

UNIVERSIDAD DE MÁLAGA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERIA TÉCNICA EN INFORMÁTICA DE SISTEMAS

Gestión Web para un Banco del Tiempo

Web Management for a Bank of Time

Realizado por

JUAN ANTONIO GARRIDO DOMÍNGUEZ

Dirigido por

JOSÉ GALINDO GÓMEZ

Departamento

**DEPARTAMENTO DE LENGUAJES Y
CIENCIAS DE LA COMPUTACIÓN**

Málaga, Marzo de 2011

UNIVERSIDAD DE MÁLAGA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

Reunido el tribunal examinador en el día de la fecha, constituido por:

Presidente Dº/Dª. _____

Secretario Dº/Dª. _____

Vocal Dº/Dª. _____

para juzgar el proyecto Fin de Carrera titulado:

Gestión Web para un Baco del Tiempo

del alumno Dº/Dª. **Juan Antonio Garrido Domínguez**

dirigido por Dº/Dª. **José Galindo Gómez**

ACORDÓ POR _____ OTORGAR LA CALIFICACIÓN DE

Y PARA QUE CONSTE, SE EXTIENDE FIRMADA POR LOS COMPARECIENTES DEL TRIBUNAL, LA PRESENTE DILIGENCIA.

Málaga, a ____ de _____ de _____

El Presidente _____ El Secretario _____ El Vocal _____

Fdo: _____ Fdo: _____ Fdo: _____

Agradecimientos

Llegado el momento, toca acordarse de todos los que directa o indirectamente me han acompañado en este largo y duro camino agradeciéndoles su apoyo.

- A mis padres y hermanos, por su comprensión y apoyo en estos años de carrera.
- A mis amigos y compañeros por su interés y sus consejos para la realización del proyecto.
- A David Chapman, administrador del sistema de moneda complementaria Málaga Común, por su dedicación y paciencia, para que el proyecto se haya podido convertir en una realidad.
- A mi tutor D. José Galindo Gómez por la confianza depositada en el proyecto, su dedicación y ánimo.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN	9
1.1 MOTIVACIÓN	9
1.1.1 <i>El problema con el dinero</i>	9
1.1.2 <i>El problema sugiere la solución</i>	10
1.2 OBJETIVOS.....	11
1.3 SOFTWARE PARA LA GESTIÓN DE SISTEMAS LETS	12
1.3.1 <i>Cyclos</i>	12
1.3.2 <i>UniLETIM</i>	13
1.3.3 <i>WebLets</i>	13
1.4 MÁLAGA COMÚN, SOLUCIÓN PROPUESTA	14
1.5 MEDIOS MATERIALES	14
1.6 ORGANIZACIÓN DE LA MEMORIA	15
HERRAMIENTAS EMPLEADAS	17
2.1 SISTEMA OPERATIVO	17
2.1.1 <i>Linux</i>	17
2.1.2 <i>Windows Server 2003</i>	18
2.2 SERVIDORES WEB	18
2.2.1 <i>Apache</i>	18
2.2.2 <i>Microsoft Internet Information Server</i>	20
2.2.3 <i>Sun ONE Web Server</i>	21
2.2.4 <i>Comparativas de uso de servidores web</i>	21
2.3 SISTEMAS DE GESTIÓN DE BASES DE DATOS.....	21
LOS SISTEMAS DE GESTIÓN DE BASES DE DATOS (EN INGLÉS DATABASE MANAGEMENT SYSTEM, ABREVIADO.. 21 DBMS) SON UN TIPO DE SOFTWARE MUY ESPECÍFICO, DEDICADO A SERVIR DE INTERFAZ ENTRE LA BASE DE DATOS, EL USUARIO Y LAS APLICACIONES QUE LA UTILIZAN [5].....	22
2.3.1 <i>MySQL</i>	22
2.3.2 <i>Microsoft SQL Server</i>	23
2.3.3 <i>Oracle</i>	24
2.3.4 <i>PostgreSQL</i>	25
2.4 LENGUAJES DE PROGRAMACIÓN	26
2.4.1 <i>PHP</i>	26
2.4.2 <i>JSP</i>	27
2.4.3 <i>ASP & ASP.NET</i>	27
2.4.4 <i>HTML</i>	28
2.4.5 <i>CSS</i>	29
2.4.6 <i>JavaScript</i>	29
2.4.7 <i>SQL</i>	29
2.5 SISTEMA GESTOR DE CONTENIDOS WEB (CMS)	30
2.5.1 <i>¿Qué es un CMS?</i>	30
2.5.2 <i>Necesidad de un CMS</i>	32
2.5.3 <i>Criterios de selección de un CMS</i>	34
2.5.4 <i>Silverstripe</i>	35
2.5.5 <i>Otros Sistemas Gestores de Contenidos Web</i>	36
2.6 CONCLUSIONES SOBRE LAS HERRAMIENTAS A UTILIZAR PARA EL DESARROLLO.....	36
ANÁLISIS DEL SISTEMA.....	39
3.1 ANÁLISIS DE REQUISITOS.....	39
3.1.1 <i>Requisitos para la gestión de sistemas</i>	40
<i>La aplicación de gestión de la moneda complementaria no debe ser un sistema cerrado y único, sino que debe permitirse la existencia de varios sistemas paralelos, cada uno con la posibilidad de tener su propio tipo de moneda o unidad de intercambio (créditos, horas, etc.)</i>	40
3.1.2 <i>Requisitos para la gestión de usuarios</i>	40
3.1.3 <i>Requisitos para la gestión de servicios</i>	41

3.1.4	Requisitos para la gestión de facturas	42
3.1.5	Requisitos para la gestión de mensajes	42
3.2	DIAGRAMAS DE ACTIVIDADES	43
3.2.1	Diagrama de actividades para el proceso de Autenticación.....	43
3.2.2	Diagrama de actividades para el proceso de Creación de un nuevo Servicio.....	45
3.2.3	Diagrama de actividades para el proceso Solicitud de un Servicio.....	45
3.3	EL MODELO DE DATOS	46
3.3.1	Perfil de Modelado de Datos UML.....	47
3.3.2	Modelo de datos de la aplicación.....	48
3.4	EL MODELO DE CASOS DE USO	50
3.4.1	Escenarios para los casos de uso.....	50
3.4.2	Descripción de los actores.....	50
3.4.3	Caso de uso Gestión de Sistemas	52
3.4.4	Caso de uso Gestión de Usuarios.....	57
3.4.5	Caso de uso Gestión de Servicios.....	60
3.4.6	Caso de uso Gestión de Facturas.....	63
3.4.7	Caso de uso Gestión de Mensajes.....	68
3.5	DIAGRAMA DE CLASES	71
3.5.1	Diagrama de clases para el escenario Componentes.....	71
3.5.2	Diagrama de clases para el escenario Acciones.....	72
3.5.3	Diagrama de clases para el escenario Extensiones	72
3.5.4	Diagrama de clases para el escenario Controles.....	74
3.5.5	Diagrama de clases para el escenario Principal.....	75
3.6	DIAGRAMAS DE SECUENCIAS	75
3.6.1	Diagrama de secuencias para la creación y modificación de datos	76
3.6.2	Diagrama de secuencias para el borrado de datos.....	77
3.6.3	Diagrama de secuencias para acciones genéricas.....	77
DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN		79
4.1	LIBRO DE ESTILO WEB	79
4.1.1	División del espacio de las páginas.....	80
4.1.2	Maqueta HTML.....	82
4.1.3	Usabilidad y accesibilidad web.....	82
4.2	EL MODELO MVC	83
4.2.1	El modelo de datos	84
4.2.2	La vista	87
4.2.3	El controlador	90
4.3	ESTRUCTURA DE LA APLICACIÓN	92
4.3.1	Acciones para las opciones de menú.....	93
4.3.2	Permisos para las opciones.....	95
4.4	MÓDULO DE LA APLICACIÓN	98
4.4.1	Estructura del módulo de la aplicación.....	98
4.4.2	Clases para el modelo de datos.....	99
4.4.3	Páginas y controles de la aplicación.....	100
4.4.4	Instancias de páginas	100
4.4.5	Controles para las acciones a realizar.....	103
4.4.6	Acciones para los menús de opciones.....	104
4.4.7	Javascript del módulo.....	106
4.4.8	Plantillas	107
4.5	CONTROLES DHTMLX.....	108
PARA MEJORAR EL ASPECTO Y LA INTERACCIÓN CON EL USUARIO DE LA WEB SE HAN INCLUIDO UNA SERIE DE CONTROLES GRATUITOS REALIZADOS POR LA EMPRESA DHTMLX (HTTP://WWW.DHTMLX.COM) EN JAVASCRIPT.....		108
DESPLIEGUE Y USO DE LA APLICACIÓN		109
5.1	INSTALACIÓN Y PASO A PRODUCCIÓN.....	109
5.1.1	Despliegue en un servidor web.....	110
5.1.2	Problemas de integración.....	111
5.2	PRUEBAS DE RENDIMIENTO Y OPTIMIZACIÓN	112
5.2.1	Tipos de pruebas de rendimiento.....	113

5.2.1.1 Pruebas de carga.....	113
5.2.3 Análisis de carga de la aplicación Málaga Común.....	118
5.3 CONFIGURACIÓN DE LA APLICACIÓN.....	120
5.3.1 Creación de un administrador.....	120
5.3.2 Traspasos de créditos.....	120
5.4 MANTENIMIENTO.....	121
5.4.1 Mantenimiento en la aplicación Málaga Común.....	122
MANUALES PARA LOS DISTINTOS TIPOS DE USUARIOS.....	123
6.1 INSTALACIÓN DE LA APLICACIÓN.....	124
6.2 PÁGINA WEB PRINCIPAL.....	125
6.2.1 Málaga Común.....	125
8.2.1 Intercambios.....	126
8.2.2 Participantes.....	128
8.2.3 Contactar.....	128
6.2.6 Opciones del menú inferior.....	129
6.3 SISTEMA DE GESTIÓN DE LA MONEDA COMPLEMENTARIA.....	131
6.3.1 Registro de un usuario en el sistema.....	131
6.3.2 Restablecer una nueva contraseña.....	133
6.3.3 Acceso a los datos de un usuario registrado.....	134
6.4 ADMINISTRACIÓN DE LA MONEDA COMPLEMENTARIA.....	171
6.4.1 Datos del sistema.....	171
6.4.3 Gestión de permisos.....	184
6.4.4 Mensajes de difusión.....	186
6.5 DESARROLLO DE NUEVAS FUNCIONALIDADES.....	187
6.5.1 Configuración.....	188
6.5.2 Administración del contenido estático de la aplicación.....	192
6.6 EJEMPLO DE USO.....	194
6.6.1 Ejemplo de creación de un nuevo sistema.....	194
6.6.2 Ejemplo de registro de un nuevo usuario con sus ofertas mínimas.....	196
6.6.3 Ejemplo de creación, solicitud y realización de una oferta.....	199
6.6.4 Ejemplo de creación, solicitud y realización de una demanda.....	201
6.6.5 Ejemplo de envío de un mensaje.....	203
CONCLUSIONES Y LÍNEAS FUTURAS.....	205

CAPÍTULO 1

INTRODUCCIÓN

En este primer capítulo describiremos brevemente los aspectos generales del proyecto. Primero mostraremos las motivaciones que nos han llevado a realizar este trabajo. En segundo lugar, explicaremos cuáles son los objetivos que se han establecido sobre él. En tercer lugar estudiaremos las características de algunos de los programas existentes. En cuarto lugar, los medios materiales a usar. Por último, comentaremos las secciones en las que está dividida esta memoria, junto a una descripción breve de su contenido.

1.1 Motivación

1.1.1 El problema con el dinero

El mundo está en serios problemas, tanto ecológica como económicamente [1] siendo el dinero convencional y la forma que toma una de sus principales causas.

Cada comunidad moderna depende de la circulación de la moneda nacional a través de su economía interna. Produciéndose flujos de entrada con las exportaciones, los visitantes y el gasto público; y flujos salientes con las importaciones, los viajes y los impuestos.

Cuando la industria local pierde un mercado de exportación, cuando menos visitantes llegan o cuando los gobiernos reducen el gasto, el dinero que sale no se sustituye.

Como la cantidad de dinero que circula en la comunidad cae, también lo hace el nivel de negocio. Disminuyen los negocios y las personas pierden sus empleos, no porque no tienen nada que ofrecer, sino porque no hay dinero suficiente para todos.

En el concurso de esta oferta de trabajo limitada, la gente trabaja de manera que pueda peligrar a su salud, al medio ambiente y al bienestar de la comunidad.

La gente está dispuesta a hacer casi cualquier cosa por dinero porque lo necesita para tomar parte en el juego. Esta es la fuente del problema, puesto que el dinero, en virtud de su propia estructura, es escaso y difícil de conseguir.

Fundamentalmente hay tres razones para ello:

1. Hay una cantidad limitada en circulación.
2. Se puede ir prácticamente a cualquier lugar del mundo, y de hecho así lo hace.
3. No se puede emitir por sí mismo.

Las comunidades de todo el mundo sufren de una escasez de dinero simplemente porque no permanece en ellas, se ha ido a otros lugares.

Cuando se piensa en ello, esta situación no tiene sentido. El dinero es sólo un medio de intercambio, un conjunto de entradas, un número en su cuenta bancaria. No tiene valor en sí mismo, no se puede comer, usar o construir cualquier cosa con él [2, 3, 4].

1.1.2 El problema sugiere la solución

Podemos solucionar este problema mediante la creación de fondos locales para financiar las necesidades locales, para generar riqueza y que nos proteja de la pobreza.

Una moneda local no puede salir de la comunidad que atiende, de manera que garantiza las conexiones de intercambio de conocimientos entre las personas, bienes y servicios. Con una moneda local, la comunidad se ve menos afectada por las fluctuaciones en la oferta de dinero externo.

Las monedas locales han sido comunes en toda la historia, emergiendo cada vez que una comunidad necesita proteger su economía interna de las perturbaciones externas, tales como la guerra o la depresión. El movimiento social de crédito fue un ejemplo, y los sistemas de más éxito se utilizaron en Austria antes de la segunda guerra mundial.

No es sorprendente que el actual clima económico haya dado lugar a varios sistemas que van desde pequeñas redes de ayuda informal a cientos de trueques comerciales. Siendo el crecimiento de estas redes extraordinario.

En la actualidad los LETS (Local Exchange Trading System, Sistemas de Comercio por Intercambio Local) son la forma más avanzada de sistemas de intercambio de monedas locales.

Los LETS son fáciles, rentables, simples, seguros y además cumplen con los requisitos de mantener a la moneda dentro de la comunidad a la que sirve, ser expedido por las personas que lo utilizan, y existir en cantidad suficiente como para satisfacer las necesidades de la comunidad.

Funciona como un banco o una sociedad de construcción. Todo el mundo tiene una cuenta, pero en lugar de transferir dinero de un banco a otro, todos los intercambios están dentro de un único sistema.

Los intercambios se realizan a través de una moneda ficticia, incluso puede que sea virtual, es decir, no tenga ninguna representación física. Pudiendo ser una moneda con un nombre propio, un sistema de puntos, un sistema en el que la moneda sea una unidad de tiempo, etc.

En el caso en que la moneda se base en una unidad de tiempo es cuando tendremos un “*Banco del tiempo*” propiamente dicho [2, 3, 4].

Además este tipo de sistemas suelen ser totalmente transparente, todos los usuarios pueden ver las actividades que se han realizado en el sistema, de esta forma se evita el fraude.

1.2 Objetivos

El objetivo de este proyecto es la creación de un portal web para la gestión de un sistema de moneda complementaria y su implantación en Málaga.

Para cumplir este objetivo la web se va a dividir en dos partes bien diferenciadas: una pública donde se expondrán los objetivos del sistema de moneda complementaria, su funcionamiento, historia, etc. Y otra privada, accesible a través de unas credenciales específicas para cada usuario, donde se gestionará el crédito, los servicios disponibles, etc.

Además la zona privada contendrá diferentes tipos de usuarios, dependiendo de los cuales se podrán acceder a unas zonas u otras. Por ejemplo, un administrador podrá crear sistemas adicionales (añadiendo nuevos tipos de monedas), mientras que un usuario simple sólo podrá gestionar su crédito y servicios.

1.3 Software para la gestión de sistemas LETS

En los sistemas LETS, al no existir una moneda física, la gestión se convierte en una tarea crítica que puede resolverse satisfactoriamente utilizando un software adecuado. Este deberá como mínimo permitir la creación de sistemas de riqueza alternativa (tipo de monedas a utilizar), y la gestión de usuarios y su crédito, preferiblemente manteniendo en todo momento la transparencia hacia el resto de usuarios.

Entre los programas que cumplen las características anteriores destacan principalmente tres: Cyclos, uniLETIM y Weblets. Además de estos programas hay que notar que los sistemas LETS son cada vez más populares, y como consecuencia de este aumento de popularidad cada vez aparecen más programas para su gestión. La gran mayoría son aplicaciones específicas para algún que otro sistema particular.

1.3.1 Cyclos

Cyclos es un software iniciado por la fundación Strohalm de Holanda (<http://project.cyclos.org>) para administrar proyectos de monedas alternativas en países del tercer mundo. Entre sus principales características destacan:

- Es un software con una carrera dilatada.
- Es un software libre, es decir, cualquiera que lo desee puede descargarlo, adaptarlo, personalizarlo y usarlo completamente gratis, además se puede descargar el código fuente. Esto lo convierte en un proyecto colaborativo y hace que no se dependa de ninguna empresa.
- Es un software probado en muchos lugares del mundo.
- En su desarrollo priman la robustez y seguridad del sistema.

Estas características lo han convertido en el software más popular para la gestión tanto de los sistemas LETS en general como de los bancos del tiempo en particular.

Pese a adaptarse completamente a los requerimientos de nuestro sistema existe una gran desventaja. Este se encuentra desarrollado en Java, un lenguaje en el que es difícil encontrar hospedajes gratuitos o en su defecto lo más barato posible.

1.3.2 UniLETIM

UniLetim es un software gratuito desarrollado en PHP. Diseñado para administrar a los usuarios del sistema y manipular las ofertas y servicios disponibles, puede funcionar en tres modos diferentes:

- **LETS en modo multiusuario:** cada usuario del sistema puede modificar sus datos, tanto personales como los referidos a su cuenta de usuario.
- **LETS en modo administrador:** existe un administrador encargado de gestionar todos los datos del sistema.
- **Modo Banco del Tiempo:** al igual que en el modo administrador, existe un único usuario encargado de gestionar todo el sistema. Sin embargo, a diferencia del sistema anterior, en éste no se puede definir un precio para los servicios, sino que se definen según las horas dedicadas.

Como principal ventaja de este software cabe destacar su sencillez, que hace que consuma muy pocos recursos y pueda instalarse en entornos muy baratos. Sin embargo se echan en falta muchas características como pueden ser la posibilidad de evaluar a usuarios, o un entorno traducido al español.

1.3.3 WebLets

WebLets es un programa desarrollado para el sistema LETS de Ottawa que esta disponible gratuitamente.

Al igual que ocurre con uniLETIM se encuentra desarrollado en PHP, sin embargo es poco flexible y en su funcionalidad se echan en falta muchas características, tales como un entorno traducido al español, o la posibilidad de evaluar a los usuarios.

1.4 Málaga Común, solución propuesta

Después de evaluar y descartar los anteriores programas para la gestión de los sistemas LETS, *Málaga Común* es la propuesta para la gestión del sistema de moneda complementaria de Málaga, y tendrá las siguientes características principales:

- Debe estar escrito en un lenguaje ligero, del que existan hospedajes gratuitos o bien bastante baratos en su defecto. Igualmente debe ocurrir con la base de datos y cualquier otra tecnología complementaria que se utilice.
- Debe poder integrarse fácilmente con la web a desarrollar, manteniendo el mismo diseño que ésta.
- Se tiene que poder definir diferentes tipos de riquezas (tiempo, créditos, monedas, etc.).
- Se tiene que poder evaluar los servicios ofrecidos, así como a los usuarios que los ofrecen.
- El sistema tiene que ser transparente, es decir, cualquier usuario debe poder ver todos los servicios realizados por los otros usuarios, de forma que se evite el fraude.

1.5 Medios materiales

Para montar el proyecto vamos a necesitar un servicio de hospedaje web que nos permita desplegar tanto la web como la aplicación a desarrollar. También será recomendable registrar un dominio, cosa que se evaluará junto con el tipo de hospedaje a utilizar (gratuito o de pago) una vez se conozca el presupuesto disponible para el proyecto.

Los usuarios simplemente necesitarán un navegador web que cumpla con los estándares para visitar la parte web, y un navegador que soporte JavaScript para el acceso a la aplicación (zona privada).

1.6 Organización de la memoria

El resto de la memoria se organiza como sigue:

- **Capítulo 2 – Herramientas empleadas.** En este capítulo señalaremos las características más destacadas de las herramientas que se usarán para realizar el proyecto así como posibles alternativas.
- **Capítulo 3 – Análisis del sistema.** En este capítulo se analizan los requisitos del sistema, y se crearan los diferentes esquemas para su desarrollo.
- **Capítulo 4 – Diseño e implementación de la aplicación.** En este capítulo vamos a ver los métodos empleados para diseñar e implementar la aplicación.
- **Capítulo 5 – Despliegue y uso de la aplicación.** Desde este capítulo se describen los pasos necesarios para la puesta en producción del sistema desarrollado.
- **Capítulo 6 – Manuales para los distintos tipos de usuarios.** Este capítulo consta de los manuales para los distintos tipos de usuarios que existen en la aplicación desarrollada.

CAPÍTULO 2

HERRAMIENTAS EMPLEADAS

En este punto señalaremos las características más destacadas de las herramientas que se usarán para realizar el proyecto así como posibles alternativas. Al final de esta muestra determinaremos las razones que nos conducen a la selección de los recursos.

2.1 Sistema operativo

Un Sistema operativo (SO) es un software que actúa de interfaz entre los dispositivos de hardware y los programas usados por el usuario para utilizar un computador. Es responsable de gestionar, coordinar las actividades y llevar a cabo el intercambio de los recursos y actúa como estación para las aplicaciones que se ejecutan en la máquina [5].

