (no), lila (si) y blanco (portador)	Combinación valida
N, N, S	El usuario selecciona morado (no), lila (no) y blanco (si)	Combinación invalida de caracteres
N, N, N	El usuario selecciona morado (no), lila (no) y blanco (no)	Combinación invalida de caracteres
N, N, P	El usuario selecciona morado (no), lila (no) y blanco (portador)	Combinación invalida de caracteres
N, P, S	El usuario selecciona morado (no), lila (portador) y blanco (si)	Combinación invalida de caracteres
N, P, N	El usuario selecciona morado (no), lila (portador) y blanco (no)	Combinación invalida de caracteres
N, P, P	El usuario selecciona morado (no), lila (portador) y blanco (portador)	Combinación invalida de caracteres

REGLAS:

El gen para el carácter del pecho no va ligado al sexo por tanto los casos de prueba son los mismos tanto para el macho como para la hembra.
Solo se podrán seleccionar un carácter como dominante y otro como portador, el resto deberán de estar chequeados a no.

CUERPO

Leyenda: S – Chequeado Si
N – Chequeado No
P – Chequeado Portador
SF – Chequeado SF (simple factor)

Orden: Rojo Negro Naranja
Morado Lila Blanco
Verde Amarillo Azul

ENTRADAS MACHO	CASOS DE PRUEBA	SALIDA ESPERADA
Cuerpo Macho	El usuario no selecciona alguno de los caracteres.	El sistema impide continuar y advierte del error al cliente.
S, S, S	El usuario selecciona verde(si), amarillo (si) y azul (si) a la vez	Combinación invalida de caracteres
S, S, N	El usuario selecciona verde (si), amarillo (si) y azul (no)	Combinación invalida de caracteres
S, S, P	El usuario selecciona verde (si), amarillo (si) y azul (portador)	Combinación invalida de caracteres
S, N, S	El usuario selecciona verde (si), amarillo (no) y azul (si)	Combinación valida
S, N, N	El usuario selecciona verde (si), amarillo (no) y azul (no)	Combinación valida
S, N, P	El usuario selecciona verde (si), amarillo (no) y azul (portador)	Combinación valida
S, SF, S	El usuario selecciona verde (si), amarillo	Combinación invalida de caracteres

	(portador)y azul (si)	
S, SF, N	El usuario selecciona verde (si), amarillo (portador)y azul (no)	Combinación invalida de caracteres
S, SF, P	El usuario selecciona verde (si), amarillo (portador) y azul (portador)	Combinación invalida de caracteres
N, S, S	El usuario selecciona verde (no), amarillo (si) y azul (si)	Combinación valida
N, S, N	El usuario selecciona verde (no), amarillo (si) y azul (no)	Combinación valida
N, S, P	El usuario selecciona verde (no), amarillo (si) y azul (portador)	Combinación valida
N, N, S	El usuario selecciona verde (no), amarillo (no) y azul (si)	Combinación invalida de caracteres
N, N, N	El usuario selecciona verde (no), amarillo (no) y azul (no)	Combinación invalida de caracteres
N, N, P	El usuario selecciona verde (no), amarillo (no) y azul (portador)	Combinación invalida de caracteres
N, SF, S	El usuario selecciona verde (no), amarillo (portador) y azul (si)	Combinación invalida de caracteres
N, SF, N	El usuario selecciona verde (no), amarillo (portador) y azul (no)	Combinación invalida de caracteres
N, SF, P	El usuario selecciona verde (no), amarillo (portador) y azul (portador)	Combinación invalida de caracteres
SF, S, S	El usuario selecciona verde (SF), amarillo (si) y azul (si)	Combinación invalida de caracteres
SF, S, N	El usuario selecciona verde (SF), amarillo (si) y azul (no)	Combinación invalida de caracteres
SF, S, P	El usuario selecciona verde (SF), amarillo (si) y azul (portador)	Combinación invalida de caracteres
SF, N, S	El usuario selecciona verde (SF), amarillo (no) y azul (si)	Combinación invalida de caracteres
SF, N, N	El usuario selecciona verde (SF), amarillo (no) y azul (no)	Combinación invalida de caracteres
SF, N, P	El usuario selecciona verde (SF), amarillo (no) y azul (portador)	Combinación invalida de caracteres
SF, SF, S	El usuario selecciona verde (SF), amarillo (portador) y azul (si)	Combinación valida
SF, SF, N	El usuario selecciona verde (SF), amarillo (portador) y azul (no)	Combinación valida
SF, SF, P	El usuario selecciona verde (SF), amarillo (portador) y azul (portador)	Combinación valida

ENTRADAS HEMBRA	CASOS DE PRUEBA	SALIDA ESPERADA
Cuerpo Hembra	El usuario no selecciona alguno de los caracteres.	El sistema impide continuar y advierte del error al cliente.
S, S, S	El usuario selecciona verde(si), amarillo (si) y azul (si) a la vez	Combinación invalida de caracteres
S, S, N	El usuario selecciona verde (si), amarillo (si) y azul (no)	Combinación invalida de caracteres
S, S, P	El usuario selecciona verde (si), amarillo (si) y azul (portador)	Combinación invalida de caracteres
S, N, S	El usuario selecciona verde (si), amarillo (no) y azul (si)	Combinación valida
S, N, N	El usuario selecciona verde (si), amarillo (no) y azul (no)	Combinación valida
S, N, P	El usuario selecciona verde (si), amarillo (no) y azul (portador)	Combinación valida
N, S, S	El usuario selecciona verde (no), amarillo (si) y azul (si)	Combinación valida
N, S, N	El usuario selecciona verde (no), amarillo (si) y azul (no)	Combinación valida
N, S, P	El usuario selecciona verde (no), amarillo (si) y azul (portador)	Combinación valida
N, N, S	El usuario selecciona verde (no), amarillo (no) y azul (si)	Combinación invalida de caracteres
N, N, N	El usuario selecciona verde (no), amarillo (no) y azul (no)	Combinación invalida de caracteres
N, N, P	El usuario selecciona verde (no), amarillo (no) y azul (portador)	Combinación invalida de caracteres

REGLAS:

Para el carácter del cuerpo sólo se pueden chequear si para el carácter verde o amarillo.

Se debe chequear al menos un carácter como si para el amarillo o verde o en su defecto verde sf y amarillo sf.

La hembra solo puede ser portadora del gen de cabeza azul para el resto de caracteres el carácter será puro.

2.8. Gestión de clientes

Pantalla en la que el administrador podrá hacer consultas sobre clientes.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Código Cliente	1. Caracteres alfanuméricos 2. XX0001 <= código <=XX9999	3. Caracteres extraños. 4. XX0001 > código 5. XX9999< código 6. Dígitos negativos
Primer Apellido	7. Campo de texto alfabético. 8. 0<=apellido <=20	9. Caracteres extraños o numéricos. 10. 0> apellido 11. 20< apellido e
Segundo Apellido	12. Campo de texto alfabético. 13. 0<= apellido <=20	14. Caracteres extraños o numéricos. 15. 0> apellido 16. 20< apellido
DNI	17. Campo de texto alfanumérico 18. Tamaño DNI = 9 19. Formato xxxxxxxxA	20. Caracteres extraños 21. Tamaño DNI != 9 22. No cumple el formato
Estado	23. Selecciona una opción	24. Campo vacío
Fecha Inicio	25. Los caracteres permitidos son dígitos y “/”. 26. Tamaño fecha = 10 27. Fecha valida. 28. Formato dd/mm/aaaa	29. Caracteres extraños o alfabéticos. 30. Tamaño fecha != 10 31. Fecha incorrecta. Ejemplo 31/02/2008 32. Formato incorrecto
Fecha Fin	33. Los caracteres permitidos son dígitos y “/”. 34. Tamaño fecha = 10 35. Fecha valida. 36. Formato dd/mm/aaaa	37. Caracteres extraños o alfabéticos. 38. Tamaño fecha != 10 39. Fecha incorrecta. Ejemplo 31/02/2008 40. Formato incorrecto

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Código cliente	El cliente no existe.	El sistema informa al usuario.
Código cliente	El cliente existe en la base de datos.	El sistema cargara el nombre del cliente en el campo adyacente
Código cliente	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Primer Apellido	El usuario introduce caracteres validos	El sistema realiza la búsqueda normalmente.
Primer Apellido	El usuario introduce caracteres extraños o dígitos	El sistema informa al usuario del error.
Primer Apellido	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Segundo Apellido	El usuario introduce caracteres validos	El sistema realiza la búsqueda normalmente.
Segundo Apellido	El usuario introduce caracteres extraños o dígitos	El sistema informa al usuario del error.
Segundo Apellido	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
DNI	El usuario introduce el DNI en un formato valido	El sistema realiza la búsqueda normalmente.
DNI	El usuario introduce caracteres extraños o un formato incorrecto	El sistema informa al usuario del error.
DNI	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Estado Pedido	El usuario selecciona una opción	El sistema realiza la búsqueda normalmente

Estado Pedido	El usuario no selecciona ninguna opción.	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Fechas	El usuario introduce las fechas validas y en un formato valido	El sistema realiza la búsqueda normalmente.
Fechas	El usuario introduce caracteres extraños o un formato incorrecto o una fecha no valida	El sistema informa al usuario del error.
Fechas	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
	El usuario no introduce ningún criterio de búsqueda.	El sistema realiza la búsqueda de todos los pedidos de clientes.

2.9 Gestión de cliente

Pantalla en la que el administrador podrá modificar el estado del pedido de un cliente o darlo de baja.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Estado Pedido	1. Selecciona una opción	2. Campo vacio

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Estado Pedido	El usuario selecciona una opción.	El sistema guarda la información.
	El usuario no selecciona ninguna opción	El sistema impide esta acción

2.10. Gestión de proveedor

Pantalla en la que el administrador podrá hacer consultas sobre proveedores.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Código Proveedor	1. Caracteres alfanuméricos 2. XX0001 <= código <=XX9999	3. Caracteres extraños. 4. XX0001 > código 5. XX9999< código 6. Dígitos negativos
DNI	7. Campo de texto alfanumérico 8. Tamaño DNI = 9 9. Formato xxxxxxxxA	10. Caracteres extraños 11. Tamaño DNI != 9 12. No cumple el formato
Primer Apellido	3. Campo de texto alfabético. 4. 0<=apellido <=20	5. Caracteres extraños o numéricos. 6. 0> apellido 7. 20< apellido e
Segundo Apellido	8. Campo de texto alfabético. 9. 0<= apellido <=20	10. Caracteres extraños o numéricos. 11. 0> apellido 12. 20< apellido

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Código Proveedor	El proveedor no existe.	El sistema informa al usuario.
Código Proveedor	El proveedor existe en la base de datos.	El sistema cargara el nombre del proveedor en el campo adyacente
Código Proveedor	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
DNI	El usuario introduce el DNI en un formato valido	El sistema realiza la búsqueda normalmente.
DNI	El usuario introduce caracteres extraños o un formato incorrecto	El sistema informa al usuario del error.
DNI	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Primer Apellido	El usuario introduce caracteres validos	El sistema realiza la búsqueda normalmente.
Primer Apellido	El usuario introduce caracteres extraños o dígitos	El sistema informa al usuario del error.
Primer Apellido	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.

Segundo Apellido	El usuario introduce caracteres validos	El sistema realiza la búsqueda normalmente.
Segundo Apellido	El usuario introduce caracteres extraños o dígitos	El sistema informa al usuario del error.
Segundo Apellido	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.

2.11. Alta y modificación de proveedor

Pantalla en la que el usuario podrá introducir los datos de un nuevo proveedor o modificar los ya existentes.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Tipo de proveedor	1. Elige un valor (campo chequeado)	2. No selecciona ningún valor (campo no chequeado)
Nombre	3. Campo de texto alfabético. 4. 1<=Nombre <=20	5. Caracteres extraños o numéricos. 6. 1>Nombre 7. 20<Nombre
Primer Apellido	8. Campo de texto alfabético. 9. 1<=apellido <=20	10. Caracteres extraños o numéricos. 11. 1> apellido 12. 20< apellido e
Segundo Apellido	13. Campo de texto alfabético. 14. 1<= apellido <=20	15. Caracteres extraños o numéricos. 16. 1> apellido 17. 20< apellido
Correo electrónico	18. 1<= correo <=40 19. Formato nombre@dominio	20. 1> correo 21. 40< correo 22. Formato incorrecto
Teléfono	23. Tamaño teléfono = 9 24. Formato correcto (empieza por 6) 25. Caracteres numéricos	26. Tamaño distinto a 9 27. Formato incorrecto 28. Caracteres alfabéticos o extraños
Calle	29. Caracteres alfabéticos 30. 1<= calle <=50	31. Caracteres numéricos o extraños. 32. 1> calle 33. 50 < calle
Numero	55. Caracteres numéricos 54. 1<= numero <= 999	56. Caracteres extraños o alfabéticos 57. 1 > numero 58. 999 < numero
Población	68. Caracteres alfabéticos 69. 1<= población <= 40	70. Caracteres numéricos o extraños. 71. 1 > población 72. 40 < población
Código Postal	73. Caracteres numéricos 74. 00001 <= código <=99999	75. Caracteres extraños o alfabéticos. 76. 00001 > código postal 77. 99999 < código postal
País	78. Selecciona una opción	79. Campo vacio
Provincia	80. Selecciona una opción	81. Campo vacio

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Datos de proveedor	El formulario cumple las clases de equivalencia validas.	El sistema da de alta al proveedor y muestra el modulo de cliente.
	El formulario no cumple las clases de equivalencia validas.	El sistema muestra el error y pide la entrada de valores nuevos.
	Todos los campos obligatorios del formulario están completos	El sistema da de alta al proveedor y muestra el modulo de cliente.
	Campos obligatorios vacios	El sistema muestra el error y pide la entrada de los valores

2.12. Gestión de productos

Pantalla para la búsqueda de productos dados de alta en la aplicación.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Producto (Código)	1. Campo de texto numérico. 2. 0<= producto <=999999	3. Caracteres extraños o alfabéticos. 4. producto < 0 5. producto > 999999
Stock	6. Caracteres numéricos 7. 0 <= stock = 999	8. Caracteres extraños o alfabéticos. 9. producto < 0 10. producto > 999
Tipo	11. Selecciona una opción 12. No selecciona ninguna opción	

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Código Proveedor	El producto no existe.	El sistema informa al usuario.
Código Proveedor	El producto existe en la base de datos.	El sistema cargara el nombre del producto en el campo adyacente
Código Proveedor	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Stock	El usuario introduce caracteres validos	El sistema realiza la búsqueda normalmente.
Stock	El usuario introduce caracteres extraños o alfaba béticos	El sistema informa al usuario del error.
Stock	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Tipo	El usuario selecciona una opción.	El sistema realiza la búsqueda normalmente
	El usuario no selecciona ninguna opción	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.

2.13. Modificación de productos

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Nombre	1. Campo de texto alfabético. 2. 1<=Nombre <=20	3. Caracteres extraños o numéricos. 4. 1>Nombre 5. 20<Nombre
Familia	6. Selecciona una opción	7. Campo Vacio
Fecha Alta	Campo deshabilitado	Campo deshabilitado
IVA	8. Selecciona una opción	9. Campo vacio
Precio Compra	No se modifica	No se modifica
Precio Venta	10. Caracteres numéricos 11. 0.0 <= stock = 9999.99	12. Caracteres extraños o alfabéticos. 13. producto < 0.0 14. producto > 9999.00
Tipo Oferta	15. Selecciona una opción	16. Campo vacio
Stock	17. Caracteres numéricos 18. 0 <= stock = 999	19. Caracteres extraños o alfabéticos. 20. producto < 0 21. producto > 999
Peso	22. Caracteres numéricos 23. 0 <= peso = 80	24. Caracteres extraños o alfabéticos. 25. peso < 0.0 26. peso > 9999.00
Características	27. 1 <= código <=300	28. 1 > características 29. 300 < características

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Datos del producto	El formulario cumple las clases de equivalencia validas.	El sistema modifica el producto.
	El formulario no cumple las clases de equivalencia validas.	El sistema muestra el error y pide la entrada de valores nuevos.
	Todos los campos obligatorios del formulario están completos	El sistema modifica el producto.
	Campos obligatorios vacios	El sistema muestra el error y pide la entrada de los valores

2.14. Pedidos

Pantalla para la búsqueda de productos dados de alta en la aplicación.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Proveedor (Código)	1. Campo de texto numérico. 2. 0<= proveedor <=999999	3. Caracteres extraños o alfabéticos. 4. proveedor < 0 5. proveedor > 999999
Producto (Código)	6. Campo de texto numérico. 7. 0<= producto <=999999	8. Caracteres extraños o alfabéticos. 9. producto < 0 10. producto > 999999
Unidades	11. Caracteres numéricos 12. 0 <= unidades = 999	8. Caracteres extraños o alfabéticos. 9. unidades < 0 10. unidades > 999

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Código Proveedor	El proveedor no existe.	El sistema informa al usuario.
Código Proveedor	El proveedor existe en la base de datos.	El sistema cargara el nombre del proveedor en el campo adyacente
Código Proveedor	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Código Producto	El producto no existe.	El sistema informa al usuario.
Código Producto	El producto existe en la base de datos.	El sistema cargara el nombre del producto en el campo adyacente
Código Producto	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Unidades	El usuario introduce caracteres validos	El sistema realiza la búsqueda normalmente.
Unidades	El usuario introduce caracteres extraños o alfabeticos	El sistema informa al usuario del error.
Unidades	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
	El usuario no introduce ningún criterio de búsqueda.	El sistema realiza la búsqueda de todos los pedidos de clientes.

2.15. Gestión de pedido

Pantalla en la que el administrador podrá acceder a la consulta de pedidos de clientes o proveedores.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Opción pedido	1. Selecciona una opción	2. No seleccionar ninguna opción

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Opción pedido	El usuario selecciona una opción.	El sistema lleva al usuario a la opción deseada, pedidos a clientes o a proveedores.
	El usuario no selecciona ninguna opción	El sistema impide continuar

2.16. Gestión de pedidos a clientes

Pantalla en la que el administrador podrá hacer consultas sobre los pedidos de los clientes.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Código Pedido	1. Caracteres numéricos 2. 1 <= código <=999999	3. Caracteres extraños. 4. 1 > código 5. 999999< código 6. Dígitos negativos
Estado	7. Selecciona una opción	8. Campo vacio
Código Cliente	9. Caracteres numéricos 10. 1 <= código <=999999	11. Caracteres extraños. 12. 1 > código 13. 999999< código 14. Dígitos negativos
Código Producto	15. Caracteres alfanuméricos 16. 1 <= código <=999999	17. Caracteres extraños. 18. 1 > código 19. 999999< código 20. Dígitos negativos

Fecha Inicio	21. Los caracteres permitidos son dígitos y "p". 22. Tamaño fecha = 10 23. Fecha valida. 24. Formato dd/mm/aaaa	25. Caracteres extraños o alfabéticos. 26. Tamaño fecha != 10 27. Fecha incorrecta. Ejemplo 31/02/2008 28. Formato incorrecto
Fecha Fin	29. Los caracteres permitidos son dígitos y "p". 30. Tamaño fecha = 10 31. Fecha valida. 32. Formato dd/mm/aaaa	33. Caracteres extraños o alfabéticos. 34. Tamaño fecha != 10 35. Fecha incorrecta. Ejemplo 31/02/2008 36. Formato incorrecto

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Código pedido	El pedido no existe.	El sistema informa al usuario.
Código pedido	El pedido existe en la base de datos.	El sistema cargara el código del pedido en el campo de texto correspondiente
Código pedido	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Estado Pedido	El usuario selecciona una opción	El sistema realiza la búsqueda normalmente
Estado Pedido	El usuario no selecciona ninguna opción.	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Código cliente	El cliente no existe.	El sistema informa al usuario.
Código cliente	El cliente existe en la base de datos.	El sistema cargara el nombre del cliente en el campo de texto adyacente
Código cliente	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Código producto	El producto no existe.	El sistema informa al usuario.
Código producto	El producto existe en la base de datos.	El sistema cargara el nombre del producto en el campo de texto adyacente
Código producto	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Fechas	El usuario introduce las fechas validas y en un formato valido	El sistema realiza la búsqueda normalmente.
Fechas	El usuario introduce caracteres extraños o un formato incorrecto o una fecha no valida	El sistema informa al usuario del error.
Fechas	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
	El usuario no introduce ningún criterio de búsqueda.	El sistema realiza la búsqueda de todos los pedidos de clientes.

2.17. Gestión de pedidos a proveedores

Pantalla en la que el administrador podrá hacer consultas sobre los pedidos de los proveedores.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Código Pedido	1. Caracteres alfanuméricos 2. 1 <= código <=999999	3. Caracteres extraños. 4. 1 > código 5. 999999< código 6. Dígitos negativos
Código Proveedor	9. Caracteres numéricos 10. 1 <= código <=999999	11. Caracteres extraños. 12. 1 > código 13. 999999< código

Código Producto	15. Caracteres numéricos 16. 1 <= código <=999999	14. Dígitos negativos 17. Caracteres extraños. 18. 1 > código 19. 999999< código 20. Dígitos negativos
Fecha Inicio	21. Los caracteres permitidos son dígitos y “/” 22. Tamaño fecha = 10 23. Fecha valida. 24. Formato dd/mm/aaaa	25. Caracteres extraños o alfabéticos. 26. Tamaño fecha != 10 27. Fecha incorrecta. Ejemplo 31/02/2008 28. Formato incorrecto
Fecha Fin	29. Los caracteres permitidos son dígitos y “/” 30. Tamaño fecha = 10 31. Fecha valida. 32. Formato dd/mm/aaaa	33. Caracteres extraños o alfabéticos. 34. Tamaño fecha != 10 35. Fecha incorrecta. Ejemplo 31/02/2008 36. Formato incorrecto

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Código pedido	El pedido no existe.	El sistema informa al usuario.
Código pedido	El pedido existe en la base de datos.	El sistema cargara el código del pedido en el campo de texto correspondiente
Código pedido	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Código cliente	El cliente no existe.	El sistema informa al usuario.
Código cliente	El cliente existe en la base de datos.	El sistema cargara el nombre del cliente en el campo de texto adyacente
Código cliente	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Código producto	El producto no existe.	El sistema informa al usuario.
Código producto	El producto existe en la base de datos.	El sistema cargara el nombre del producto en el campo de texto adyacente
Código producto	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Fechas	El usuario introduce las fechas validas y en un formato valido	El sistema realiza la búsqueda normalmente.
Fechas	El usuario introduce caracteres extraños o un formato incorrecto o una fecha no valida	El sistema informa al usuario del error.
Fechas	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
	El usuario no introduce ningún criterio de búsqueda.	El sistema realiza la búsqueda de todos los pedidos de clientes.

2.18. Gestión de centros

Pantalla en la que el administrador podrá hacer consultas sobre los centros de la empresa

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Código Tienda	1. Caracteres numéricos 2. 1 <= código <=999999	3. Caracteres extraños. 4. 1 > código 5. 999999< código 6. Dígitos negativos
País	7. Selecciona una opción 8. Campo vacio	
Provincia	9. Selecciona una opción 10. Campo vacio	

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Código Tienda	El centro no existe.	El sistema informa al usuario.
Código Tienda	El centro existe en la base de datos.	El sistema cargara el nombre del centro en el campo de texto adyacente
Código Tienda	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
País	El usuario selecciona una opción	El sistema realiza la búsqueda normalmente
Provincia	El usuario no selecciona ninguna opción.	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.

2.19. Alta y modificación de centros

Pantalla en la que el usuario podrá añadir y modificar los centros.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Nombre	1. Campo de texto alfabético. 2. 1<=Nombre <=20	3. Caracteres extraños o numéricos. 4. 1>Nombre 5. 20<Nombre
Calle	6. Caracteres alfabéticos 7. 1<= calle <=50	8. Caracteres numéricos o extraños. 9. 1> calle 10. 50 < calle
Numero	11. Caracteres numéricos 12. 1<= numero <= 999	13. Caracteres extraños o alfabéticos 14. 1 > numero 15. 999 < numero
Bloque	16. Caracteres numéricos 17. 1<= numero <= 999	18. Caracteres extraños o alfabéticos 19. 1 > numero 20. 999 < numero
Población	21. Caracteres alfabéticos 22. 1<= población <= 40	23. Caracteres numéricos o extraños. 24. 1 > población 25. 40 < población
Código Postal	26. Caracteres numéricos 27. 00001 <= código <=99999	28. Caracteres extraños o alfabéticos. 29. 00001 > código postal 30. 99999 < código postal
País	31. Selecciona una opción	32. Campo vacío
Provincia	33. Selecciona una opción	34. Campo vacío
Correo electrónico	35. 1<= correo <=40 36. Formato nombre@dominio	37. 1> correo 38. 40< correo 39. Formato incorrecto
Teléfono	40. Tamaño teléfono = 9 41. Formato correcto (empieza por 6) 42. Caracteres numéricos	43. Tamaño distinto a 9 44. Formato incorrecto 45. Caracteres alfabéticos o extraños
Calle	46. Caracteres alfabéticos 47. 1<= calle <=50	48. Caracteres numéricos o extraños. 49. 1> calle 50. 50 < calle
Numero	51. Caracteres numéricos 52. 1<= numero <= 999	53. Caracteres extraños o alfabéticos 54. 1 > numero 55. 999 < numero
Login	56. Campo de texto alfanumérico. 57. 1<=login<=10	58. Caracteres extraños. 59. Campo vacío 60. login < 1 61. login > 10
Password	62. Campo de texto alfanumérico. 63. 1<= password <=10	64. Caracteres extraños. 65. Campo vacío 66. password < 1 67. password > 10

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Datos de proveedor	El formulario cumple las clases de equivalencia validas.	El sistema da de alta al usuario y muestra el modulo de cliente.
	El formulario no cumple las clases de equivalencia validas.	El sistema muestra el error y pide la entrada de valores nuevos.

Todas las entradas a excepción del bloque tienen que ser rellenadas, en caso contrario el sistema muestra un mensaje de error.

2.20. Consulta de stock

Pantalla en la que el administrador podrá hacer consultas sobre el stock de los productos del almacén.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Código Tienda	1. Caracteres alfanuméricos 2. 1 <= código <=999999	3. Caracteres extraños. 4. 1 > código 5. 999999< código 6. Dígitos negativos
Código Producto	7. Caracteres numéricos 8. 1 <= código <=999999	9. Caracteres extraños. 10. 1 > código 11. 999999< código 12. Dígitos negativos

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Código Centro	El centro no existe.	El sistema informa al usuario.
Código Centro	El centro existe en la base de datos.	El sistema cargara el nombre del centro en el campo de texto adyacente
Código Centro	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Código Producto	El producto no existe.	El sistema informa al usuario.
Código Producto	El producto existe en la base de datos.	El sistema cargara el nombre del producto en el campo de texto adyacente
Código Producto	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.

2.21. Nueva oferta

Pantalla en la que el administrador podrá añadir productos como ofertados.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Código Producto	1. Caracteres alfanuméricos 2. 1 <= código <=999999	3. Caracteres extraños. 4. 1 > código 5. 999999< código 6. Dígitos negativos
Características	7. 1 <= código <=300	8. 1 > características 9. 300 < características
Precio Unitario	10. Caracteres numéricos 11. 0 <= precio <= 999999	12. Caracteres alfabéticos o extraños. 13. 0 > precio 14. 999999 < precio

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Código Producto	El producto no existe.	El sistema informa al usuario.
Código Producto	El producto existe en la base de datos.	El sistema cargara el nombre del producto en el campo de texto adyacente
Código Producto	El usuario no introduce los datos	El sistema muestra un mensaje con el error.
Características	El usuario introduce datos	
Características	El usuario no introduce datos	El sistema muestra un mensaje con el error.
Precio Unitario	El usuario introduce un precio que cumple las clases validas	
Precio Unitario	El usuario introduce caracteres alfabéticos o extraños	El sistema muestra un mensaje con el error.

Precio Unitario	El usuario no introduce los datos	El sistema muestra un mensaje con el error.
------------------------	-----------------------------------	---

2.22. Modificación oferta

Pantalla en la que el administrador podrá modificar productos ofertados.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Nombre	1. Caracteres alfanuméricos 2. 1 <= nombre <= 20	3. Caracteres extraños 4. 1 > nombre 5. 20 < nombre
Precio Unitario	6. Caracteres numéricos 7. 0 <= precio <= 999999	8. Caracteres alfabéticos o extraños. 9. 0 > precio 10. 999999 < precio

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Nombre	El usuario introduce datos	El sistema impide modificar este campo
Nombre	El usuario no introduce datos	El sistema impide modificar este campo
Precio Unitario	El usuario introduce un precio que cumple las clases validas	
Precio Unitario	El usuario introduce caracteres alfabéticos o extraños	El sistema muestra un mensaje con el error.
Precio Unitario	El usuario no introduce los datos	El sistema muestra un mensaje con el error.

2.23. Visualización de gráficos

Pantalla en la que el administrador podrá hacer consultas sobre las ventas del almacén o de las tiendas.

ENTRADAS	CLASES VALIDAS	CLASES INVALIDAS
Familia	1. Caracteres alfanuméricos 2. 1 <= código <=999999	3. Caracteres extraños. 4. 1 > código 5. 999999< código 6. Dígitos negativos
Código Centro	7. Selecciona una opción	8. Campo vacío
Fecha Inicio	9. Los caracteres permitidos son dígitos y "/". 10. Tamaño fecha = 10 11. Fecha valida. 12. Formato dd/mm/aaaa	13. Caracteres extraños o alfabéticos. 14. Tamaño fecha != 10 15. Fecha incorrecta. Ejemplo 31/02/2008 16. Formato incorrecto
Fecha Fin	17. Los caracteres permitidos son dígitos y "/". 18. Tamaño fecha = 10 19. Fecha valida. 20. Formato dd/mm/aaaa	21. Caracteres extraños o alfabéticos. 22. Tamaño fecha != 10 23. Fecha incorrecta. Ejemplo 31/02/2008 24. Formato incorrecto

ENTRADAS	CASOS DE PRUEBA	SALIDA ESPERADA
Familia	No selecciona ninguna opción	
Familia	Selecciona una opción	El sistema realiza la búsqueda por el resto de parámetros de búsqueda
Código Centro	El usuario no introduce los datos. Visualizar Productos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.
Código Centro	El usuario no introduce los datos. Visualizar Productos por tienda	El sistema solicita la introducción de un centro.
Código Centro	El centro no existe.	El sistema informa al usuario.
Código Centro	El centro existe en la base de datos.	El sistema cargara el nombre del centro en el campo de texto adyacente
Fechas	El usuario introduce las fechas validas y en un formato valido	El sistema realiza la búsqueda normalmente.

Fechas	El usuario introduce caracteres extraños o un formato incorrecto o una fecha no valida	El sistema informa al usuario del error.
Fechas	El usuario no introduce los datos	El sistema realiza la búsqueda sobre el resto de criterios de búsqueda.

3. Ejecución de las pruebas de integración

El objetivo de las pruebas de integración es verificar el correcto ensamblaje entre los distintos componentes una vez que han sido probados unitariamente, con el fin de comprobar que interactúan correctamente a través de sus interfaces, tanto internas como externas, cubren la funcionalidad establecida y se ajustan a los requisitos no funcionales especificados en las verificaciones correspondientes.

Hemos establecido el tipo de integración incremental para el diseño y desarrollo de las pruebas de integración. Para ello se han diferenciado los distintos módulos en un orden jerárquico donde el siguiente componente en la jerarquía se prueba con el conjunto de componentes ya probados. De este modo los posibles errores que surjan serán probablemente debido al último componente añadido.

La estrategia de integración elegida es “**de arriba abajo (top down)**” donde el primer componente que se prueba es el más alto de la jerarquía para después ir añadiendo los componentes de niveles inferiores.

El nivel más alto de la jerarquía en nuestro caso será el alta de usuario. Esta elección es debida a que para comenzar a ejecutar la aplicación deberemos tener algún usuario. Posteriormente se han añadido los módulos según estén disponibles a partir del módulo anterior en la jerarquía.

Los componentes contacto y centro están disponibles desde el inicio del proceso. Para el caso de Centros, estarán disponibles en el momento en que este dado de alta un usuario con perfil administrador. Por tanto están considerados como módulos independientes del resto.

La ventaja de este método es que las clases de la parte alta de la jerarquía que son las que van a ser más usadas, van a ser también las más probadas ya que se continúa probando en la siguiente etapa.

3.1 Diseño pruebas de integración

DISEÑO PRUEBAS DE INTEGRACIÓN

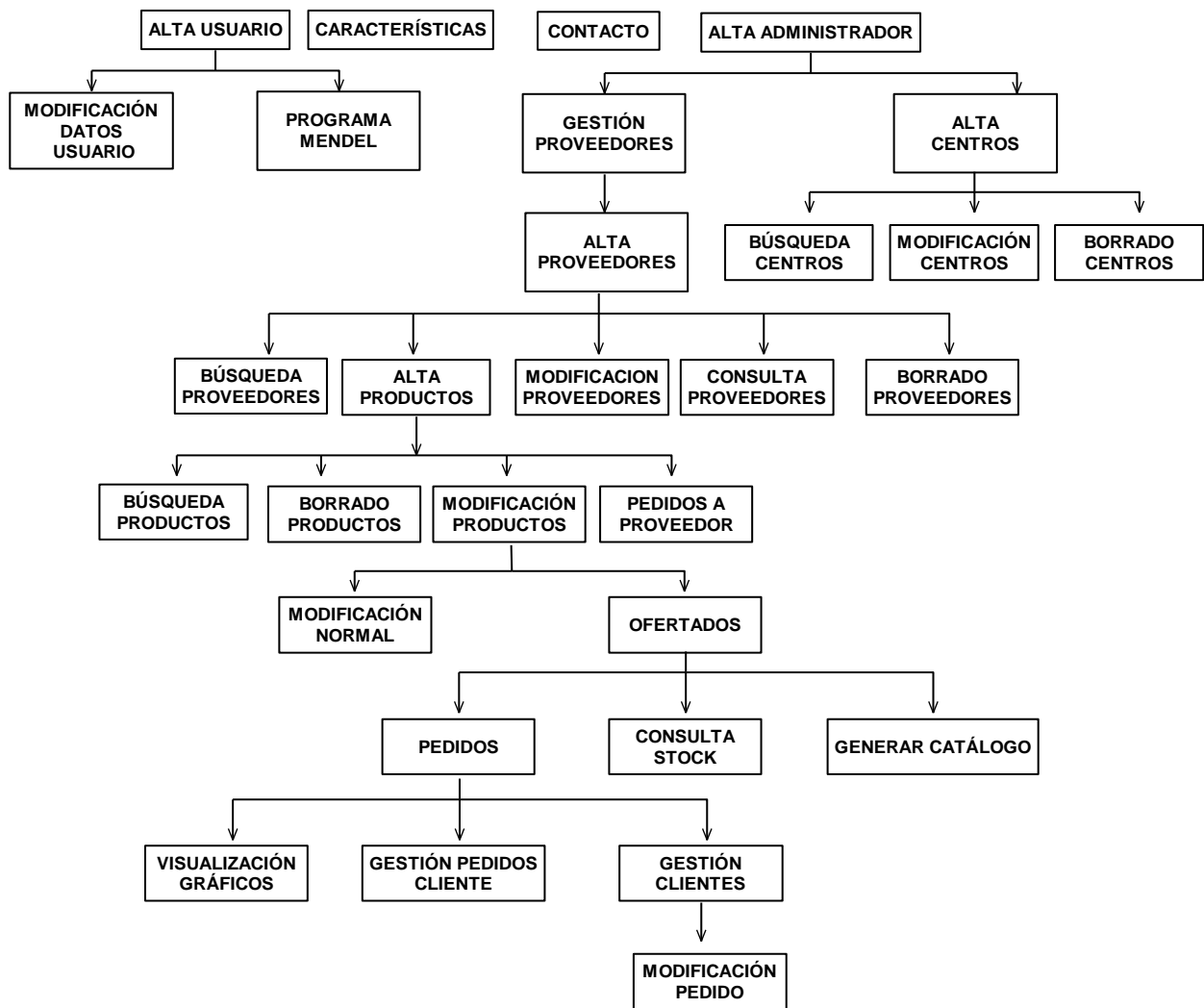


Figura 1. Diseño de pruebas de integración

3.2. Definición de las pruebas

3.2.1. Alta usuario

Las validaciones necesarias para el alta o modificación de nuevos usuarios fueron revisadas durante la parte de las pruebas unitaria.

Modulo prueba	Caso de Prueba	Resultado Esperado	Resultado
Alta Usuario	Login utilizado por otro cliente	El sistema informa del error.	Correcto
	Guardar	El sistema da de alta al cliente y almacena sus datos en la base de datos y muestra un mensaje con el resultado de la operación. Posteriormente el sistema devuelve al cliente a la pantalla de inicio de sesión.	Correcto
	Casos especiales	El sistema permite la recuperación de clientes antiguos (clientes que se habían dado de baja). El sistema reconoce un cliente antiguo si coinciden su nombre, dni, login y password.	Correcto

3.2.2. Modificación datos usuario

Las validaciones necesarias para el alta o modificación de nuevos usuarios fueron revisadas durante la parte de las pruebas unitaria.

Modulo Prueba	Caso de Prueba	Resultado Esperado	Resultado
Modificacion usuario	Login utilizado por otro cliente	El sistema informa del error.	Correcto
	Guardar	El sistema actualiza los datos modificados por el cliente y muestra un mensaje con el resultado de la operación	Correcto
	Comprobar disponibilidad	El sistema comprueba la disponibilidad del login elegido por el cliente y muestra un mensaje con el resultado de la operación. El sistema mostrara como disponible cualquier login no usado por otros clientes o el del propio cliente.	Correcto
	Comprobar disponibilidad (casos especiales)	El sistema mostrara como disponible cualquier login no usado por otros clientes o el del propio cliente. Dado que el sistema permite la recuperación de clientes eliminados tampoco estarán disponibles los login de clientes antiguos.	Correcto
	Darse de baja	El sistema lo elimina del registro de clientes y muestra un mensaje con el resultado de la operación.	Correcto
	El usuario se da de baja y se vuelve a dar de alta	El sistema permite la recuperación de ese cliente.	Correcto

3.2.3. Programa Mendel

Las pruebas necesarias para la validación de entrada de datos en el programa Mendel ya han sido especificadas en las pruebas unitarias.

Para comprobar el correcto funcionamiento del programa se han realizado varias pruebas completas, aquí vamos a especificar una de ellas.

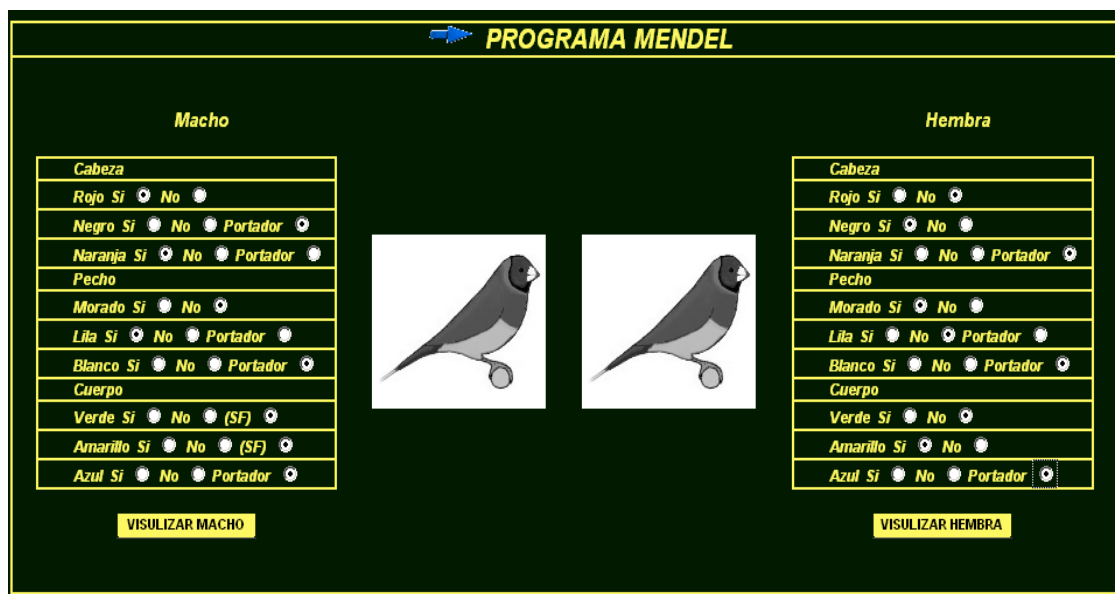


Figura 2. Programa Mendel

En el ejemplo se ha seleccionado para el ejemplar macho los colores rojo, naranja y portador de negro para el carácter de la cabeza, lila y portador de blanco para el carácter del pecho y verde simple factor, amarillo simple factor y portador de azul para el carácter del cuerpo. En el caso de la hembra, cabeza negra y portador de cabeza naranja. Pecho morado y portador de cabeza blanca. Cuerpo amarillo y portador de cuerpo azul.

Análisis Fenotipo macho:

El gen rojo de la cabeza es dominante sobre el negro por tanto la cabeza sería roja, pero debido a que tiene los dos genes responsables de cabeza naranja el color para la cabeza será naranja. El gen lila es dominante sobre el blanco por tanto el color del pecho será lila. El gen verde para el cuerpo es parcialmente dominante sobre el gen amarillo, pero en esto solo se da en el caso de que sean de pecho morado. En nuestro caso tenemos un macho de pecho lila por lo que el color del cuerpo será amarillo. Por otro lado también es portador del color azul y el gen azul afecta al color de la cabeza aclarándola por lo que la cabeza del pájaro será de una tonalidad entre naranja y salmón debido a que solo es portador de azul.

Análisis Fenotipo hembra:

La hembra será de cabeza negra y portadora de cabeza naranja. En cuanto al carácter del pecho va a ser morado y portador de pecho blanco. La espalda de la hembra es amarilla y portadora de azul, dado que el gen amarillo suprime la expresión negra, la cabeza será blanca.

Por tanto la composición de los caracteres cabeza, pecho y cuerpo de los pájaros será:

Macho – Naranja/Salmon, lila, amarillo.

Hembra – Blanco, morado, amarillo.

Figura 3. Análisis genotipos

Análisis Genotipo Macho

Cabeza – ZRZr oo
 Pecho – ll wb
 Cuerpo – G+A bl bl+

Análisis: Genotipo Hembra

Cabeza – ZrW o+o
 Pecho – wb+wb
 Cuerpo – AW bl+bl+

Cálculo de probabilidades para el carácter de la cabeza:

MACHO ZR Zr oo

HEMBRA ZrW o+o

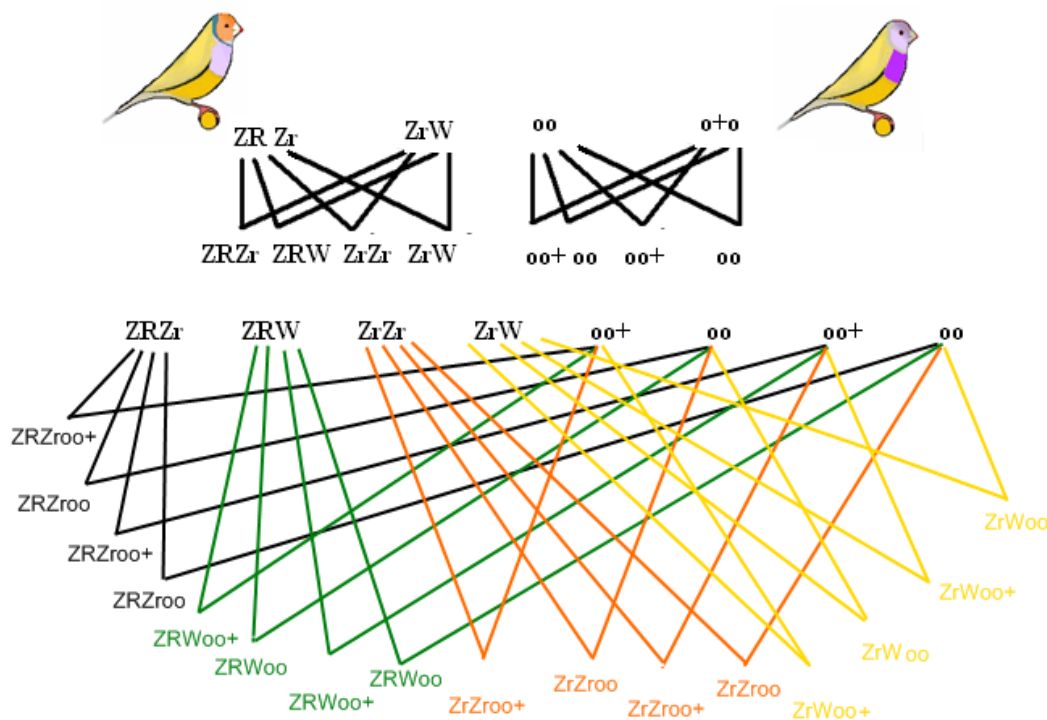


Figura 4. Combinaciones genotipos cabeza

Cada combinación tiene 6,25% de probabilidades de aparición, el siguiente paso es agrupar aquellas que se repiten y calcular sus probabilidades de aparición.

- ZRZroo + – 12,5% Macho de cabeza roja portador de cabeza negra y naranja.
- ZRZroo – 12,5% Macho de cabeza naranja portador de cabeza negra.
- ZRWoo + – 12,5% Hembra de cabeza roja portador de cabeza naranja.
- ZRWoo – 12,5% Hembra de cabeza naranja.
- ZrZroo + – 12,5% Macho de cabeza negra portador de naranja.
- ZrZroo – 12,5% Macho de cabeza negra y naranja (EPA).
- ZrWoo + – 12,5% Hembra de cabeza negra portador de cabeza naranja.
- ZrWoo – 12,5% Hembra de cabeza negra y naranja (EPA).

El gen para la cabeza naranja es autosómico (no va ligado al sexo) y solo se manifiesta en los pájaros de cabeza roja cuando se da en la forma oo.

Para los pájaros de cabeza negra que tengan además los genes oo para la cabeza naranja serán un tipo de pájaros de cabeza negra especiales a los que se denominan EPA.

Fenotipo Macho : naranja/lila/amarillo		Fenotipo Hembra : blanco/morado/amarillo	
Genotipo Macho : ZRZroo/lwbl/ZG+ZABl+bl		Genotipo Hembra : ZrWo+o/wb+wb/ZAWbl+bl	
Porcentaje	Sexo	Cabeza	Genotipo
12.5%	macho	rojo	Cabeza - ZRZro+o
12.5%	macho	naranja	Cabeza - ZRZroo
12.5%	hembra	rojo	Cabeza - ZRWo+o
12.5%	hembra	naranja	Cabeza - ZRWoo
12.5%	macho	negro	Cabeza - ZrZro+o
12.5%	macho	negro EPA	Cabeza - ZrZroo
12.5%	hembra	negro	Cabeza - ZrWo+o
12.5%	hembra	negro EPA	Cabeza - ZrWoo

Figura 5. Resultados cabeza

Cálculo de probabilidades para el carácter del pecho

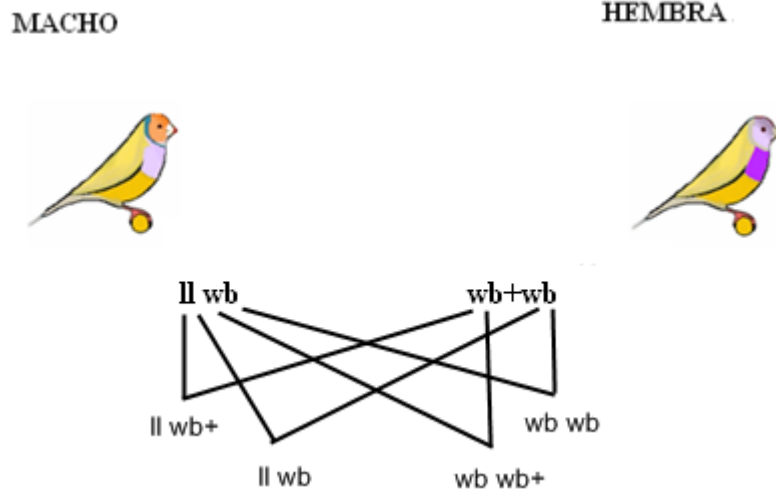


Figura 6. Combinaciones genotipo pecho

No se repite ningún resultado por lo que cada combinación tiene un 25% de probabilidades de aparición.

Los genes para el carácter del pecho son autósomicos por los que las probabilidades de aparición son las mismas tanto para el macho como para la hembra

- $ll\ wb+$ - 25% Unisex pecho morado portador de lila.
- $ll\ wb$ - 25% Unisex pecho lila portador de blanco.
- $wb\ wb+$ - 25% Unisex pecho morado portador de blanco.
- $wb\ wb$ - 25% Unisex pecho blanco.

Fenotipo Macho : naranja-salmon/lila/amarillo		Fenotipo Hembra : blanco/morado/amarillo	
Genotipo Macho : $ZRZroo/llwb/ZG+ZAbI+bl$		Genotipo Hembra : $ZrWo+o/wb+wb/ZAWbl+bl$	
Porcentaje	Sexo	Pecho	Genotipo
25%	unisex	morado	Pecho - $wb+ll$
25%	unisex	lila	Pecho - $llwb$
25%	unisex	morado	Pecho - $wb+wb$
25%	unisex	blanco	Pecho - $wbwb$

Figura 7. Resultados pecho

Cálculo de probabilidades para el carácter del cuerpo

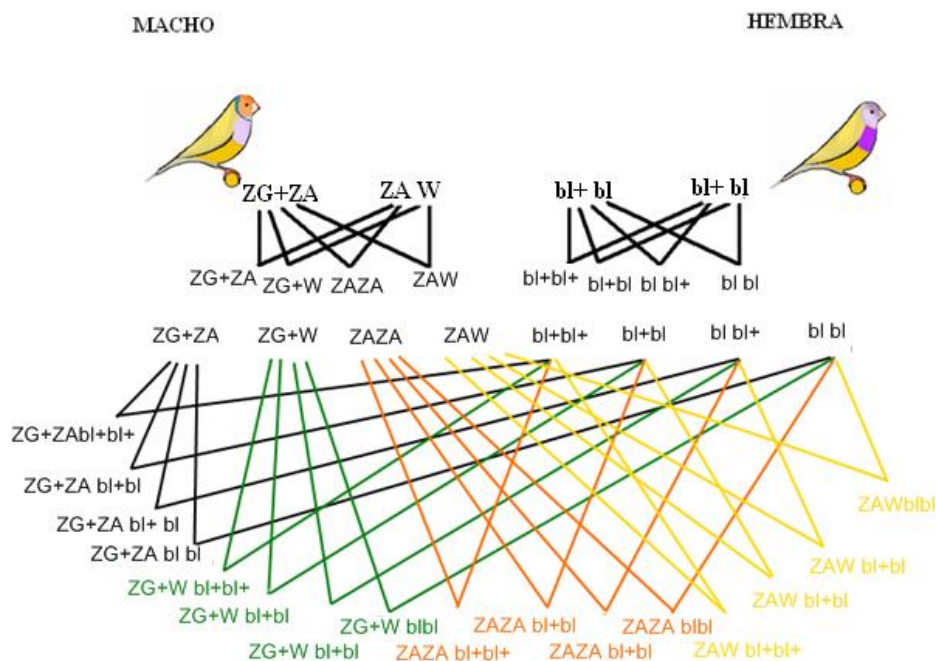


Figura 8. Combinaciones genotipo cuerpo

Cada combinación tiene 6,25% de probabilidades de aparición, el siguiente paso es agrupar aquellas que se repiten y calcular sus probabilidades de aparición.