2.1.1 Linux

Es un sistema operativo basado en los estándares Unix, surgió a principios de los 90 a partir de las inquietudes de Linux Torvalds por mejorar y ampliar Minix (otra implementación gratuita de Unix desarrollada por Andy Tanenbaum, dirigida al ámbito educativo). Desde entonces, ha ido incrementándose de forma espectacular el número de desarrolladores desinteresados que se han implicado en su desarrollo a lo largo y ancho del mundo.

Lo que es propiamente Linux [5, 6] es el núcleo del sistema operativo, que ha ido implementando soporte para una gran parte del hardware actual (USB, cámaras digitales, escáneres, impresoras, grabadoras, redes, etc.). Dicho núcleo viene arropado por librerías y utilidades distribuidas bajo licencia libre GPL o similares (de aquí la denominación GNU/Linux).

Su excelente relación calidad-precio le ha granjeado la admiración e incondicional apoyo de muchísimos usuarios alrededor del mundo. Su adopción en el ámbito de los

servidores web ha sido espectacular. Estadísticas recientes demuestran que su empuje es cada vez mayor en este campo y todos los relacionados con Internet (como, por ejemplo, los servidores de espacio web e ISP). Por ejemplo, se usa en Google y Amazon.

Linux, entre muchas otras características, es multitarea, multiusuario, multiplataforma, multiprocesador, tiene protección de la memoria entre procesos, soporta muchísimos tipos de sistemas de archivos, dispone de una amplia variedad de protocolos de red soportados en el núcleo y, finalmente, permite compartir por red ficheros e impresoras incluso con otros sistemas operativos. La potencia, estabilidad, gratuidad, posibilidad de modificación y portabilidad de Linux lo hacen el sistema operativo perfecto, y ya tiene una posición líder en el ámbito de Internet, siendo cuestión de tiempo que la tenga en el resto de ámbitos informáticos.

2.1.2 Windows Server 2003

Es el producto estrella de Microsoft, siendo una versión mejorada de Windows 2000 y utilizado en una topología de red servidor. Proporciona su interfaz intuitivo, que es un reclamo a la facilidad de uso, pero esta facilidad no significa que su configuración sea fácil.

2.2 Servidores web

Un servidor web es un programa que está diseñado para transferir hipertextos, páginas web o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. El programa implementa el protocolo HTTP (HyperText Transfer Protocol) que pertenece a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa [5].

2.2.1 Apache

Apache es el servidor web más extendido en el mercado, debido en gran parte a su robusta implementación.

Originalmente Apache eran una serie de parches para el servidor web de la NCSA, estos parches fueron creados por webmasters para este servidor y después de que se

abandonara el desarrollo de dicho servidor varios de ellos se pusieron de acuerdo para coordinar sus esfuerzos y crear un único servidor web. Para ello crearon el grupo Apache (APACHE Software Project). Actualmente se distribuyen las versiones 1.3.X y 2.X.

Las características más destacables de apache son:

- **Fiabilidad:** alrededor del 90% de los servidores con más alta disponibilidad funcionan con Apache.
- **Multiplataforma:** Apache funciona en casi todos los sistemas operativos del mercado, lo que lo hace prácticamente universal.
- **Estándar:** se ajusta a los estándares del mercado.
- **Extensibilidad:** se pueden añadir módulos para ampliar las ya de por sí amplias capacidades de Apache. Hay una amplia variedad de módulos, que permiten desde generar contenido dinámico (con PHP, Java, Perl, Python,...), monitorizar el rendimiento del servidor, atender peticiones encriptadas por SSL, hasta crear servidores virtuales por IP (Internet Protocol o Protocolo de Internet) o por nombre (varias direcciones web son manejadas en un mismo servidor) y limitar el ancho de banda por cada uno de ellos. Dichos módulos incluso pueden ser creados por cualquier persona con conocimientos de programación.
- **Soporte a lenguajes de programación:** Apache trabaja con Perl, PHP y otros lenguajes de script. También trabaja con Java y páginas JSP. Teniendo todo el soporte que se necesita para tener páginas dinámicas.
- **Escalabilidad:** Apache da soporte a múltiples alojamientos mediante host virtuales.
- **Fácil configuración en la creación y gestión de logs:** es fácil de configurar para la creación y gestión de *logs*. Apache permite la creación de ficheros de *log* a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en el servidor.

- **Autenticación de usuarios:** Apache permite la autenticación mediante el uso de base de datos DBM, posibilitando la restricción de acceso a determinadas páginas de forma sencilla y cómoda de mantener.
- **Errores personalizados:** Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- **Software libre:** Apache es una tecnología libre de código fuente abierto. El hecho de ser libre es importante pero no tanto como que se trate de código fuente abierto. Esto le da una transparencia a este software de manera que si se desea ver que es lo que se está instalando como servidor, se puede saber, sin ningún secreto, sin ninguna puerta trasera.

2.2.2 Microsoft Internet Information Server

Es el servidor web de Microsoft. Actualmente se encuentra en su versión 6.0. Es el segundo servidor más usado en el mercado, en gran parte debido a que ya viene incorporado con los sistemas Windows y su instalación y configuración es cómoda. También su incursión en el mercado se debe a las mejoras proporcionadas desde sus versiones iniciales lo que lo ha convertido en un servidor estable, con el inconveniente de ser solo para plataformas Windows, y ser software propietario.

Algunas de las características más destacables son:

- **Detección de estado:** puede detectar el estado de los procesos de trabajo, así como reciclar los procesos de trabajo en base a diversos factores, como el rendimiento, una planificación designada, el número de solicitudes y el consumo de memoria. O puede reciclar los procesos de trabajo bajo demanda.
- **Escalabilidad de los sitios:** soporta múltiples alojamientos.
- **Integración de ASP.NET:** ofrece a los desarrolladores unos elevados niveles de funcionalidad, como el desarrollo rápido de aplicaciones (RAD) y una amplia selección de lenguajes.

2.2.3 Sun ONE Web Server

Servidor web de Sun Microsystems. Actualmente se encuentra en su versión 7.0 la cual ha pasado a ser gratuita, con lo que puede llegar a ser un serio competidor. Es un servidor multiplataforma que ofrece total conectividad con la plataforma Java 2 Enterprise Edition (J2EE).

Algunas de las características más destacables y que han sido liberadas y se integran con Sun ONE son:

- **Directory Server:** permite administrar y proteger la identidad de la red en múltiples plataformas.
- **SunScreen Firewall:** un firewall empresarial que permite el control del acceso al sistema, así como seguridad perimetral en modo furtivo.
- **Solares Containers:** permiten que los servidores de la red se particionen de manera flexible en entornos de ejecución independiente, proporcionando una mayor utilización de los recursos.
- **Sun ONE Portal Server:** proporciona a las comunidades de usuarios, acceso seguro a cualquier tipo de aplicación o servicio web con solamente un navegador.

2.2.4 Comparativas de uso de servidores web

Como vemos en la Figura 2.1 que proporciona Netcraft mensualmente, el servidor que domina es Apache mientras que ISS ha aumentado en gran medida su uso. Sin embargo la utilización de un servidor u otro depende de muchos factores, como son el económico, técnico, la fiabilidad, etc. En nuestro caso nos decantamos por la fiabilidad y transparencia de Apache.

2.3 Sistemas de gestión de bases de datos

Los sistemas de gestión de bases de datos (en inglés database management system, abreviado

DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan [5].

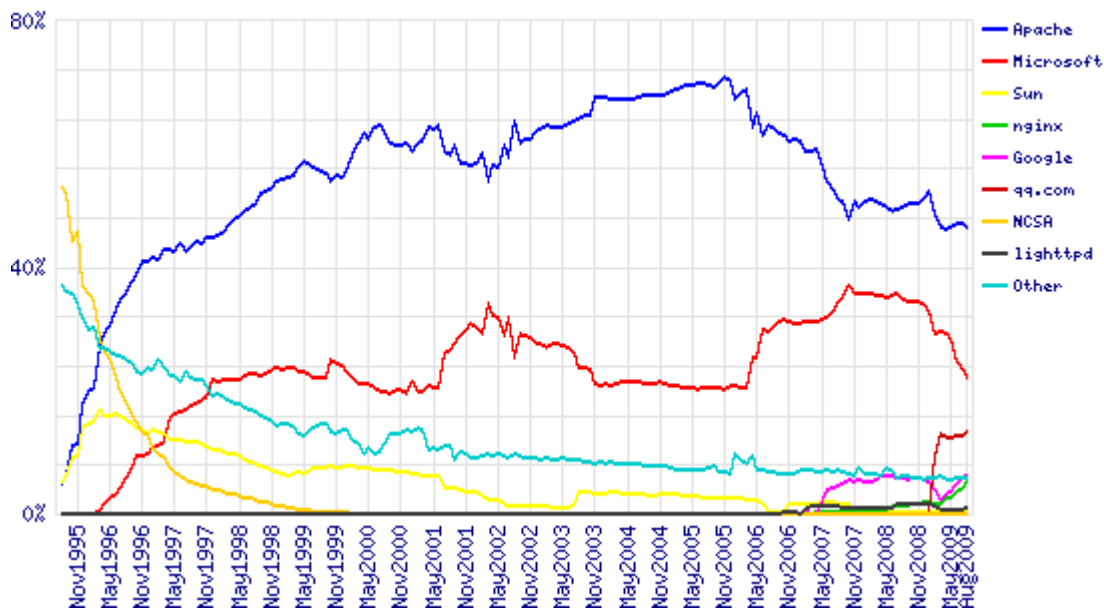


Figura 2.1: Evolución de servidores web según Netcraft

2.3.1 MySQL

MySQL es un servidor de bases de datos relacionales muy rápido y robusto. Actualmente es el SGBD más extendido en Internet.

Es software libre, publicado bajo licencia GPL (GNU Public License). Este gestor se creó con la rapidez en mente, de modo que no tiene muchas de las características de los gestores comerciales más importantes, como Oracle, Sybase o SQL Server. No obstante, eso no ha impedido que sea el más indicado para aplicaciones que requieren muchas lecturas y pocas escrituras y no necesiten de características muy avanzadas, como es el caso de las aplicaciones web. Poco a poco los elementos faltantes en MySQL están siendo incorporados tanto por desarrollos internos, como por los desarrolladores de software libre.

Las características más destacables de MySQL son:

- **Multiplataforma:** funciona en casi todos los sistemas operativos del mercado lo que lo hace prácticamente universal.
- **Amplio subconjunto del lenguaje SQL:** no se ajusta totalmente al estándar SQL/92, pero agrupa un gran subconjunto de las instrucciones que este ofrece.

- **Diferentes opciones de almacenamiento:** según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- **Transacciones y claves foráneas:** (con borrado y actualización en cascada). A partir de la versión 4.0.12 ya incorpora estas mejoras, con lo que MySQL ha conseguido equipararse en estabilidad a otras plataformas existentes en el mercado.
- **Indexación de campos de texto:** con lo que mejora el rendimiento en búsquedas de forma plausible.
- **Otras mejoras:** como son el bloqueo a nivel de fila, caché de consultas, la instrucción UNION y el borrado y actualización multitable.
- **Software libre:** el hecho de ser libre ha hecho que la comunidad se vuelque sobre el proporcionándole nuevas capacidades así como herramientas para su administración, lo que facilita en gran medida el uso de este SGBD.

2.3.2 Microsoft SQL Server

Es uno de los SGBD creados por Microsoft, junto con Access. Inicialmente SQL Server, en su versión 6, se basaba en el motor de base de datos Sybase, sobre el que Microsoft tenía un acuerdo de colaboración, pero sobre el que apenas podía desarrollar mejoras. Así tras un largo desarrollo, y comenzando desde cero, Microsoft desarrolló un motor propio que vio la luz a partir de la versión 7 del producto, lo cual provocó un cambio radical de la versión 6.5 a la 7. A partir de este punto de inflexión se lanzaron nuevas versiones de SQL Server con importantes mejoras sobre el motor de base de datos.

Algunas de las características más destacables son:

- **Facilidad de instalación, distribución y utilización:** SQL Server incluye un conjunto de herramientas administrativas y de desarrollo que mejoran la capacidad para instalar, distribuir, administrar y utilizar SQL Server entre varios sitios.
- **Escalabilidad:** puede utilizarse el mismo motor de base de datos a través de plataformas que van desde equipos portátiles que ejecutan Microsoft Windows XP

hasta grandes servidores con varios procesadores que ejecutan Windows NT, Enterprise Edition, etc.

- **Almacenamiento de datos:** SQL Server incluye herramientas para extraer y analizar datos resumidos para el proceso analítico en línea (OLAP, Online Analytical Processing). SQL Server incluye también herramientas para diseñar gráficamente las bases de datos y analizar los datos mediante preguntas en el lenguaje normal.
- **Integración del sistema con otro software de servidor:** SQL Server se integra con el correo electrónico, Internet y Windows.

2.3.3 Oracle

Oracle surgió a final de los años 70 y principio de los años 80 y actualmente es el SGBD más potente del mercado.

Algunas de las características más destacables son:

- **Multiplataforma:** puede ejecutarse en todas las plataformas, desde un PC hasta un supercomputador y en muchos de los sistemas operativos más utilizados como pueden ser Windows (en todas sus modalidades) o sistemas UNIX.
- **Copias de seguridad y recuperación:** Oracle proporciona sofisticados procedimientos de realización de copias de seguridad y recuperación de los datos.
- **Gestión de espacio:** Oracle ofrece una gestión flexible del espacio. Se puede asignar un cierto espacio de disco para el almacenamiento de los datos, y controlar las subsiguientes asignaciones instruyendo a Oracle sobre cuánto espacio debe reservar para los requerimientos futuros. También tiene una serie de características que fueron diseñadas teniendo en cuenta las necesidades de las bases de datos de gran tamaño.
- **Conectividad de carácter abierto:** Oracle proporciona conectividad hacia y desde paquetes software de otros fabricantes. Utilizando extensiones a la base de datos Oracle, se puede trabajar con información almacenada con otros sistemas de bases de

datos, como DB2 de IBM o Microsoft Access. También se pueden almacenar los datos en la base de datos de Oracle y acceder a ellos desde otros paquetes software.

- **Mecanismo de integridad:** el servidor Oracle también se encarga de la integridad de los datos. Si se produce cualquier tipo de fallo mientras un usuario está cambiando los datos en una base de datos, este tiene la capacidad de deshacer o cancelar cualquier transacción sospechosa.
- **Componente de consulta en paralelo:** se utiliza con computadoras de múltiples UCP, y permite que el procesamiento de las consultas sea repartido entre múltiples procesos servidores.
- **PL/SQL:** mediante este lenguaje de programación se pueden mantener procedimientos almacenados (o segmentos de código) en la base de datos, disparadores de base de datos (*triggers* o disparadores) y paquetes.

2.3.4 PostgreSQL

PostgreSQL es un Sistema Manejador de Bases de Datos Objeto-Relacional, liberado bajo la licencia BSD (Berkeley Software Distribution en español, Distribución de Software Berkeley). Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL), así como a sistemas propietarios como Oracle. Tiene las ventajas y apoyos de los sistemas de código abierto y ha alcanzado la potencia de los sistemas propietarios. La versión original (proyecto Ingres en la Universidad de Berkeley) fue abandonada, pero su licencia permitió que los desarrolladores de código abierto continuarán su evolución, actualmente se encuentra en es el gestor de bases de datos de código abierto más avanzado. A pesar de esto PostgreSQL tiene un problema ante otros SGBD libres, su lentitud y la mayor carga del sistema.

Algunas de las características más destacables son:

- **Vistas:** es capaz de generar vistas sobre las tablas.
- **Disparadores (*triggers*):** puede asociar *triggers* a las tablas aumentando su capacidad de autogestión.

- **Acceso concurrente multiversión:** no se bloquean las tablas, ni siquiera las filas, cuando un proceso escribe.
- **Múltiples lenguajes:** capacidad de albergar programas en el servidor en varios lenguajes.
- **Herencia de tablas:** es posible reutilizar las tablas del sistema operativo.

2.4 Lenguajes de programación

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana[1]. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación.

2.4.1 PHP

PHP, acrónimo recursivo de “PHP Hypertext Preprocessor”, es un lenguaje de *script* de código abierto de alto nivel e interpretado. Está especialmente pensado para el desarrollo de sitios webs dinámicos debido a su robustez, potencia y velocidad, pudiendo ser embebido en código HTML.

- Entre sus principales características destacan el gran número de bases de datos para las que ofrece soporte.
- Su sintaxis basada en Perl y C hace su aprendizaje bastante rápido para programadores de estos lenguajes.
- El código embebido en HTML permite desarrollar interfaces rápidamente.

- La versión 5 del lenguaje es orientado a objetos y posee características modernas como pueden ser la reflexión y numerosos *Frameworks* de desarrollo.

2.4.2 JSP

JSP (Java Server Pages) es un lenguaje de Sun Microsystem basado en la tecnología Java. Se basa principalmente en los *servlets*, que son clases Java que amplían la funcionalidad de un servidor web. Un motor se encarga de administrar la carga y descarga de los *servlets* y colabora con el servidor web para enviar la respuesta a los clientes.

Los *servlets* aportan las siguientes características al lenguaje:

- **Rendimiento:** los *servlets* se cargan la primera vez que son llamados y permanecen indefinidamente en memoria.
- **Simplicidad:** los *applets* de Java se ejecutan en la máquina virtual del cliente, lo cual da problemas de seguridad y compatibilidad. Los *servlets*, sin embargo lo hacen en la máquina virtual del servidor, con lo cual no es necesario que el cliente tenga software especial, solo necesita HTTP para comunicarse.
- **Sesiones HTTP:** la clase `HTTPSession` da la posibilidad de recordar resultados de peticiones previas del cliente.
- **Acceso a tecnología Java:** al ser aplicaciones Java, los *servlets* tienen acceso a toda la gama de características Java, como el uso de subprocesos, acceso a redes y conectividad a base de datos.

Así las páginas JSP, que se convierten automáticamente en *servlets*, heredan todas esas ventajas.

2.4.3 ASP & ASP.NET

ASP (Active Server Pages) es el lenguaje de Microsoft así como ASP.NET la evolución de este como solución al desarrollo web. Es parte del Internet Information Server (IIS) desde la versión 3.0. El desarrollo que se ha venido dando a lo que es ASP ha sido bastante amplio.

Entre sus funciones principales están el acceso a base de datos, envío de correo electrónico, creación dinámica de gráficos, entre otros.

Algunas de las mejoras en ASP.NET son:

- **Rendimiento:** la aplicación se compila una sola vez al lenguaje nativo, luego, en cada petición tiene una compilación Just In Time, es decir se compila desde el código nativo, lo que permite mejorar bastante el rendimiento. También permite el almacenamiento del caché en el servidor.
- **Rapidez en programación:** mediante diversos controles, podemos con unas pocas líneas y en menos de 5 minutos mostrar toda una base de datos y hacer rutinas complejas.
- **Servicios web:** trae herramientas para compartir datos e información entre distintos sitios.
- **Seguridad:** tiene diversas herramientas que garantizan la seguridad de las aplicaciones.

2.4.4 HTML

El HTML, acrónimo inglés de Hyper Text Markup Language (lenguaje de marcación de hipertexto), es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

Gracias a Internet y a los navegadores del tipo Explorer o Firefox, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

HTML es hijo de SGML, aunque hay unas versiones de XHTML que son descendientes de XML.

HTML utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determinan la forma en la que deben aparecer en su navegador el texto, así como también las imágenes y los demás elementos, en la pantalla del ordenador.

Toda etiqueta se identifica porque está encerrada entre menor y mayor qué (<>), y algunas tienen atributos que pueden tomar algún valor.

2.4.5 CSS

Las hojas de estilo en cascada (en inglés Cascading Style Sheets), es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

2.4.6 JavaScript

JavaScript es un lenguaje interpretado orientado a las páginas web, con una sintaxis semejante a la del lenguaje Java.

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que fabricó los primeros navegadores de Internet comerciales.

Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Se utiliza en páginas web HTML, para realizar tareas y operaciones en el marco de la aplicación cliente, como pueden ser la validación de formularios.

2.4.7 SQL

El Lenguaje de Consulta Estructurado (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Aúna características de álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos, de una forma sencilla.

En la actualidad el SQL es el estándar de facto de la inmensa mayoría de los SGBD comerciales. Y, aunque la diversidad de añadidos particulares que incluyen las distintas implementaciones comerciales del lenguaje es amplia, el soporte al estándar SQL-92 es general y muy amplio.

No obstante el éxito del SQL2 en estos momentos se trabaja en un nuevo estándar, el SQL3, que pretende dar soporte a las nuevas características derivadas de la generalización de los conceptos objeto-relacionales en las bases de datos.

El SQL proporciona una rica funcionalidad más allá de la simple consulta (o recuperación) de datos. Asume el papel de lenguaje de definición de datos (LDD), lenguaje de definición de vistas (LDV) y lenguaje de manipulación de datos (LMD).

Además permite la concesión y denegación de permisos, la implementación de restricciones de integridad y controles de transacción, y la alteración de esquemas.

Este lenguaje dispone, además, de una extensión llamada FSQL (Fuzzy SQL) para el tratamiento de información imprecisa usando lógica difusa, en bases de datos difusas [7].

2.5 Sistema Gestor de Contenidos Web (CMS)

La aplicación no se va a desarrollar desde cero, sino que se va a desarrollar bajo un sistema gestor de contenidos Web.

2.5.1 ¿Qué es un CMS?

Los sistemas de gestión de contenidos (Content Management System, CMS) son software que se utiliza principalmente para facilitar la gestión de webs, ya sean en Internet o en una intranet, y por eso también son conocidos como gestores de contenido web (Web Content Management o WCM). Hay que tener en cuenta, sin embargo, que la aplicación de los CMS no se limita sólo a las webs.

La funcionalidad de los sistemas de gestión de contenidos se puede dividir en cuatro categorías: creación de contenido, gestión de contenido, publicación y presentación.