La única combinación que se repite es bl+bl que tiene el doble de probabilidades de aparición que el resto.

- ZG+ZA bl+bl+ – 6,25% Macho verde-amarillo
- ZG+ZA bl+bl – 12,5 % Macho verde – amarillo azul simple factor
- ZG+ZA bl bl – 6,25% Macho verde – amarillo azul doble factor
- ZG+W bi+bl+ – 6,25% Hembra verde
- ZG+W bi+bl – 12,5% Hembra verde azul simple factor
- ZG+W bl bl+ – 6,25% Hembra verde azul doble factor
- ZAZA bi+bl+ – 6,25% Macho amarillo
- ZAZA bi+bl – 12,5% Macho amarillo azul simple factor
- ZAZA bl bl+ – 6,25% Macho amarillo azul doble factor
- ZAZW bi+bl+ – 6,25% Hembra amarilla
- ZAZW bi+bl – 12,5% Hembra amarilla azul simple factor
- ZAZW bl bl+ – 6,25% Hembra amarilla azul doble factor

Dado que el color del cuerpo afecta y se ve afectado por los caracteres del pecho y la cabeza los colores expuestos son orientativos.

Fenotipo Macho : naranja-salmon/lila/amarillo		Fenotipo Hembra : blanco/morado/amarillo	
Genotipo Macho : ZRZroo/llwb/ZG+ZAbi+bl		Genotipo Hembra : ZrWo+o/wb+wb/ZAWbi+bl	
Porcentaje	Sexo	Cuerpo	Genotipo
6.25%	macho	verdeAmarillo	Cuerpo - ZG+ZAbi+bl+
12.5%	macho	verdeAmarilloAzu(SF)	Cuerpo - ZG+ZAbi+bl
6.25%	macho	verdeAmarilloAzu(DF)	Cuerpo - ZG+ZAbibl
6.25%	hembra	verde	Cuerpo - ZG+Wbi+bl+
12.5%	hembra	verdeAzu(SF)	Cuerpo - ZG+Wbi+bl
6.25%	hembra	verdeAzu(DF)	Cuerpo - ZG+Wbibl
6.25%	macho	amarillo	Cuerpo - ZAZAbi+bl+
12.5%	macho	amarilloAzu(SF)	Cuerpo - ZAZAbi+bl
6.25%	macho	amarilloAzu(DF)	Cuerpo - ZAZAbibl
6.25%	hembra	amarillo	Cuerpo - ZAWbi+bl+
12.5%	hembra	amarilloAzu(SF)	Cuerpo - ZAWbi+bl
6.25%	hembra	amarilloAzu(DF)	Cuerpo - ZAWbibl

Figura 9. Resultados cuerpo

El siguiente paso es calcular las probabilidades conjuntas de los caracteres cabeza, pecho, cuerpo. Para ello se utiliza la formula

De este modo se obtiene que las probabilidades de obtener por ejemplo un pájaro macho de cabeza roja pecho morado y cuerpo verde-amarillo son de 0,391%

Fenotipo Macho : naranja-salmon/lila/amarillo Genotipo Macho : ZRZroo/lwb/ZG+ZAbl+bl		Fenotipo Hembra : blanco/morado/amarillo Genotipo Hembra : ZrWo+o/wb+wb/ZAWbl+bl				
	Porcentaje	Sexo	Cabeza	Pecho	Cuerpo	Genotipo
●	0.391%	macho	rojo	morado	verdeAmarillo	ZRZro+owb+llZG+ZAbl+bl+
●	0.781%	macho	rojo	morado	verdeAmarilloAzu(SF)	ZRZro+owb+llZG+ZAbl+bl
●	0.391%	macho	rojo	morado	verdeAmarilloAzu(DF)	ZRZro+owb+llZG+ZAblbl
●	0.391%	macho	rojo	morado	amarillo	ZRZro+owb+llZAZAbl+bl+
●	0.781%	macho	rojo	morado	amarilloAzu(SF)	ZRZro+owb+llZAZAbl+bl
●	0.391%	macho	rojo	morado	amarilloAzu(DF)	ZRZro+owb+llZAZAblbl
●	0.391%	macho	rojo	lila	verdeAmarillo	ZRZro+ollwbZG+ZAbl+bl+
●	0.781%	macho	rojo	lila	verdeAmarilloAzu(SF)	ZRZro+ollwbZG+ZAbl+bl
●	0.391%	macho	rojo	lila	verdeAmarilloAzu(DF)	ZRZro+ollwbZG+ZAblbl
●	0.391%	macho	rojo	lila	amarillo	ZRZro+ollwbZAZAbl+bl+

◀ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 ▶

VISUALIZAR

Figura 10. Resultados completos

PRUEBAS SOBRE LOS BOTONES DE LA APLICACIÓN MENDEL

Modulo Prueba	Caso de Prueba	Resultado Esperado	Resultado
Visualizar Macho	Formulario incompleto o con combinaciones no validas.	El sistema informa del error.	Correcto
	Formulario correcto	El sistema carga la imagen correspondiente al fenotipo suministrado por el cliente.	Correcto
Visualizar Hembra	Formulario incompleto o con combinaciones no validas.	El sistema informa del error.	Correcto
	Formulario correctO	El sistema carga la imagen correspondiente al fenotipo suministrado por el cliente.	Correcto
Botones	Borrar	El sistema borra todos los valores introducidos por el cliente.	Correcto
	Calcular	El sistema calcula las probabilidades para el carácter de la cabeza.	Correcto
	Volver	El sistema devuelve al cliente a la pantalla anterior.	Correcto
	Continuar	El sistema continua con el proceso mostrando la siguiente pantalla.	Correcto
	Visualizar	El sistema abre un popup con la imagen e información acerca del fenotipo, sexo y genotipo de la cría seleccionada por el cliente.	Correcto

3.2.4. Gestión de proveedores

Modulo Prueba	Caso de Prueba	Resultado Esperado	Resultado
Búsqueda de proveedores	Buscar	El sistema realiza la búsqueda de proveedores que se ajusten a los parámetros de búsqueda.	Correcto
	Borrar	El sistema borra todos los parámetros de búsqueda introducidos por el cliente.	Correcto
	Listado Proveedores	El listado muestra un check para la selección de proveedores, el código, nombre, tipo de proveedor (empresa o particular) y un enlace a los productos suministrados por el proveedor.	Correcto
	Enlace producto	El sistema muestra un popup con el listado de productos suministrados por el proveedor, el listado muestra el código, nombre y precio del producto.	Correcto
Alta proveedores	Formulario Proveedor	El sistema muestra el formulario para añadir nuevos proveedores.	Correcto
		Un mismo proveedor puede ser añadido varias	Correcto

veces, se diferenciara por su código de proveedor.

El sistema permite añadir los productos asociados a un proveedor. El formulario de alta de productos solicitara la introducción de la familia a la que pertenece un producto, el nombre y el precio de compra al proveedor. Si existe el producto en base de datos, es decir, existe un producto con el mismo nombre y que pertenece a la misma familia, el sistema recuperara el código del producto, sino lo marcará como indefinido y en el momento del alta definitiva del proveedor se darán de alta los nuevos productos asignándoles un código.

Modulo Prueba	Caso de Prueba	Resultado Esperado	Resultado
Alta de Productos	El usuario solicita añadir un nuevo producto a la lista de productos suministrados por el proveedor	El sistema muestra el formulario	Correcto
	El producto no existe en la base de datos	El sistema lo marca como indefinido y lo muestra junto con el nombre y el precio en el listado de productos.	Correcto
	El producto existe en la base de datos	El sistema recupera su código y lo muestra junto con el nombre y el precio en el listado de productos.	Correcto
	(*) El producto existe en la base de datos pero está marcado como borrado	El sistema lo marca como recuperado y lo muestra junto con el nombre y el precio en el listado de productos.	Correcto
	El usuario solicita eliminar un producto de la lista	El sistema lo elimina y muestra nuevamente el listado actualizado	Correcto
	Alta	El sistema añade el nuevo proveedor y el listado de productos suministrados por este guardándolos en la base de datos.	Correcto

* El sistema permite la recuperación de productos que anteriormente habían sido eliminados, se considerara que se trata del mismo producto si coincide su nombre y la familia a la que pertenece.

Modulo Prueba	Caso de Prueba	Resultado Esperado	Resultado
Modificacion proveedores	Formulario Proveedor	El sistema muestra el formulario con los datos del proveedor.	Correcto
Alta de productos	El usuario solicita añadir un nuevo producto a la lista de productos suministrados por el proveedor	El sistema muestra el formulario	Correcto
	El producto no existe en la base de datos	El sistema lo marca como indefinido y lo muestra junto con el nombre y el precio en el listado de productos.	Correcto
	El producto existe en la base de datos	El sistema recupera su código y lo muestra junto con el nombre y el precio en el listado de productos.	Correcto
	(*) El producto existe en la base de datos pero está marcado como borrado	El sistema lo marca como recuperado y lo muestra junto con el nombre y el precio en el listado de productos.	Correcto
	El usuario solicita eliminar un producto de la lista	El sistema lo elimina y muestra nuevamente el listado actualizado	Correcto
	Alta	El sistema modifica el proveedor y el listado de productos suministrados por este guardándolos en la base de datos.	Correcto
Consulta de proveedores	de El usuario solicita la consulta de los datos de un proveedor	El sistema muestra los datos del proveedor en modo consulta	Correcto
Borrado de proveedores	de El usuario solicita la eliminación del proveedor.	El sistema marca al proveedor como borrado en la base de datos y borra la lista de productos suministrados por este.	Correcto
	El proveedor es el único proveedor de un determinado producto. Dos casos		
	El stock del producto es menor o igual que cero	El sistema marca el producto como eliminado	Correcto
	El stock del producto es superior a cero	El sistema no elimina el producto y lo marca como "producto sin proveedor"	Correcto
		Un proveedor eliminado, al contrario que clientes o productos, no puede ser recuperado.	Correcto

3.2.5. Gestión de centros

Modulo Prueba	Caso de Prueba	Resultado Esperado	Resultado
Búsqueda de centros	Buscar	El sistema realiza la búsqueda de los centros que se ajusten a los parámetros de búsqueda.	Correcto
	Borrar	El sistema borra todos los parámetros de búsqueda introducidos por el cliente.	Correcto
	Listado centros	El listado muestra el código, nombre y provincia de los centros, siendo el código un enlace para ir a la modificación de los centros.	Correcto
	Enlace producto	El sistema muestra la pantalla de modificación de centros.	Correcto
Alta centros	Formulario centro	El sistema muestra el formulario para añadir nuevos centros.	Correcto
	Cada centro es único y no pueden existir dos centros con el mismo nombre	El sistema informa de error al usuario.	Correcto
	Cada centro deberá tener asociado un login para poder realizar pedidos. Si el login está siendo usado por otro cliente el sistema debe informar del error.	El sistema informa de error al usuario.	Correcto
	Alta de centro que había sido borrado anteriormente.	Si coincide el nombre del centro, el login elegido, teléfono y correo el sistema permite la recuperación del centro.	Correcto
	El login elegido esta reservado para un centro que fue eliminado	Si no coincide el nombre, login, teléfono y correo el sistema informa que ese login está reservado para la recuperación del centro al que perteneció.	Correcto
	Modificación de centros	Formulario centro	El sistema muestra el formulario con los datos del centro.
	Cada centro es único y no pueden existir dos centros con el mismo nombre	El sistema informa de error al usuario.	Correcto
	Cada centro deberá tener asociado un login para poder realizar pedidos. Si el login está siendo usado por otro cliente el sistema debe informar del error.	El sistema informa de error al usuario.	Correcto
	El login elegido esta reservado para un centro borrado anteriormente.	El sistema informa del error y no permite continuar con el proceso de modificación.	Correcto
	Los datos nuevos coinciden con los de un centro borrado anteriormente.	El sistema en la parte de modificación no permite la recuperación de centros borrados.	Correcto
Borrado de centros	Usuario elimina centro	El sistema elimina el centro y todos sus datos asociados. El usuario asociado al centro en el momento de creación de este se marca como eliminado.	Correcto
	Recuperación de centros.	Un centro borrado puede ser recuperado. En este caso se volverá a insertar el centro en la tabla de centro y se marcará su usuario como no eliminado. Se considera centro recuperado cuando coincide su login, nombre, teléfono y dirección de correo electrónico.	Correcto

3.2.6. Gestión de productos

Modulo Prueba	Caso de Prueba	Resultado Esperado	Resultado
Búsqueda de productos	Buscar	El sistema realiza la búsqueda de los productos que se ajusten a los parámetros de búsqueda.	Correcto
	Borrar	El sistema borra todos los parámetros de búsqueda introducidos por el cliente.	Correcto
	Listado productos	El listado muestra el código, nombre, fecha, stock y tipo siendo el nombre un enlace para ir a la modificación del producto. Para aquellos productos que no tengan proveedor asociado aparecerán marcados.	Correcto
	Enlace producto	El sistema muestra la pantalla de modificación del producto.	Correcto
Modificación de productos	Formulario producto	El sistema muestra el formulario con los datos del producto.	Correcto
	El usuario modifica el nombre del producto por otro que pertenece a	El sistema informa de error al usuario.	Correcto

	otro producto		
	El producto se modifica como ofertado. (OFERTADOS)	El producto ya está disponible en el catalogo de productos para poder ser adquirido por los clientes	Correcto
Borrado de productos	Usuario elimina un producto.	El sistema lo marca como eliminado y lo elimina de la lista de productos.	Correcto
	Recuperación de productos.	Cuando el administrador este añadiendo productos a la lista de productos suministrados por el proveedor, si uno de estos es un producto marcado como eliminado permitirá la recuperación de este.	Correcto

3.2.7. Pedidos a proveedores

Modulo Prueba	Caso de Prueba	Resultado Esperado	Resultado
Selección productos pedido	Buscar Productos	Los productos que aparecerán serán todos aquellos que estén dados de alta en la base de datos y que tengan un proveedor asociado.	Correcto
		En el listado aparecerá el código, nombre y listado del precios de los distintos proveedores para ese producto.	Correcto
	Buscar Productos por Proveedor	Si se ha seleccionado un proveedor aparecerá el listado de productos suministrados por ese proveedor.	Correcto
		En el listado aparecerá el código, nombre y precio del producto para ese proveedor.	
	Buscar Proveedor	Aparecerá el listado de proveedores que estén dados de alta en base de datos.	Correcto
	Buscar Proveedor por Producto	Si se ha seleccionado un producto aparecerá el listado de proveedores que suministran ese producto.	Correcto
	Añadir	El sistema ira añadiendo uno a uno los productos del pedido.	Correcto
	Generar Factura	El sistema calcula la factura asociada al pedido y la mostrara al usuario para su aceptación.	Correcto
	Aceptar Factura	El sistema solicitara al usuario la aceptación de la factura para terminar con el proceso de pedido, una vez aceptada se realizara el pedido y este podrá ser consultado en el apartado de "pedidos a proveedores". Además en este punto el sistema permite al usuario imprimir los datos de la factura.	Correcto

3.2.8. Pedidos de clientes

Modulo Prueba	Caso de Prueba	Resultado Esperado	Resultado
Pedido cliente	Buscar Productos	Los productos que aparecerán serán todos aquellos que estén dados de alta en la base y estén disponibles para su venta.	Correcto
		En el listado aparecerá el código, nombre y el stock y el precio.	Correcto
	Añadir	El sistema ira añadiendo uno a uno los productos del pedido. En el listado aparecerá el código, nombre, número de unidades solicitadas, stock y precio unitario	Correcto
	Eliminar	El sistema permite la eliminación de productos del carrito.	Correcto
	Formulario de pagos	El sistema cargara los datos que el cliente introdujo en el formulario de alta de cliente, permitiendo su modificación.	Correcto
		Si el usuario modifica estos datos será para que el pedido sea enviado a otra dirección, pero no implica la modificación de sus datos de usuario.	Correcto
		El sistema solicita en este punto la selección de la modalidad de pago.	Correcto
	Formulario de Pagos II	El sistema muestra al cliente un resumen con los datos del pedido. Estos incluyen los datos de la factura, de envío y la selección del banco en caso de que la forma de pago seleccionada sea mediante transferencia bancaria.	Correcto
	Cálculo de los gastos de envío del pedido	El sistema calcula los gastos de envío del pedido y los muestra en los datos de la factura.	Correcto
	Confirmar	El sistema realiza el pedido almacenando los datos	Correcto

		de este en la base de datos. En este punto el pedido podrá ser consultado o se podrá modificar su estado por el administrador.	
	Generar Factura	El sistema calcula la factura asociada al pedido y la mostrara al usuario para su aceptación.	Correcto
	Aceptar Factura	El sistema solicitará al usuario la aceptación de la factura para terminar con el proceso de pedido, una vez aceptada se realizara el pedido y este podrá ser consultado en el apartado de "pedidos a proveedores". Además en este punto el sistema permite al usuario imprimir los datos de la factura.	Correcto
Pedido Centro	Buscar Productos	El comportamiento es el mismo que en clientes. Salvo que el precio de venta para los centros es el más alto del precio al que venden ese producto los proveedores.	Correcto
	Añadir	El comportamiento es el mismo que en clientes.	Correcto
	Eliminar	El comportamiento es el mismo que en clientes.	Correcto
	Formulario de pagos	El sistema cargara solo los datos de los que disponga para ese centro, el resto de datos solicitados no serán necesarios para realizar el pedido.	Correcto
	Formulario de Pagos II	El comportamiento es el mismo que en clientes.	Correcto
	Cálculo de los gastos de envío del pedido	El sistema al identificar al usuario como un centro no cobrara gastos de envío para ese pedido	Correcto
	Confirmar	El comportamiento es el mismo que en clientes.	Correcto
	Generar Factura	El comportamiento es el mismo que en clientes.	Correcto
	Aceptar Factura	El comportamiento es el mismo que en clientes.	Correcto
	Inserción datos distribución	Los productos pedidos pasaran a ser aquellos que el centro distribuye. Por un lado se almacenaran el número de unidades pedidas que será el stock de ese producto en ese centro, y por otro lado la suma de todas las unidades pedidas que serán las ventas de ese producto en ese centro.	Correcto

3.2.9. Consulta de stock

Modulo Prueba	Caso de Prueba	Resultado Esperado	Resultado
Consulta Stock	Buscar Productos	Los productos que aparecerán serán todos aquellos que estén dados de alta en la base y estén disponibles para su venta.	Correcto
	Listado productos	Mostrara un listado con el código del producto, nombre y stock.	Correcto
	Buscar centros	Realizara una búsqueda sobre todos aquellos centros que estén dados de alta en el sistema	Correcto
	Introduce centro	La búsqueda se realizara por el stock disponible en un determinado centro	Correcto
	No introduce centro	La búsqueda se realizara por el stock disponible en el almacén	Correcto

3.2.10. Generar catálogo

Modulo Prueba	Caso de Prueba	Resultado Esperado	Resultado
Catálogo de Ofertas	Listado Ofertas	Mostrará un listado con el catálogo de ofertas. Por cada productos mostrará el código, link con el nombre del producto, características, precio y botón para visualizar la imagen del producto	Correcto
		El link sobre producto lleva a una pantalla que permite modifica el precio del producto ofertado	Correcto
		El botón visualizar permite visualizar en un popup el producto ofertado en caso de existir imagen	Correcto
	Añadir Oferta	Muestra popup para la búsqueda del producto. Permite modificar las características y el precio de venta. Al añadir el producto se chequea como ofertado (campo promoción a 1)	Correcto
		Un vez añadido el producto ofertado aparecerá en el listado de productos	Correcto

3.2.11. Gestión clientes

Modulo Prueba	Caso de Prueba	Resultado Esperado	Resultado
Búsqueda de clientes	de Buscar cliente	Listado de pedidos de clientes filtrados por código de cliente, apellidos, dni, estado del pedido y/o fecha de pedido. El listado muestra código de cliente, nombre, apellidos, estado del pedido, fecha del pedido y código del pedido	Correcto
		El código de cliente será un link lleva a una pantalla con los datos del cliente en modo lectura salvo el estado del pedido	Correcto
	Modificar estado pedido cliente	Modifica el estado del pedido	Correcto
	Baja cliente	A través del botón dar de baja da de baja al cliente	Correcto

3.2.12. Visualización de gráficos

Modulo Prueba	Caso de Prueba	Resultado Esperado	Resultado
Visualizar producto	por Visualizar producto	Muestra un gráfico de las unidades vendidas por producto filtradas por familia o fecha de venta del producto	Correcto
Visualizar por tienda	Visualizar por tienda	Muestra un gráfico de las unidades vendidas por producto en un determinado centro filtradas por centro o fecha de venta del producto	Correcto

Documentación de Usuario

Índice de Contenido

Manual de Usuario	
1. Introducción	230
2. Requisitos mínimos	231
2.1. Requisitos hardware	231
2.2. Requisitos software	231
3. Manual de usuario	231
3.1. Presentación de la aplicación.....	231
3.2. Funcionalidades de la aplicación	231
3.3. Accediendo a PAJARRACO'S PAJARERÍA	232
3.4. Menú de usuario	233
3.4.1. Cómo registrarse en la web.....	233
3.4.2. Visualizar catálogo de productos.....	235
3.4.3. Conocer información acerca de numerosas aves	236
3.4.4. Consultar el listado de centros	237
3.4.5. Cómo ponerse en contacto con la empresa.....	237
3.5. Menú cliente.....	238
3.5.1. Cómo realizar un pedido a través de la web.....	239
3.5.2. Gestionar datos de usuario	243
3.5.3. Programa Mendel.....	244
3.6. Menú administrador	250
3.6.1. Gestión de clientes	251
3.6.2. Gestión de proveedores.....	253
3.6.3. Gestión de productos.....	255
3.6.4. Gestión de pedidos	260
3.6.5. Consulta de stock	262
3.6.6. Catálogo de ofertas	263
3.6.7. Visualización gráfico.....	266
4. Manual de administración - parte servidora	267
4.1. Instalación Java.....	267
4.2. Instalación del Servidor de Aplicaciones	267
4.3. Instalación de la herramienta de desarrollo	268
4.4. Instalación del Sistema Gestor de Base de Datos	268
4.5. Instalación del Servidor Web.....	268
4.6. Instalación de la aplicación	269
4.7. Reinicio de la aplicación.....	274
4.8. Export / Import de la Base de Datos	275
4.9. Export / Import de la Base de Datos	275

1. Introducción

El presente documento tiene como finalidad acercar al usuario la aplicación web Pajarraco's Pajarería. Podemos distinguir dos grandes bloques dentro del manual de usuario:

- **Requisitos mínimos:** en este apartado se desglosarán todos los distintos tipos de requisitos que son imprescindibles para poder acceder a la web de la pajarería desde cualquier pc.
- **Manual de uso de la aplicación web:** aquí se desglosarán todas las pantallas de la web y se detallará el funcionamiento de cada una de ellas.

2. Requisitos mínimos

2.1. Requisitos hardware

La aplicación que hemos desarrollado se trata de una aplicación web. Este tipo de aplicaciones no se instalan en la máquina del cliente que la está utilizando, sino que estos clientes acceden a la aplicación a través de una "ip" (dirección web), que introduce en la barra de un navegador web. La aplicación estará corriendo en una máquina servidora que será la encargada de servir la web a los distintos clientes que la soliciten. Por tanto los clientes que accedan a la aplicación no deberán de tener ningún requisito hardware específico de su propia máquina.

2.2. Requisitos software

Esta aplicación ha sido optimizada para ser usada por los navegadores Firefox, IE 7 e inferior y Chrome.

3. Manual de usuario

3.1. Presentación de la aplicación

Actualmente cada vez es más común el uso de las nuevas tecnologías como forma de interactuar con el medio que nos rodea. Es por ello por lo que el uso de aplicaciones web se ha convertido en el medio más usado por las empresas para darse a conocer a los diferentes sectores de la sociedad. PAJARRACO'S PAJERÍA es una aplicación web que surge de la necesidad de acercar de una forma eficiente (tanto a clientes de cartera como potenciales), una empresa dedicada fundamentalmente tanto a la comercialización de especies de aves así como de los productos necesarios para sus cuidados. Es una manera cómoda, fácil y rápida para el usuario de comprar productos, resolver dudas etc únicamente conectándose a internet desde cualquier ordenador. De la misma manera el uso de esta aplicación web en un negocio supone tanto un ahorro de costes de infraestructura (ya que deja de ser necesario el tener locales físicos para la venta con todos los gastos que eso conlleva), además de una importante herramienta para gestionar el negocio y conocer de manera rápida y cómoda los resultados

3.2. Funcionalidades de la aplicación

Las funcionalidades de la aplicación variarán en función del rol que tenga el usuario.

- Para un usuario sin estar registrado podrá realizar las siguientes funcionalidades:

- Visualizar el catálogo de productos.
- Listado de centros y contacto.
- Información acerca de numerosas aves.
- Registrarse en la web.

- En caso de que el usuario sea cliente y esté logado además podrá:

- Comprar cualquier producto disponible en el catálogo.
- Modificar sus datos personales.

Acceder al foro de la aplicación.
Podrá acceder al programa Mendel.

- Los empleados de la empresa tendrán un perfil especial de administrador con el que podrán llevar a cabo las siguientes acciones:

Gestionar todo lo referente a pedidos de la empresa (tanto los realizados por la empresa a los proveedores que tiene como los pedidos realizados por los clientes a través de la web).

Gestionar el catálogo de productos de la empresa y el stock de cada uno de ellos.

Gestionar los proveedores que suministran los productos a la empresa.

Realizar labores de moderador del foro.

Gestionar y actualizar la información referente a los centros que tiene abiertos la empresa.

Conocer la situación de ventas en cada momento.

3.3. Accediendo a PAJARRACO'S PAJARERÍA

Como ya hemos comentado anteriormente, nuestro proyecto se basa en una aplicación web y para poder acceder a él deberá abrir una ventana de internet explorer y pegar la URL o dirección web, correspondiente a la aplicación.

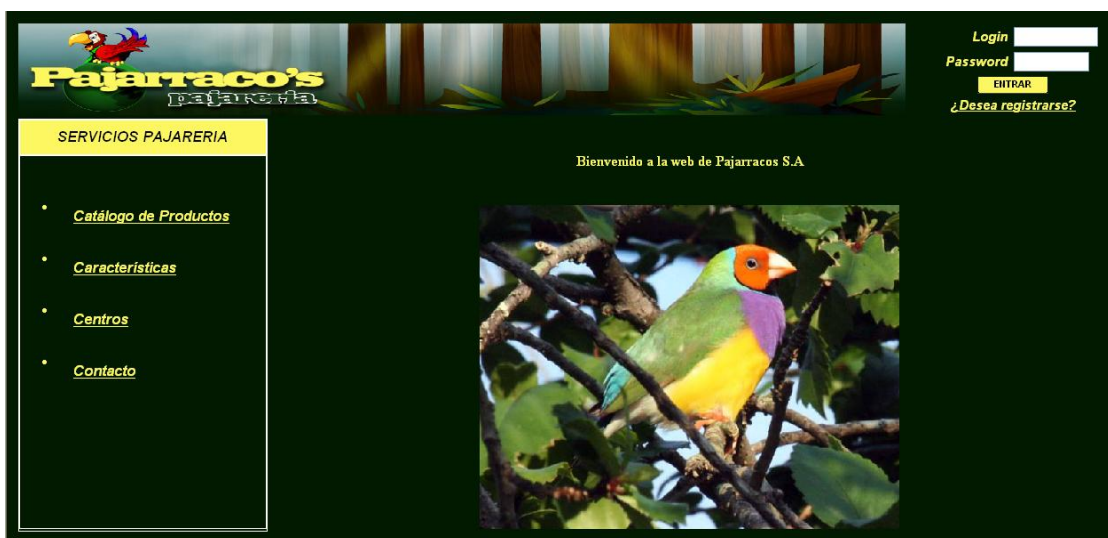


Figura 1. Pantalla de inicio de Pajarraco's S.A

El navegador le redirigirá automáticamente a la web mostrándole su pantalla de inicio. Desde esta pantalla, podrá acceder a todas las funcionalidades del programa.

Esta pantalla se divide en tres partes:

- **Encabezado:** que muestra la imagen con el logo de la empresa. Como imagen corporativa de la empresa se mostrará en todas las pantallas de la web.



- **Área de registro:** situada en el margen superior derecho y al igual que el encabezado se mostrará en todas las pantallas de la web. En esta región aparecerán dos cajas de texto donde los usuarios registrados podrán logarse introduciendo su login y password y pinchando en el botón "ENTRAR". Además se muestra un link "¿Desea registrarse?", donde redirigirá a los usuarios no registrados, a un formulario podrán darse de alta en la web. En caso de que un usuario se haya logado, esta región se modificará apareciendo su login y un link Cerrar Sesión para des-logarse.



- **Menú lateral:** situado en el lateral izquierdo de la pantalla. Se mostrará también en todas la pantallas de la aplicación. Cambia según el rol. Si el usuario que está navegando por la web no está registrado se mostrará el menú de usuario, por el contrario para usuarios registrados el menú que se mostrará es el de cliente (para clientes de la tienda) o menú de administrados (para el personal autorizado o empleados).

3.4. Menú de usuario

El menú usuario es accesible desde la página principal de la aplicación y desde el cualquier usuario podrán realizar las siguientes acciones:

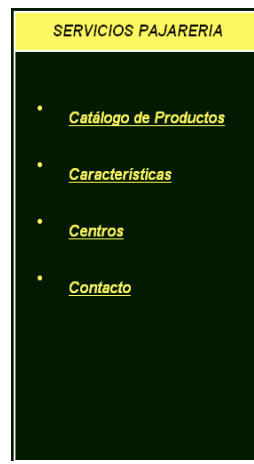


Figura 2. Menú usuario

- 1.- Registrarse en la web.
- 2.- Visualizar el catálogo de productos.
- 3.- Información acerca de numerosas aves.
- 4.- Consultar el listado de centros.
- 5.- Ponerse en contacto con la empresa.

3.4.1. Cómo registrarse en la web

Esta funcionalidad no aparece dentro del menú lateral sino en el área de registro, pero al ser una funcionalidad propia de cualquier usuario se ha decidido englobar dentro de las funcionalidades del menú de usuario.

La pantalla para poder registrarse en la web contiene un formulario con la siguiente información a rellenar:

Figura 3. Alta usuario

A continuación pasamos a describir brevemente la información que se deberá introducir en cada campo del formulario:

Campo	Tipo de Campo	Descripción
Tratamiento	Radio	Describe el tratamiento que se dará al cliente que se esté registrando en la web. Podrán ser Sr o Sra. Sólo podrá seleccionarse uno.
Nombre	Campo de texto	Campo de texto obligatorio de 20 caracteres, que contendrá el nombre del cliente. Sólo admite caracteres alfabéticos.
Primer Apellido	Campo de texto	Campo de texto obligatorio de 20 caracteres, contendrá el primer apellido del cliente. Sólo admite caracteres alfabéticos.
Segundo Apellido	Campo de texto	Campo de texto obligatorio de 20 caracteres, contendrá el segundo apellido del cliente. Sólo admite caracteres alfabéticos.
Tipo de documento	Lista desplegable	Lista desplegable con todos los tipos de documentos identificativos que podrá elegir el cliente. Será un campo obligatorio y no tendrá ningún valor marcado por defecto.
Nº Documento	Campo de texto	Campo de texto obligatorio de 9 caracteres, validar para cada una de las distintas opciones. Para el tipo de documento DNI la validación que se llevará a cabo será 8 dígitos y una letra del alfabeto.
Fecha de Nacimiento	Calendario	Calendario.
Nacionalidad	Campo de texto	Campo de texto editable y obligatorio de 25 caracteres alfabéticos.
Correo electrónico	Campo de texto	Campo de texto opcional de 40 caracteres con la dirección de correo electrónico del cliente. Los correos que admite son de la forma correo@XXX o correo@XXX.es.
Teléfono de contacto	Campo de texto	Campo de texto editable de 9 caracteres numéricos, obligatorio con el teléfono de contacto del cliente
Tipo de cliente	Radio	Radio con dos opciones, particular o empresa. Se trata de un campo obligatorio.
País	Lista desplegable	Lista desplegable en la que el usuario deberá seleccionar su país de residencia. Actualmente sólo hay registrados en el sistema dos países, España y Francia ya que únicamente hay centros físicos abiertos en estos países.
Calle	Campo de texto	Campo de texto obligatorio de 50 caracteres, contendrá el nombre de la calle donde reside cliente. Admite únicamente caracteres alfabéticos.
Número	Campo de texto	Campo de texto obligatorio de 3 caracteres numéricos. Contendrá el número

Campo	Tipo de Campo	Descripción
		de la vivienda en la que reside el cliente.
Piso/Letra	Lista desplegable	Listas desplegables y obligatorias en el que se informará el piso y la letra de la vivienda del cliente. El combo piso contendrá los números del 1 al 9 y la letra de la "A" – "I". En ningún combo aparece seleccionado ningún valor por defecto.
Bloque	Campo de texto	Campo de texto opcional de dos dígitos, contendrá el número del bloque de la vivienda del cliente
Información adicional	Campo de texto	Campo de texto opcional de 50 caracteres para cualquier aclaración sobre la dirección del cliente. Admite caracteres alfanuméricos.
Código Postal	Campo de texto	Campo de texto obligatorio de 5 dígitos, que informa del código postal de la residencia del cliente.
Población	Campo de texto	Campo de texto obligatorio de 40 caracteres, contendrá el nombre de la población del cliente. Admite únicamente caracteres alfabéticos.
Provincia	Lista desplegable	Lista desplegable en la que el usuario deberá seleccionar su provincia de residencia. Este listado permanecerá vacío hasta que el usuario seleccione un país. En ese momento se cargará un listado con las provincias pertenecientes al país seleccionado.
Login	Campo de texto	Campo de texto obligatorio de 10 caracteres, contendrá el login elegido por el cliente.
Password	Campo de texto	Campo de texto obligatorio de 10 caracteres, contendrá el password elegido por el cliente.
Darse de alta	Botón	Botón que el usuario deberá pinchar para guardar el formulario de registro.
Borrar	Botón	Botón para borrar los campos informados.

3.4.2. Visualizar catálogo de productos

Un cliente que quiera consultar el catálogo de productos de la empresa deberá seleccionar la opción del menú de usuario correspondiente a Catálogo de Productos. Esta pantalla presenta un formulario con los campos de Familia, Producto y Visualizar Oferta en caso de que se quiera filtrar la búsqueda de productos. Si no se rellena ningún campo se mostrará el catálogo de productos completo. En caso de que la búsqueda que se esté realizando no produzca ningún resultado se mostrará un mensaje por pantalla indicándolo "No existe ningún producto en el catálogo que se ajuste a la búsqueda".

La pantalla presenta la siguiente información:

	Código	Nombre	Precio Unitario	Stock
■	1	DIAMANTE DE GOULD	15.30	100
■	2	DIAMANTE BICHENOW	12.00	250
■	3	DIAMANTE MANDARIN	12.90	45
■	4	DIAMANTE TRICOLOR	25.00	93
■	5	JAULA DORADA	12.95	50
■	6	CANARIO	10.95	10
■	7	ALPISTE LIGTH	6.95	100
■	8	ALPISTE MULTICEREALES	7.50	250

Figura 4. Visualizar catálogo de productos

A continuación se detalla el funcionamiento de cada campo del formulario:

Campo	Tipo de Campo	Descripción
Familia	Lista desplegable	Campo opcional con un valor vacío por defecto. Será una lista desplegable con todas las categorías en las que pueden agruparse los productos disponibles en la tienda. Actualmente las categorías existentes son: aves, alimentación, medicamentos, jaulas, juguetes y otros.
Producto	Campos de texto y lupa	Campo de texto opcional que muestra el código y la descripción de un producto. Permite introducir un código de producto que se quiere visualizar, cargándose automáticamente la descripción en el campo correspondiente. Otra forma de realizar la búsqueda de un producto consistirá en pinchar sobre la imagen de la lupa, que abrirá un popup (llamado popup productos) donde el usuario podrá elegir un producto que desea consultar. Para ver funcionamiento consultar anexo de guía de popups, buscar producto.
Visualizar Ofertas	Checkbox	Cuando el usuario marca el check de Visualizar Ofertas, la aplicación mostrará únicamente aquellos productos que se encuentren ofertados en ese momento.
Listado de Productos	Listado	Es el listado correspondiente al resultado de realizar una búsqueda. Contendrá el código, nombre, el precio y el stock del producto/s mostrados. Si no se introduce ningún criterio de búsqueda mostrará todo el catálogo de productos existentes en la tienda. Si la búsqueda de productos no devuelve ningún resultado, se mostrará un mensaje de aviso indicándolo.
Borrar	Botón	Borra la información introducida en el formulario.
Buscar	Botón	Botón para realizar la búsqueda de productos. Dependiendo de los filtros de búsqueda, se mostrará un listado de productos resultado para cada caso.

3.4.3. Conocer información acerca de numerosas aves

Cualquier cliente puede hacer consultas acerca de varias aves a través del enlace de características que aparece en la pantalla principal de la web. Los temas que se abordan en este apartado van desde conocer cuidados básicos de cada tipo de ave hasta cómo curar enfermedades habituales o curiosidades y habilidades particulares de cada especie. La página presenta un menú horizontal en la parte superior que muestra los temas que puede consultar el cliente. Dentro de cada tema aparecen varias familias de aves con distintos subespecies de cada una.

La pantalla de características se muestra de la siguiente manera:

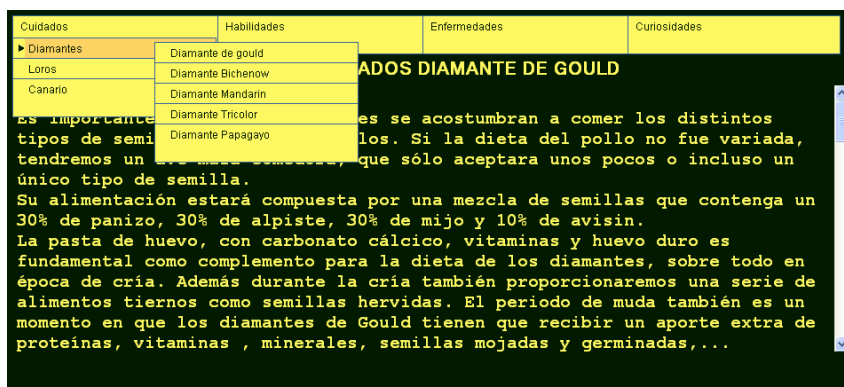


Figura 5. Información aves

Campo	Tipo de Campo	Descripción
Cuidados, habilidades, enfermedades,	Link	Como ya se ha explicado anteriormente, cada una de estas opciones será un menú desplegable donde se obtendrá información acerca del tema y

Campo	Tipo de Campo	Descripción
curiosidades		especie que se halla seleccionado.

3.4.4. Consultar el listado de centros

Para poder consultar los datos de una tienda, se deberá seleccionar la opción del menú de usuario “centros”. La web presentará un mapa del territorio español dividido por provincias. En ese momento se elegirá la provincia de la cual se quiere conocer los centros, y a continuación aparecerá una etiqueta donde se detallará la información de todos los centros que se ubiquen en la provincia elegida. En caso de que en la provincia que se haya seleccionado no exista ningún centro se mostrará un mensaje advirtiéndolo “No existe ningún centro en ésta provincia”.

A continuación mostramos la pantalla de centros:



Figura 6. Centros

Campo	Tipo de Campo	Descripción
Mapa	Imagen	Mapa de España delimitado por comunidades. Al pinchar sobre cada provincia saldrá la información de contacto relativa a los centros que se hallen en esa provincia.
Provincia, dirección, teléfono, e-mail	Etiqueta	Datos de contacto relativos a los centros seleccionados.

3.4.5. Cómo ponerse en contacto con la empresa

La empresa pone a disposición de cualquier usuario la posibilidad de ponerse en contacto con la empresa a través de la web. Para ello dispone de un formulario de contacto.

A continuación se muestra dicho formulario de contacto:

Figura 7. Contacto

Campo	Tipo de Campo	Descripción
Nombre	Campo de texto	Campo de texto obligatorio y editable de 20 caracteres (no admite números) que contendrá el nombre del usuario para poder identificarse.
Email	Campo de texto	Campo de texto obligatorio y editable con 40 caracteres, contendrá la dirección de email del centro con el que desea contactar. El formato del correo debe ser de la forma xxx@correo.com o xxx@correo . En caso de que el correo que se introduzca no cumpla este formato la aplicación arrojará un mensaje advirtiéndolo “La dirección de Correo Electrónico es incorrecta, el formato debe ser ejemplo@correo.com o ejemplo@correo ”. Además se comprobará que el email introducido corresponde con el de algún centro. En caso contrario, la aplicación mostrará un mensaje de error “El e-mail que ha introducido no corresponde con ningún centro existente”.
Motivo de contacto	Campo de texto	Campo de texto obligatorio y editable que admite hasta 40 caracteres en el que el usuario dará una breve descripción del motivo de contacto
Comentarios	Caja de texto	Caja de texto obligatorio y editable con 400 caracteres en la que se expondrá el motivo por el cual el usuario contactará con la empresa.
Enviar	Botón	Botón para enviar el formulario de contacto. Por defecto el envío se realizará a la dirección de correo del almacén. Tras Pulsar el botón se abrirá una ventana de nuevo mensaje de Outlook en la que se volcará la información del formulario de contacto para que el usuario pueda finalmente enviar el mensaje. En caso de fallo al abrir el mensaje en Outlook se mostrará un mensaje de error indicándolo “No se puede iniciar Microsoft Office Outlook”.
Borrar	Botón	Botón para borrar los datos introducidos en el formulario de contacto.

3.5. Menú cliente

El menú de cliente sólo está disponible para aquellos usuarios que previamente se hayan registrado en la web. Este menú será accesible tras logarse en la web. En el podrán accederse a las siguientes funcionalidades:

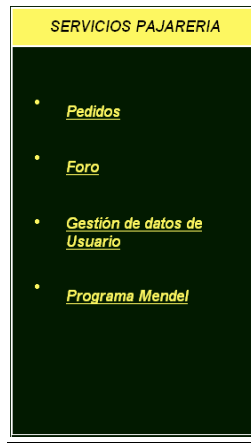


Figura 8. Cliente

- 1.- Realizar un pedido.
- 2.- Intercambiar mensajes en el foro de la aplicación.
- 3.- Modificar sus datos de usuario.
- 4.- Acceder al programa Mendel de descendencia genética.

3.5.1. Cómo realizar un pedido a través de la web

La realización de un pedido podrá comenzarse pinchando en el enlace de “Pedidos”. En ese momento la aplicación mostrará un formulario donde el usuario podrá comenzar a añadir productos a su carrito de la compra.

A continuación mostramos la pantalla inicial de pedidos:



Figura 9. Pedido

Campo	Tipo de Campo	Descripción
Producto	Campo de texto y lupa	Campo obligatorio que muestra el código y la descripción de un producto. Permite introducir un código de producto que se quiere visualizar, cargándose automáticamente la descripción en el campo correspondiente. Otra forma de realizar la búsqueda de un producto consistirá en pinchar sobre la imagen de la lupa, que abrirá un popup (llamado popup productos) donde el usuario podrá elegir un producto que desea consultar. Para ver funcionamiento consultar anexo de guía de popups, buscar producto.
Producto		
N unidades	Campo de texto	Campo obligatorio que admite 3 números como máximo.
Listado de productos	Listado	Este listado muestra los productos que se van añadiendo al carrito. Para cada producto se mostrará el código y nombre del producto, número de unidades que se van a adquirir del producto, stock disponible del producto y precio por unidad del mismo.

Campo	Tipo de Campo	Descripción
Añadir	Botón	Botón que se habilitará cuando se haya elegido un producto del catálogo y se haya introducido el número de unidades que se van a adquirir del producto. En caso de que el número de unidades que se quiere comprar supere el stock del producto no se añadirá al listado de productos del carrito, al igual que si se intenta añadir un producto que ya está en el carrito.
Borrar	Botón	Borra los campos del formulario que hayan sido informados.
Eliminar	Botón	Botón sólo habilitado en caso de que exista algún producto en el carrito y se haya marcado por lo menos uno de ellos. Borrará los productos marcados del carrito. Si se eliminan todos los productos del carrito se mostrará un mensaje de advertencia y no se podrá continuar con el proceso de compra,
Continuar	Botón	Botón sólo habilitado en caso de que haya algún producto en el carrito de la compra. Permitirá avanzar a la siguiente pantalla.

Tras pulsar el botón de continuar se mostrará una pantalla parecida a la siguiente:

CARRITO					
<input checked="" type="checkbox"/>	Código	Nombre	N unidades	Precio Unitario	Importe total
<input checked="" type="checkbox"/>	1	DIAMANTE DE GOULD	1	15.30	15.3
<input checked="" type="checkbox"/>	2	DIAMANTE BICHENOW	13	12.00	156
<input checked="" type="checkbox"/>	3	DIAMANTE MANDARIN	23	12.90	296.7

ELIMINAR

VOLVER CONTINUAR

Figura 10. Carrito

Campo	Tipo de Campo	Descripción
Listado de productos	Listado	Resumen del carrito. Descripción de los productos que se han adquirido. Para cada uno de ellos se mostrará su código, nombre, número de unidades que se van a comprar, precio por unidad y precio total por productos.
Eliminar	Botón	Este botón únicamente se habilitará cuando se marque algún producto del carrito. En este punto también se podrán eliminar los productos que finalmente no se quiera adquirir.
Volver	Botón	Permitirá regresar a la pantalla anterior, en este caso la pantalla del catálogo de productos para comprar.
Continuar	Botón	Botón sólo habilitado en caso de que haya algún producto en el carrito de la compra. Permitirá avanzar a la siguiente pantalla.

Continuamos avanzando en el proceso de compra. En este punto se le mostrará la siguiente pantalla con sus datos personales (aquellos con los que se registró en la web), y la posibilidad de escoger el método de pago para realizar el pedido. Todos los campos del formulario aparecerán deshabilitados salvo el método de pago.

FORMULARIO DE PAGOS

Tratamiento Sr. Sra.

Nombre Primer Apellido Segundo Apellido

Tipo de documento Nº Documento

Correo Electrónico Teléfono de Contacto

Calle Número

Piso/Letra Bloque

Población Código Postal

País Provincia

Información Adicional

Elija la forma de pago : Transferencia Bancaria Pago Contrareembolso

Figura 11. Pagos

Campo	Tipo de Campo	Descripción
Tratamiento	Radio	Indica el tratamiento que marcó el que el cliente se dio de alta en la web.
Nombre	Campo de texto	Nombre del cliente que está realizando la compra.
Primer apellido	Campo de texto	Apellido del cliente que está realizando la compra.
Segundo apellido	Campo de texto	Apellido del cliente que está realizando la compra.
Tipo de documento	Lista desplegable	Muestra el tipo de documento con el que se registró en la web.
Nº de documento	Campo de texto	Número de documento del cliente.
Calle	Campo de texto	Calle donde se sitúa la residencia del cliente que está realizando el pedido.
Número	Campo de texto	Número de la residencia del cliente que está realizando el pedido.
Piso/Letra	Campo de texto	Piso y letra de la residencia del cliente que está realizando el pedido.
Bloque	Campo de texto	Bloque de la residencia del cliente que está realizando el pedido.
Población	Campo de texto	Población donde reside el cliente.
Código Postal	Campo de texto	Código postal de la zona donde vive el cliente.
Provincia	Campo de texto	Provincia donde vive el cliente.
País	Campo de texto	País donde vive el cliente.
Información adicional	Campo de texto	Datos adicionales que introdujera el cliente en su formulario de alta.
Correo electrónico	Campo de texto	Correo electrónico de contacto con el cliente.
Teléfono de contacto	Campo de texto	Teléfono de contacto del cliente.
Forma de pago	Radio	Indica la forma de pago que ha elegido el cliente para realizar el pago del pedido.
Volver	Botón	Botón habilitado que permite regresar a la pantalla anterior (Carrito de la compra).

Campo	Tipo de Campo	Descripción
Continuar	Botón	Botón habilitado únicamente cuando se ha escogido un método de pago del pedido y que permite seguir con el proceso de compra.