2.5.1.1 Creación de contenido

Un CMS aporta herramientas para que los creadores sin conocimientos técnicos en páginas webs puedan concentrarse en el contenido. Lo más habitual es proporcionar un editor de texto WYSIWYG (What You See If What You Get), en el que el usuario ve el resultado final mientras escribe, al estilo de los editores comerciales, pero con un rango de formatos de texto limitado. Esta limitación tiene sentido, ya que el objetivo es que el creador pueda poner énfasis en algunos puntos, pero sin modificar mucho el estilo general del sitio web.

Hay otras herramientas como la edición de los documentos en XML, utilización de aplicaciones ofimáticas con las que se integra el CMS, importación de documentos existentes y editores que permiten añadir marcas, habitualmente HTML, para indicar el formato y estructura de un documento.

Un CMS puede incorporar una o varias de estas herramientas, pero siempre tendría que proporcionar un editor WYSIWYG por su facilidad de uso y la comodidad de acceso desde cualquier ordenador con un navegador y acceso a Internet.

Para la creación del sitio propiamente dicho, los CMS aportan herramientas para definir la estructura, el formato de las páginas, el aspecto visual, uso de patrones, y un sistema modular que permite incluir funciones no previstas originalmente.

2.5.1.2 Gestión de contenido

Los documentos creados se depositan en una base de datos central donde también se guardan el resto de datos de la web, cómo son los datos relativos a los documentos (versiones hechas, autor, fecha de publicación y caducidad, etc.), datos y preferencias de los usuarios, la estructura de la web, etc.

La estructura de la web se puede configurar con una herramienta que, habitualmente, presenta una visión jerárquica del sitio y permite modificaciones. Mediante esta estructura se puede asignar un grupo a cada área, con responsables, editores, autores y usuarios con diferentes permisos. Eso es imprescindible para facilitar el ciclo de trabajo (*workflow*) con un circuito de edición que va desde el autor hasta el responsable final de la publicación. El CMS permite la comunicación entre los miembros del grupo y hace un seguimiento del estado de cada paso del ciclo de trabajo.

2.5.1.3 Publicación

Una página aprobada se publica automáticamente cuando llega la fecha de publicación, y cuando caduca se archiva para futuras referencias. En su publicación se aplica el patrón definido para toda la web o para la sección concreta donde está situada, de forma que el resultado final es un sitio web con un aspecto consistente en todas sus páginas.

Esta separación entre contenido y forma permite que se pueda modificar el aspecto visual de un sitio web sin afectar a los documentos ya creados y libera a los autores de preocuparse por el diseño final de sus páginas.

2.5.1.4 Presentación

Un CMS puede gestionar automáticamente la accesibilidad de la web, con soporte de normas

internacionales de accesibilidad como WAI (Web Accessibility Initiative), y adaptarse a las preferencias o necesidades de cada usuario. También puede proporcionar compatibilidad con los diferentes navegadores disponibles en todas las plataformas (Windows, Linux, Mac, Palm, etc.) y su capacidad de internacionalización lo permite adaptarse al idioma, sistema de medidas y cultura del visitante.

El sistema se encarga de gestionar muchos otros aspectos como son los menús de navegación o la jerarquía de las página actual dentro del web, añadiendo enlaces de forma automática. También gestiona todos los módulos, internos o externos, que incorpore al sistema. Así por ejemplo, con un módulo de noticias se presentarían las novedades aparecidas en otro web, con un módulo de publicidad se mostraría un anuncio o mensaje animado, y con un módulo de foro se podría mostrar, en la página principal, el título de los últimos mensajes recibidos. Todo eso con los enlaces correspondientes y, evidentemente, siguiendo el patrón que los diseñadores hayan creado.

2.5.2 Necesidad de un CMS

En el apartado anterior se han presentado bastantes motivos para ver la utilidad de un sistema que gestione un entorno web, pero se podría pensar que no es necesario para una web relativamente pequeña o cuando no se necesitan tantas funcionalidades. Eso sólo podría ser cierto para una web con unas pocas páginas estáticas para el que no se prevea un crecimiento futuro ni muchas actualizaciones, lo que no es muy realista. En cualquier otro caso, la flexibilidad y escalabilidad que permiten estos sistemas, justifican su utilización en prácticamente cualquier tipo de web.

Muchos usuarios particulares utilizan CMS gratuitos para elaborar y gestionar sus webs personales, obteniendo webs dinámicos llenos de funcionalidades. El resultado que obtienen es superior al de algunas empresas que se limitan a tener páginas estáticas que no aportan ningún valor añadido.

Éstos son algunos de los puntos más importantes que hacen útil y necesaria la utilización de un CMS:

- **Inclusión de nuevas funcionalidades en el web:** esta operación puede implicar la revisión de multitud de páginas y la generación del código que aporta las funcionalidades. Con un CMS eso puede ser tan simple como incluir un módulo

realizado por terceros, sin que eso suponga muchos cambios en la web. El sistema puede crecer y adaptarse a las necesidades futuras.

- **Mantenimiento de gran cantidad de páginas:** en una web con muchas páginas hace falta un sistema para distribuir los trabajos de creación, edición y mantenimiento con permisos de acceso a diferentes áreas. También se tienen que gestionar los “*metadatos*” de cada documento, las versiones, la publicación y caducidad de páginas y los enlaces rotos, entre otros aspectos.
- **Reutilización de objetos o componentes:** un CMS permite la recuperación y reutilización de páginas, documentos, y en general de cualquier objeto publicado o almacenado.
- **Páginas interactivas:** las páginas estáticas llegan al usuario exactamente como están almacenadas en el servidor web. En cambio, las páginas dinámicas no existen en el servidor tal como se reciben en los navegadores, sino que se generan según las peticiones de los usuarios. De esta manera cuando por ejemplo se utiliza un buscador, el sistema genera una página con los resultados que no existían antes de la petición. Para conseguir esta integración, los CMS conectan con una base de datos que hace de repositorio central de todos los datos de la web.
- **Cambios de aspecto de la web:** si no hay buena separación entre contenido y presentación, un cambio de diseño puede comportar la revisión de muchas páginas para su adaptación. Los CMS facilitan los cambios con la utilización, por ejemplo, del estándar CSS (apartado 2.4.5) con lo que se consigue la independencia de presentación y contenido.
- **Consistencia de la web:** la consistencia en un web no quiere decir que todas las páginas sean iguales, sino que hay un orden (visual) en vez de caos. Un usuario nota enseguida cuándo una página no es igual que el resto de las de la misma web por su aspecto, la disposición de los objetos o por los cambios en la forma de navegar. Estas diferencias provocan sensación de desorden y dan a entender que el web no lo han diseñado profesionales. Los CMS pueden aplicar un mismo estilo en todas las páginas con el mencionado CSS, y aplicar una misma estructura mediante patrones de páginas.

- **Control de acceso:** controlar el acceso de un web no consiste simplemente al permitir la entrada a la web, sino que comporta gestionar los diferentes permisos a cada área de la web aplicados a grupos o individuos.

2.5.3 Criterios de selección de un CMS

Antes de empezar el proceso de selección de un CMS concreto, hay que tener claros los objetivos de la web, teniendo en cuenta al público destinatario, y estableciendo una serie de requerimientos que tendría que poder satisfacer el CMS.

- Ser de código abierto, para poder ampliarse en cualquier momento.
- Tiene que ser fiable y permitir la escalabilidad del sistema para adecuarse a futuras necesidades con módulos. También tiene que haber una separación de los conceptos de contenido, presentación y estructura que permita la modificación de uno de ellos sin afectar a los otros. Es recomendable, pues, que se utilicen hojas de estilo (CSS) y patrones de páginas.
- Debe de estar maduro y disponer de módulos que le añadan funcionalidades.
- La herramienta tiene que tener soporte tanto por parte de los creadores como por otros desarrolladores. De esta manera se puede asegurar de que en el futuro habrá mejoras de la herramienta y que se podrá encontrar respuesta a los posibles problemas.
- Una herramienta poco conocida puede ser muy buena, pero hay que asegurar de que tiene cierto futuro. También son importantes las opiniones de los usuarios expertos.
- La herramienta tiene que ser fácil de utilizar y aprender. Los usuarios no siempre serán técnicos, por lo tanto hace falta asegurar que podrán utilizar la herramienta sin muchos esfuerzos y sacarle el máximo rendimiento.

- Parar asegurar la accesibilidad de una web, el CMS tendría que cumplir un estándar de accesibilidad. El más extendido es WAI (Web Accessibility Initiative) del World Wide Web Consortium.
- Teniendo en cuenta que no todos los usuarios disponen de líneas de alta velocidad, las páginas se tendrían que cargar rápidamente o dar la opción.

2.5.4 Silverstripe

Silverstripe es un sistema administrador de contenidos (CMS, apartado 2.5.1) de código abierto ideal para implementar websites o intranets.

Sus principales características son:

- Interfase simple, intuitiva y amigable con el usuario.
- Desarrollado bajo el patrón de diseño MVC (Modelo Vista Controlador).
- Es escalable.
- Soporte nativo para *Ajax*.
- Modular.
- Sistema de plantilla (*template*).
- Licencia BSD.
- Multiplataforma: Windows, Linux y Mac.
- Fácil de instalar.
- Desarrollado en PHP5.

2.5.5 Otros Sistemas Gestores de Contenidos Web

Los sistemas gestores de contenidos han adquirido gran relevancia en el mundo del desarrollo web, debido a esta relevancia se ha producido una gran proliferación de ellos, entre los que destacan algunos como:

- **Joomla:** es uno de los más poderosos Sistemas de Gestión de Contenidos de código abierto. Se utiliza para todo, desde sitios web simples hasta complejas aplicaciones corporativas. Joomla es fácil de instalar, fácil de administrar y fiable.
- **CMS Made Simple:** es de código abierto (GPL), construido usando PHP proporciona a los desarrolladores web un medio simple para editar desde decenas a cientos de páginas, semi-páginas web estáticas, etc.
- **PHP-Fusion:** es de código abierto, ligero y escrito en PHP4. Utiliza una base de datos MySQL para almacenar el contenido de su sitio e incluye un sistema de administración simple pero completo.
- **Drupal:** es un software que permite a un individuo o una comunidad de usuarios publicar fácilmente, gestionar y organizar una gran variedad de contenido en un sitio web.

2.6 Conclusiones sobre las herramientas a utilizar para el desarrollo

Con lo expuesto anteriormente, llegamos a la conclusión de las herramientas que ofrece el mercado y las características que cada una ofrecen. Para este proyecto se ha decidido usar software libre, por lo que se ha decantado por Apache como servidor web, MySQL como base de datos y PHP como lenguaje de script. Esta decisión es debida en gran parte al gran auge de estas herramientas, provocado en el mercado que se integren en un paquete denominado LAMP (acrónimo de Linux, Apache, MySQL y PHP).

LAMP está considerada como una de las mejores herramientas disponibles para que cualquier organización o individuo pueda emplear un servidor web versátil y potente. Aunque creados por separado, cada una de las tecnologías que lo forman disponen de una serie de

características comunes. Especialmente interesante es el hecho que estos cuatro productos pueden funcionar en una amplia gama de hardware, con requerimientos relativamente pequeños sin perder estabilidad. Esto ha convertido a LAMP en la alternativa más adecuada para pequeñas y medianas empresas.

Existen, no obstante, multitud de variaciones de software libre. La L de Linux puede ser sustituida por FreeBSD, NetBSD u OpenBSD. En lugar de la M de MySQL también podemos encontrar PostgreSQL. La P sirve para PHP, Perl, Python o Ruby.

Algunas de las ventajas que se obtienen de utilizar LAMP son:

- **Múltiples arquitecturas:** soporte a gran cantidad de arquitecturas, como son Intel y compatibles, SPARC, Mips y PPC (Macintosh).
- **Sencillez:** código relativamente sencillo y con pocos cambios de una plataforma a otra.
- **Seguridad:** parches generados en poco tiempo después de encontrarse un agujero de seguridad.
- **Actualización cómoda:** actualizaciones del software vía Internet.
- **Mejoras propias:** posibilidad de incrementar los servicios y funciones desde el código fuente.

Sin embargo, y pese a todas estas ventajas existen una serie de desventajas:

- **Poco intuitivo:** es muy distinto de Windows, lo que dificulta el trabajo a quienes estén acostumbrados a el.
- **Conocimientos:** las actualizaciones requieren en ocasiones tener conocimientos profundos del sistema.
- **Tiempo:** configurar algunos servicios de red requiere de más tiempo que en Windows.

Por último y como sistema gestor de contenidos se va a utilizar SilverStripe debido a su flexibilidad para desarrollar aplicaciones web, además de gestionar sus contenidos.

CAPÍTULO 3

ANÁLISIS DEL SISTEMA

El análisis de la aplicación comienza por la definición de los requisitos de la aplicación extraídos del documento específico donde se detalla la oferta (en nuestro caso el anteproyecto) y de las entrevistas con el cliente. Este paso es fundamental ya que marcará el desarrollo de la aplicación.

Una vez obtenidos los requisitos que deben cumplirse al finalizar la aplicación y entendido mejor el problema, se genera el modelo de datos que contendrá los datos de la aplicación.

Por último, se genera la documentación necesaria para que el programador pueda desarrollar la aplicación. Esta especificación consta de los diagramas UML que detallan el comportamiento de la aplicación. En nuestro caso, no detallamos todos los diagramas existentes ya que creemos que con los que se aportan son suficientes para la comprensión total de la aplicación debido a su sencillez.

3.1 Análisis de requisitos

La definición de requisitos se divide en los funcionales, que son los que describen el comportamiento de la aplicación, y que nos ayudarán a generar los casos de uso así como la realización de pruebas, y los no funcionales que describen aspectos relevantes o restricciones de la aplicación.

A continuación describiremos los requisitos funcionales obtenidos mediante las entrevistas con el cliente, y estudio de los sistemas LETS ya existentes. Estos requisitos han sido agrupados en diferentes apartados con la finalidad de facilitar su comprensión.

3.1.1 Requisitos para la gestión de sistemas

La aplicación de gestión de la moneda complementaria no debe ser un sistema cerrado y único, sino que debe de permitirse la existencia de varios sistemas paralelos, cada uno con la posibilidad de tener su propio tipo de moneda o unidad de intercambio (créditos, horas, etc.).

A partir de esta idea se extraen los siguientes requisitos funcionales:

1.- Debe existir la gestión (creación, edición, borrado y listado) de diferentes tipos de monedas o unidades de intercambio.

2.- Debe existir la gestión de diferentes tipos de sistemas, cada uno asociado a un único tipo de moneda o unidad de intercambio.

3.1.2 Requisitos para la gestión de usuarios

La aplicación para la gestión de moneda complementaria estará compuesta por usuarios que tendrán asociados diferentes sistemas en los que puede participar. Además en cada sistema de los que puede formar parte un usuario le será asignado un crédito o nivel de confianza. Por otro lado todos los datos de registro del usuario deberán ser validados por un administrador del sistema antes de que sean visibles para el resto de usuarios.

Por último se debe de poder restringir los permisos de cada usuario de cara a la realización de algunas operaciones dependiendo de su perfil en el sistema.

A partir de esta idea se extraen los siguientes requisitos funcionales:

3.- Debe existir la posibilidad de que un usuario se registre introduciendo sus datos desde una zona pública de la web (accesible por cualquier usuario no autenticado), en uno de los sistemas en los que este registrado.

4.- Debe existir la posibilidad de que un administrador del sistema pueda activar o desactivar a un usuario previamente registrado.

5.- Debe existir la gestión de los usuarios (datos introducidos por el propio usuario en el registro, sistema al que pertenece, perfil de confianza que posee en dicho sistema, permisos para realizar operaciones, etc.) por parte de un administrador del sistema.

6.- Debe existir la gestión de perfiles de confianza asociado a un usuario y un sistema determinado.

7.- Debe existir la gestión de grupos asociados a un determinado usuario, que definan los permisos y restricciones sobre determinadas operaciones del sistema.

3.1.3 Requisitos para la gestión de servicios

Cada usuario tendrá la posibilidad de publicar ofertas o demandas de servicios, estas podrán ser vistas por cualquier persona no autenticada aunque para solicitarlos se debe estar dado de alta en el sistema concreto al que pertenecen. Además antes de que pueda ser vista por otros usuarios diferentes al autor, todos los servicios deben ser revisados por un administrador del sistema. El administrador podrá acreditar a un usuario para que el mismo no precise validación en sus anuncios, aunque éstos figurarán como “no validados” hasta que no lo haga el administrador.

A partir de esta idea se extraen los siguientes requisitos funcionales:

8.- Debe existir la gestión de servicios (ofertas y demandas) por parte de un usuario.

9.- Debe existir la posibilidad de que un administrador del sistema pueda publicar o eliminar un servicio determinado, además de modificar sus datos en caso de ser necesario.

10.- Debe existir la posibilidad de clonar un servicio (replicar los datos en un nuevo servicio), en caso de no poderse editar por existir peticiones de ese servicio por parte de algún usuario.

11.- Debe existir un área pública (sin necesidad de autenticación) en el sistema donde publicar los servicios existentes.

3.1.4 Requisitos para la gestión de facturas

Los servicios van a tener asociadas facturas, que serán los contratos por los cuales se formaliza la solicitud y realización de los servicios. Éstas, dependiendo de su fase de ejecución podrán encontrarse en estado pendiente de aceptar, rechazadas o realizadas, siendo sólo en el último caso cuando se hará efectivo el cobro del servicio. Además son las facturas y no los servicios los que deben tener asociados los datos de realización del servicio, ya que una misma oferta o demanda se podrá realizar en múltiples ocasiones. Cada factura va a tener asociada dos opiniones, una del que realiza el servicio y otra del que lo recibe. Estas opiniones previamente deberán ser validadas por un administrador del sistema.

A partir de esta idea se extraen los siguientes requisitos funcionales:

12.- Debe existir gestión de facturas, donde la creación se realizará mediante la petición de un determinado servicio.

13.- Debe existir la posibilidad de aceptar o rechazar una factura.

14.- Debe existir la posibilidad de pagar una factura previamente aceptada.

15.- Debe existir un calendario, agenda o algún otro interfaz similar para introducir la fecha de realización del servicio.

16.- Debe existir la posibilidad de crear y modificar opiniones por partes de los interesados en la factura una vez el servicio se encuentre realizado.

17.- Debe existir la posibilidad de aceptar o rechazar una opinión por parte del administrador del sistema, además de poder modificarla.

3.1.5 Requisitos para la gestión de mensajes

Para posibilitar la comunicación entre los miembros del sistema debe existir la posibilidad de que estos se puedan enviar mensajes. Además deben existir mensajes automáticos que se envíen como alertas dependiendo de determinadas acciones. Por ejemplo una alerta se le

puede enviar a un administrador en el momento que tenga pendiente de aceptar una opinión realizada sobre un servicio realizado.

A partir de esta idea se extraen los siguientes requisitos funcionales:

18.- Debe existir una bandeja de entrada donde lleguen los mensajes tanto de otros usuarios como alertas del sistema.

19.- Debe existir la posibilidad de enviar y contestar mensajes a otros usuarios.

3.2 Diagramas de actividades

Los diferentes escenarios identificados en el análisis de requisitos interaccionan entre sí atendiendo a determinadas reglas. Una buena manera de describir dichas interacciones es mediante los diagramas de actividades. Los diagramas de actividades muestran el flujo de trabajo desde el punto de inicio hasta el punto final detallando muchas de las rutas de decisiones que existen en el progreso de eventos contenidos en la actividad. Estos también pueden usarse para detallar situaciones donde el proceso paralelo puede ocurrir en la ejecución de algunas actividades. Los Diagramas de Actividades son útiles para el Modelado de Negocios donde se usan para detallar el proceso involucrado en las actividades de negocio.

A continuación se detallan diagramas de actividades para algunas de los flujos detectados en la toma de requisitos para la aplicación.

3.2.1 Diagrama de actividades para el proceso de Autenticación

En la Figura 3.1 se muestra un diagrama de actividades, en el podemos ver el flujo de acciones necesarios para el proceso de autenticación en la aplicación.

El diagrama se ha dividido en dos partes, la superior donde se definen las acciones que necesitan de interacción por parte de un usuario, y la inferior con las acciones que realiza automáticamente el sistema.

El flujo se inicia cuando un usuario accede a la página de autenticación, que será una página genérica donde se le solicite sus credenciales, en este caso un nombre de usuario y una contraseña. Una vez introducido dichos datos, será el sistema el encargado de validarlos, en

caso de que sean correctos se realizará la autenticación de forma correcta y terminará el flujo de acciones descrito en este diagrama.

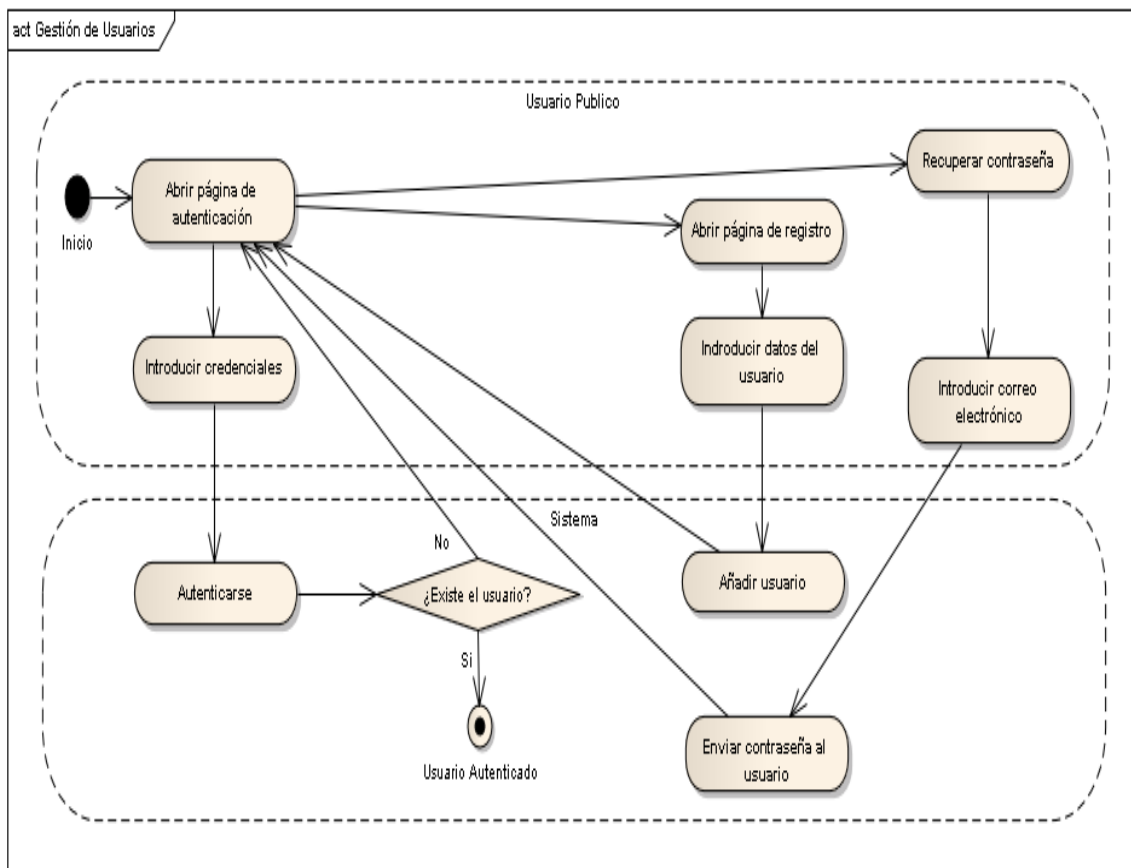


Figura 3.1: Diagrama de actividades para el proceso de Autenticación

Si las credenciales introducidas por el usuario no son correctas, el sistema redirigirá de nuevo a la página de autenticación, donde además de indicar que se ha producido un error, se le dará la posibilidad al usuario de volver a introducir unas nuevas credenciales, o bien recuperar su contraseña en caso de haberla olvidado, o registrarse en el sistema si el usuario aún no lo está.

En caso de seleccionar la opción de recuperar la contraseña, habrá que introducir la dirección de correo electrónico facilitado en el registro del usuario, y en caso de ser correcto, el sistema le enviará la contraseña a dicha dirección, redirigiendo nuevamente a la página de autenticación.

En caso de seleccionar la opción de registro, se redirecciona a una página donde el usuario deba introducir sus datos para que el sistema pueda dar de alta al usuario. Una vez dado de alta se vuelve a redireccionar a la página de autenticación.

3.2.2 Diagrama de actividades para el proceso de Creación de un nuevo Servicio

En la Figura 3.2 podemos ver un proceso simplificado para la creación de un nuevo servicio, en ella se parte de la condición de que el usuario ya que se encuentra autenticado en el sistema sobre el que se va a crear el servicio. Una vez autenticado se selecciona la opción “Crear nuevo servicio” e inserta los datos que se le exigen a través de un formulario. Tras estos pasos ya se ha creado el servicio, aunque solo puede ser visto por el propio autor. Para hacerlo público al resto de usuarios debe ser publicado por un usuario administrador, manteniendo de esta forma un control sobre los servicios que se publican. Además y aunque no se encuentre especificado en el diagrama de actividades, en cualquier momento se puede quitar la publicación del servicio, de forma que quede oculto a cualquier usuario.

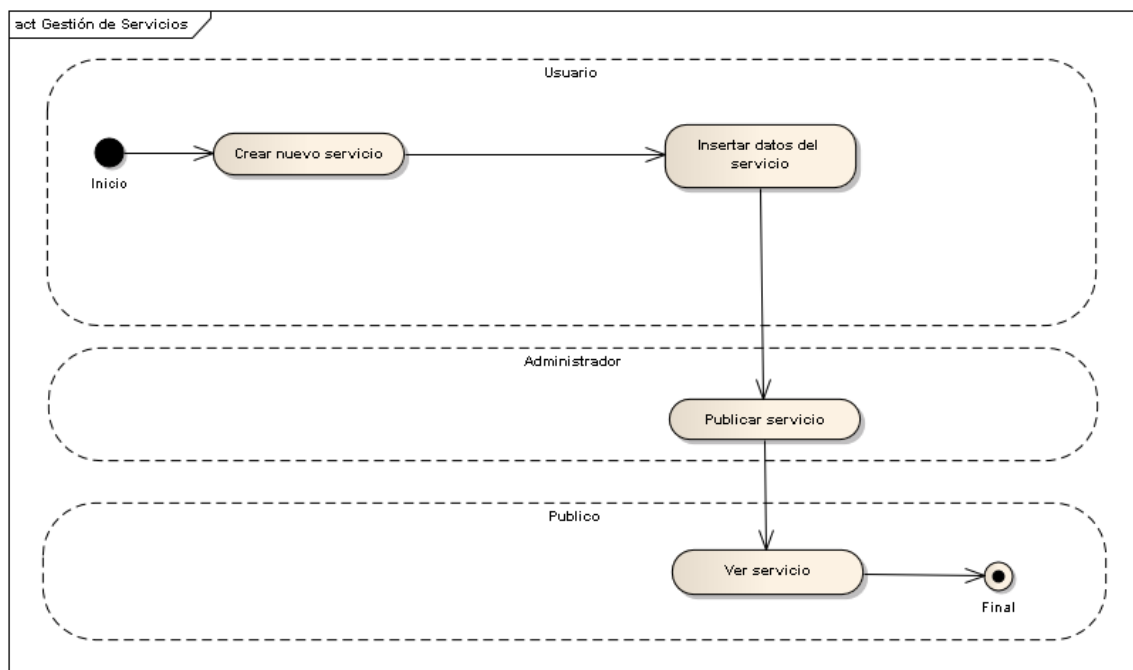


Figura 3.2: Diagrama de actividades para el proceso de Creación de un Servicio

3.2.3 Diagrama de actividades para el proceso Solicitud de un Servicio

En la Figura 3.3 podemos ver el flujo de acciones necesarios para el proceso de solicitud de un servicio existente en la aplicación.

El flujo se inicia cuando un usuario accede a la parte pública de la aplicación donde se listan los servicios disponibles. Selecciona una oferta para ver sus detalles, momento en el que se comprueba si está autenticado, si no lo está se salta al flujo de autenticación. Si por el

contrario el usuario si que se encuentra autenticado se comprueba si pertenece al mismo sistema del servicio seleccionado, si no pertenece a dicho servicio se muestra un mensaje de error, y termina el flujo de tramitación.

Si el usuario se encuentra autenticado y además pertenece al sistema del servicio seleccionado, el sistema cambiará automáticamente a dicho servicio en caso de que aún no lo estuviese, y muestra los datos del servicio.

Una vez en el detalle del servicio, si el usuario tiene crédito suficiente para solicitarlo podrá hacerlo, de lo contrario no podrá realizar ninguna otra acción y por tanto termina el flujo de acciones.

Al solicitar un servicio éste queda pendiente de validación por parte del otro usuario implicado en él, que además de aceptarlo podrá rechazarlo o editarlo. Si decide editarlo de nuevo pasa a estar pendiente de alguna de estas tres acciones pero esta vez por parte del otro usuario.

Una vez rechazado el servicio, ninguno de los dos usuarios implicados en él podrán hacer nada y por tanto se termina el flujo de acciones.

Si se acepta el servicio por alguno de los usuarios queda pendiente de pago. El pago sólo lo puede realizar el usuario que ha recibido el servicio, por tanto el flujo de acciones queda a la espera de que este usuario lo realice. Por último una vez pagado el servicio ambos usuarios podrán opinar sobre el servicio realizado, momento en el que termina el flujo de acciones.

3.3 El modelo de datos

El modelo de datos es una representación que tiene dos componentes: las propiedades estáticas que se definen en un esquema y las propiedades dinámicas que se definen como especificaciones de transacciones, consultas e informes. Un esquema consiste en una definición de todos los tipos de objetos de la aplicación, incluyendo sus atributos, relaciones y restricciones estáticas.

En nuestro caso, el modelo de datos muestra las tablas necesarias para recoger los datos de la aplicación junto con la relación entre ellas por sus claves foráneas, pero en ningún caso es un esquema entidad-relación, sino que se basa en el Perfil de Modelado de Datos del UML que aunque no es actualmente un estándar ratificado; sin embargo tiene un soporte de

industria amplio y es un método útil para conectar la brecha que hay entre el modelado de base de datos relacional convencional y UML.

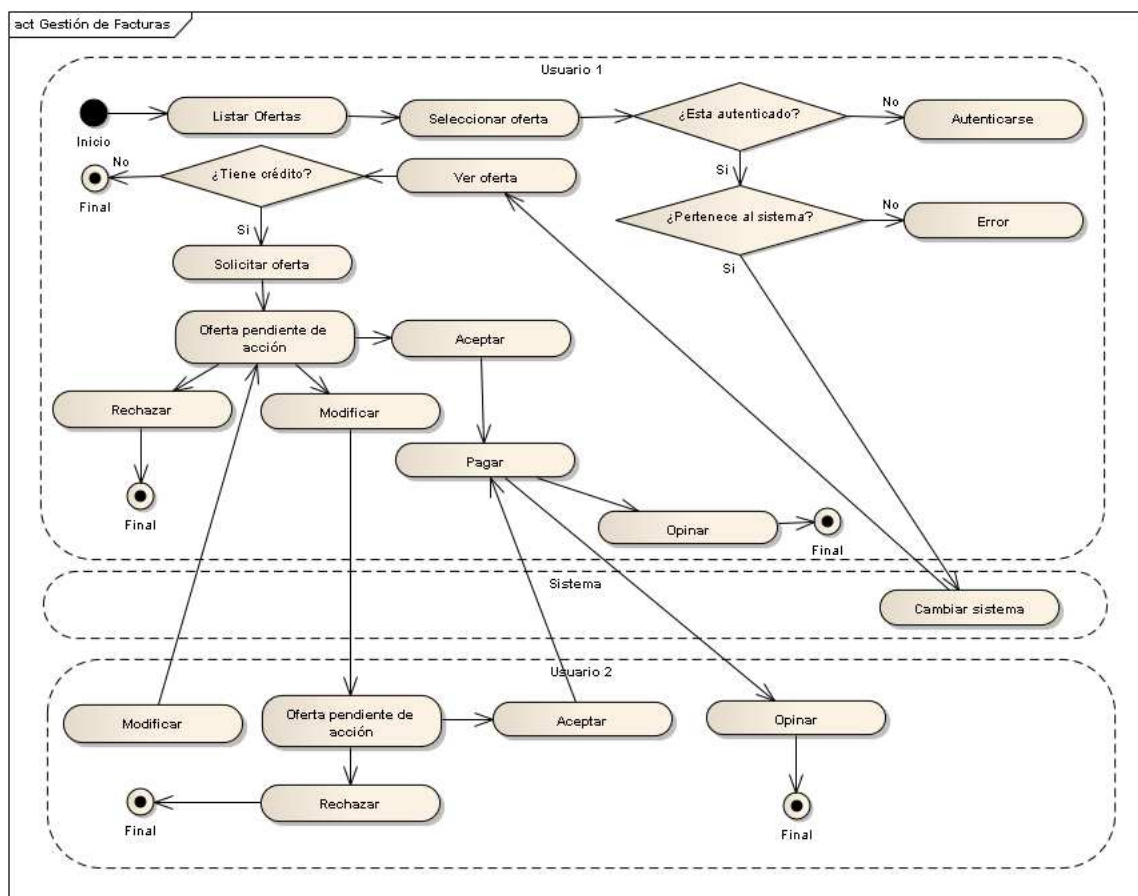


Figura 3.3: Diagrama de actividades para el proceso de Solicitud de un Servicio

3.3.1 Perfil de Modelado de Datos UML

El Modelado de base de datos y diseño de base de datos no son explícitamente cubiertos por las especificaciones del UML, pero se alcanzan con extensiones fáciles de usar y entender para el UML estándar, mapeando los conceptos de base de datos de tablas y relaciones en los conceptos del UML de clases y asociaciones. Estas extensiones también permiten modelar las claves de base de datos, disparadores, limitaciones, RI y otras características de base de datos relacionadas.

Las tareas de modelado de datos típicas son las que se indican en los siguiente apartados.

3.3.1.1 Tablas y columnas

La estructura de modelado básica de una base de datos relacional es la tabla, que representa un conjunto de registros, o filas, con la misma estructura. El elemento de organización básico de una base de datos relacional es la columna. Cada ítem individual de datos ingresado en una base de datos relacional se representa por un valor en una columna de una fila en una tabla.

El perfil de modelado de datos UML representa:

- Tablas como clases estereotipadas; es decir, los elementos clase con un estereotipo de tabla.
- Columnas como atributos estereotipados; es decir, atributos con un estereotipo de columna.

3.3.1.2 Claves de base de datos

Se usan dos tipos de claves para acceder a las tablas: claves primarias y claves foráneas. Una clave primaria identifica únicamente un registro de una tabla, mientras una clave foránea accede los datos en alguna otra tabla relacionada a través de su clave primaria.

Una clave primaria consiste de una o más columnas; una clave primaria simple (columna simple) se identifica como el atributo de una operación estereotipada. Una clave primaria compleja (muchas columnas) se define como la operación estereotipada por si misma.

Una clave foránea es una colección de columnas (atributos) que juntos tienen algún significado operacional (ellos refuerzan una relación a una clave primaria en otra tabla). Las claves foráneas se representan como operaciones con el estereotipo FK; los parámetros de las operaciones se convierten en las columnas involucradas en la clave.

3.3.2 Modelo de datos de la aplicación

En la Figura 3.4 se muestra el modelo de datos del núcleo de la aplicación, este modelo solo contempla las tablas que se han creado para el software de gestión de la moneda complementaria Málaga Común, no contemplando las tablas ya existentes en los *frameworks* y el gestor de contenidos sobre el que se va a desarrollar la aplicación.

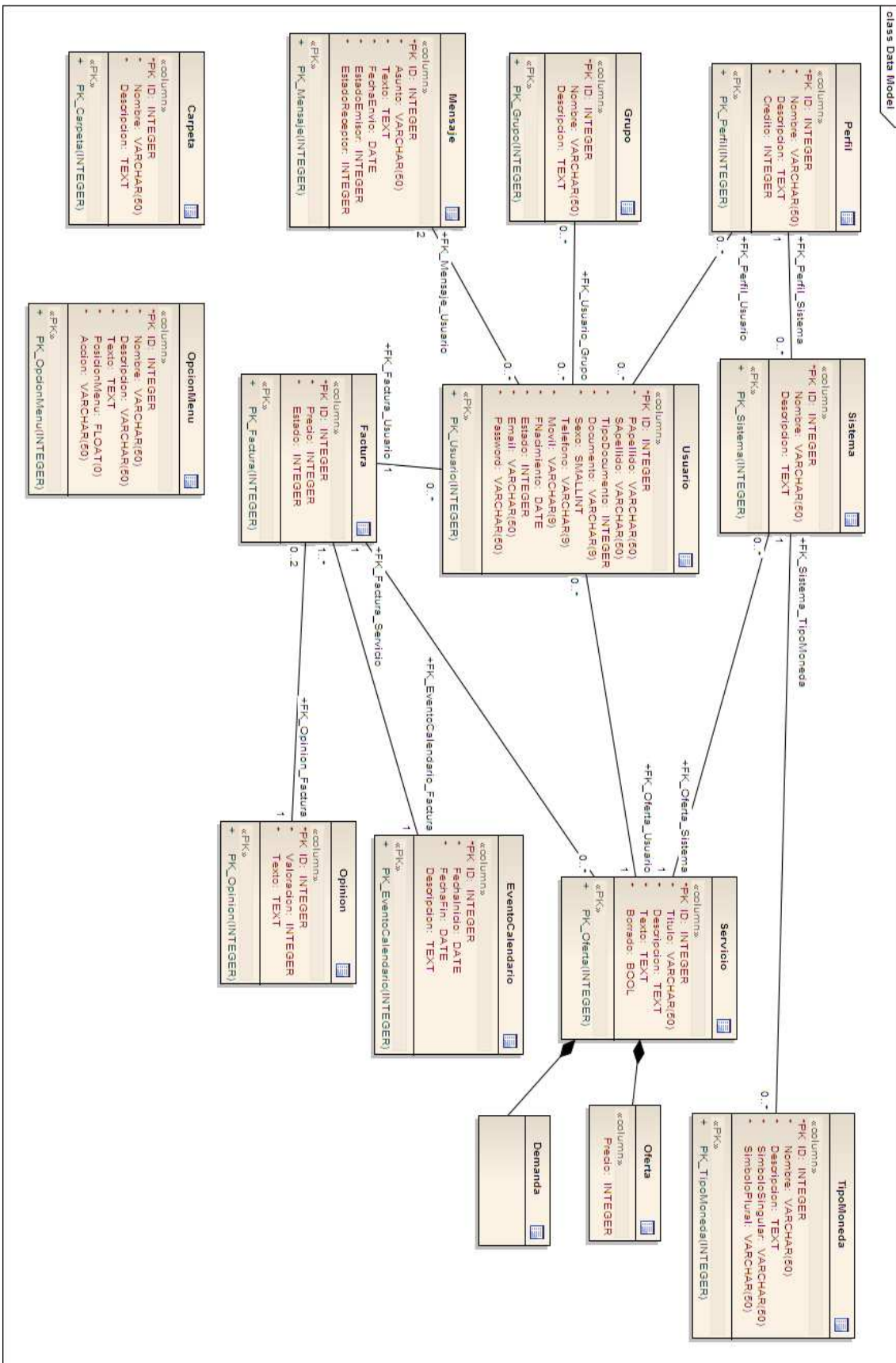


Figura 3.4: Modelo de datos del núcleo de la aplicación

3.4 El modelo de casos de uso

En ingeniería del software, un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. Normalmente, en los casos de usos se evita el empleo de jergas técnicas, prefiriendo en su lugar un lenguaje más cercano al usuario final. En ocasiones, se utiliza a usuarios sin experiencia junto a los analistas para el desarrollo de casos de uso.

En otras palabras, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo la especialización y la generalización son relaciones. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo.

3.4.1 Escenarios para los casos de uso

Para la aplicación de gestión de la moneda complementaria Málaga Común los diferentes escenarios se han agrupado en los casos que se pueden ver en la Figura 3.5.

Los actores involucrados en nuestro sistema, así como la descripción de cada caso de uso y los escenarios que comprende se detallarán en los apartados siguientes.

3.4.2 Descripción de los actores

A continuación se describen los diferentes actores que conforman el sistema.

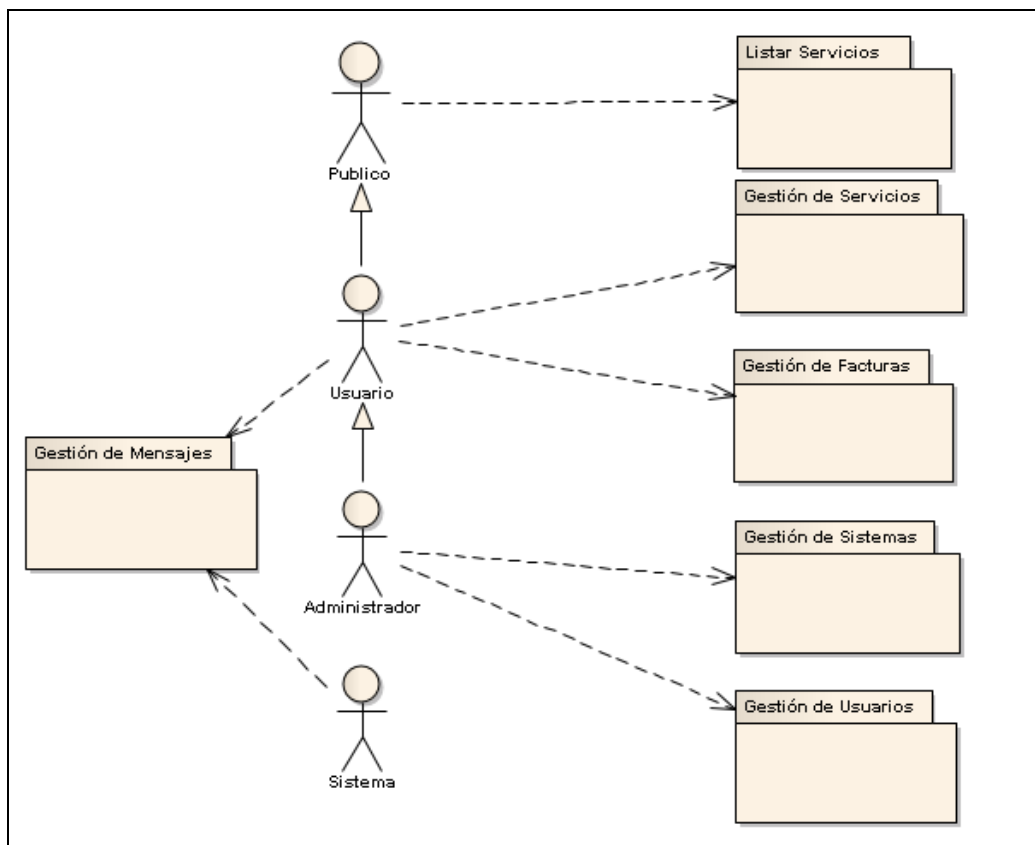


Figura 3.5: Escenarios para los casos de uso

ACT-01	Publico
Descripción	Este actor representa a un usuario no autenticado en el sistema. Es decir, a cualquier persona que visite la web.
ACT-02	Usuario
Descripción	Este actor representa a un usuario autenticado en el sistema. Previamente se ha tenido que registrar en el sistema.
ACT-03	Administrador
Descripción	Este actor representa a un usuario (previamente autenticado) con permisos especiales de administración.
ACT-04	Sistema
Descripción	Este actor representa al sistema y será el encargado de realizar tareas donde no se requiera la intervención de ninguno de los otros actores existentes.

3.4.3 Caso de uso Gestión de Sistemas

Tal como vemos en la Figura 3.6 vamos a describir los distintos casos de uso para la gestión de un sistema determinado.