En caso de que el método de pago elegido sea por Transferencia bancaria se mostrará la siguiente pantalla:

SERVICIOS PAJARERIA

- Pedidos
- Foro
- Gestión de datos de Usuario
- Programa Mendel

Pago Mediante Transferencia Bancaria

Datos del Pedido

Código	Nombre	N unidades	Precio Unitario	Importe
001	ort1	10	12.23	122.30000000000001
001	ort1	10	12.23	122.38000000000001
Importe total:				244.68000000000002

Datos del Envío

Datos del Cliente	Datos de la Entrega	Datos de Contacto	Fecha
mana sanchez postigo	torreón coronel fernandez de castro / 11, s/ojiva segovia - 09460	69665696@marzipos@hotmail.com	19/06/2008

Datos Bancarios

Banco o Caja	Entidad	Oficina	Dc	Cuenta
Banquepar	1111	1111	11	1111111111
Caja Duero	2222	2222	22	2222222222

Volver Confirmar

Figura 12. Resumen Pagos

Campo	Tipo de Campo	Descripción
Datos de pedidos	Listado	Listado donde el usuario podrá ver la factura desglosada de su pedido, de cada producto seleccionado podrá ver el código , nombre, número de unidades, precio unitario e importe además del importe total del pedido.
Datos de envío	Listado	Listado donde el usuario podrá ver un resumen de los datos de envío del pedido. Contendrá los datos del cliente con su nombre y apellido, los datos de entrega con la dirección, los datos de contacto del cliente, y la fecha del pedido.
Datos bancarios	Listado	Listado donde el usuario podrá consultar los datos bancarios donde realizar el pago. Contendrá el banco o caja, entidad, oficina, dc, y número de cuenta.
Volver	Botón	Botón habilitado que permite regresar a la pantalla anterior (Formulario de pago).
Confirmar	Botón	Botón habilitado que permite realizar el pedido en caso de que el formulario se halla rellenado correctamente y permite al usuario acceder a las siguientes pantallas según la opción de forma de pago elegida.

Una vez completado el pedido la web presentará al usuario un resumen del pedido que acaba de realizar. Se visualizará una pantalla similar a la que se muestra a continuación:

SERVICIOS PAJARERÍA

- Pedidos
- Foro
- Gestión de datos de Usuario
- Programa Mendiol

Finalizar Pago

Datos del Pedido

Código	Nombre	N unidades	Precio Unitario	Importe
001	art1	10	12.23	122.30000000000001
001	art1	10	12.23	122.30000000000001
Importe total:		244.60000000000002		

Datos del Envío

Datos del Cliente	Datos de la Entrega	Datos de Contacto	Fecha
maria sanchez postigo	torrealeto conuel fernandez de castro 7 1, segovia segovia - 09400	686668666 mariaaripos@icicamail.com	19/06/2008

Su pedido ha sido realizado con éxito. Gracias

Figura 13. Resumen Pedidos

Campo	Tipo de Campo	Descripción
Datos de pedidos	Listado	Listado donde el usuario podrá ver la factura desglosada de su pedido, de cada producto seleccionado podrá ver el código, nombre, número de unidades, precio unitario e importe además del importe total del pedido.
Datos de envío	Listado	Listado donde el usuario podrá ver un resumen de los datos de envío del pedido. Contendrá los datos del cliente con su nombre y apellido, los datos de entrega con la dirección, los datos de contacto del cliente, y la fecha del pedido.
Mensaje de éxito	Label	Mensaje de éxito de la operación.
Volver	Botón	Botón habilitado que permite regresar a la pantalla anterior (Formulario de pago).

3.5.2. Gestionar datos de usuario

A través de este formulario el usuario podrá modificar sus datos personales además de darse de baja como cliente de la web.

Pasamos a describir los distintos campos y botones que formarán parte de este formulario.

Campo	Tipo de Campo	Descripción
Tratamiento	Checkbox	Los tratamientos que se darán a cada cliente podrán ser Sr o Sra. Sólo podrá seleccionarse uno
Nombre	Campo de texto	Campo de texto obligatorio de 20 caracteres, contendrá el nombre del cliente
Primer Apellido	Campo de texto	Campo de texto obligatorio de 20 caracteres, contendrá el primer apellido del cliente
Segundo Apellido	Campo de texto	Campo de texto obligatorio de 20 caracteres, contendrá el segundo apellido del cliente
Tipo de documento	Lista desplegable	Lista desplegable con los tipos de documentos identificativos del cliente, será un campo obligatorio y su valor por defecto será el dni
Número de	Campo de texto	Campo de texto obligatorio de 9 caracteres, validar para cada una de las

Campo	Tipo de Campo	Descripción
documento		distintas opciones
Fecha de Nacimiento	Calendario	Calendario
Nacionalidad	Campo de texto	Campo de texto editable de 25 caracteres.
Datos de contacto	Etiqueta	Etiqueta con el nombre "datos de contacto"
Correo electrónico	Campo de texto	Campo de texto obligatorio de 20 caracteres con la dirección de correo electrónico del cliente
Teléfono de contacto	Campo de texto	Campo de texto editable de 9 caracteres obligatorio con el teléfono de contacto del cliente
Tipo de cliente	Checkbox	Checkbox con dos opciones, particular o empresa
País	Campo de texto	Campo de texto editable de 25 caracteres
Calle	Campo de texto	Campo de texto obligatorio de 50 caracteres, contendrá el nombre de la calle del cliente
Número	Campo de texto	Campo de texto obligatorio de 3 caracteres, contendrá el número de vivienda del cliente
Piso/Letra	Lista desplegable	Listas desplegables y obligatorias en el que se informará el piso y la letra a la que se realizará el pedido. El combo piso contendrá los números del 1 al 20 y la letra de la "a" – "f". En ningún combo aparece seleccionado ningún valor por defecto.
Bloque	Campo de texto	Campo de texto opcional de dos dígitos, contendrá el número del bloque de la vivienda del cliente
Información adicional	Campo de texto	Campo de texto opcional de 50 caracteres para cualquier aclaración sobre la dirección del cliente
Código Postal	Campo de texto	Campo de texto obligatorio de 5 caracteres, contendrá el código postal del cliente
Población	Campo de texto	Campo de texto obligatorio de 40 caracteres, contendrá el nombre de la población del cliente
Provincia	Campo de texto	Campo de texto obligatorio de 20 caracteres, contendrá el nombre de la provincia del cliente
Login	Campo de texto	Campo de texto obligatorio de 10 caracteres, contendrá el login elegido por el cliente
Password	Campo de texto	Campo de texto obligatorio de 10 caracteres, contendrá el password elegido por el cliente
Comprobar disponibilidad	Botón	Botón para comprobar la disponibilidad del login elegido por el cliente
Darse de baja como cliente	Botón	Botón para darse de baja como cliente. En este punto saldrá un mensaje de confirmación para el cliente. Una vez recibida la confirmación por parte del cliente será eliminado de la base de datos y volverá a la página de inicio de la aplicación con el perfil de usuario

3.5.3. Programa Mendel

La web dispone de una aplicación a través de la cual se puede conocer, de antemano, la posible descendencia de los diamantes de gould a partir de los fenotipos proporcionados por cualquier usuario de la aplicación. Esta aplicación es lo que hemos denominado Programa Mendel.

En la pantalla mostrada a continuación se puede ver la imagen de los dos progenitores y todas las combinaciones de colores

que podemos elegir para la cabeza, cuerpo y pecho de cada uno de ellos.

Una vez elegidos todos los colores de ambos progenitores, se pulsará el botón de calcular para conocer los porcentajes de probabilidades de todas las posibles descendencias.

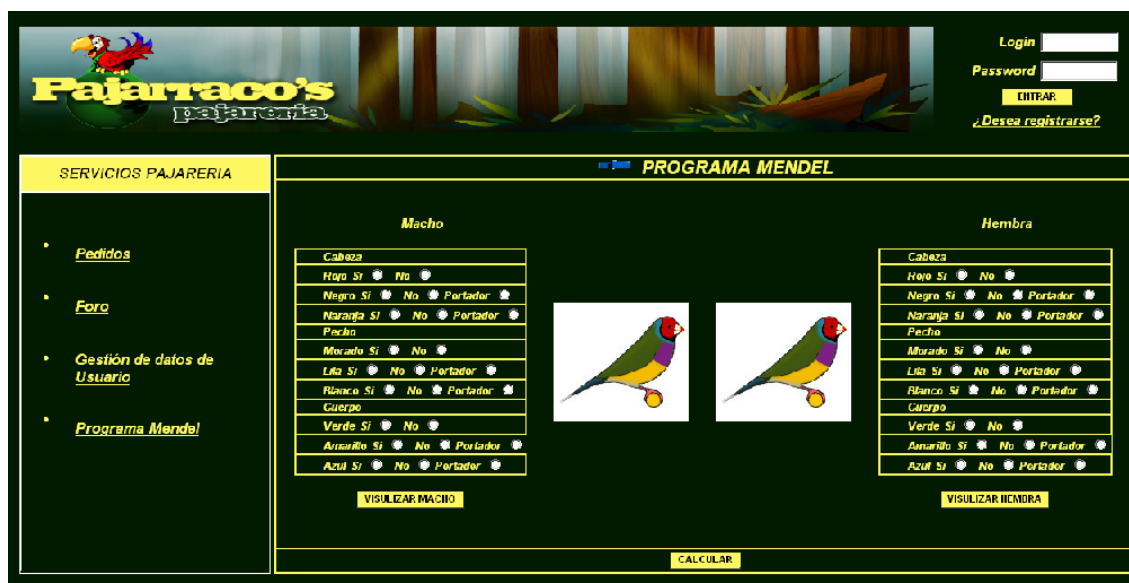


Figura 14. Programa Mendel

Campo	Tipo de Campo	Descripción
Sexo	Etiqueta	Cabecera para la introducción de los datos del macho. "Macho"
Cabeza	Etiqueta	Cabecera para la introducción del fenotipo de la cabeza. "Cabeza"
Color Rojo	Radio button	Campo de tipo check con dos posibles opciones, Si No. En caso de chequear Si se deshabilitará el campo check Si del color negro y naranja.
Color Negro	Radio button	Campo de tipo check con tres posibles opciones, Si No y Portador. En caso de chequear Si se deshabilitará el campo check Si del color rojo.
Color Naranja	Radio button	Campo de tipo check con tres posibles opciones, Si No y Portador.
Pecho	Etiqueta	Cabecera para la introducción del fenotipo del Pecho. "Pecho"
Color Morado	Radio button	Campo de tipo check con dos posibles opciones, Si No. En caso de chequear Si se deshabilitará el campo check Si del color blanco y lila
Color Blanco	Radio button	Campo de tipo check con tres posibles opciones, Si No y Portador. En caso de chequear Si se deshabilitará el campo check Si del color morado y lila además del radio portador del color lila.
Color Lila	Radio button	Campo de tipo check con tres posibles opciones, Si No y Portador. En caso de chequear Si se deshabilitará el campo check Si del color morado y blanco además del radio portador del color blanco.
Cuerpo	Etiqueta	Cabecera para la introducción del fenotipo del Cuerpo. "Cuerpo"
Color Verde	Radio button	Campo de tipo check con dos posibles opciones, Si No. En caso de chequear Si se deshabilitará el campo check Si del color amarillo y azul.
Color Amarillo	Radio button	Campo de tipo check con tres posibles opciones, Si No y Portador. En caso de chequear Si se deshabilitará el campo check Si del color verde y azul además del radio portador del color azul.
Color Azul	Radio button	Campo de tipo check con tres posibles opciones, Si No y Portador. En caso de chequear Si se deshabilitará el campo check Si del color morado y amarillo además del radio portador del color amarillo.

Campo	Tipo de Campo	Descripción
Visualizar macho	Botón	Al pulsar el botón visualizar se cargará la imagen del pájaro resultado de la consulta. Estará deshabilitado hasta completar la selección de los colores.
Ilustración del macho	Imagen	Imagen del pájaro macho resultado de los colores seleccionados. Todos los posibles estarán almacenados en la BBDD
Ilustración de la hembra	Imagen	Imagen del pájaro hembra resultado de los colores seleccionados. Todos los posibles estarán almacenados en la BBDD
Cabeza	Etiqueta	Cabecera para la introducción del fenotipo de la cabeza. "Cabeza"
Color Rojo	Radio button	Campo de tipo check con dos posibles opciones, Si No. En caso de chequear Si se deshabilitará el campo check Si del color negro y naranja.
Color Negro	Radio button	Campo de tipo check con tres posibles opciones, Si No y Portador. En caso de chequear Si se deshabilitará el campo check Si del color rojo.
Color Naranja	Radio button	Campo de tipo check con tres posibles opciones, Si No y Portador.
Pecho	Etiqueta	Cabecera para la introducción del fenotipo del Pecho. "Pecho"
Color Morado	Radio button	Campo de tipo check con dos posibles opciones, Si No. En caso de chequear Si se deshabilitará el campo check Si del color negro y naranja.
Color Blanco	Radio button	Campo de tipo check con dos posibles opciones, Si No. En caso de chequear Si se deshabilitará el campo check Si del color blanco y lila
Color Lila	Radio button	Campo de tipo check con tres posibles opciones, Si No y Portador. En caso de chequear Si se deshabilitará el campo check Si del color morado y lila además del radio portador del color lila.
Cuerpo	Etiqueta	Campo de tipo check con tres posibles opciones, Si No y Portador. En caso de chequear Si se deshabilitará el campo check Si del color morado y blanco además del radio portador del color blanco.
Color Verde	Radio button	Cabecera para la introducción del fenotipo del Cuerpo. "Cuerpo"
Color Amarillo	Radio button	Campo de tipo check con dos posibles opciones, Si No. En caso de chequear Si se deshabilitará el campo check Si del color amarillo y azul.
Color Azul	Radio button	Campo de tipo check con tres posibles opciones, Si No y Portador. En caso de chequear Si se deshabilitará el campo check Si del color verde y azul además del radio portador del color azul.
Visualizar hembra	Botón	Al pulsar el botón visualizar se cargará la imagen del pájaro resultado de la consulta. Estará deshabilitado hasta completar la selección de los colores.
Calcular	Botón	Tras pulsar el botón calcular se cargarán los Cromosomas resultado y la tabla resultado con los porcentajes resultantes de la consulta en la siguiente ventana.
Cromosomas Macho	Campo de texto	Combinación de cromosomas para el fenotipo del macho seleccionado.
Cromosomas hembra	Campo de texto	Combinación de cromosomas para el fenotipo de la hembra seleccionado.
Tabla de resultados	Listado	Listado con los resultados de la consulta en forma de porcentajes. Estará formado por la probabilidad en porcentaje, la parte del cuerpo (cabeza, pecho o cuerpo) y la combinación de cromosomas.
Gráficos	Botón	Tras pulsar este botón podremos visualizar el diagrama de barras y de queso de los resultados.

Tras pinchar en el botón calcular se mostrará un listado con los resultados en probabilidades correspondientes a la cabeza.

Figura 15. Resultados Cabeza

Campo	Tipo de Campo	Descripción
Fenotipo/Genotipo Macho	Campo de texto	Combinación de cromosomas para el genotipo y fenotipo del macho seleccionado.
Fenotipo/Genotipo hembra	Campo de texto	Combinación de cromosomas para el genotipo y fenotipo de la hembra seleccionado.
Tabla de resultados	Listado	Listado con los resultados de la consulta en forma de porcentajes. Estará formado por la probabilidad en porcentaje para la parte de la cabeza.
Continuar	Botón	Botón que permite al usuario continuar con los resultados.
Volver	Botón	Botón que permite al usuario volver a la pantalla de programa Mendel.

Si se pulsa el botón de continuar se mostrará otro listado correspondiente esta vez a los porcentajes correspondientes al pecho.

Figura 16. Resultados Pecho

Campo	Tipo de Campo	Descripción
Fenotipo/Genotipo Macho	Campo de texto	Combinación de cromosomas para el genotipo y fenotipo del macho seleccionado.
Fenotipo/Genotipo hembra	Campo de texto	Combinación de cromosomas para el genotipo y fenotipo de la hembra seleccionado.
Tabla de resultados	Listado	Listado con los resultados de la consulta en forma de porcentajes. Estará formado por la probabilidad en porcentaje para la parte del pecho.
Continuar	Botón	Botón que permite al usuario continuar con los resultados.
Volver	Botón	Botón que permite al usuario volver a la pantalla que visualiza los resultados de la cabeza.

Si continuamos obtendremos el listado de probabilidades correspondientes al cuerpo.

Bienvenido : mspostigo
[Cerrar Sesión](#)

SERVICIOS PAJARERIA

- [Pedidos](#)
- [Foro](#)
- [Datos de Usuario](#)
- [Programa Mendel](#)

Resultados Cuerpo

Fenotipo Macho : negroEPA,lila/azul
Genotipo Macho : ZzZrzoIiwB/ZG·ZG·bIbI

Fenotipo Hembra : salmos/blanco/azul
Genotipo Hembra : ZRWo·o·/wbwb/ZG·WbIbI

Porcentaje	Sexo	Cuerpo	Genotipo
50%	macho	verdeA.zu(1F)	Cuerpo - ZG·ZG·bIbI
50%	hembra	verdeA.zu(1F)	Cuerpo - ZG·WbIbI

[Volver](#) [continuar](#)

Figura 17. Resultado Pecho

Campo	Tipo de Campo	Descripción
Fenotipo/Genotipo Macho	Campo de texto	Combinación de cromosomas para el genotipo y fenotipo del macho seleccionado.
Fenotipo/Genotipo hembra	Campo de texto	Combinación de cromosomas para el genotipo y fenotipo de la hembra seleccionado.
Tabla de resultados	Listado	Listado con los resultados de la consulta en forma de porcentajes. Estará formado por la probabilidad en porcentaje para la parte de la cuerpo.
Continuar	Botón	Botón que permite al usuario continuar con los resultados.
Volver	Botón	Botón que permite al usuario volver a la pantalla de resultados del pecho.

Finalmente obtendremos un listado completo para obtener una visión global de todos los resultados mostrados en los listados anteriores.

Bienvenido : mspostigo
Cerrar Sesión

SERVICIOS PAJARERIA

- [Pedidos](#)
- [Foro](#)
- [Datos de Usuario](#)
- [Programa Mendel](#)

Resultados

Fenotipo Macho : negroEPA/lila/azul
Genotipo Macho : ZiZroo/lhw/ZG-ZG-bibl

Fenotipo Hembra : salmon/blanco/azul
Genotipo Hembra : ZRWo-o-lwbw/ZG-Wbibl

<input checked="" type="checkbox"/>	Porcentaje	Sexo	Cabeza	Pecho	Cuerpo	Genotipo
<input type="checkbox"/>	25%	macho	rojo	lila	verdeAzu(DF)	ZRZro-ollwbZG-ZG-bibl
<input type="checkbox"/>	25%	macho	rojo	blanco	verdeAzu(DF)	ZRZro-owbwbZG-ZG-bibl
<input type="checkbox"/>	25%	hembra	negro	lila	verdeAzu(DF)	ZRWo-ollwbZG-Wbibl

Volver Visualizar

Figura 18. Resultados

Campo	Tipo de Campo	Descripción
Fenotipo/Genotipo Macho	Campo de texto	Combinación de cromosomas para el genotipo y fenotipo del macho seleccionado.
Fenotipo/Genotipo hembra	Campo de texto	Combinación de cromosomas para el genotipo y fenotipo de la hembra seleccionado.
Tabla de resultados	Listado	Listado con los resultados de la consulta en forma de porcentajes. Estará formado por la probabilidad en porcentaje para la parte de la cabeza , pecho y cuerpo, y la combinación de cromosomas.
Continuar	Botón	Botón que permite al usuario continuar con los resultados.
Volver	Botón	Botón que permite al usuario volver a la pantalla de los resultados referentes al carácter del pecho.

Además desde este listado se puede ver una imagen de cada una de las posibles combinaciones de crías que se pueden obtener a partir de los fenotipos introducidos inicialmente.



Figura 19. Visualizar Cría

Campo	Tipo de Campo	Descripción
Sexo	Campo de texto	Sexo de la cría seleccionada.
Genotipo	Campo de texto	Información genética, conjunto de cromosomas que definen los caracteres del color de la cabeza, pecho y cuerpo.
Fenotipo	Listado	Manifestación visible del genotipo, es decir, los colores relativos a cabeza, pecho y cuerpo.
Imagen	Imagen	Imagen de la cría.
Continuar	Botón	Botón que permite al usuario continuar con los resultados.
Volver	Botón	Botón que permite al usuario volver a la pantalla de programa Mendel.

3.6. Menú administrador

El menú de administrador sólo está disponible para aquellos usuarios registrados en la web con perfil de administrador. Este menú será accesible tras logarse en la web. En el podrán accederse a las siguientes funcionalidades:

- 1.- Gestión de clientes
- 2.-Gestión de proveedores
- 3.- Gestión de productos
- 4.- Gestión de envíos a clientes
- 5.- Gestión de pedidos
- 6.- Consulta de stock
- 7.- Gestión foro
- 8.- Generar catálogo
- 9.- Visualización de gráficos

3.6.1. Gestión de clientes

A través de la gestión de clientes el administrador podrá realizar búsquedas de los pedidos de los clientes a través de su código de cliente, nombre o estado de alguno de sus pedidos.

A continuación vamos a ir describiendo los distintos formularios para la gestión de los clientes.



Figura 20. Búsqueda clientes

Campo	Tipo de Campo	Descripción
Código de cliente	Campo de texto	Campo editable y opcional en el que el administrador introducirá un código de cliente a buscar. Admite 10 dígitos.
Nombre del cliente	Campo de texto	Campo editable y opcional en el que el administrador introducirá un nombre del cliente a buscar. Admite 20 letras.
Primer Apellido	Campo de texto	Campo editable y opcional en el que el administrador introducirá un primer apellido del cliente a buscar. Admite 20 letras.
Segundo Apellido	Campo de texto	Campo editable y opcional en el que el administrador introducirá un segundo apellido del cliente a buscar. Admite 20 letras.
DNI	Campo de texto	Campo editable y opcional que permitirá buscar a un cliente por DNI. Admite 9 caracteres.
Estado cliente	Combo desplegable	Lista de posibles estados en los que puede encontrarse un pedido de un determinado cliente (Pendiente, realizado, pagado). Inicialmente no se cargará con ningún valor. Campo opcional.
Fecha Inicio	Calendario	Calendario para seleccionar la fecha de inicio para la búsqueda de pedido del cliente en un rango de fechas
Fecha Fin	Calendario	Calendario para seleccionar la fecha de fin para la búsqueda de pedido del cliente en un rango de fechas
Código Cliente	Link	Enlace con el que el administrador podrá acceder a todos los datos del cliente correspondiente a ese código.
Nombre Cliente	Etiqueta	Campo protegido que mostrará el nombre del cliente.
Apellidos	Etiqueta	Campo protegido que mostrará los apellidos del cliente.
Estado pedido	Etiqueta	Campo protegido que indica el estado del pedido de un determinado

Campo	Tipo de Campo	Descripción
		cliente.
Fecha pedido	Etiqueta	Campo protegido que muestra la fecha en la que se realizó el pedido.
Buscar	Botón	Botón habilitado que permite realizar búsquedas en función de los filtros. Si no se introduce ningún filtro de búsqueda se visualizará un listado con todos los clientes dados de alta hasta el momento.
Borrar	Botón	Botón para reiniciar los campos de búsqueda del cliente

Tras realizar la búsqueda, si el usuario selecciona alguno de los pedidos, la aplicación mostrará un formulario con todos los datos de contacto del cliente que ha realizado el pedido y el estado del mismo. Desde este formulario se podrá modificar el estado del pedido o incluso dar de baja a un cliente del sistema.

The screenshot shows a web application interface for 'Pajaraco's pajareria'. The main content area is titled 'DATOS CLIENTE' and contains a form with the following sections:

- 1.-Datos Personales:** Includes fields for 'Nombre' (with sub-fields for 'Primer Apellido' and 'Segundo Apellido'), 'Tipo de documento', 'Nº Documento', 'Fecha de nacimiento', and 'Nacionalidad'.
- 2.-Datos de Contacto:** Includes fields for 'Correo Electrónico' and 'Teléfono de Contacto'.
- 3.-Lugar de Residencia:** Includes a radio button for 'Tipo de cliente' (Particular or Empresa), and fields for 'Calle', 'Número', 'Piso/Letra', 'Bloque', 'Población', 'Código Postal', 'Provincia', and 'País'. There is also a dropdown for 'Estado Pedido' set to 'REALIZADO'.

At the bottom of the form are buttons for 'VOLVER', 'MODIFICAR', and 'DAR DE BAJA'. The left sidebar lists various services under 'SERVICIOS PAJARERIA', and the top right corner shows a user login 'Bienvenido : admin' and a 'Cerrar Sesión' link.

Figura 21. Detalle Cliente y Estado Pedido

Campo	Tipo de Campo	Descripción
Nombre Cliente	Etiqueta	Campo protegido con el nombre del cliente.
Primer Apellido	Etiqueta	Campo protegido con el primer apellido del cliente.
Segundo Apellido	Etiqueta	Campo protegido con el segundo apellido del cliente.
Tipo de documento	Etiqueta	Campo protegido que indica el tipo de documento del cliente.
Número de documento	Etiqueta	Campo protegido con el número de documento del cliente.
Fecha de Nacimiento	Etiqueta	Campo que muestra la fecha de nacimiento del cliente.
Tipo cliente	Radio	Campo protegido en el que se muestran dos posibilidades: particular o empresa, si el cliente es particular aparecerá chequeado el radio situado al lado de esta etiqueta y sino el situado al lado de empresa. Sólo puede estar seleccionado uno de los dos.
Calle	Etiqueta	Campo protegido que informa de la calle en la que reside actualmente el cliente.
Número	Etiqueta	Campo protegido que informa del número en el que reside actualmente el cliente.