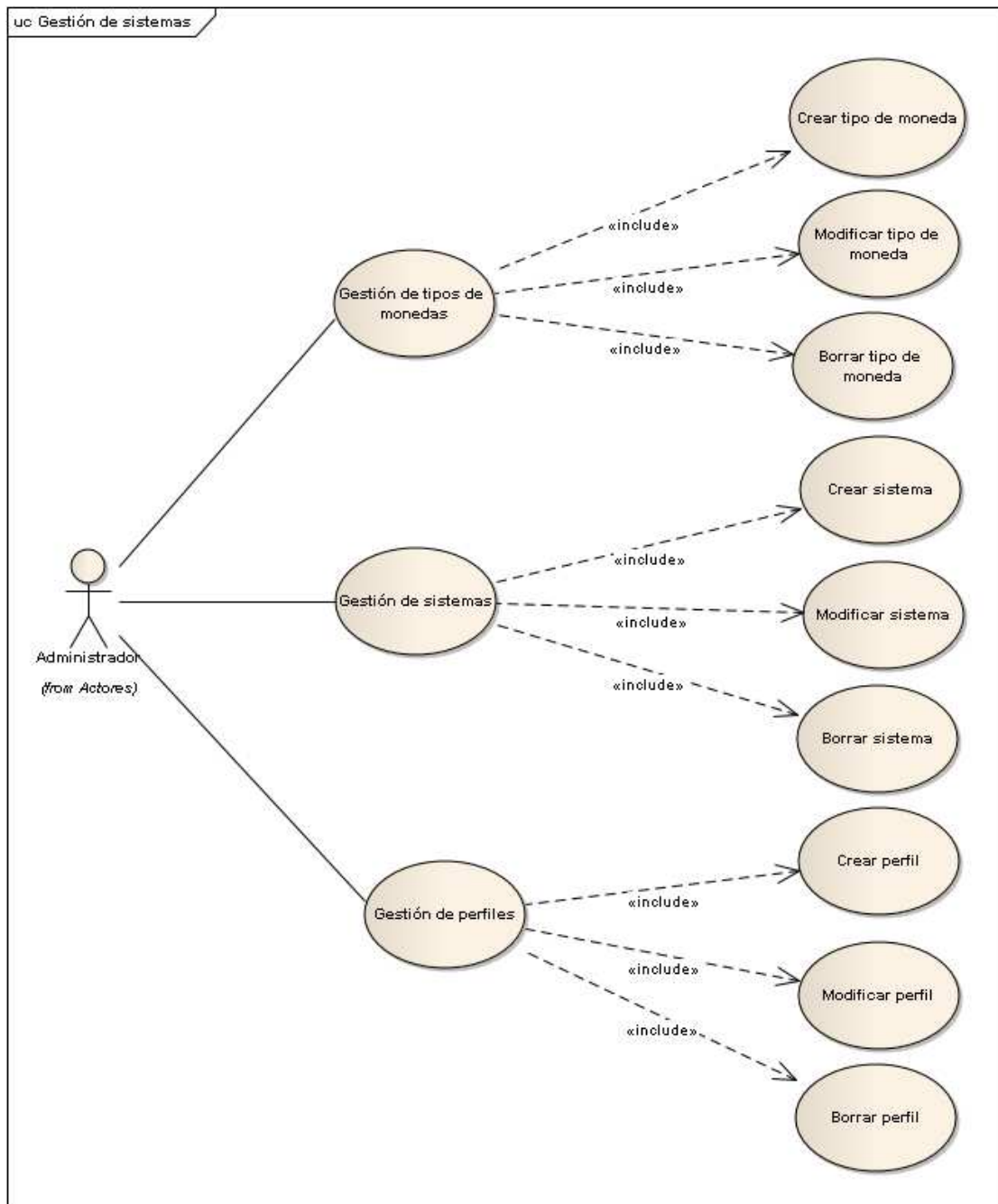


Figura 3.6: Diagrama de casos de usos para el escenario Gestión de Sistemas

3.4.3.1 Casos de uso para la Gestión de tipos de monedas

En la Tabla 3.1, Tabla 3.2, y Tabla 3.3 se describen los casos de uso para crear un nuevo tipo de moneda, editar un tipo de moneda existente, y borrar un tipo de moneda, respectivamente.

Tabla 3.1: Caso de uso para crear un tipo de moneda

Caso de Uso	Crear tipo de moneda
Objetivo	Crea un nuevo tipo de moneda.
Actores:	Administrador (A), Sistema (S)
Pasos:	1.- A: El caso de uso se inicia cuando el Administrador se autentica mediante sus credenciales en el sistema. 2.- A: Selecciona la opción “Tipo de Monedas” dentro del menú de opciones. 3.- A: Selecciona la opción “Crear nuevo” dentro del menú de opciones. 4.- A: Rellena los datos necesarios. 5.- S: Muestra los datos del nuevo tipo de moneda creado.
Extensiones:	4.1.- Los datos introducidos no son correctos: 4.1.1.- S: Indica el error. 4.1.2.- A: Vuelve a introducir datos (paso 4). 4.2.- A: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

Tabla 3.2: Caso de uso para modificar un tipo de moneda

Caso de Uso	Modificar tipo de moneda
Objetivo	Modifica un tipo de moneda.
Actores:	Administrador (A), Sistema (S)
Precondiciones:	El tipo de moneda a modificar debe existir previamente.
Pasos:	1.- A: El caso de uso se inicia cuando el Administrador se autentica mediante sus credenciales en el sistema. 2.- A: Selecciona la opción “Tipo de Monedas” dentro del menú de opciones. 3.- A: Selecciona la opción “Editar” dentro del menú de opciones. 4.- A: Modificar los datos deseados. 5.- S: Muestra los nuevos datos del tipo de moneda modificado.
Extensiones:	4.1.- Los datos introducidos no son correctos: 4.1.1.- S: Indica el error. 4.1.2.- A: Vuelve a modificar los datos (paso 4). 4.2.- A: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

Tabla 3.3: Caso de uso para borrar un tipo de moneda

Caso de Uso	Borrar tipo de moneda
Objetivo	Borrar un tipo de moneda.
Actores:	Administrador (A), Sistema (S)
Precondiciones:	El tipo de moneda a borrar debe existir previamente, y no estar usándose en ningún sistema.
Pasos:	1.- A: El caso de uso se inicia cuando el Administrador se autentica mediante sus credenciales en el sistema. 2.- A: Selecciona la opción “Tipo de Monedas” dentro del menú de opciones. 3.- A: Selecciona la opción “Borrar” dentro del menú de opciones. 4.- S: Pide confirmación sobre el borrado. 5.- S: Borra el tipo de moneda. 6.- S: Retorna a la opción de menú.
Extensiones:	4.1.- El tipo de moneda esta siendo utilizado por algún sistema: 4.1.1.- S: Indica el error. 4.1.2.- S: Retorna a la opción de menú. 4.2.- A: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

3.4.3.2 Casos de uso para la Gestión de sistemas

En la Tabla 3.4, Tabla 3.5, y Tabla 3.6 se describen los casos de uso para crear un nuevo sistema, editar un sistema existente, y borrar un sistema, respectivamente.

Tabla 3.4: Caso de uso para crear un sistema

Caso de Uso	Crear sistema
Objetivo	Crea un nuevo sistema.
Actores:	Administrador (A), Sistema (S)
Pasos:	1.- A: El caso de uso se inicia cuando el Administrador se autentica mediante sus credenciales en el sistema. 2.- A: Selecciona la opción “Sistemas” dentro del menú de opciones. 3.- A: Selecciona la opción “Crear nuevo” dentro del menú de opciones. 4.- A: Rellena los datos necesarios. 5.- S: Muestra los datos del nuevo sistema creado.
Extensiones:	4.1.- Los datos introducidos no son correctos: 4.1.1.- S: Indica el error. 4.1.2.- A: Vuelve a introducir datos (paso 4). 4.2.- A: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

Tabla 3.5: Caso de uso para modificar un sistema

Caso de Uso	Modificar sistema
Objetivo	Modifica un sistema.
Actores:	Administrador (A), Sistema (S)
Precondiciones:	El sistema a modificar debe existir previamente.
Pasos:	1.- A: El caso de uso se inicia cuando el Administrador se autentica mediante sus credenciales en el sistema. 2.- A: Selecciona la opción “Sistemas” dentro del menú de opciones. 3.- A: Selecciona la opción “Editar” dentro del menú de opciones. 4.- A: Modificar los datos deseados. 5.- S: Muestra los nuevos datos del sistema modificado.
Extensiones:	4.1.- Los datos introducidos no son correctos: 4.1.1.- S: Indica el error. 4.1.2.- A: Vuelve a modificar los datos (paso 4). 4.2.- A: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

Tabla 3.6: Caso de uso para borrar un sistema

Caso de Uso	Borrar sistema
Objetivo	Borrar un sistema.
Actores:	Administrador (A), Sistema (S)
Precondiciones:	El sistema a borrar debe existir previamente.
Pasos:	1.- A: El caso de uso se inicia cuando el Administrador se autentica mediante sus credenciales en el sistema. 2.- A: Selecciona la opción “Sistemas” dentro del menú de opciones. 3.- A: Selecciona la opción “Borrar” dentro del menú de opciones. 4.- S: Pide confirmación sobre el borrado. 5.- S: Borra el sistema. 6.- S: Retorna a la opción de menú.
Extensiones:	4.1.- El sistema esta siendo utilizado: 4.1.1.- S: Indica el error. 4.1.2.- S: Retorna a la opción de menú. 4.2.- A: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

2.4.3.3 Casos de uso para la Gestión de perfiles

En la Tabla 3.7, Tabla 3.8, y Tabla 3.9 se describen los casos de uso para crear un nuevo perfil, editar un perfil existente, y borrar un perfil, respectivamente.

Tabla 3.7: Caso de uso para crear un perfil

Caso de Uso	Crear perfil
Objetivo	Crea un nuevo perfil.
Actores:	Administrador (A), Sistema (S)
Pasos:	1.- A: El caso de uso se inicia cuando el Administrador se autentica mediante sus credenciales en el sistema. 2.- A: Selecciona la opción “Perfiles” dentro del menú de opciones. 3.- A: Selecciona la opción “Crear nuevo” dentro del menú de opciones. 4.- A: Rellena los datos necesarios. 5.- S: Muestra los datos del nuevo perfil creado.
Extensiones:	4.1.- Los datos introducidos no son correctos: 4.1.1.- S: Indica el error. 4.1.2.- A: Vuelve a introducir datos (paso 4). 4.2.- A: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

Tabla 3.8: Caso de uso para modificar un perfil

Caso de Uso	Modificar perfil
Objetivo	Modifica un perfil.
Actores:	Administrador (A), Sistema (S)
Precondiciones:	El perfil a modificar debe existir previamente.
Pasos:	1.- A: El caso de uso se inicia cuando el Administrador se autentica mediante sus credenciales en el sistema. 2.- A: Selecciona la opción “Perfiles” dentro del menú de opciones. 3.- A: Selecciona la opción “Editar” dentro del menú de opciones. 4.- A: Modificar los datos deseados. 5.- S: Muestra los nuevos datos del perfil modificado.
Extensiones:	4.1.- Los datos introducidos no son correctos: 4.1.1.- S: Indica el error. 4.1.2.- A: Vuelve a modificar los datos (paso 4). 4.2.- A: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

Tabla 3.9: Caso de uso para borrar un perfil

Caso de Uso	Borrar perfil
Objetivo	Borrar un perfil.
Actores:	Administrador (A), Sistema (S)
Precondiciones:	El perfil a borrar debe existir previamente.

Tabla 3.9: Caso de uso para borrar un perfil (continuación)

Caso de Uso	Borrar perfil
Pasos:	1.- A: El caso de uso se inicia cuando el Administrador se autentica mediante sus credenciales en el sistema. 2.- A: Selecciona la opción “Perfiles” dentro del menú de opciones. 3.- A: Selecciona la opción “Borrar” dentro del menú de opciones. 4.- S: Pide confirmación sobre el borrado. 5.- S: Borra el perfil. 6.- S: Retorna a la opción de menú.
Extensiones:	4.1.- El perfil esta siendo utilizado: 4.1.1.- S: Indica el error. 4.1.2.- S: Retorna a la opción de menú. 4.2.- A: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

3.4.4 Caso de uso Gestión de Usuarios

En la Figura 3.7 se ven los distintos diagramas de uso para el escenario de gestión de usuarios. En la Tabla 3.10 y la Tabla 3.11 podemos ver el caso de uso para registrar a un usuario, y el caso de uso para modificar sus datos respectivamente. En Tabla 3.12, y Tabla 3.13 se describen los casos de uso para activar y desactivar a un usuario respectivamente. Por último la Tabla 3.14 y la Tabla 3.15 muestran los casos de uso para asignar perfiles y asignar grupos respectivamente.

Tabla 3.10: Caso de uso para registrar a un usuario

Caso de Uso	Registrarse
Objetivo	Registrar a un nuevo usuario.
Actores:	Publico (P), Administrador (A), Sistema (S)
Precondiciones:	
Pasos:	1.- P: El actor Público entra en la zona de registro de la Web. 2.- P: Rellena los datos necesarios. 3.- S: Le envía un mensaje al Administrador indicando que un usuario se ha registrado en el sistema. 4.- S: Redirige al usuario a la página de autenticación.
Extensiones:	2.1.- Los datos introducidos no son correctos: 2.1.1.- S: Indica el error. 2.1.2.- P: Vuelve a introducir datos (paso 2). 2.2.- P: Cancela la operación: 2.2.1: S: Retorna a la página principal.

Tabla 3.11: Caso de uso para modificar los datos de un usuario

Caso de Uso	Modificar datos de un usuario.
Objetivo	Modifica los datos de un usuario.
Actores:	Administrador (A), Sistema (S)
Precondiciones:	El usuario a modificar debe existir previamente.
Pasos:	1.- A: El caso de uso se inicia cuando el Administrador se autentica mediante sus credenciales en el sistema. 2.- A: Selecciona la opción “Usuarios” dentro del menú de opciones. 3.- A: Selecciona la opción “Editar” dentro del menú de opciones. 4.- A: Modificar los datos deseados. 5.- S: Muestra los nuevos datos del usuario modificado.
Extensiones:	4.1.- Los datos introducidos no son correctos: 4.1.1.- S: Indica el error. 4.1.2.- A: Vuelve a modificar los datos (paso 4). 4.2.- A: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

Tabla 3.12: Caso de uso para activar a un usuario

Caso de Uso	Activar usuario
Objetivo	Activa a un usuario previamente registrado.
Actores:	Administrador (A), Sistema (S)
Precondiciones:	El usuario a activar debe estar previamente registrado.
Pasos:	1.- A: El caso de uso se inicia cuando el Administrador se autentica mediante sus credenciales en el sistema. 2.- A: Selecciona la opción “Usuarios” dentro del menú de opciones. 3.- A: Selecciona la opción “Activar” dentro del menú de opciones. 4.- S: Envía un mensaje al usuario indicando su cuenta se ha activado. 6.- S: Retorna a la opción de menú.
Extensiones:	

Tabla 3.13: Caso de uso para desactivar a un usuario

Caso de Uso	Desactivar usuario
Objetivo	Desactiva a un usuario previamente activado.
Actores:	Administrador (A), Sistema (S)

Tabla 3.13: Caso de uso para desactivar a un usuario (continuación)

Caso de Uso	Desactivar usuario
Precondiciones:	El usuario a desactivar debe estar activo.
Pasos:	1.- A: El caso de uso se inicia cuando el Administrador se autentica mediante sus credenciales en el sistema. 2.- A: Selecciona la opción “Usuarios” dentro del menú de opciones. 3.- A: Selecciona la opción “Desactivar” dentro del menú de opciones. 4.- S: Envía un mensaje al usuario indicando que su cuenta se ha desactivado. 6.- S: Retorna a la opción de menú.

Tabla 3.14: Caso de uso para asignar perfiles

Caso de Uso	Asignar perfiles
Objetivo	Asigna perfiles a un usuario existente.
Actores:	Administrador (A)
Precondiciones:	El usuario debe existir previamente.
Pasos:	1.- A: El caso de uso se inicia cuando el Administrador se autentica mediante sus credenciales en el sistema. 2.- A: Selecciona la opción “Usuarios” dentro del menú de opciones. 3.- A: Selecciona la opción “Editar” dentro del menú de opciones. 4.- A: Se selecciona los perfiles deseados. 6.- S: Retorna a la opción de menú.

Tabla 3.15: Caso de uso para asignar permisos

Caso de Uso	Asignar permisos
Objetivo	Asigna permisos a un usuario existente.
Actores:	Administrador (A)
Precondiciones:	El usuario debe existir previamente.
Pasos:	1.- A: El caso de uso se inicia cuando el Administrador se autentica mediante sus credenciales en el sistema. 2.- A: Selecciona la opción “Usuarios” dentro del menú de opciones. 3.- A: Selecciona la opción “Editar” dentro del menú de opciones. 4.- A: Se selecciona los permisos deseados. 6.- S: Retorna a la opción de menú.

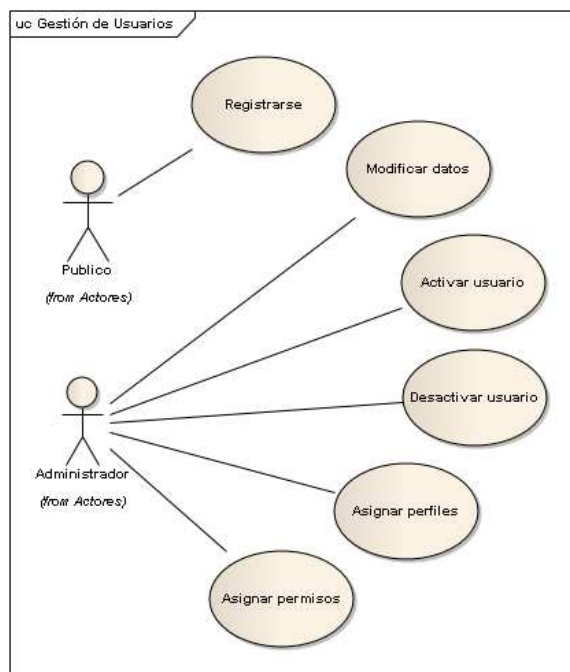


Figura 3.7: Diagrama de casos de usos para el escenario Gestión de Usuarios

3.4.5 Caso de uso Gestión de Servicios

En la Figura 3.8 se ven los distintos diagramas de uso para el escenario de gestión de usuarios. La Tabla 3.16, la Tabla 3.17, y la Tabla 3.18 describen los casos de uso para crear un servicio, modificar un servicio existente, y borrar un servicio respectivamente. A su vez la Tabla 3.19 describe el caso de uso para clonar un servicio. Y la Tabla 3.20, y la Tabla 3.21 describen los casos de uso para publicar y dejar de hacer público un servicio respectivamente.

Tabla 3.16: Caso de uso para crear un servicio

Caso de Uso	Crear servicio
Objetivo	Crear un nuevo servicio.
Actores:	Usuario (U), Sistema (S)
Pasos:	1.- U: El caso de uso se inicia cuando el Usuario se autentica mediante sus credenciales en el sistema. 2.- U: Selecciona la opción "Servicios" dentro del menú de opciones. 3.- U: Selecciona la opción "Crear nuevo" dentro del menú de opciones. 4.- U: Rellena los datos necesarios. 5.- S: Muestra los datos del nuevo perfil creado. 6.- S: Envía un mensaje al Administrador informando de la creación del servicio.

Tabla 3.16: Caso de uso para crear un servicio (continuación)

Caso de Uso	Crear servicio
Extensiones:	4.1.- Los datos introducidos no son correctos: 4.1.1.- S: Indica el error. 4.1.2.- U: Vuelve a introducir datos (paso 4). 4.2.- U: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

Tabla 3.17: Caso de uso para modificar un servicio

Caso de Uso	Modificar servicio
Objetivo	Modifica un servicio.
Actores:	Usuario (U), Sistema (S)
Precondiciones:	El servicio a modificar debe existir previamente. No pueden existir facturas asociadas al servicio a modificar.
Pasos:	1.- U: El caso de uso se inicia cuando el Usuario se autentica mediante sus credenciales en el sistema. 2.- U: Selecciona la opción “Servicios” dentro del menú de opciones. 3.- U: Selecciona la opción “Editar” dentro del menú de opciones. 4.- U: Modificar los datos deseados. 5.- S: Muestra los nuevos datos del perfil modificado. 6.- S: Quita la publicación del servicio. 7.- S: Envía un mensaje al Administrador información sobre la modificación del servicio.
Extensiones:	4.1.- Los datos introducidos no son correctos: 4.1.1.- S: Indica el error. 4.1.2.- U: Vuelve a modificar los datos (paso 4). 4.2.- U: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

Tabla 3.18: Caso de uso para borrar un servicio

Caso de Uso	Borrar servicio
Objetivo	Borrar un servicio.
Actores:	Usuario (U), Sistema (S)
Precondiciones:	El servicio a borrar debe existir previamente. No pueden existir facturas asociadas al servicio a modificar.
Pasos:	1.- U: El caso de uso se inicia cuando el Usuario se autentica mediante sus credenciales en el sistema. 2.- U: Selecciona la opción “Servicios” dentro del menú de opciones. 3.- U: Selecciona la opción “Borrar” dentro del menú de opciones. 4.- S: Pide confirmación sobre el borrado.

Tabla 3.18: Caso de uso para borrar un servicio (continuación)

Caso de Uso	Borrar servicio
Pasos (continuación):	5.- S: Borra el servicio. 6.- S: Retorna a la opción de menú.
Extensiones:	4.1.- El servicio esta siendo utilizado: 4.1.1.- S: Indica el error. 4.1.2.- S: Retorna a la opción de menú. 4.2.- U: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

Tabla 3.19: Caso de uso para clonar un servicio

Caso de Uso	Clonar servicio
Objetivo	Clona un servicio.
Actores:	Usuario (U), Sistema (S)
Precondiciones:	El servicio a clonar debe existir previamente.
Pasos:	1.- U: El caso de uso se inicia cuando el Usuario se autentica mediante sus credenciales en el sistema. 2.- U: Selecciona la opción “Servicios” dentro del menú de opciones. 3.- U: Selecciona la opción “Clonar” dentro del menú de opciones. 6.- S: Retorna a la opción de menú.

Tabla 3.20: Caso de uso para publicar un servicio

Caso de Uso	Publicar servicio
Objetivo	Publicar un servicio.
Actores:	Administrador (A), Sistema (S)
Precondiciones:	El servicio a publicar debe existir previamente. El servicio a publicar no puede estar ya publicado.
Pasos:	1.- A: El caso de uso se inicia cuando el Administrador se autentica mediante sus credenciales en el sistema. 2.- A: Selecciona la opción “Servicios” dentro del menú de opciones. 3.- A: Selecciona la opción “Publicar” dentro del menú de opciones. 6.- S: Retorna a la opción de menú.

Tabla 3.21: Caso de uso para dejar de hacer público un servicio

Caso de Uso	Dejar de hacer público un servicio
Objetivo	Dejar de hacer público un servicio.
Actores:	Usuario (U), Sistema (S)

Tabla 3.21: Caso de uso para dejar de hacer público un servicio (continuación)

Caso de Uso	Dejar de hacer público un servicio
Precondiciones:	El servicio debe existir previamente. El servicio debe estar publicado.
Pasos:	1.- U: El caso de uso se inicia cuando el Usuario se autentica mediante sus credenciales en el sistema. 2.- U: Selecciona la opción “Servicios” dentro del menú de opciones. 3.- U: Selecciona la opción “Despublicar” dentro del menú de opciones. 6.- S: Retorna a la opción de menú.

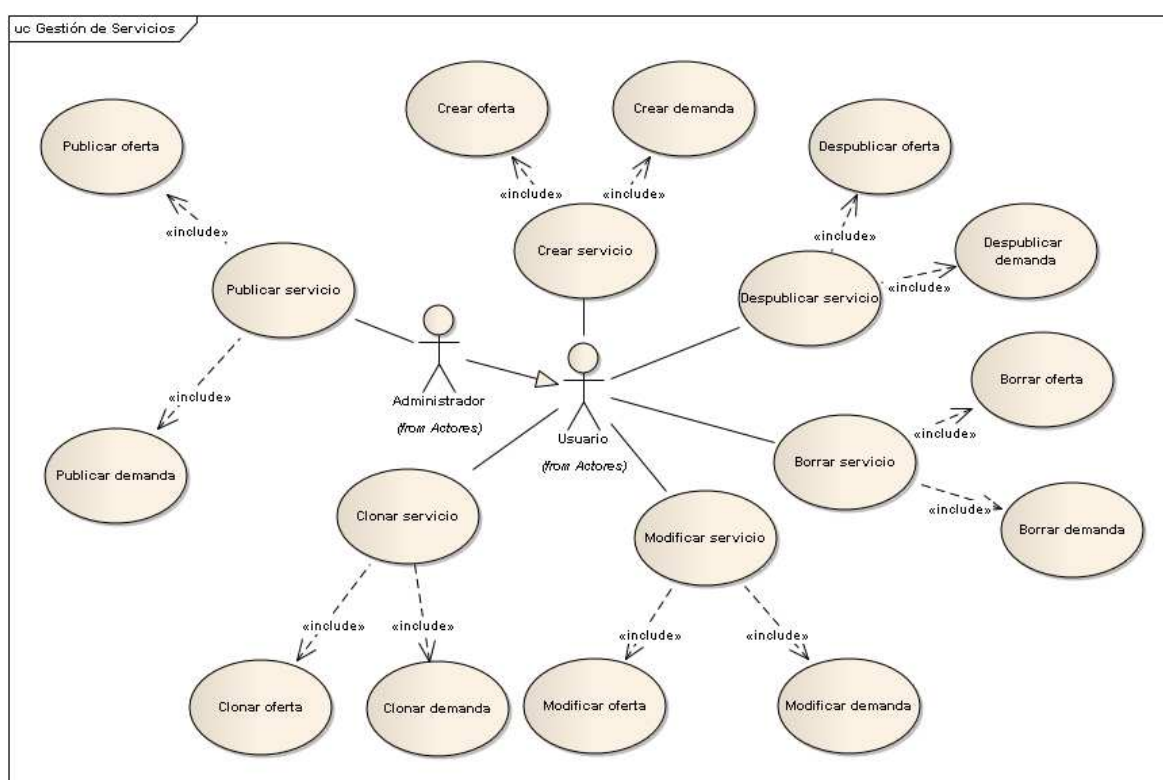


Figura 3.8: Diagrama de casos de usos para el escenario Gestión de Servicios

3.4.6 Caso de uso Gestión de Facturas

En la Figura 3.9 podemos ver los casos de uso para gestionar las facturas en el sistema. La Tabla 3.22, la Tabla 3.23, la Tabla 3.24, la Tabla 3.25, y la Tabla 3.26 muestran los casos de uso para crear una factura, modificar una existente, aceptarla, rechazarla y pagarla respectivamente. A si mismo la Tabla 3.27, la Tabla 3.28, la Tabla 3.29, la Tabla 3.30, y la Tabla 3.31 muestran los casos de uso para crear una opinión sobre una factura realizada, modificarla, borrarla, hacerla publica, y dejar de hacerla publica.

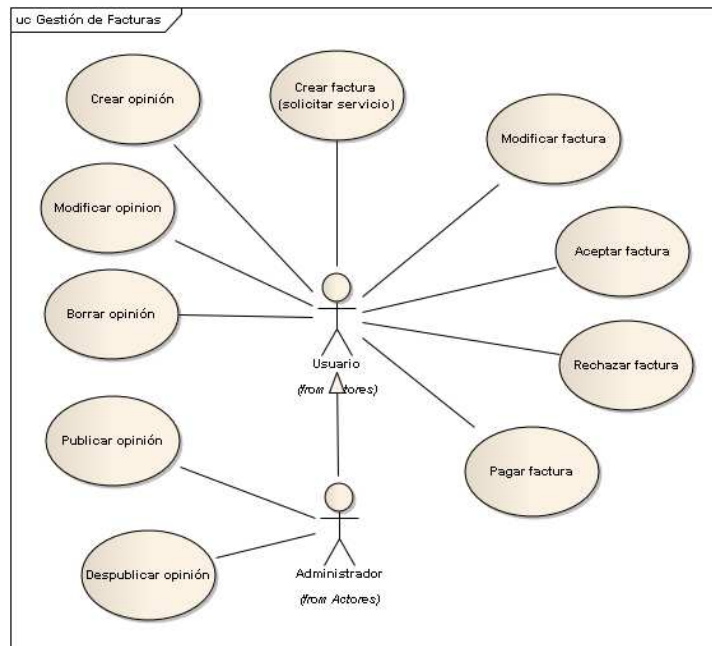


Figura 3.9: Diagrama de casos de usos para el escenario Gestión de Facturas

Tabla 3.22: Caso de uso para crear una factura

Caso de Uso	Crear factura
Objetivo	Crea una nueva factura (solicita un servicio).
Actores:	Usuario (U), Sistema (S)
Precondiciones:	El usuario que debe realizar el pago debe tener crédito suficiente, para solicitar el servicio.
Pasos:	<ol style="list-style-type: none"> 1.- U: El caso de uso se inicia cuando el Usuario se autentica mediante sus credenciales en el sistema. 2.- U: Selecciona la opción "Servicios" dentro del menú de opciones. 3.- U: Selecciona la opción "Solicitar servicio" dentro del menú de opciones. 4.- U: Rellena los datos necesarios. 5.- S: Muestra los datos de la factura creada. 6.- S: Envía un mensaje al otro usuario implicado en el servicio informando de la creación de la factura.
Extensiones:	<ol style="list-style-type: none"> 4.1.- Los datos introducidos no son correctos: <ol style="list-style-type: none"> 4.1.1.- S: Indica el error. 4.1.2.- U: Vuelve a introducir datos (paso 4). 4.2.- U: Cancela la operación: <ol style="list-style-type: none"> 4.2.1: S: Retorna a la opción de menú.

Tabla 3.23: Caso de uso para modificar una factura

Caso de Uso	Modificar factura
Objetivo	Modifica una factura.

Tabla 3.23: Caso de uso para modificar una factura (continuación)

Caso de Uso	Modificar factura
Actores:	Usuario (U), Sistema (S)
Precondiciones:	La factura a modificar debe existir previamente. La factura aún no debe estar aceptada.
Pasos:	1.- U: El caso de uso se inicia cuando el Usuario se autentica mediante sus credenciales en el sistema. 2.- U: Selecciona la opción “Facturas” dentro del menú de opciones. 3.- U: Selecciona la opción “Editar” dentro del menú de opciones. 4.- U: Modificar los datos deseados. 5.- S: Muestra los nuevos datos de la factura modificada. 6.- S: Envía un mensaje al otro usuario implicado en la factura información sobre la modificación de la factura.
Extensiones:	4.1.- Los datos introducidos no son correctos: 4.1.1.- S: Indica el error. 4.1.2.- U: Vuelve a modificar los datos (paso 4). 4.2.- U: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

Tabla 3.24: Caso de uso para aceptar una factura

Caso de Uso	Aceptar factura
Objetivo	Acepta una factura.
Actores:	Usuario (U), Sistema (S)
Precondiciones:	La factura a aceptar debe existir previamente. La factura aún no debe haber sido aceptada ni rechazada.
Pasos:	1.- U: El caso de uso se inicia cuando el Usuario se autentica mediante sus credenciales en el sistema. 2.- U: Selecciona la opción “Facturas” dentro del menú de opciones. 3.- U: Selecciona la opción “Aceptar” dentro del menú de opciones. 6.- S: Retorna a la opción de menú. 7.- S: Envía un mensaje al otro usuario implicado en la factura informando de su nuevo estado.
Extensiones:	

Tabla 3.25: Caso de uso para rechazar una factura

Caso de Uso	Rechazar factura
Objetivo	Rechaza una factura.
Actores:	Usuario (U), Sistema (S)
Precondiciones:	La factura a rechazar debe existir previamente. La factura aún no debe haber sido aceptada ni rechazada.
Pasos:	1.- U: El caso de uso se inicia cuando el Usuario se autentica mediante sus credenciales en el sistema. 2.- U: Selecciona la opción “Facturas” dentro del menú de opciones. 3.- U: Selecciona la opción “Rechazar” dentro del menú de opciones. 6.- S: Retorna a la opción de menú. 7.- S: Envía un mensaje al otro usuario implicado en la factura informando de su nuevo estado.
Extensiones:	

Tabla 3.26: Caso de uso para pagar una factura

Caso de Uso	Pagar factura
Objetivo	Paga una factura.
Actores:	Usuario (U), Sistema (S)
Precondiciones:	La factura a pagar debe existir previamente. La factura debe haber sido aceptada previamente.
Pasos:	1.- U: El caso de uso se inicia cuando el Usuario se autentica mediante sus credenciales en el sistema. 2.- U: Selecciona la opción “Facturas” dentro del menú de opciones. 3.- U: Selecciona la opción “Pagar” dentro del menú de opciones. 6.- S: Retorna a la opción de menú. 7.- S: Envía un mensaje al otro usuario implicado en la factura informando de su nuevo estado. 8.- S: Resta el crédito de la factura al usuario receptor del servicio y se lo añade al usuario prestador del servicio.
Extensiones:	

Tabla 3.27: Caso de uso para crear una opinión

Caso de Uso	Crear opinión
Objetivo	Crea una nueva opinión.
Actores:	Usuario (U), Sistema (S)
Precondiciones:	La factura sobre la que se va a opinar debe haber sido previamente pagada.

Tabla 3.27: Caso de uso para crear una opinión (continuación)

Caso de Uso	Crear opinión
Objetivo	Crea una nueva opinión.
Actores:	Usuario (U), Sistema (S)
Precondiciones:	La factura sobre la que se va a opinar debe haber sido previamente pagada.

Tabla 3.28: Caso de uso para modificar una opinión

Caso de Uso	Modificar opinión
Objetivo	Modifica una opinión sobre una factura.
Actores:	Usuario (U), Sistema (S)
Precondiciones:	La opinión a modificar debe existir previamente.
Pasos:	<ol style="list-style-type: none"> 1.- U: El caso de uso se inicia cuando el Usuario se autentica mediante sus credenciales en el sistema. 2.- U: Selecciona la opción “Opiniones” dentro del menú de opciones. 3.- U: Selecciona la opción “Editar” dentro del menú de opciones. 4.- U: Modificar los datos deseados. 5.- S: Muestra los nuevos datos de la opinión. 6.- S: Despublica la opinión. 7.- S: Envía un mensaje al Administrador información sobre la modificación de la opinión.
Extensiones:	<ol style="list-style-type: none"> 4.1.- Los datos introducidos no son correctos: <ol style="list-style-type: none"> 4.1.1.- S: Indica el error. 4.1.2.- U: Vuelve a modificar los datos (paso 4). 4.2.- U: Cancela la operación: <ol style="list-style-type: none"> 4.2.1: S: Retorna a la opción de menú.

Tabla 3.29: Caso de uso para borrar una opinión

Caso de Uso	Borrar opinión
Objetivo	Borrar una opinión.
Actores:	Usuario (U), Sistema (S)
Precondiciones:	La opinión a borrar debe existir previamente.
Pasos:	<ol style="list-style-type: none"> 1.- U: El caso de uso se inicia cuando el Usuario se autentica mediante sus credenciales en el sistema. 2.- U: Selecciona la opción “Opiniones” dentro del menú de opciones. 3.- U: Selecciona la opción “Borrar” dentro del menú de opciones. 4.- S: Pide confirmación sobre el borrado. 5.- S: Borra la opinión.

Tabla 3.29: Caso de uso para borrar una opinión (continuación)

Caso de Uso	Borrar opinión
Pasos (continuación):	6.- S: Retorna a la opción de menú.
Extensiones:	4.1.- U: Cancela la operación: 4.1.1: S: Retorna a la opción de menú.

Tabla 3.30: Caso de uso para publicar una opinión

Caso de Uso	Publicar opinión
Objetivo	Publica una opinión.
Actores:	Administrador (A), Sistema (S)
Precondiciones:	La opinión a publicar debe existir previamente. La opinión a publicar no puede estar ya publicada.
Pasos:	1.- A: El caso de uso se inicia cuando el Administrador se autentica mediante sus credenciales en el sistema. 2.- A: Selecciona la opción “Opiniones” dentro del menú de opciones. 3.- A: Selecciona la opción “Publicar” dentro del menú de opciones. 6.- S: Retorna a la opción de menú.
Extensiones:	

Tabla 3.31: Caso de uso para eliminar una opinión publicada

Caso de Uso	Eliminar opinión ya publicada
Objetivo	Una opinión deja de estar publicada.
Actores:	Administrador (A), Sistema (S)
Precondiciones:	La opinión a que deja de estar publicada debe existir previamente. La opinión a que deja de estar publicada debe estar publicada.
Pasos:	1.- A: El caso de uso se inicia cuando el Administrador se autentica mediante sus credenciales en el sistema. 2.- A: Selecciona la opción “Opiniones” dentro del menú de opciones. 3.- A: Selecciona la opción que va a dejar de estar publicada dentro del menú de opciones. 6.- S: Retorna a la opción de menú.
Extensiones:	

3.4.7 Caso de uso Gestión de Mensajes

En la Figura 3.10 podemos ver los casos de uso para la gestión de los mensajes.

La Tabla 3.32, la Tabla 3.33, y la Tabla 3.34 muestran los casos de uso para crear, modificar, y borrar un mensaje del sistema respectivamente. Mientras que la Tabla 3.35 muestra el caso de uso para enviar un mensaje.

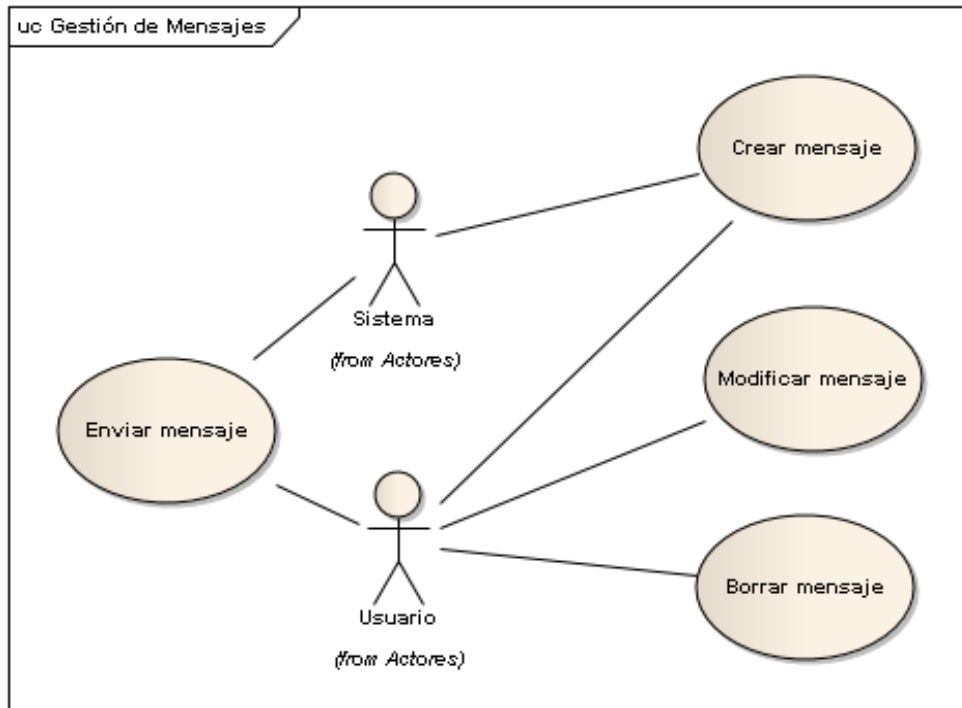


Figura 3.10: Diagrama de casos de usos para el escenario Gestión de Mensajes

Tabla 3.32: Caso de uso para crear un mensaje

Caso de Uso	Crear mensaje
Objetivo	Crea un nuevo mensaje.
Actores:	Usuario (U), Sistema (S)
Precondiciones:	
Pasos:	1.- U: El caso de uso se inicia cuando el Usuario se autentica mediante sus credenciales en el sistema. 2.- U: Selecciona la opción “Mensajes en Borrador” dentro del menú de opciones. 3.- U: Selecciona la opción “Crear nuevo” dentro del menú. 4.- U: Rellena los datos necesarios. 5.- S: Muestra los datos del nuevo mensaje creado.
Extensiones:	4.1.- Los datos introducidos no son correctos: 4.1.1.- S: Indica el error. 4.1.2.- U: Vuelve a introducir datos (paso 4). 4.2.- U: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

Tabla 3.33: Caso de uso para modificar un mensaje

Caso de Uso	Modificar mensaje
Objetivo	Modifica un mensaje.
Actores:	Usuario (U), Sistema (S)
Precondiciones:	El mensaje a modificar debe existir previamente.
Pasos:	1.- U: El caso de uso se inicia cuando el Usuario se autentica mediante sus credenciales en el sistema. 2.- U: Selecciona la opción “Mensajes en Borrador” dentro del menú de opciones. 3.- U: Selecciona la opción “Editar” dentro del menú de opciones. 4.- U: Modificar los datos deseados. 5.- S: Muestra los nuevos datos del mensaje modificado.
Extensiones:	4.1.- Los datos introducidos no son correctos: 4.1.1.- S: Indica el error. 4.1.2.- U: Vuelve a modificar los datos (paso 4). 4.2.- U: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

Tabla 3.34: Caso de uso para borrar un mensaje

Caso de Uso	Borrar mensaje
Objetivo	Borrar un mensaje.
Actores:	Usuario (U), Sistema (S)
Precondiciones:	El mensaje a borrar debe existir previamente.
Pasos:	1.- U: El caso de uso se inicia cuando el Usuario se autentica mediante sus credenciales en el sistema. 2.- U: Selecciona la opción “Mensajes” dentro del menú de opciones. 3.- U: Selecciona la opción “Borrar” dentro del menú de opciones. 4.- S: Pide confirmación sobre el borrado. 5.- S: Borra el mensaje. 6.- S: Retorna a la opción de menú.
Extensiones:	4.1.- U: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

Tabla 3.35: Caso de uso para enviar un mensaje

Caso de Uso	Enviar mensaje
Objetivo	Envía un mensaje.
Actores:	Usuario (U), Sistema (S)
Precondiciones:	El mensaje a enviar debe existir previamente.

Tabla 3.35: Caso de uso para enviar un mensaje (continuación)

Caso de Uso	Enviar mensaje
Pasos:	1.- U: El caso de uso se inicia cuando el Usuario se autentica mediante sus credenciales en el sistema. 2.- U: Selecciona la opción "Mensajes en Borrador" dentro del menú de opciones. 3.- U: Selecciona la opción "Enviar" dentro del menú de opciones. 4.- U: Selecciona el destinatario. 5.- S: Envía el mensaje. 6.- S: Retorna a la opción de menú.
Extensiones:	4.1.- U: Cancela la operación: 4.2.1: S: Retorna a la opción de menú.

3.5 Diagrama de clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

En nuestro caso y dada la extensión del diagrama de clases se va a dividir en diferentes escenarios atendiendo a sus funcionalidades (Figura 3.11).

3.5.1 Diagrama de clases para el escenario Componentes

El gestor de contenidos utilizado para el desarrollo de la aplicación posee un ORM (Object-Relational Mapping) o "mapeo de objetos a bases de datos", que está formado por objetos que permiten acceder a los datos y que contienen en sí mismos el código necesario para hacerlo. Además, a través del *Framework Picasso* estos objetos también contendrán la información necesaria para su edición.

En el escenario Componentes que se puede ver en la Figura 3.12 se definen las clases que utilizan el ORM para la persistencia de datos. Puesto que se va a desarrollar bajo el gestor de contenidos SilverStripe y el Framework de desarrollo Picasso, estas clases se adaptan al modelo de programación propuesto por estas herramientas (capítulo 4), siendo cada clase por sí sola lo único que se necesita para acceder, editar y borrar los datos que definen.