Campo	Tipo de Campo	Descripción
Piso/Letra	Listas desplegables	Campo protegido que informa del piso y la letra del domicilio del cliente.
Bloque	Etiqueta	Campo protegido y opcional en el que se informará del bloque en el que reside el cliente.
Código Postal	Campo de texto	Campo protegido con el código postal de la ciudad en la que reside el cliente.
Localidad	Campo de texto	Campo protegido que muestra la localidad donde reside el cliente.
Provincia	Etiqueta	Campo protegido que informa de la provincia donde se sitúa la localidad.
País	Etiqueta	Campo protegido que muestra el país del cliente.
Teléfono de Contacto	Etiqueta	Campo protegido en el que se informará un teléfono de contacto con el cliente.
Correo Electrónico	Etiqueta	Campo protegido y opcional para poder ponerse en contacto con el cliente vía e-mail.
Estado del pedido	Lista desplegable	Lista desplegable que cargará por defecto el valor del estado asociado al pedido y al usuario que el administrador haya seleccionado. Podrá modificarse su valor a cualquiera de las otras opciones que se cargarán en el combo.
Volver	Botón	Botón habilitado que permite regresar a la pantalla inicial (Búsqueda de clientes).
Modificar	Botón	Botón habilitado que permite guardar los cambios en el estado del pedido del cliente.
Dar de Baja	Botón	Botón habilitado que permite dar de baja de la base de datos al cliente seleccionado.

3.6.2. Gestión de proveedores

La gestión de proveedores permite realizar la gestión de los datos de los proveedores asociados con la empresa. Se podrá dar de alta, baja o incluso modificar o consultar proveedores.

Bienvenido : admin
Cerrar Sesión

SERVICIOS PAJARERIA

- Gestion de clientes
- Gestion de Proveedores
- Gestion de Productos
- Gestion de pedidos
- Gestion de Centros
- Consulta de stock
- Gestion Foro
- Generar catalogo
- Visualizacion de graficos

GESTION DE PROVEEDORES

Proveedor

Primer Apellido

Segundo Apellido

Nº Documento

Apellido

<input type="checkbox"/>	Código	Nombre	Empresa	Productos
<input type="checkbox"/>	1	Maria	PARTICULAR	Productos
<input type="checkbox"/>	2	Pedro	PARTICULAR	Productos

Figura 22. Gestión de proveedores

Campo	Tipo de Campo	Descripción
Código	Campo de texto	Campo de texto opcional tendrá 7 caracteres, contendrá el código del proveedor.
Nombre	Campo de texto	Campo de texto opcional de 20 caracteres, contendrá el nombre de la empresa o nombre del proveedor en caso de que se tratase de un particular.
Número de documento	Campo de texto	Campo de texto opcional de 9 caracteres, contendrá el número de documento del proveedor.
Primer Apellido	Campo de texto	Campo de texto opcional de 20 caracteres, contendrá el primer apellido del proveedor.
Segundo Apellido	Campo de texto	Campo de texto opcional de 20 caracteres, contendrá el segundo apellido del proveedor.
Buscar	Botón	Botón para hacer la búsqueda de proveedores. Si no se han introducido criterios de búsqueda mostrará todos los proveedores.
Borrar	Botón	Botón para borrar los datos introducidos en el formulario de búsqueda.

Una vez realiza cualquier búsqueda, en caso de que el sistema encuentre resultados, mostrará un listado con el detalle de todos los proveedores que se ajusten a los filtros de búsqueda.

Campo	Tipo de Campo	Descripción
Listado de búsqueda	Listado	Contendrá el listado resultado de la búsqueda. Estará formado por un campo check para seleccionar un proveedor, el código de proveedor, el nombre, un campo empresa para saber si se trata de una empresa o un particular y un campo productos. El campo producto será un link que abrirá el popup con la lista de productos de un proveedor.
Añadir	Botón	Botón que abre la pantalla con el formulario para añadir un nuevo proveedor.
Borrar	Botón	Botón para borrar el o los proveedores seleccionados. Una vez pulsado el botón saldrá un mensaje para pedir confirmación y una confirmada la acción se borrará de la base de datos.
Modificar	Botón	Botón para modificar los datos de un proveedor. Este botón abrirá una pantalla igual a la del formulario de alta pero con los datos del proveedor.
Consulta	Botón	Botón para consultar los datos de un proveedor. Abrirá una pantalla con los datos del proveedor en modo consulta.

3.6.3. Gestión de productos

Cualquier administrador podrá controlar y gestionar los productos disponibles en el catálogo. Esta opción le permitirá localizar de manera rápida los productos de dicho catálogo mediante su código, nombre o número de unidades disponibles en el almacén. Una vez localizados podrá modificar sus datos o realizar el pedido de cualquiera de ellos. Por último también podrá añadir nuevos productos al catálogo disponible.

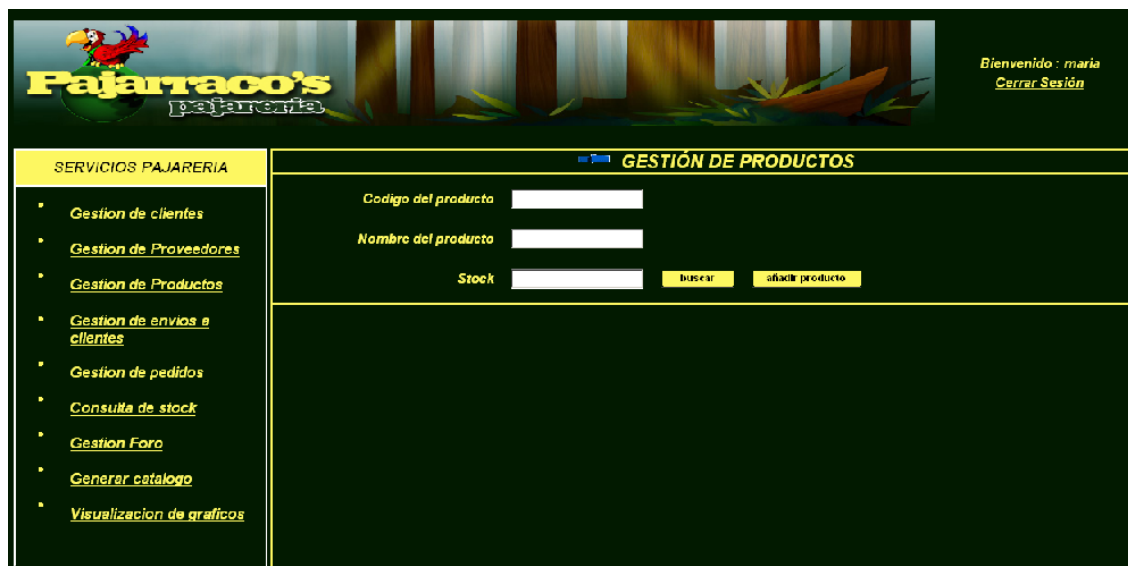


Figura 23. Gestión de productos

Campo	Tipo de Campo	Descripción
Código del producto	Campo de texto	Campo editable y opcional en el que el administrador podrá introducir el código del producto a buscar. Admite 7 caracteres.
Nombre del producto	Campo de texto	Campo editable y opcional en el que el administrador podrá introducir el nombre del producto a buscar. Admite 20 caracteres.
Stock	Campo de texto	Campo editable y opcional que sólo admite 6 dígitos y en que el administrador introducirá una cantidad. Al buscar se recuperarán los productos cuyo stock en el almacén sea menor o igual a la cantidad introducida. De esta forma el administrador sabrá los productos de los cuales debe realizar un pedido.
Listado de productos	Listado	El administrador podrá ver el listado resultante a los parámetros de búsqueda que haya proporcionado anteriormente. El listado estará formado por código, nombre, fecha de alta, precio y stock.
Código del producto	Etiqueta	Campo protegido con donde el administrador podrá ver el código del producto.
Nombre del producto	Etiqueta	Enlace con el que el administrador podrá acceder a la pantalla de modificación de ese producto.
Fecha de alta	Etiqueta	Campo protegido que mostrará la fecha en la que se dio de alta ese producto.
Precio	Etiqueta	Campo protegido que indica el precio actual del producto
Stock	Etiqueta	Campo protegido que muestra la cantidad de unidades del producto existentes en el almacén en ese momento.
Buscar	Botón	Botón habilitado que permite al administrador realizar búsquedas en función de los filtros rellenos. Si no se introduce ningún filtro de búsqueda se visualizará un listado con todos los productos dados de alta

Campo	Tipo de Campo	Descripción
		hasta el momento.
Añadir Producto	Botón	Botón habilitado que permite acceder al formulario para dar de alta un nuevo producto en la base de datos.
Realizar pedido	Botón	Botón habilitado en el caso de que se haya seleccionado un registro del listado de productos. Permite acceder a la pantalla de realización de pedidos de un producto.

A continuación se muestra el formulario para dar de alta un producto en el catálogo.

Figura 24. Alta productos

Campo	Tipo de Campo	Descripción
Nombre del producto	Campo de texto	Campo editable obligatorio en el que se podrá introducir el nombre del nuevo producto. Admite dígitos y letras y tiene una longitud máxima de 20 caracteres.
Familia	lista desplegable	Lista que muestra las distintas familias en las que se distribuyen todos los productos existentes (Alimentación, Especie, Accesorios, Medicamentos...). No se informa ningún valor por defecto. Campo obligatorio.
Fecha de alta	Etiqueta	Campo protegido que mostrará la fecha actual en la que se dio de alta ese producto. Se cargará por defecto con la fecha del día.
Precio	Campo de texto	Campo editable y obligatorio que indica el precio unitario de venta del producto a los clientes. Admite 6 dígitos (3 decimales y 3 enteros).
Stock	Campo de texto	Campo editable y obligatorio en el que se muestra el número de unidades del producto que se van a dar de alta en el almacén del producto. Tendrá 6 dígitos.
Características	Campo de texto	Campo editable y obligatorio en el que el administrador podrá introducir una pequeña descripción de las características más importantes del producto. Tiene un tamaño máximo de 300 caracteres.
Volver	Botón	Botón habilitado que permite regresar a la pantalla inicial (Búsqueda de productos).
Guardar	Botón	Botón habilitado que permite guardar los datos del nuevo producto en la base de datos si se ha rellenado el formulario correctamente. En este caso se mostrará un mensaje de éxito. En caso contrario se mostrará un

Campo	Tipo de Campo	Descripción
		mensaje de error.

Respecto al formulario de modificación de producto, cabe destacar que a este formulario se podrá acceder mediante el link que aparece en el listado de productos de la pantalla de gestión de productos. El formulario tendrá el siguiente aspecto:

The screenshot shows a web interface for 'Pajarraco's'. At the top right, there are login fields for 'Login' and 'Password', a 'CITAR' button, and a link '¿Desea registrarse?'. The main content area is titled 'Modificación de Productos'. It contains the following fields:

- Nombre del producto:** A text input field containing 'articulo1'.
- Familia / Tipo:** A dropdown menu with 'diamante' selected.
- Fecha de Alta:** A date input field showing '12/08/2008'.
- Precio:** A text input field with '14'.
- Stock:** A text input field with '5'.
- Características:** A large text area containing the text 'estas son las caracteristicas del producto 00'.

At the bottom of the form are two buttons: 'Volver' and 'Guardar'. On the left side, there is a sidebar menu under 'SERVICIOS PAJARERIA' with options like 'Gestion de clientes', 'Gestion de Proveedores', 'Gestion de Productos', etc.

Figura 25. Modificación productos

Campo	Tipo de Campo	Descripción
Nombre del producto	Etiqueta	Campo editable que muestra el nombre del producto seleccionado en el listado.
Familia	Combo desplegable	Campo editable con el valor seleccionado de la familia a la que pertenece el producto.
Fecha de alta	Etiqueta	Campo protegido que mostrará la fecha en la que se dio de alta el producto.
Precio	Campo de texto	Campo editable y obligatorio que indica el precio unitario de venta del producto a los clientes. Admite 6 dígitos (3 enteros y 3 decimales).
Stock	Etiqueta	Campo editable que se muestra el número de unidades del producto existentes en el almacén.
Características	Etiqueta	Campo editable que da una pequeña descripción de las características más importantes del producto.
Volver	Botón	Botón habilitado que permite regresar a la pantalla inicial (Búsqueda de productos).
Guardar	Botón	Botón habilitado que permite guardar los datos modificados del producto.

Cuando el stock de alguno de los productos no sea suficiente, cualquier administrador tendrá la posibilidad de realizar pedidos de los productos bajos de stock.. Para poder acceder a este formulario bastará con pulsar el botón de realizar pedidos de la pantalla de gestión de productos.

The screenshot shows the 'PEDIDOS' section of the Pajaraco's Pajarería application. On the left is a sidebar menu with options: 'Gestion de clientes', 'Gestion de Proveedores', 'Gestion de Productos', 'Gestion de envios a clientes', 'Gestion de pedidos', 'Consulta de stock', 'Gestion Foro', 'Generar catalogo', and 'Visualizacion de graficos'. The main content area has a header 'PEDIDOS' and search fields for 'Productos', 'Proveedor', and 'Unidades'. Below these is a table with columns: 'Producto', 'Proveedor', 'Unidades', 'Precio', 'Importe', and 'Fecha'. The table contains two rows of data. At the bottom of the table are buttons for 'Añadir', 'Eliminar Producto', 'Volver', and 'generar factura'. The top right of the page shows a user greeting: 'Bienvenido : maria' and a 'Cerrar Sesión' link.

Figura 26. Pedidos

Campo	Tipo de Campo	Descripción
Producto	Lupa	Campo para hacer búsquedas de productos. Se puede introducir el código del producto, nombre o stock para realizar la búsqueda. En caso de que el producto no exista se mostrará un mensaje de error, si se ha encontrado se cargará el código y la descripción en los campos correspondientes.
Proveedor	Lupa	Campo que contiene la referencia y la descripción de un proveedor. Se puede introducir el código del proveedor para realizar la búsqueda. En caso de que el proveedor no exista se mostrará un mensaje de error, si se ha encontrado se cargará el código y la descripción en los campos correspondientes. Otra posibilidad de buscar el proveedor es pulsando sobre la imagen de la lupa.
Número de unidades	Campo de texto	Campo editable y obligatorio en el que el administrador pondrá el número de unidades del producto que desea adquirir. Campo numérico
Pedido	Listado	Listado con los pedidos introducidos por el usuario. Contiene un campos de tipo check para seleccionar los productos que el administrador quiera eliminar, código de producto, código de proveedor, número de unidades, importe, la fecha y el importe total del pedido
Añadir	Botón	Botón habilitado que permite añadir los productos elegidos por el usuario a la lista
Eliminar	Botón	Botón habilitado que permite al usuario eliminar los productos de la lista que desee el usuario
Volver	Botón	Botón habilitado que permite regresar a la pantalla inicial (Búsqueda de productos).
Generar Factura	Botón	Botón habilitado que permite realizar un pedido de un producto a un determinado proveedor y guardar un registro para ese pedido en base de datos.

Tras haber realizado el pedido el administrador podrá visualizar el desglose del pedido realizado.

Bienvenido : maria
Cerrar Sesión

SERVICIOS PAJARERIA

- Gestion de clientes
- Gestion de Proveedores
- Gestion de Productos
- Gestion de envios a clientes
- Gestion de pedidos
- Consulta de stock
- Gestion Foro
- Generar catalogo
- Visualizacion de graficos

FACTURA

Producto	Proveedor	N unidades	Precio	Importe	Fecha
001	001	10	12.25	122.5	02/08/2008
002	002	20	7.45	149.0	02/08/2008
Importe total :				271.5	

Volver Aceptar

Figura 27. Factura

Campo	Tipo de Campo	Descripción
Pedido	Listado	Listado con los pedidos introducidos por el usuario. Contiene los campos código de producto, código de proveedor, número de unidades, importe, la fecha y el importe total del pedido
Volver	Botón	Botón habilitado que permite regresar a la pantalla inicial (Búsqueda de productos).
Aceptar	Botón	Botón habilitado que permite aceptar la factura asociada al pedido

Además la web proporcionará la posibilidad de imprimir el desglose de la factura del pedido.

Bienvenido : maria
Cerrar Sesión

SERVICIOS PAJARERIA

- Gestion de clientes
- Gestion de Proveedores
- Gestion de Productos
- Gestion de envios a clientes
- Gestion de pedidos
- Consulta de stock
- Gestion Foro
- Generar catalogo
- Visualizacion de graficos

FACTURA

Producto	Proveedor	N unidades	Precio	Importe	Fecha
001	001	10	12.25	122.5	02/08/2008
002	002	20	7.45	149.0	02/08/2008
Importe total :				271.5	

Volver Imprimir

Figura 28. Factura

Campo	Tipo de Campo	Descripción
Pedido	Listado	Listado con los pedidos introducidos por el usuario. Contiene los campos código de producto, código de proveedor, número de unidades, importe, la fecha y el importe total del pedido

Campo	Tipo de Campo	Descripción
Volver	Botón	Botón habilitado que permite regresar a la pantalla inicial (Búsqueda de productos).
Aceptar	Botón	Botón habilitado que permite aceptar la factura asociada al pedido

3.6.4. Gestión de pedidos

A través de la opción de gestión de pedidos, cualquier administrador del sistema podrá gestionar tanto los pedidos realizados por clientes a través de la web, como los pedidos que la empresa ha realizado a distintos proveedores.



Figura 29. Gestión de pedidos

Campo	Tipo de Campo	Descripción
Opciones	Radio	El administrador podrá elegir mediante un radio button la opción de pedidos de clientes o a proveedores

Una vez que se ha elegido los pedidos que se quieren gestionar (pedidos de clientes o pedidos a proveedores), se presentará un formularios diferente en función de la opción elegida.

Para los pedidos realizados a clientes se presentará el siguiente formulario:

Bienvenido : maria
Cerrar Sesión

SERVICIOS PAJARERIA

- Gestion de clientes
- Gestion de Proveedores
- Gestion de Productos
- Gestion de envios a clientes
- Gestion de pedidos
- Consulta de stock
- Gestion Foro
- Generar catalogo
- Visualizacion de graficos

GESTION DE PEDIDOS A CLIENTES

Codigo del pedido Estado

Cliente Tipo de Producto

Fecha Inicio Fecha Fin

Buscar

<input type="checkbox"/>	Codigo del cliente	Codigo del producto	Estado	Precio Unitario
<input type="checkbox"/>	CJ 001	0345	Pte. envio	5.43
<input type="checkbox"/>	CJ 002	0356	Enviado	5.43
<input type="checkbox"/>	CJ 003	0374	Recibido	5.43
<input type="checkbox"/>	CJ 004	0348	Cerrado	5.43

Guardar

Volver

Figura 30. Pedidos clientes

Campo	Tipo de Campo	Descripción
Código pedido	Campo de texto y lupa	Contendrá un campo de texto para introducir el código del pedido que será de 7 caracteres. Si no se conoce el código se pulsará la lupa que abrirá un popup para poder realizar la búsqueda del pedido.
Estado	Lista desplegable	Lista desplegable con los distintos tipos de estado de un pedido. Este campo será opcional y su opción por defecto estará vacía.
Código cliente	Campo de texto y lupa	Contendrá un campo de texto para introducir el código del cliente que será de 7 caracteres y/o el nombre que será de 60 caracteres. También podrá realizar una búsqueda del cliente a través de la lupa que abrirá el popup de búsqueda de clientes.
Tipo de Producto	Campo de texto y lupa	Contendrá un campo de texto para introducir el código del producto que será de 7 caracteres y o el nombre que será de 20 caracteres. También podrá realizar una búsqueda del producto a través de la lupa que abrirá el popup de búsqueda de productos.
Fecha Inicio	Calendario	Calendario para fijar el rango de fechas para la búsqueda de un pedido. Ese campo es opcional.
Fecha Fin	Calendario	Calendario para fijar el rango de fechas para la búsqueda de un pedido. Ese campo es opcional.
Listado de pedidos	Listado	Será la lista resultante de la búsqueda. Contendrá un campo de texto protegido con el código del cliente, código de Producto también protegido, estado del pedido será una lista desplegable editable y por último un campo de texto protegido con el precio.
Buscar	Botón	Tras pulsar el botón buscar saldrá el listado de pedidos resultado de los filtros de búsqueda introducidos anteriormente.
Volver	Botón	El administrador podrá volver a la pantalla de gestión de pedidos.

Para los pedidos realizados a proveedores el sistema nos mostrará la siguiente pantalla:

Figura 31. Pedidos a proveedores

Campo	Tipo de Campo	Descripción
Código pedido	Campo de texto y lupa	Contendrá un campo de texto para introducir el código del pedido que será de 7 caracteres. Si no se conoce el código se pulsará la lupa que abrirá un popup para poder realizar la búsqueda del pedido.
Estado	Lista desplegable	Lista desplegable con los distintos tipos de estado de un pedido. Este campo será opcional y su opción por defecto estará vacía.
Familia	Campo de texto y lupa	Contendrá un campo de texto para introducir el código del producto que será de 7 caracteres. Si no se conoce el código se pulsará la lupa que abrirá un popup para poder realizar la búsqueda del producto.
Proveedor	Campo de texto y lupa	Contendrá un campo de texto para introducir el código del proveedor que será de 7 caracteres y otro con el nombre del proveedor que será de 20 caracteres. Además tendrá una lupa que abrirá un popup para hacer búsquedas.
Fecha Inicio	Calendario	Calendario para fijar el rango de fechas para la búsqueda de un pedido. Ese campo es opcional.
Fecha Fin	Calendario	Calendario para fijar el rango de fechas para la búsqueda de un pedido. Ese campo es opcional.
Listado de pedidos	Listado	Será la lista resultante de la búsqueda. Contendrá un campo de texto protegido con el código de pedido, una lista desplegable con el estado que se podrá modificar, un campo de texto protegido con el tipo de producto y por último un campo de texto también protegido con el precio.
Buscar	Botón	Tras pulsar el botón buscar saldrá el listado de pedidos resultado de los filtros de búsqueda introducidos anteriormente.
Guardar	Botón	Botón que permitirá al administrador guardar los cambios introducidos.
Volver	Botón	Botón que permitirá al administrador volver a la pantalla de gestión de envíos.

3.6.5. Consulta de stock

A través de la funcionalidad de consulta de stock, los administradores del sistema podrán conocer en todo momento el stock disponibles de cada uno de los productos del catálogo. Así sabrán en todo momento si es necesario realizar el pedido de alguno de ellos.



Figura 32. Consultad de stock

Campo	Tipo de Campo	Descripción
Tienda	Lupa	Campo que contiene el código que será de 7 caracteres y la provincia que será de 25 caracteres. El administrador podrá introducir directamente el código de la tienda. En caso de que no exista el código se mostrará un mensaje de error, si el código es correcto, se cargará directamente la descripción de la tienda. También se podrá elegir un centro accediendo a la pantalla de selección de centros.
Producto	Lupa	Campo obligatorio que contiene el código que será de 7 caracteres y la descripción del producto que será 25 caracteres, asociado a la tienda seleccionada en caso de existir. El administrador podrá introducir directamente el código del producto. En caso de que no exista el código se mostrará un mensaje de error, si el código es correcto, se cargará directamente la descripción del producto. También se podrá elegir un producto accediendo a la pantalla de selección de producto.
Código del producto	Etiqueta	Campo protegido que muestra el código del producto
Nombre del producto	Etiqueta	Campo protegido que muestra el nombre del producto.
Stock	Etiqueta	Campo protegido que muestra el número de unidades existentes en el almacén de ese producto concreto.
Volver	Botón	Botón habilitado que permite regresar a la pantalla inicial (Búsqueda de productos).
Buscar	Botón	Botón habilitado que realizar búsquedas en función de los filtros de búsqueda que se hayan rellenado. En caso de no existir ninguno se muestra el stock de todos los productos existentes en el almacén.

3.6.6. Catálogo de ofertas

La web dispone de un catálogo de promociones u ofertas que irá cambiando en función de la demanda, stock y otros factores. A través de esta opción del menú de administrador se podrá gestionar todos los cambios que se necesiten realizar en el catálogo de ofertas de la web.



Figura 33. Catálogo de ofertas

Campo	Tipo de Campo	Descripción
Sel	Checkbox	Campo que permite elegir una oferta o varias.
Nombre del producto	Link	Enlace que muestra el nombre del producto en oferta y permite acceder a la pantalla de modificación de la oferta.
Características del producto	Etiqueta	Campo protegido que mostrará una breve descripción de las características más importantes del producto.
Precio	Etiqueta	Campo protegido que muestra el precio unitario del producto.
Imagen del producto	Imagen	Imagen del producto.
Añadir oferta	Botón	Botón habilitado que permite al administrador acceder a la pantalla en la cual podrá añadir nuevas ofertas.
Eliminar oferta	Botón	Botón habilitado en caso de que haya algún registro seleccionado que permite eliminar del catálogo ofertas.

Cuando se selecciona la opción de menú de añadir oferta se mostrará el siguiente formulario para incluir una nueva oferta en el catálogo.