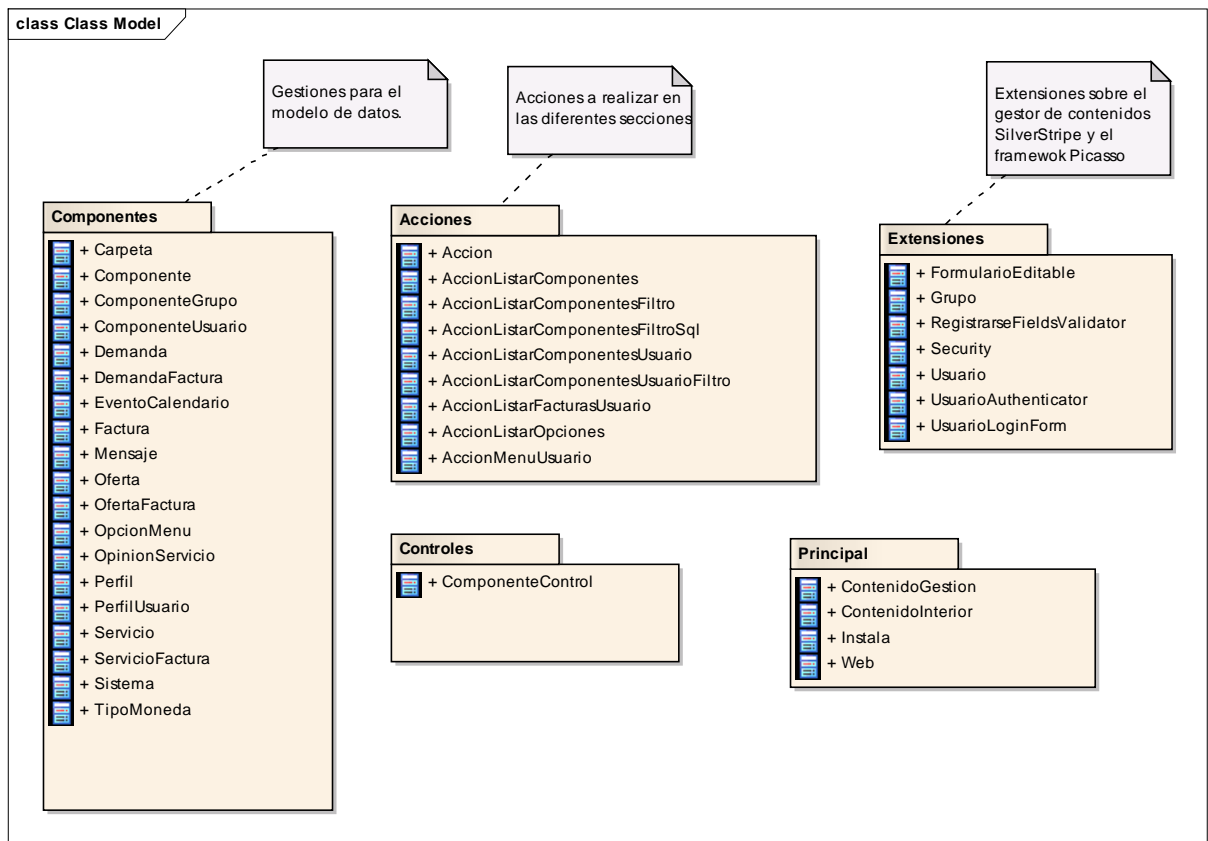


Figura 3.11: Diagrama de clases de los distintos escenarios

3.5.2 Diagrama de clases para el escenario Acciones

En el escenario definido como Acciones (Figura 3.13) se van a encontrar las clases encargadas de realizar acciones sobre los componentes del modelo. Siempre teniendo en cuenta que las acciones de inserción, modificación y obtención ya las llevan implícita los componentes y no son necesarias programarlas.

3.5.3 Diagrama de clases para el escenario Extensiones

En el escenario Extensiones (Figura 3.14) se encuentran clases que extienden el funcionamiento del gestor de contenidos SilverStripe y el Framework Picasso.

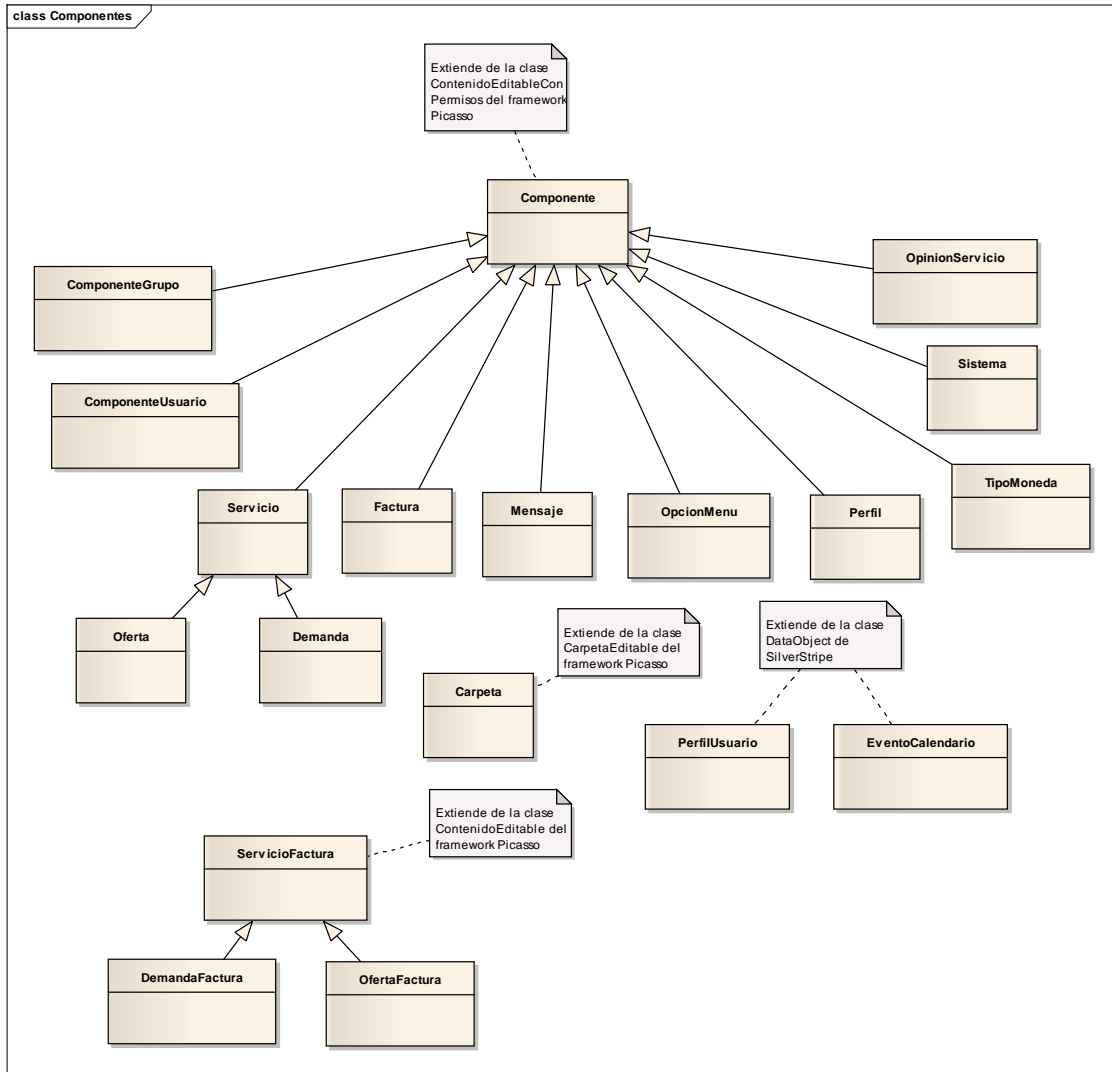


Figura 3.12: Diagrama de clases para el escenario de Componentes

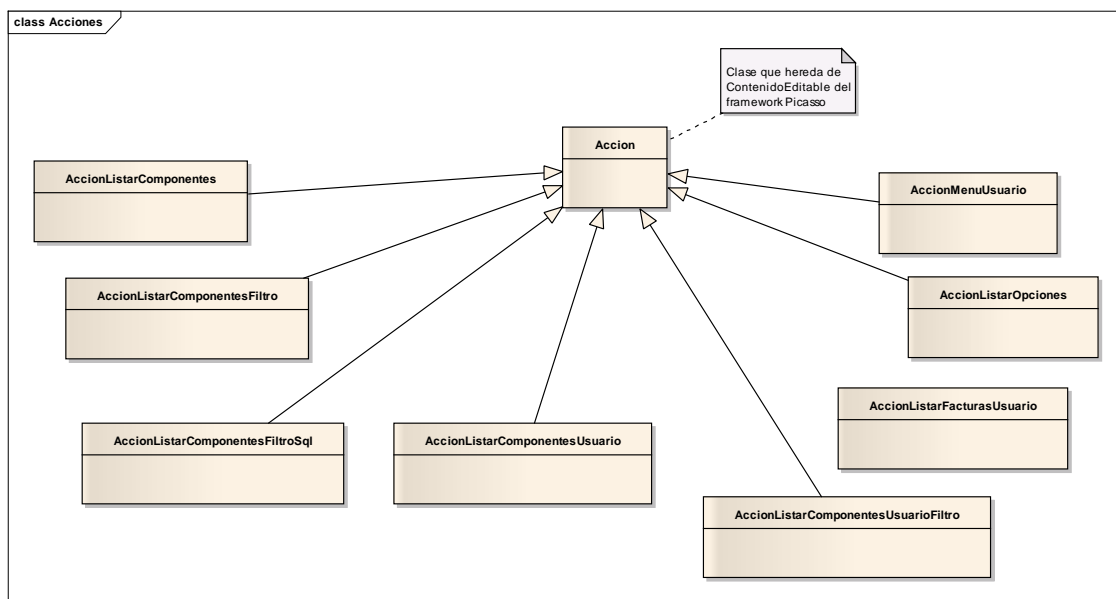


Figura 3.13: Diagrama de clases para el escenario de Acciones

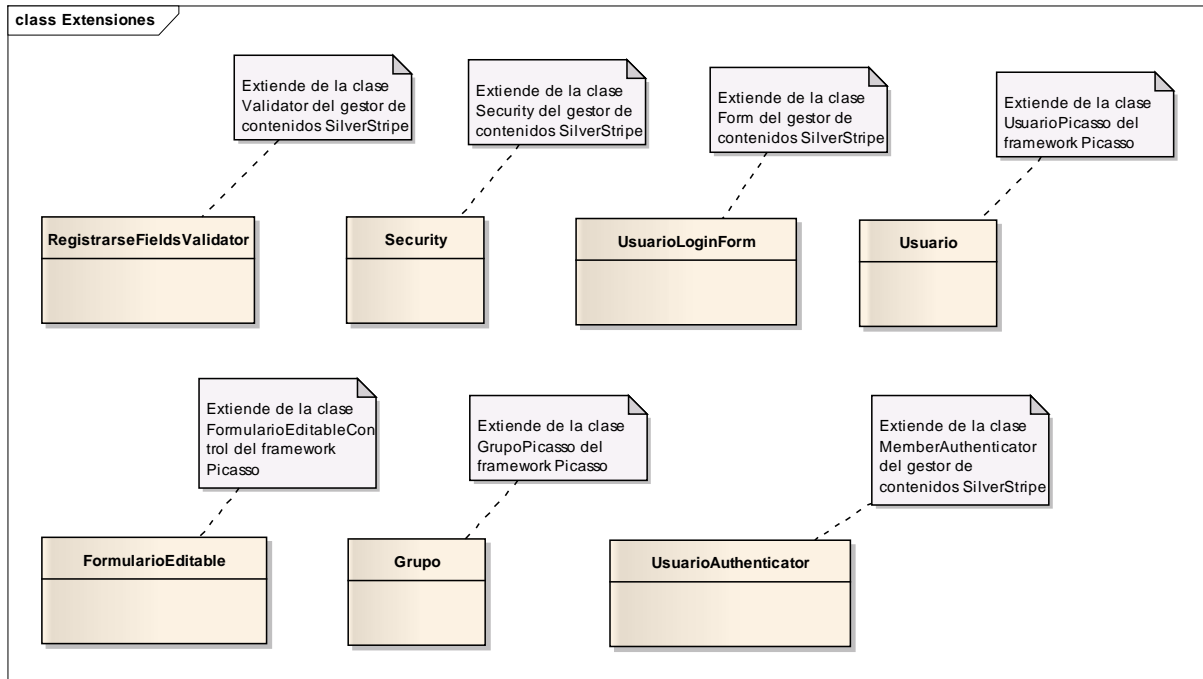


Figura 3.14: Diagrama de clases para el escenario de Extensiones

3.5.4 Diagrama de clases para el escenario Controles

La arquitectura de la aplicación está basada en MVC (Modelo-Vista-Controlador). En nuestro caso, el modelo serán las clases contenidas en el escenario Componentes, las vistas se generarán a partir de las clases del escenario Acciones, y entre estas se situará el control, que serán las clases pertenecientes a este escenario (Figura 3.15).

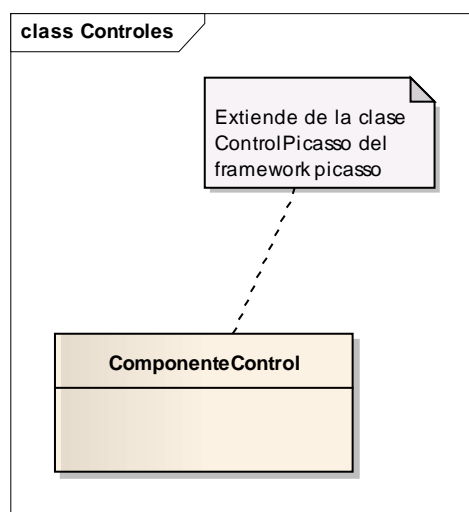


Figura 3.15: Diagrama de clases para el escenario de Controles

3.5.5 Diagrama de clases para el escenario Principal

En este escenario (Figura 3.16) se encuentran clases genéricas utilizadas con distintas finalidades.

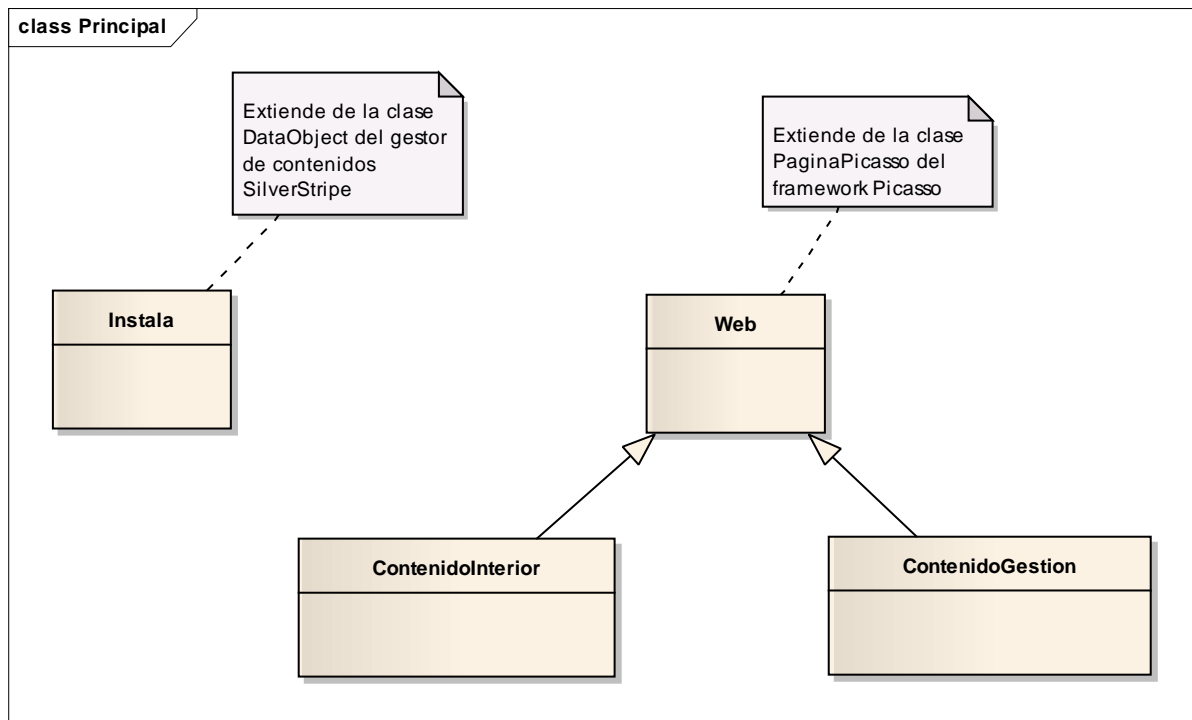


Figura 3.16: Diagrama de clases para el escenario de Principal

3.6 Diagramas de secuencias

El diagrama de interacción representa la forma en como un Cliente (Actor) u Objetos (Clases) se comunican entre si en petición a un evento. Esto implica recorrer todas las secuencias de llamadas, de donde se obtienen las responsabilidades claramente.

Un tipo de diagrama de interacción es el diagrama de secuencias, que muestra de forma explícita la secuencia de los mensajes intercambiados por los objetos. Por cada caso de uso se especifican un diagrama de secuencia, normalmente se especifica un diagrama de secuencia para el escenario normal y uno o más para el escenario de excepción.

Puesto que el desarrollo de nuestra aplicación esta basado en un gestor de contenidos web y un framework de desarrollo la mayoría de las operaciones se realizan de forma similar, por tanto en este apartado nos centraremos en una serie de diagramas genéricos.

3.6.1 Diagrama de secuencias para la creación y modificación de datos

En la Figura 3.17 podemos ver el diagrama de secuencias para la creación o modificación de los datos de un componente de la aplicación.

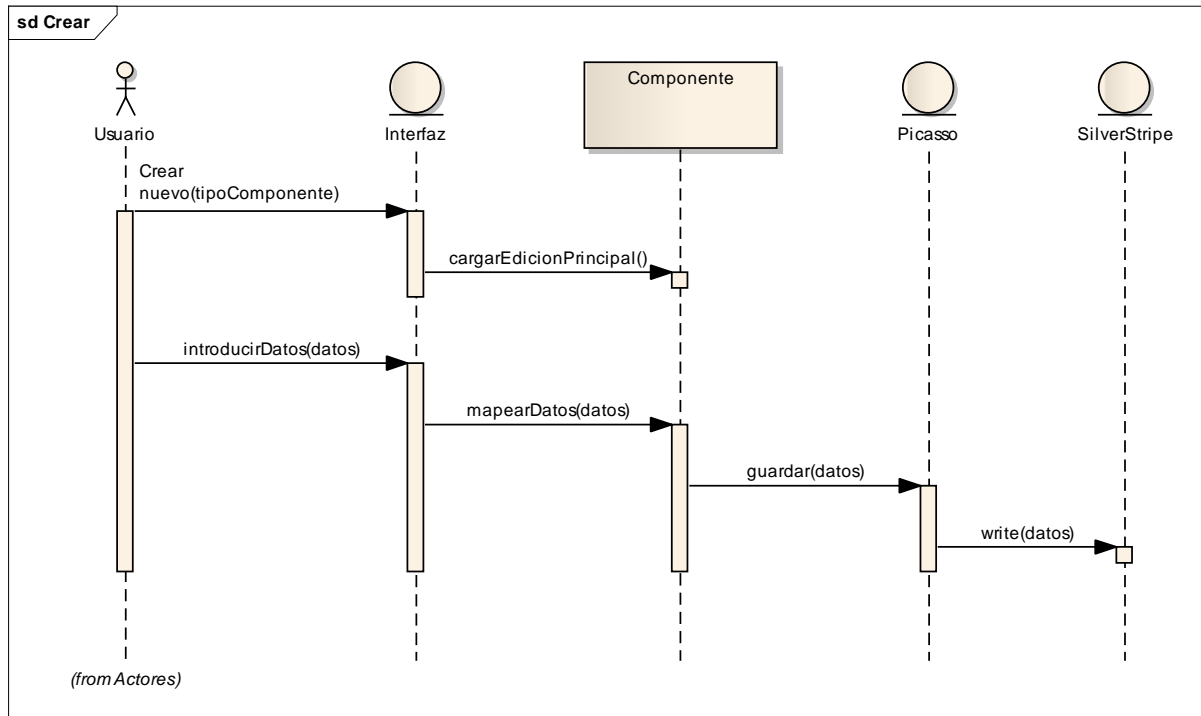


Figura 3.17: Diagrama de secuencias para la creación y modificación de datos

Para el acceso a datos existen diferentes patrones de diseño, en nuestro caso concreto vamos a utilizar el propuesto por el gestor de contenidos web SilverStripe. En este a diferencia de otros en los que por un lado tenemos el modelo de datos definido en una clase (VO o Value Object) y las funciones para la carga, edición, borrado etc. en otra (DAO o Data Access Object), vamos a tenerlo todo en la misma clase, de forma que esta contenga sus datos y las funciones para su manipulación. Además basándonos en la extensión del framework de desarrollo Picasso, también vamos a contener en la misma clase la información necesaria para la edición visual de cada componente.

Gracias a esto el proceso de creación de un nuevo componente será tan simple como llamar desde el interfaz a la función del componente a crear que contiene su información para la edición visual, introducir los datos requeridos y enviarlos a la función guardar del framework Picasso, el cual se encargará de que estos se guarden correctamente.

El proceso de modificación es muy parecido, y es por esto por lo que no se incluye un nuevo diagrama para él. La única diferencia es que en este caso la llamada a la función para obtener los datos de edición visual se realiza sobre una clase sobre la que se han recuperado los datos de un componente previamente.

3.6.2 Diagrama de secuencias para el borrado de datos

El proceso de borrado de un componente lo podemos ver en la Figura 1.18.