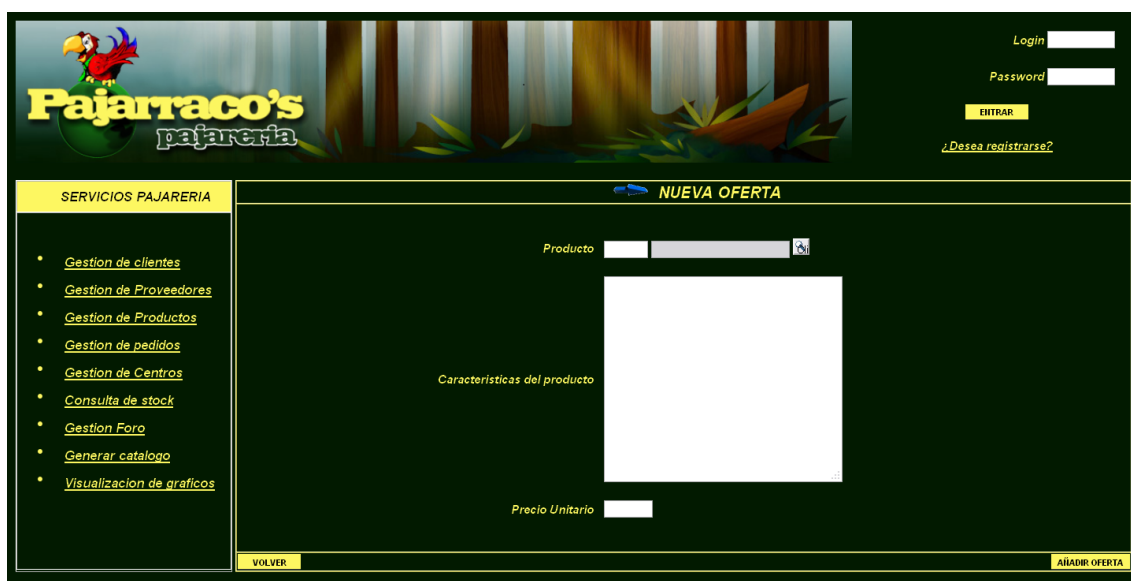


Figura 34. Nueva oferta

Campo	Tipo de Campo	Descripción
Producto	Lupa	Campo editable y obligatorio que tendrá el código que será de 7 caracteres y la descripción del producto que será de 25 caracteres (este último deshabilitado). El administrador podrá introducir directamente el código del producto y en caso de que exista, se cargará automáticamente en el otro campo su descripción. Si no existe se mostrará un mensaje de error. También podrá seleccionar el producto pinchando en la lupa la cual abrirá el popup de Productos.
Nombre del producto	Etiqueta	Campo protegido en el que se visualizará el nombre del producto en oferta.
Precio	Campo de texto	Campo editable y obligatorio que permitirá sólo dígitos. Admite 3 enteros y 3 decimales.
Volver	Botón	Botón habilitado que permite regresar a la pantalla inicial (Búsqueda de productos).
Añadir oferta	Botón	Botón habilitado que permite cargar el nuevo producto como oferta en el listado del catálogo (pantalla de catálogo de ofertas).

Desde la pantalla principal se puede acceder al formulario de modificación de una oferta pinchando en el enlace del nombre del producto. El formulario de modificación de oferta se mostrará pre rellenado con los datos de la oferta seleccionada. El administrador podrá modificar el precio de la oferta.

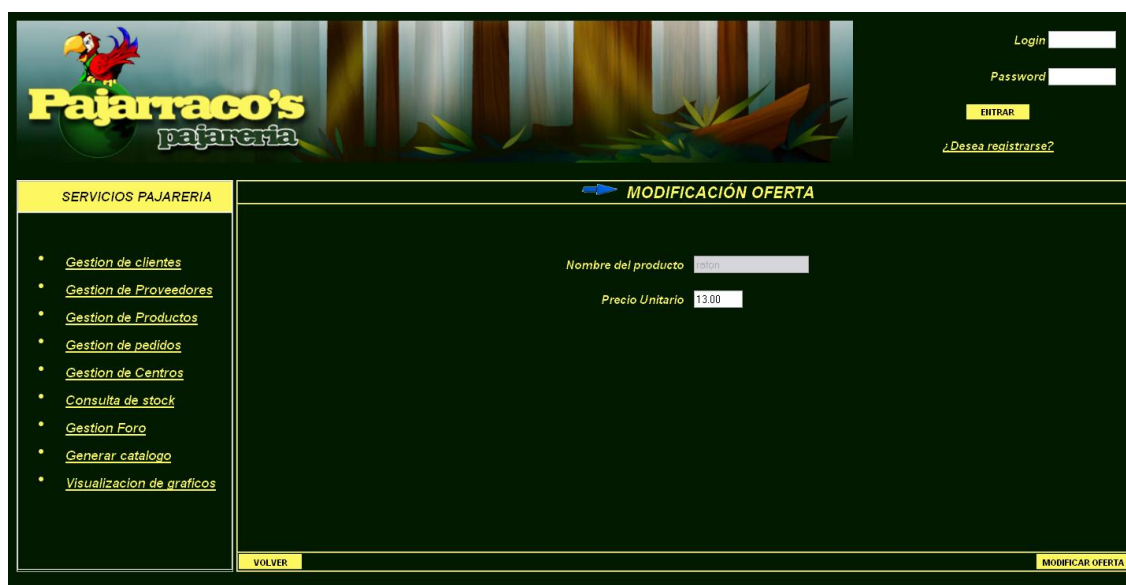


Figura 35. Modificación oferta

Campo	Tipo de Campo	Descripción
Nombre del producto	Etiqueta	Campo protegido en el que se visualizará el nombre del producto en oferta.
Precio	Campo de texto	Campo editable y obligatorio que permitirá sólo dígitos. Admite 3 enteros y 3 decimales.
Volver	Botón	Botón habilitado que permite regresar a la pantalla inicial (Búsqueda de productos).
Añadir oferta	Botón	Botón habilitado que permite cargar el producto modificado como oferta en el listado del catálogo (pantalla de catálogo de ofertas).

3.6.7. Visualización gráfico

Opción de menú a través de la cual se podrán obtener gráficos de ventas por productos o centros.

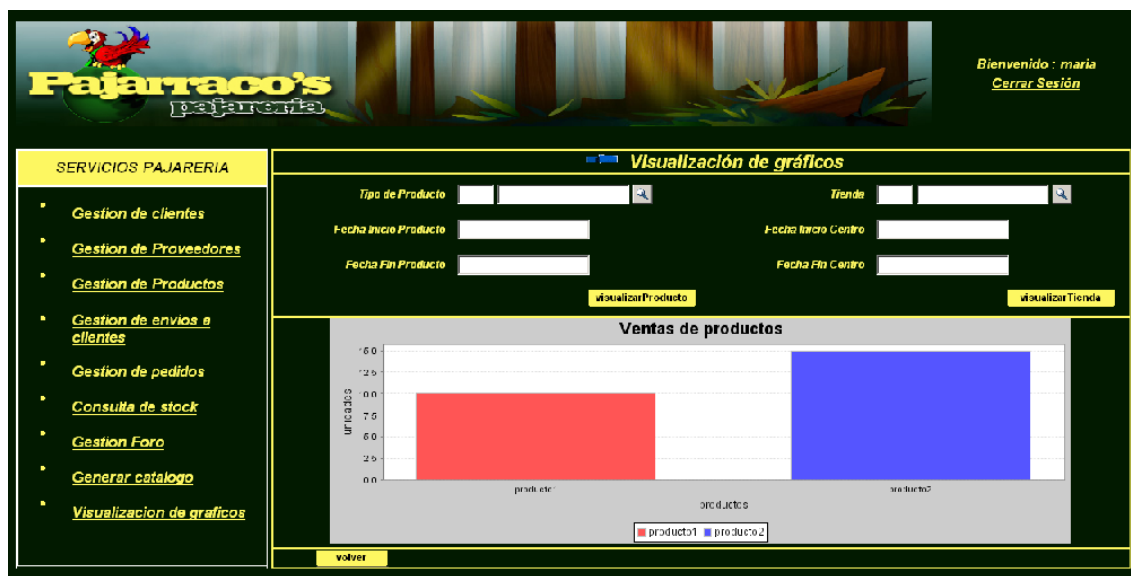


Figura 36. Visualización de gráficos

Campo	Tipo de Campo	Descripción
Código del producto	Lupa	Campo editable y obligatorio (en caso de que se visualice ese gráfico) que tendrá el código que será de 7 caracteres y la descripción del producto que será de 25 caracteres (este último deshabilitado). El administrador podrá introducir directamente el código del producto y en caso de que exista, se cargará automáticamente en el otro campo su descripción. Si no existe se mostrará un mensaje de error. También podrá seleccionar el producto pinchando en la lupa la cual abrirá el popup de Productos.
Fecha Inicio Producto	Calendario	Campo que se carga con la fecha que se seleccione en el calendario.
Fecha Fin Producto	Calendario	Campo que se carga con la fecha que se seleccione en el calendario.
Centro/Tienda	Lupa	Campo que contiene el código que será de 7 caracteres y la descripción de la tienda que será de 25 caracteres. El administrador podrá introducir directamente el código de la tienda. En caso de que no exista el código se mostrará un mensaje de error, si el código es correcto, se cargará directamente la descripción de la tienda. También se podrá elegir un centro accediendo a la pantalla de selección de centros.
Fecha Inicio Centro	Calendario	Campo que se carga con la fecha que se seleccione en el calendario.
Fecha Fin Centro	Calendario	Campo que se carga con la fecha que se seleccione en el calendario.
Visualizar Producto	Gráfico Botón	Botón que se habilita cuando introducimos un producto. Permite acceder al gráfico de producto por periodo.
Visualizar Tienda	Gráfico Botón	Botón que se habilita cuando introducimos una tienda. Permite acceder al gráfico de ventas por tienda.
Volver	Botón	Botón que permite al administrador volver a la pantalla de generar catálogo-

4. Manual de administración - parte servidora

En esta parte de la documentación encontraremos toda la información necesaria para poder configurar el entorno de trabajo necesario para poder administrar la aplicación web PAJARRACO'S PAJARERIA.

4.1. Instalación Java

Para realizar la instalación del entorno de desarrollo JavaRuntimeEnvironment6 hay que descargar la versión del siguiente enlace:

<http://www.java.com/es/download/>

Ejecutamos el instalador y seguimos las instrucciones.

4.2. Instalación del Servidor de Aplicaciones

Nuestro servidor de aplicación es Websphere Server 6.1.1. A continuación mostramos el enlace a través del cual se puede iniciar la descarga de la aplicación.

http://www14.software.ibm.com/download/data/web/en_US/trialprograms/A609944Y83419O13.html.

Una vez descargado el servidor, ejecutamos el asistente de instalación de Websphere 6.1.



Figura 37. Instalación servidor Websphere 6.1.

Una vez instalado, arrancamos el servidor y una vez arrancado el servidor ejecutamos la consola administrativa para comprobar su correcta instalación.

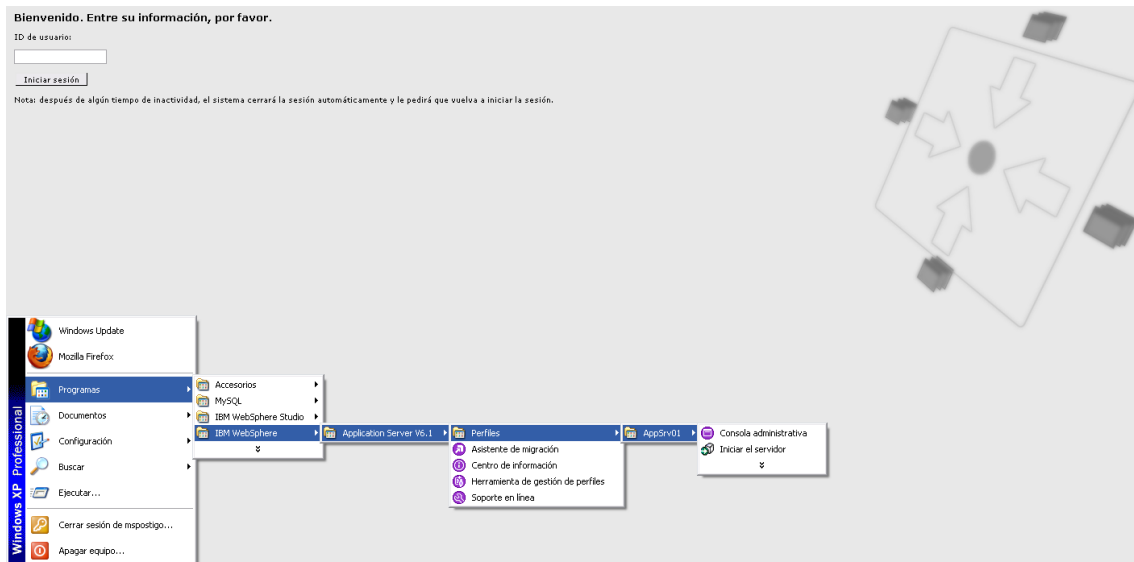


Figura 38. Ejecutar consola administrativa

4.3. Instalación de la herramienta de desarrollo

Para descargar la herramienta que emplearemos para desarrollar la aplicación, WebsphereApplicationToolkit, utilizaremos el siguiente enlace:

<http://www-01.ibm.com/support/docview.wss?uid=swg24012846>

A continuación ejecutamos el instalador y seguimos las instrucciones de este.

4.4. Instalación del Sistema Gestor de Base de Datos

Nuestro sistema de gestión de la base de datos es MySQL 5.5. Para descargar el instalador de la aplicación seguiremos el siguiente enlace:

<http://dev.mysql.com/downloads/>

Una vez instalada la aplicación, crearemos nuestra base de datos y realizaremos la carga inicial de datos mediante varios scripts.

4.5. Instalación del Servidor Web

Para realizar la instalación de nuestro servidor web, Apache 2.0, descargaremos el instalador a través del enlace:

<http://httpd.apache.org/download.cgi>

Una vez descargador el instalador, seguiremos las instrucciones del instalador. El puerto seleccionado será el puerto 80. También instalaremos el plugin de servidor Websphere para Apache 2.0. Descargamos el instalador, lo ejecutamos y seguimos las instrucciones del instalador.

Una vez instalado configuramos el virtual host del servidor web (httpd.conf) con la siguiente información

```
<VirtualHost 192.168.2.41:80>
ServerAdmin admin@gmail.com
DocumentRootstatico
SSLDisable
SSLClientAuth none
ServerName diamantedegould.no-ip.org
```


ErrorLog "logs/pajaros.log"
CustomLog "logs/pajaros-access.log" common
</VirtualHost>

donde el VirtualHost es la ip de la máquina servidora donde esté alojada la aplicación y el serverName es el dns correspondiente a la ip pública por la que se van a redireccionar las peticiones a nuestra aplicación.

4.6. Instalación de la aplicación

A través de la consola de administración, instalamos el war de la aplicación. A continuación detallamos el proceso paso a paso.

1.- En la primera pantalla seleccionamos el war de la aplicación y especificamos el context-root de la aplicación /PajareriaWeb

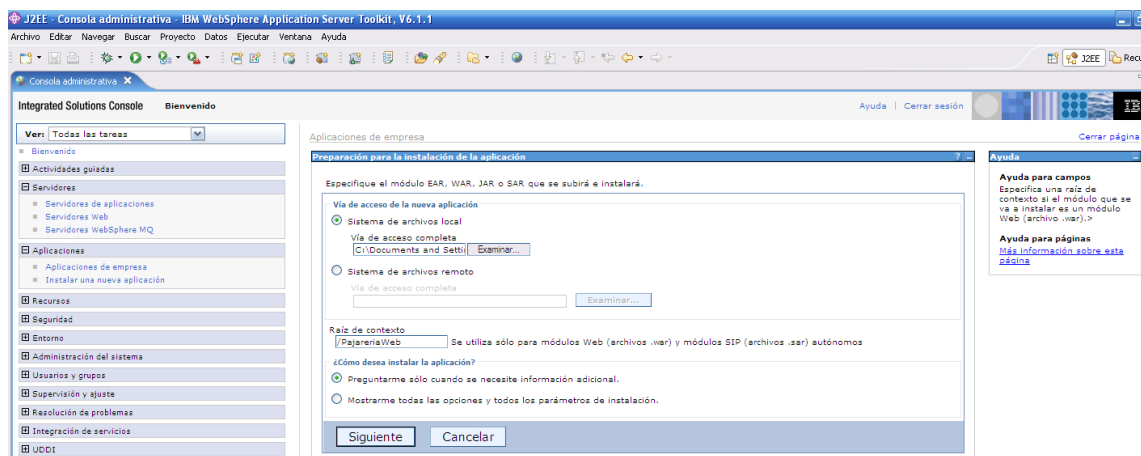


Figura 39. Selección war aplicación

2.- Especificamos la ruta de instalación sobre el servidor, el nombre de la aplicación y continuamos con el proceso de instalación.

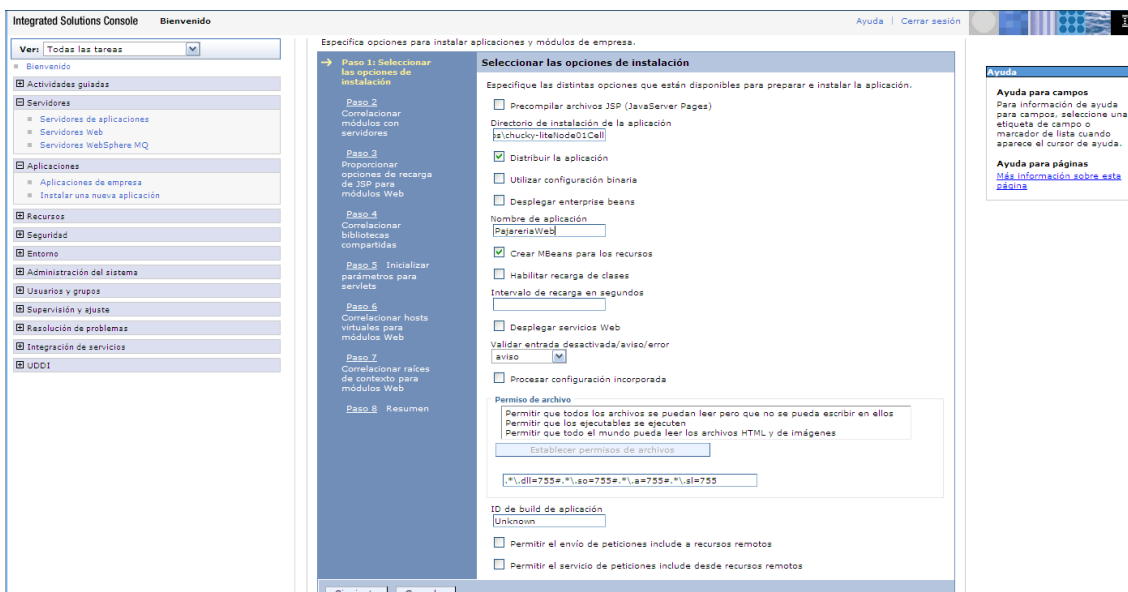


Figura 40. Especificar ruta de instalación

3.- Seleccionamos el servidor de aplicaciones y el servidor web.

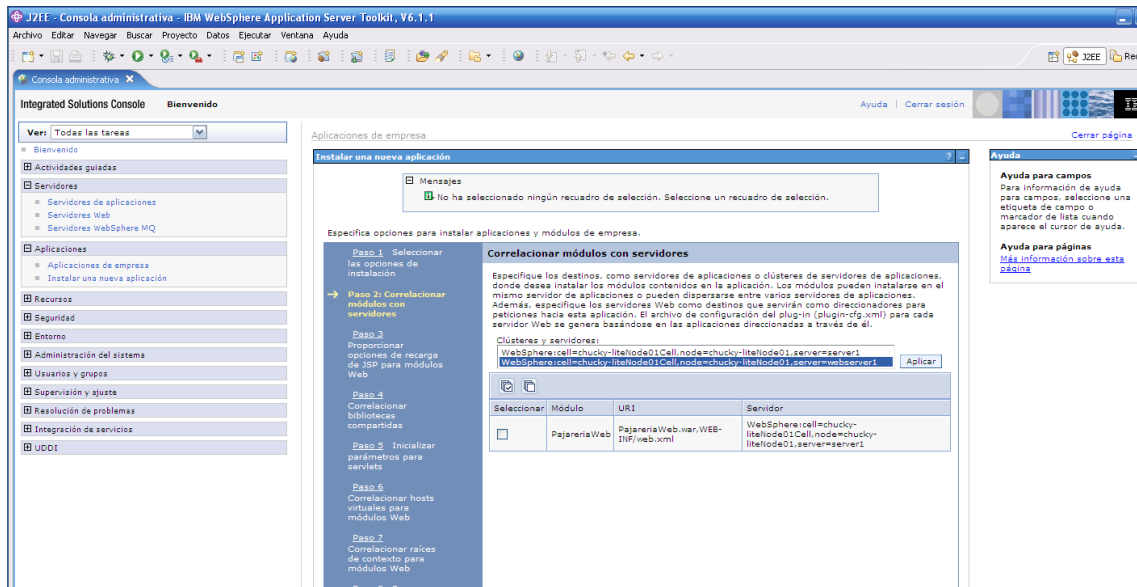


Figura 41. Elegir servidor web y de aplicaciones

4.- Indicamos el tiempo de recarga de las jsp.

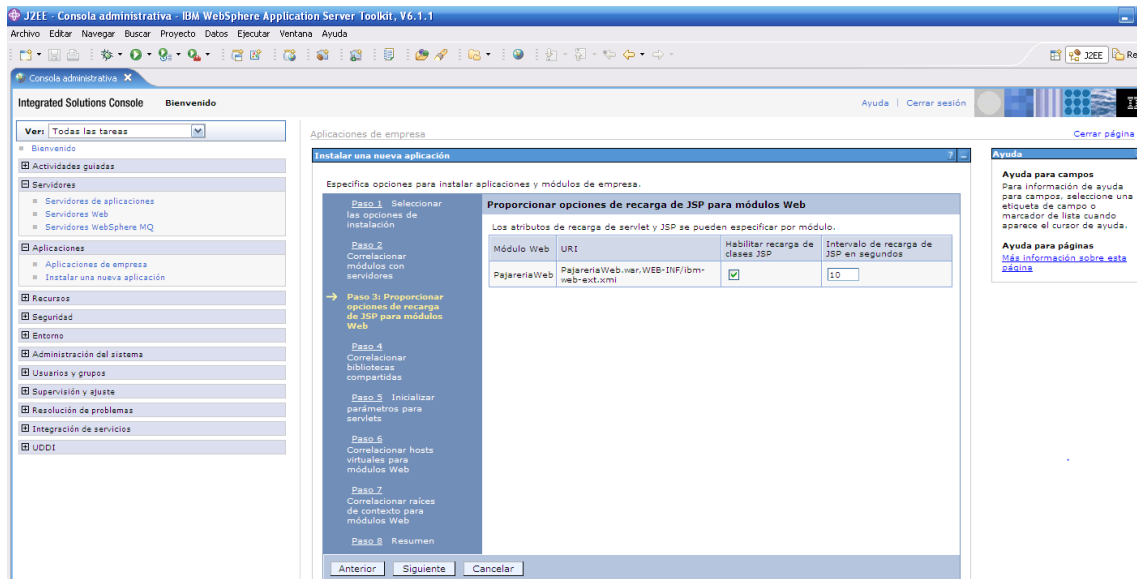


Figura 42. Configurar tiempo de recarga

5.- En nuestro caso no tendremos librerías asociadas al proyecto. En caso de tenerlas, estarán ubicadas dentro del directorio WEB-INF/lib de la aplicación.

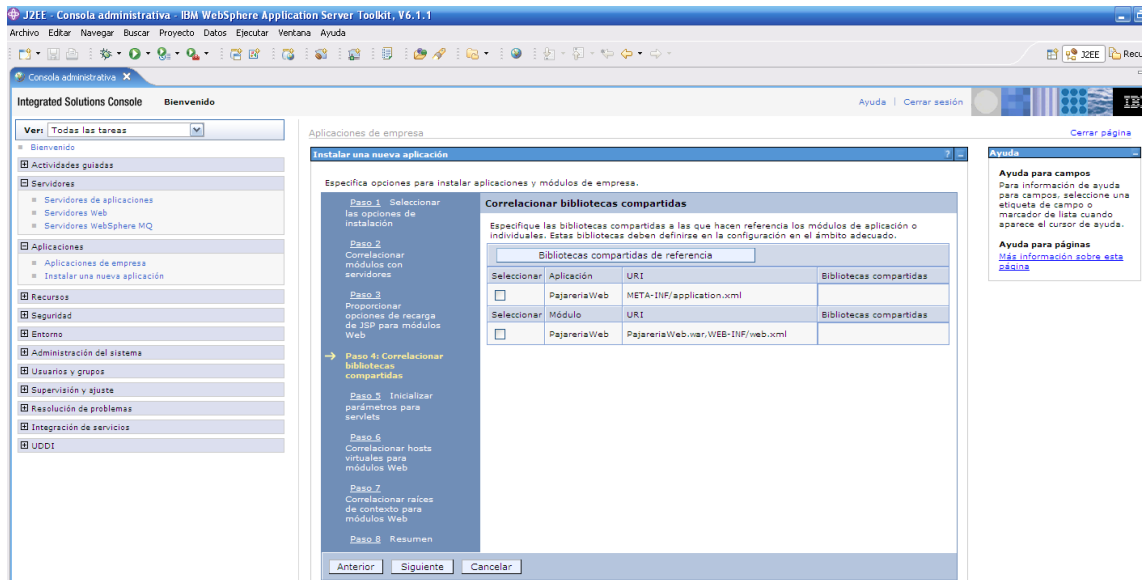


Figura 43. Ubicación librerías

6.- Revisamos la configuración especificada dentro del web.xml.

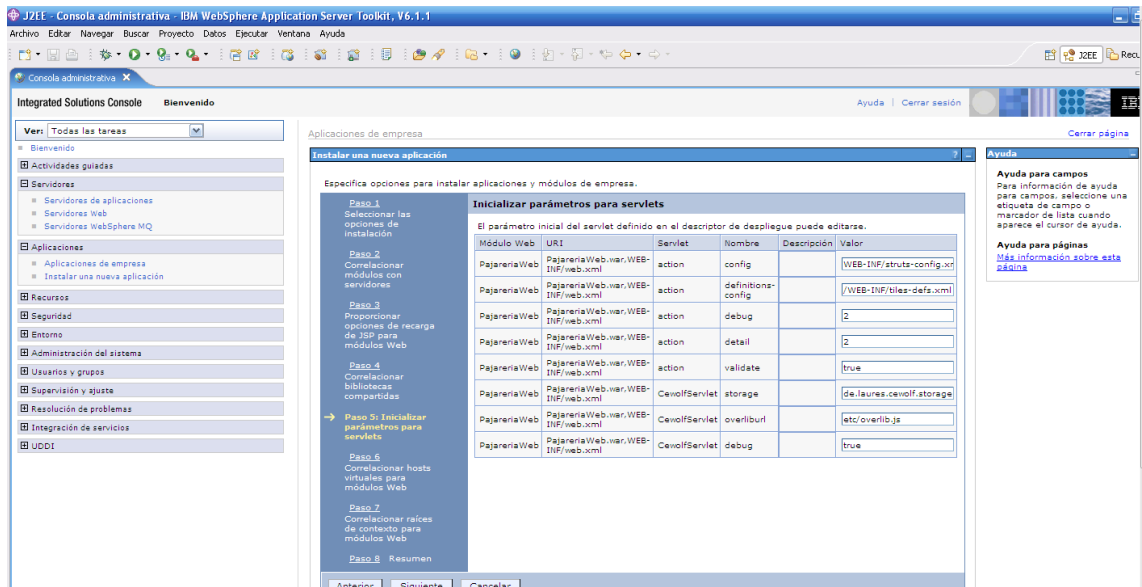


Figura 44. Configuración web

7.- Seleccionamos el virtual host por defecto.

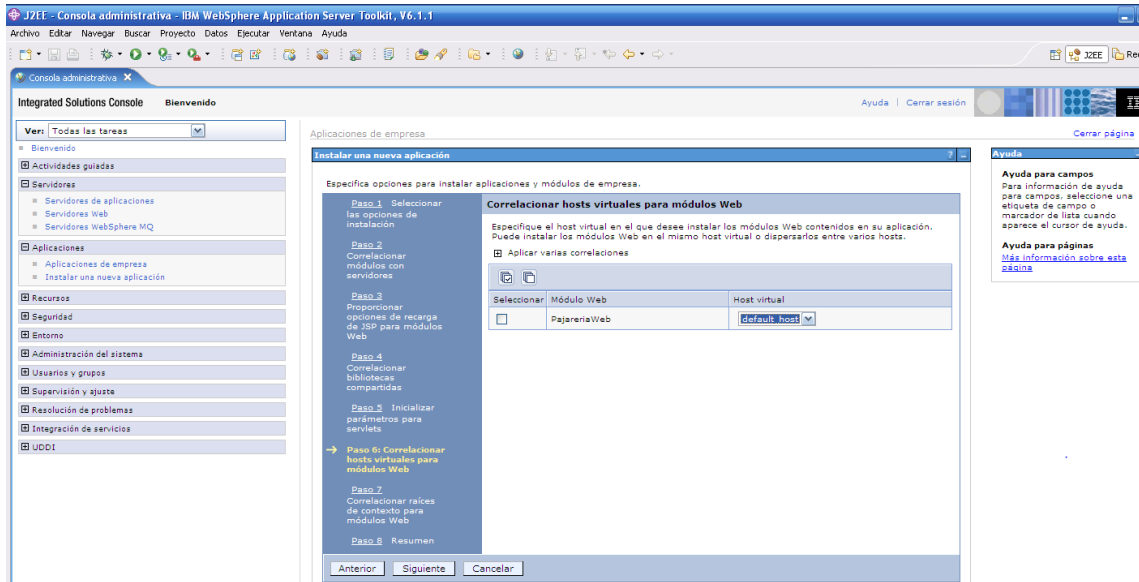


Figura 45. Virtual host

8.- Validamos el context-root de la aplicación.

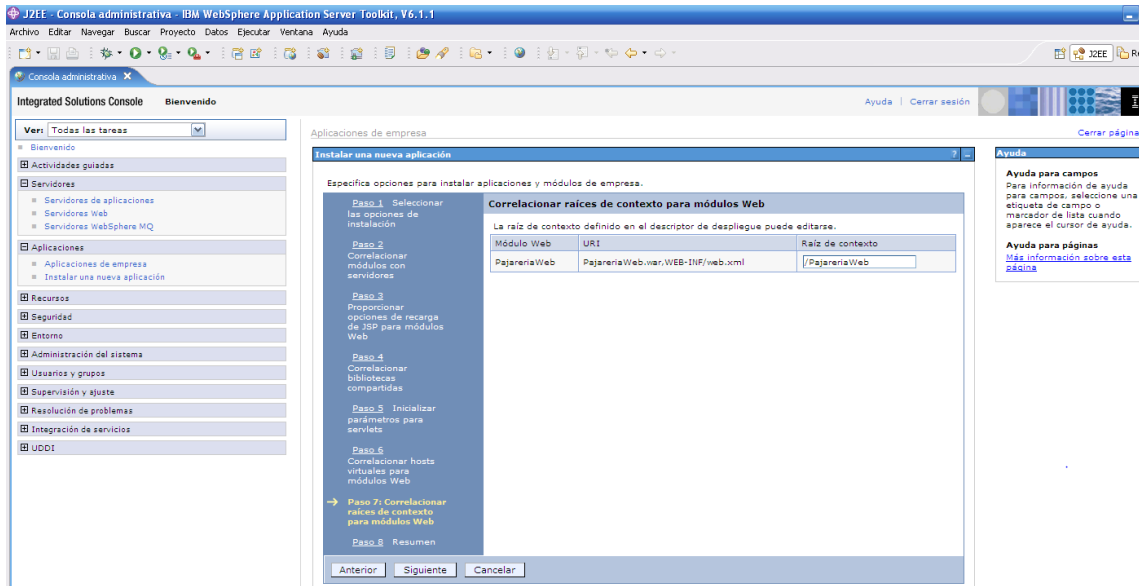


Figura 46. Validar contexto de la aplicación

9.- La siguiente pantalla muestra un resumen con los datos de la instalación.

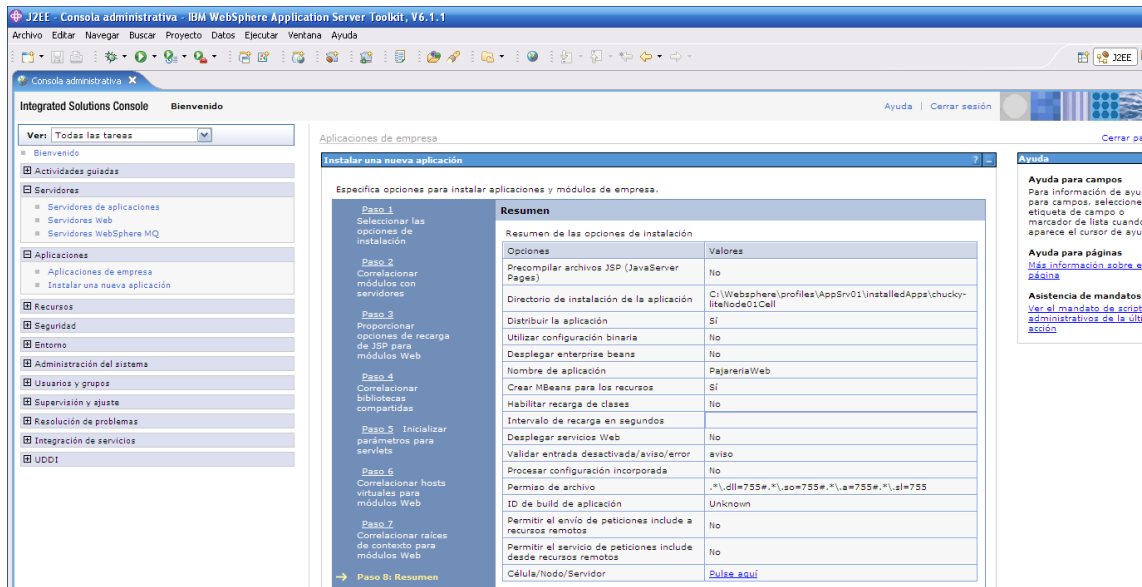


Figura 47. Resumen instalación aplicación web

10.- Una vez instalada sólo nos falta configurarla para que se carguen las librerías de la aplicación antes que el resto de librerías asociadas al servidor.

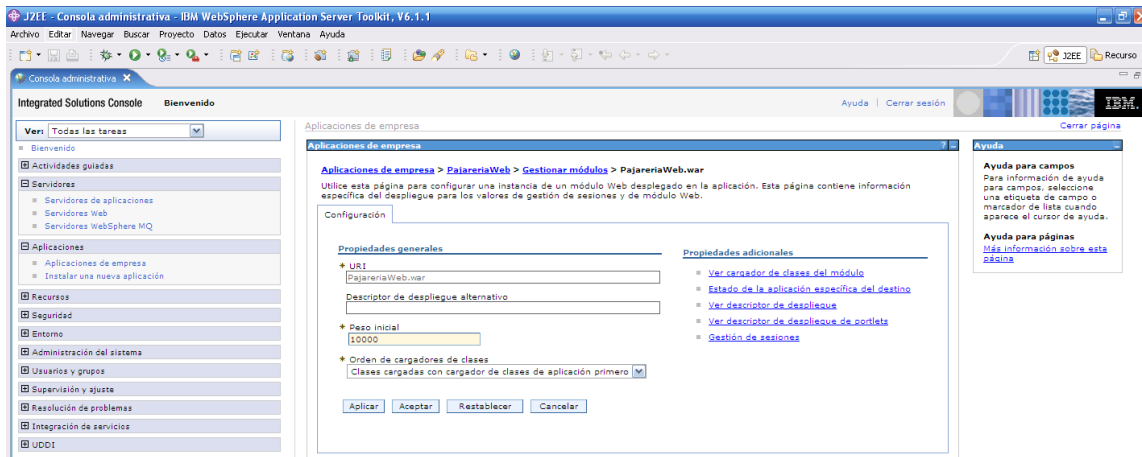


Figura 48. Configuración librerías

11.- Una vez completado el proceso de instalación comprobamos que la aplicación está corriendo en el servidor de aplicaciones correctamente.

Abrimos un navegador e introducimos <http://diamantedegould.no-ip.org/PajareriaWeb/>

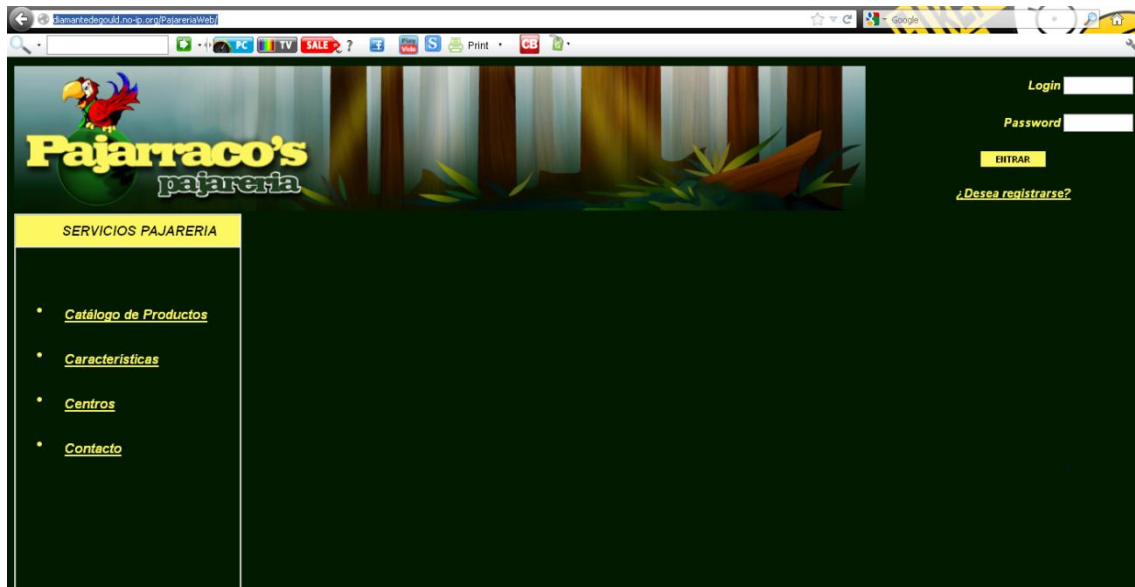


Figura 49. Aplicación web

4.7. Reinicio de la aplicación

De manera diaria se reiniciará el servidor de aplicaciones para liberar memoria. Para realizarlo, ejecutaremos el script stopServer.bat para parar la aplicación y posteriormente el script startServer.bat, que se encuentra dentro de la instalación de Websphere C:\Websphere\profiles\AppSrv01\bin.

Otra opción para realizar la parada y el arranque del aplicativo es a través del listado de aplicaciones instaladas.

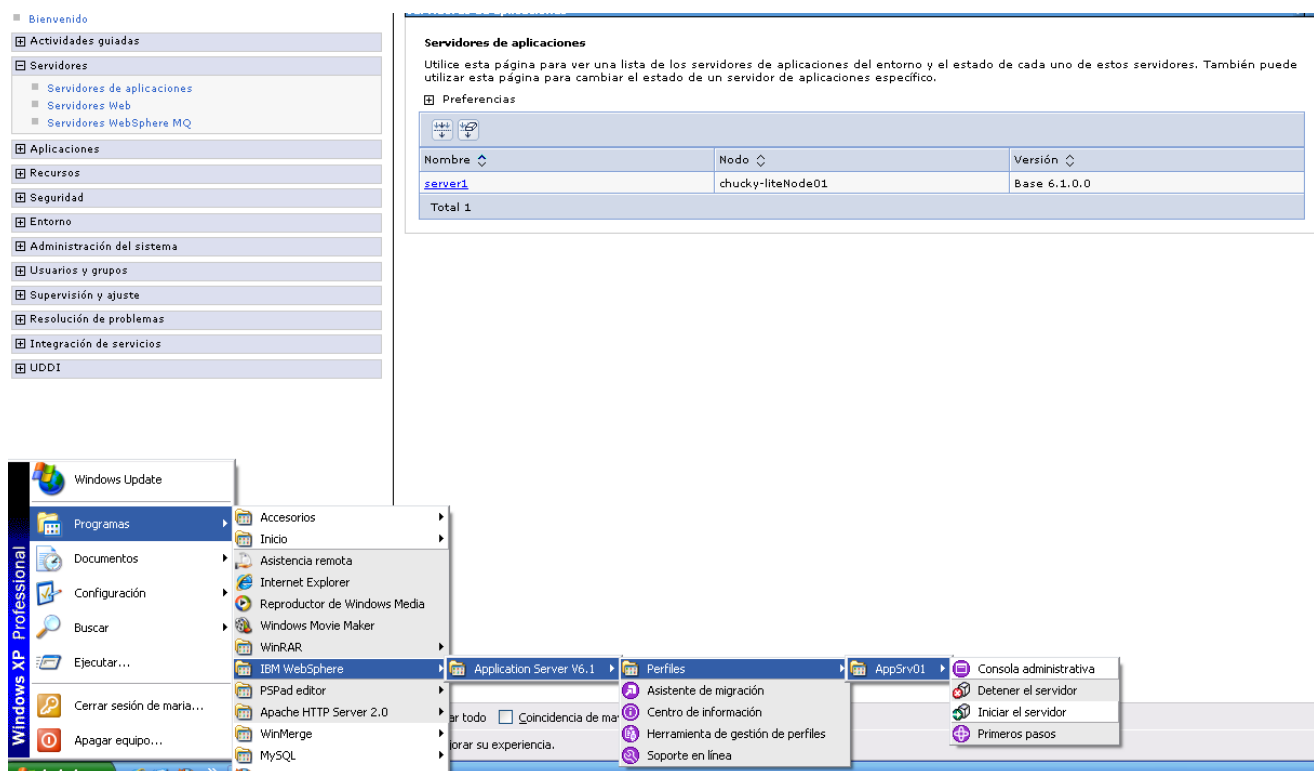


Figura 50. Reinicio aplicación

La ruta donde se almacenan los logs del arranque y ejecución de la aplicación son el SystemOut.log y SystemErr.log que se encuentra ubicados en el directorio C:\Websphere\profiles\AppSrv01\logs\server1. Las trazas de aplicación nos ayudan a comprobar el correcto arranque de la aplicación es *WSVR0001I: Servidor server1 abierto para e-business.*

4.8. Export / Import de la Base de Datos

Dentro de las tareas de mantenimiento de la aplicación, además del reinicio del servidor de aplicaciones, se realizará de manera semanal un backup de seguridad de la Base de Datos.

Para realizar el export (extracción) del contenido de la base de datos realizaremos los siguientes pasos:

- Nos posicionamos a través de una consola de msdos en el directorio de instalación de mysql, por defecto, C:\Archivos de programa\MySQL\MySQL Server 5.5\bin.
- Una vez ubicados en el directorio, ejecutaremos la sentencia: `mysqldump -h`. Esto nos genera un fichero `export.sql` con una copia de la información de la base de datos.

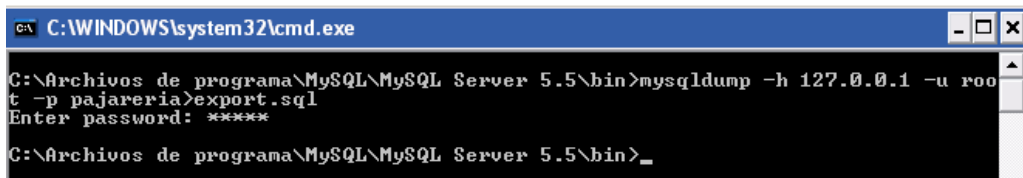


Figura 51. Consola msdos en directorio de instalación MySQL

En el caso de que tuviéramos que recuperar la información realizaríamos la operación inversa, es decir, realizaríamos el import (volcado) de la base de datos. Para ello, seguiremos los siguientes pasos:

- No conectamos al servidor con la sentencia: `mysql -h 127.0.0.1 -u root -p`.
- Creamos la base de datos con la sentencia: `createdatabase pajarería;`
- Seleccionamos la base de datos que acabamos de crear: `use pajarería;`
- Por último, ejecutamos el fichero `export.sql` que obtuvimos en el proceso anterior: `sourceexport.sql`.

4.9. Export / Import de la Base de Datos

Dentro de las tareas de mantenimiento de la aplicación, además del reinicio del servidor y las copias de seguridad de la base de datos, también es necesario adaptar la funcionalidad de la aplicación a las nuevas necesidades que vayan surgiendo. Para las actualizaciones de código y nuevos desarrollos de la aplicación, el procedimiento a seguir será la construcción del war a partir del código de la aplicación. La aplicación debe de compilar en su totalidad y el war se generará a través de AST.

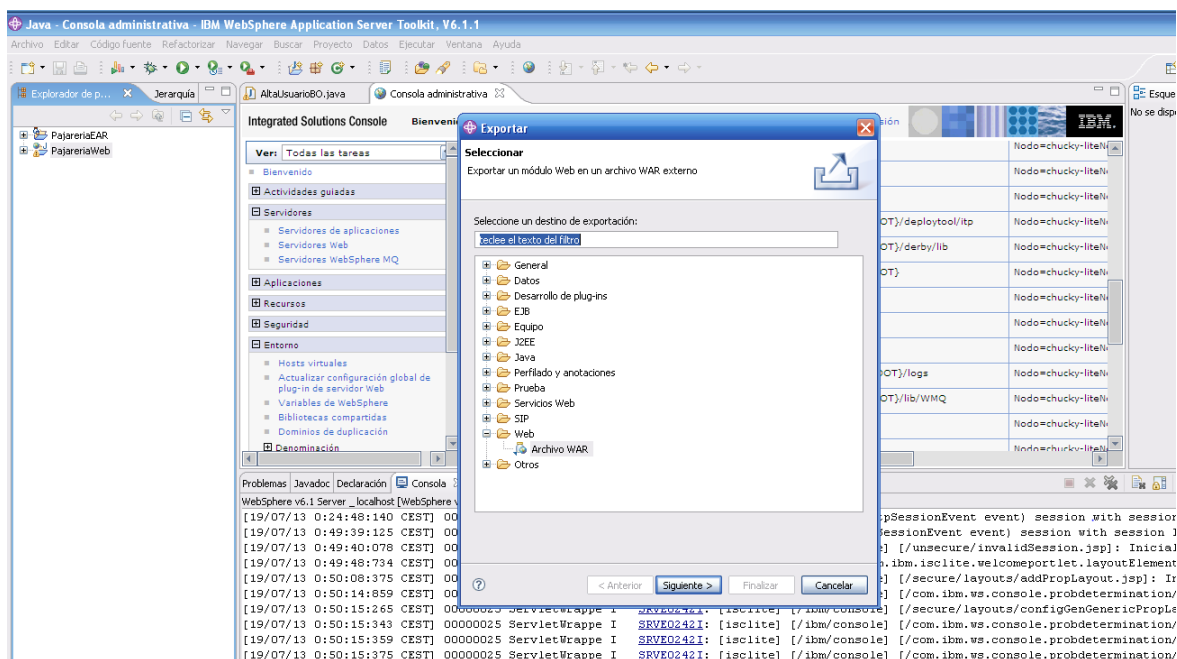


Figura 52. Exportación de código

Seleccionamos el proyecto compilado, botón derecho exportar archivo war. Una vez empaquetado continuamos con el

proceso de instalación a través de la consola administrativa de Websphere.

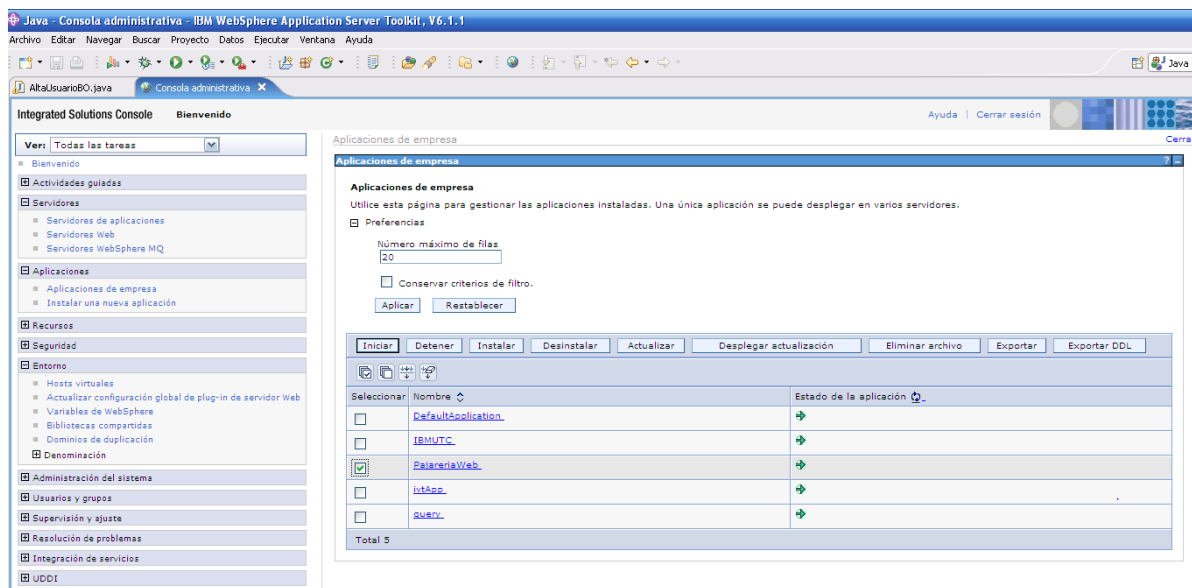


Figura 53. Instalación aplicación

Posteriormente continuamos el proceso siguiendo los pasos del punto 4.6 del documento (Instalación de la aplicación).

Bibliografía

Configuración Apache

<http://www.oracle.com/technetwork/java/index.html>

<http://www.desarrolloweb.com/manuales/41/>

<http://httpd.apache.org/docs/2.0/>

Documentación Struts

<http://struts.apache.org/development/1.x/userGuide/>

<http://struts.apache.org/release/1.2.x/api/index.html>

<http://www.jguru.com/faq/java-tools/struts>

<http://tiles.apache.org/>

http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=struts_appdemo

MySQL

<http://dev.mysql.com/doc/refman/5.0/es/index.html>

<http://www.mysqltutorial.org/>

<http://www.mysqltutorial.org/install-mysql/>

Métrica v3

http://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3#.UhXZD3-NAgk

Ingeniería del Software

<http://es.wikipedia.org/wiki/COCOMO>

<http://www.uml.org>

<http://users.dcc.uchile.cl/~psalinas/uml/modelo.html>

http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

<http://www.soyatec.com/euml2/>