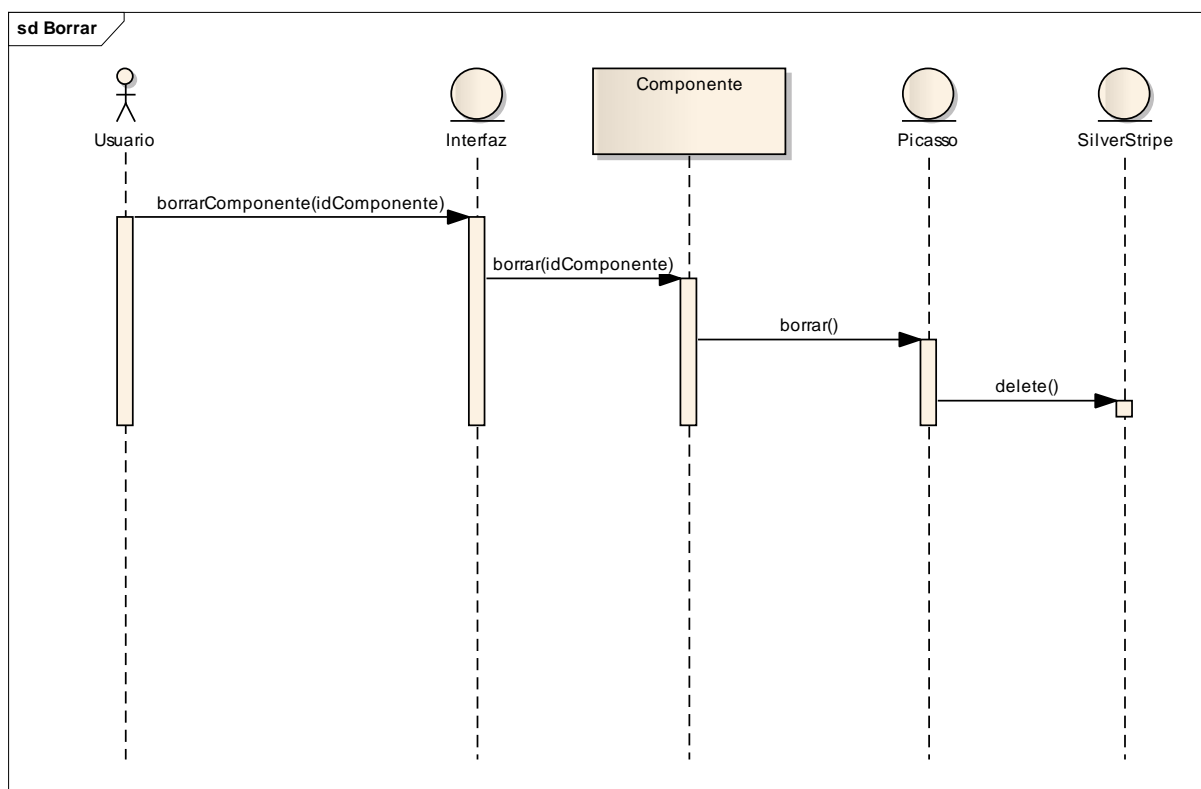


Figura 3.18: Diagrama de secuencias para el borrado de datos

Según el patrón utilizado para el acceso a datos, el proceso de borrado es tan simple como crear un nuevo objeto con los datos del componente a borrar y llamar a la función de borrado de dicho objeto.

3.6.3 Diagrama de secuencias para acciones genéricas

La Figura 3.19 refleja el proceso de ejecución de una acción. Intentando mantener el patrón utilizado para la creación, modificación y borrado de datos para el resto de operaciones se ha

creado un mecanismo genérico que permita ejecutar otro tipo de operaciones. Este se basa en una clase `OpcionMenu` que tiene asociada una clase del tipo `Accion` donde se definen las operaciones a realizar, de forma que para ejecutar distintas operaciones el proceso sea el mismo. El usuario a través del interfaz ejecuta una opción de menú, y a su vez esta ejecuta su acción asociada apoyándose en el framework Picasso y el gestor de contenidos web SilverStripe.

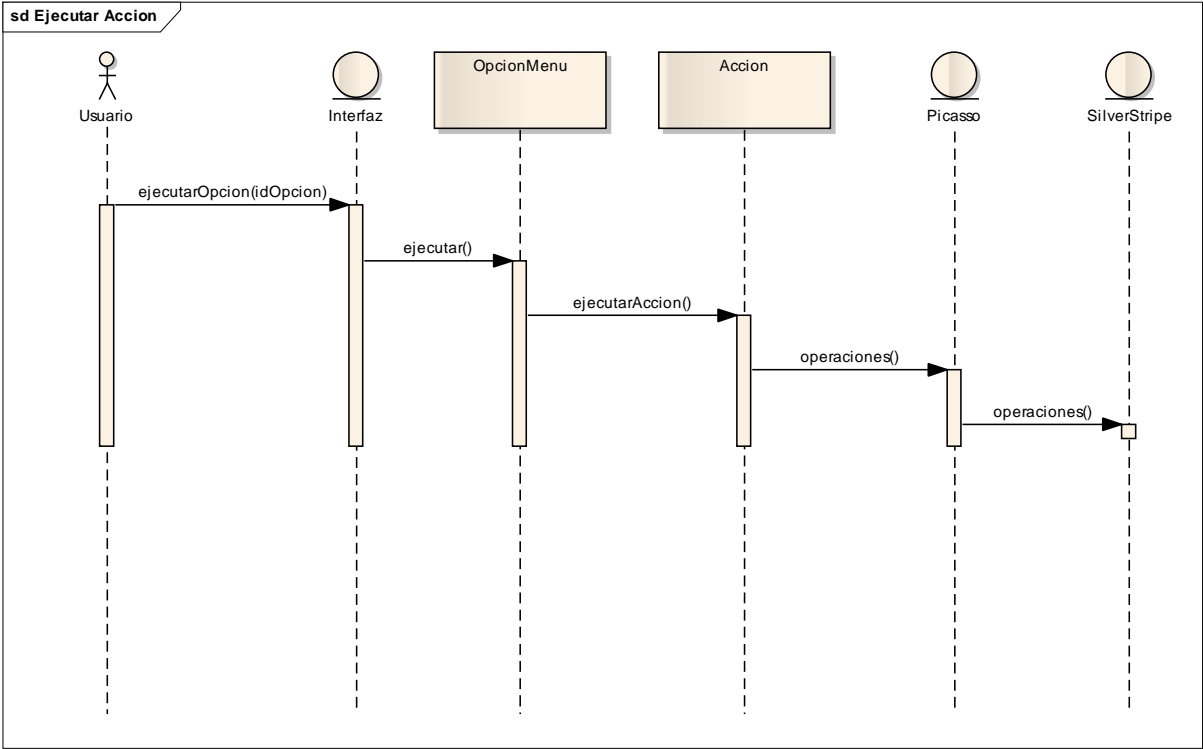


Figura 3.19: Diagrama de secuencias para acciones genéricas

CAPÍTULO 4

DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN

En este capítulo vamos a ver los métodos empleados para diseñar e implementar la aplicación. La aplicación se ha diseñado siguiendo estándares de accesibilidad, y se ha desarrollado con diversas técnicas para optimizar el rendimiento así como mejorar el interfaz del usuario. Para ello se han utilizado diversas herramientas (Capítulo 2), librerías y un sistema gestor de contenidos (CMS, apartado 2.5). Todas estas herramientas que han sido utilizadas, y la manera en que se han utilizado, serán descritas en los apartados de los que se compone el capítulo.

4.1 Libro de estilo web

El objetivo del libro de estilo es normalizar la estructura de los contenidos y el diseño del portal, homogeneizando estilos y estructuras para facilitar el desarrollo de nuevas páginas y las actualizaciones posteriores [8, 9].

Un libro de estilo estandariza el sitio web, sin embargo esto puede quedarse solo en colores y aspecto visual o definir procesos de interacción (por ejemplo, los medios de comunicación suelen tener libros de estilos para definir la ética a seguir por cada uno de sus trabajadores a la hora de publicar algún artículo).

Lo ideal es que un libro de estilo contenga ambas cosas, aunque esto conlleva una serie de ventajas e inconvenientes (Tabla 4.1).

En nuestro caso, puesto que nos vamos a centrar en una aplicación que genera el contenido a mostrar automáticamente, no vamos a necesitar definir reglas de interacción entre los editores de contenido, y simplemente nos vamos a basar en definir el formato que

deben tener todas las páginas, para facilitar de esta forma su posterior desarrollo, y mantener la integridad del diseño en futuras ampliaciones de la aplicación.

Tabla 4.1: Ventajas e inconvenientes de utilizar libros de estilos

Ventajas	Inconvenientes
Se evitan muchas discusiones porque gran parte del diseño esta establecido.	Hay que ser metódico en la elaboración del libro, un olvido o elemento no especificado crítico puede dar muchos problemas.
Cuando hay discusiones estas se refieren a contenidos, no a aspectos secundarios.	Tampoco se puede especificar absolutamente todo o el libro de estilo nunca estará listo.
Evita cometer errores.	Se debe consensuar y no tratar de imponerlo, lo que requiere mucho tiempo.
Facilita el trabajo de diseño que es mucho más rápido y estandarizado.	Se ha de hacer respetar el libro de estilo y contar con la implicación de los cargos directivos.
Hay un punto de inicio para introducir mejoras.	Debe haber algún sistema de validación de que las páginas creadas son acordes al libro de estilo. Esto es problemático porque a nadie le gusta que le critiquen el trabajo.
En general la calidad de un sitio que tiene un libro de estilo es mucho mayor.	

4.1.1 División del espacio de las páginas

En la Figura 4.1 podemos ver la estructura que van a tener todas las páginas de la aplicación, cualquier diseño posterior que se realice deberá seguir estas pautas. Es importante recalcar que estas pautas son orientativas y que el diseño final puede variar según el tema utilizado (apartado 4.2.2), atendiendo a razones estéticas, siempre que se respeten las diferentes divisiones en zonas.

El tamaño de la página será de 940 *pixeles* y estará centrada en la pantalla, dejando un

espacio variable a izquierda y derecha que dependerá de la resolución de la pantalla donde se muestra la página.



Figura 4.1: División del espacio de las páginas

Seguidamente tendremos una cabecera común a todas las páginas, independientemente de la sección donde nos encontremos esta será siempre igual, y en ella constarán el logotipo y el nombre de la aplicación.

El menú principal se situará justamente debajo de la cabecera (aunque también se podrá situar en la parte superior como caso especial si el diseño lo requiere), definiéndose en él una serie de opciones que se mantendrán visibles en todo momento, independientemente de la sección en la que nos encontremos.

Cada sección del menú principal tendrá asociada otra serie de subsecciones que se van a encontrar definidas en el menú lateral o zona de recursos.

La información de cada subsección (zona de recursos) concreta se verá en la zona de contenido.

Por último, existe un pie de página que al igual que la cabecera será común a todas las secciones.

4.1.2 Maqueta HTML

Siguiendo las pautas marcadas por el libro de estilo se ha desarrollado una maqueta HTML con la estructura definida, ya que es mucho más sencillo llevar a nuestra aplicación una estructura HTML que ya se ve correctamente a desarrollarla desde cero. En este caso y dada la sencillez del diseño sólo se ha creado una maqueta que respeta el diseño de la Figura 4.1. Además, se ha realizado de forma accesible, manteniendo en todo momento el estándar de accesibilidad WAI-AA (apartado 4.1.3), lo cual no quiere decir que la versión final lo cumpla.

4.1.3 Usabilidad y accesibilidad web

La accesibilidad web significa que personas con algún tipo de discapacidad van a poder hacer uso de la web.

En concreto, al hablar de accesibilidad web se está haciendo referencia a un diseño web que va a permitir que estas personas puedan percibir, entender, navegar e interactuar de forma efectiva con la web, así como crear y aportar contenido [11].

Existen millones de personas con discapacidad que no pueden utilizar algunas web. Actualmente, la mayoría de los sitios web presentan barreras de accesibilidad, lo que hace difícil e incluso imposible su utilización. Sin embargo, si los sitios web y el software web, fueran accesibles, personas con alguna discapacidad podrían utilizar la web de forma eficaz.

La accesibilidad web envuelve muchos tipos de discapacidades, incluyendo problemas visuales, auditivos, físicos, del habla, cognitivos y neurológicos. Pero la accesibilidad web beneficia también a organizaciones y a personas sin discapacidad. Por ejemplo, un principio básico de la accesibilidad web es la flexibilidad con el objetivo de satisfacer diferentes necesidades, situaciones y preferencias.

La flexibilidad va a beneficiar a todas aquellas personas que utilizan la web, incluyendo personas que no tienen ninguna discapacidad pero que, debido a determinadas situaciones, tienen dificultades para acceder a la web (por ejemplo, una conexión lenta), también estaríamos hablando de aquellas personas que sufren una incapacidad transitoria (por ejemplo, un brazo roto), y de personas de edad avanzada.

El esfuerzo necesario para hacer accesible un sitio web depende de muchos factores, incluyendo tipo de contenido, el tamaño, la complejidad del sitio web, así como las herramientas de desarrollo utilizadas y el entorno.

Muchos de los elementos que hacen un sitio web accesible se implementan de forma sencilla cuando se tienen en cuenta al comienzo de un proyecto de desarrollo o de rediseño. Sin embargo, la modificación de algunos sitios web puede requerir un gran esfuerzo, sobre todo aquellos que no se han “etiquetado” correctamente con etiquetas estándares de XHTML, y sitios con cierto tipo de contenido, como multimedia.

Cuando se desarrolla o rediseña un sitio web, la evaluación de la accesibilidad de forma temprana y a lo largo del desarrollo, permite encontrar al principio problemas de accesibilidad, cuando es más fácil resolverlos, en lugar de esperar hasta el final. Técnicas sencillas, como es cambiar la configuración en un buscador, pueden determinar si una página web cumple algunas de las pautas de accesibilidad. Una evaluación exhaustiva, para determinar el cumplimiento de las pautas, es mucho más compleja.

En definitiva, las entidades que participan en Internet, deberían tratar de mejorar todos los aspectos de accesibilidad analizados e incidir en el cumplimiento de las pautas de accesibilidad propuestas por la WAI (Iniciativa de Accesibilidad a la Web).

El sitio web de WAI (<http://wai.org>) provee información y material de ayuda para implementar las pautas, tales como “Guía breve para crear sitios Web accesibles”. Otras organizaciones proporcionan también recursos para ayudar a hacer la web accesible.

Las “Pautas de Accesibilidad al Contenido en la Web 1.0” de la W3C (<http://w3c.org>) proporciona una lista de todos los puntos de revisión de las citadas pautas, organizados por conceptos, como lista de revisión para los desarrolladores de contenidos en la web. Cada punto de verificación tiene un nivel de prioridad asignado por el grupo de trabajo del W3C fundamentado en su impacto sobre la accesibilidad.

4.2 El modelo MVC

SilverStripe se basa en el patrón de diseño MVC (Modelo Vista Controlador). El modelo MVC es un modelo de arquitectura de software que separa los datos de la aplicación, el interfaz del usuario, y la lógica de control en tres componentes distintos (Figura 4.2).

A continuación veremos como han sido tratados cada uno de estos componentes:

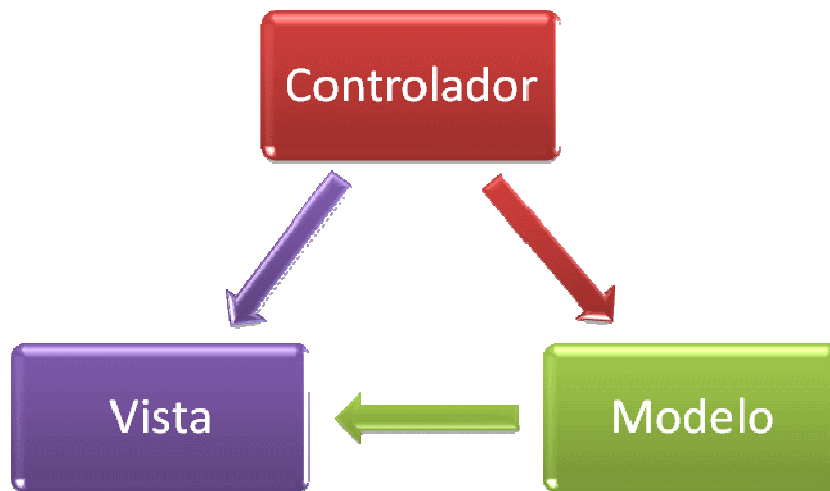


Figura 4.2: Modelo Vista Controlador

4.2.1 El modelo de datos

Para el modelo de datos SilverStripe se basa en un ORM (Object-Relational Mapping o en español Mapeo Objeto-Relacional) que es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos, y el utilizado en una base de datos relacional, con las siguientes características:

- Cada tabla de la base de datos es mapeada a una clase PHP.
- Cada fila de la tabla es un objeto (instancia) de la clase que representa a su tabla.
- Cada columna de la tabla es mapeada a una propiedad de la clase PHP.

Todas las tablas en SilverStripe se definen como subclases de la clase `DataObject`. De forma que al crear una clase que extienda de esta se genera la tabla en el esquema de base de datos automáticamente, y obtenemos las funciones para consultar, guardar y borrar.

Por ejemplo, para crear la tabla que define el tipo de monedas tal como se ha visto en el modelo de datos definido en el análisis del sistema (apartado 3.3) se crea una nueva clase llamada `TipoMoneda` que extienda de la clase `DataObject` (Figura 4.3), con lo que se va a crear esta tabla en la base de datos.

```
class TipoMoneda extends DataObject {  
    }  
}
```

Figura 4.3: Clase para generar la tabla TipoMoneda

Ahora para las columnas de la base de datos se definen en la clase tres *arrays* con un significado especial.

- **\$db:** este *array* define cada columna de la base de datos que no es una clave foránea a ninguna otra tabla. La manera de indicar cada elemento del *array* es de forma nominativa mediante el nombre de la columna y su tipo en SilverStripe [10]. Siendo los tipos uno de los existentes en el gestor de contenidos, que se pueden ver en su documentación (<http://doc.silverstripe.org>).
- **\$has_one:** este *array* define las relaciones uno a uno o uno a muchos, es decir indica que la columna es una clave foránea a otra tabla. Igual que \$db cada elemento se define indicando el nombre de la columna, seguido esta vez del tipo de la tabla a donde apunta (nombre de la clase PHP).
- **\$has_many:** este *array* define las relaciones muchos a muchos. Y sus elementos se definen igual que los anteriores (\$has_one).

En la Figura 4.4 vemos como sería el código para generar la tabla que define el tipo de monedas junto con sus columnas. Podemos observar que en este caso no se utilizan ni el *array* \$has_one ni el *array* \$has_many ya que esta tabla no tiene claves foráneas.

Para la edición de contenidos SilverStripe posee una serie de componentes denominados *fields* [10] que definen campos de edición para cada columna del modelo. Estos campos deben ser devueltos en el método `getCMSFields` de cada modelo (sobrescribiendo el método padre). Sin embargo este método sólo se utiliza en el entorno de administración del sistema gestor de contenidos.

El *framework* de desarrollo Picasso para SilverStripe posee un potente gestor de formularios que nos va a permitir poder editar cualquier componente de forma automática aprovechando los campos de edición proporcionados por SilverStripe.

```

class TipoMoneda extends DataObject {

    static $db = array(
        'Nombre' => 'Varchar',
        'Descripcion' => 'Text',
        'SimboloSingular' => 'Varchar',
        'SimboloPlural' => 'Varchar'
    );
}

```

Figura 4.4: Definición completa de la tabla TipoMoneda

Cualquier clase que herede de `ContenidoEditable` va a tener en su definición toda la información necesaria para su edición. Para ello se debe sobrescribir el método `cargarEdicionPrincipal` que devuelve un conjunto de campos de edición para la tabla (tipo `CamposEditable`).

Un tipo de campo editable (tipo `CampoEditable`), define una serie de características que va a tener el campo de edición a mostrar en el formulario, como son:

- **Nombre:** indica el nombre del campo editable.
- **Field:** indica el campo de edición de SilverStripe (véase la documentación de SilverStripe en <http://doc.silverstripe.org>).
- **MinOccurs:** indica el tipo mínimo de veces que debe aparecer el campo. Por defecto es 1, se utiliza para campos que pueden tener múltiples ocurrencias.
- **MaxOccurs:** indica el tipo máximo de veces que puede aparecer el campo. Por defecto es 1, se utiliza para campos que pueden tener múltiples ocurrencias.
- **SubCampos:** muchas veces un campo es una referencia a otra tabla, de forma que lo que se debe editar son los campos de la otra tabla, en este caso se debe indicar los campos de la otra tabla.
- **TipoSubCampo:** cuando el campo ha de editar los campos de otra tabla se debe indicar el tipo de esta.

- **CampoSubCampo:** cuando el campo ha editar son campos de otra tabla se debe indicar el nombre de la columna donde se almacena la referencia a esa tabla.
- **TextoAyuda:** define un texto a mostrar en el formulario donde se explica la finalidad del campo o se describe brevemente el tipo de contenido a introducir.
- **PonerHeredados:** este es un campo *booleano* (verdadero o falso) que indica si al crear nuevas instancias del campo (solo para campos con más de una ocurrencia) se permite seleccionar tipos que hereden del que se va a crear.
- **Visible:** es un valor *booleano* que indica si el campo será visible en el formulario de edición.
- **Validador:** indica la función de validación que se va ha realizar al introducir algún dato.

Hay que notar que no es necesario indicar todos lo campos, ya que vienen con valores por defecto y en la mayoría de los casos sólo se deberán indicar un número muy reducido de ellos.

En la Figura 4.5 se puede ver el código de la clase `TipoMoneda` que contiene los datos de definición de la tabla, sus columnas y su código de edición.

El formulario de edición que se genera tanto para crear, como para editar el `TipoMoneda` podemos verlo en la Figura 4.6.

El *framework* Picasso tiene diferentes aspectos para el formulario de edición, pudiéndose modificar en todo momento el aspecto final de este.

De esta forma, y tal como se muestra en el código de la Figura 4.4, se puede crear cada tabla del modelo de datos, junto con sus funciones de carga, edición y borrado.

4.2.2 La vista

La vista consiste en la presentación por pantalla de los resultados. Esta se va a mostrar con el aspecto definido en un tema. Un tema es un conjunto de plantillas agrupadas en una carpeta con el mismo nombre que el tema y que se sitúa dentro del directorio *themes* de SilverStripe, como podemos ver en la Figura 4.7.

```

class TipoMoneda extends ContenidoEditable {

    static $db = array(
        'Nombre' => 'Varchar',
        'Descripcion' => 'Text',
        'SimboloSingular' => 'Varchar',
        'SimboloPlural' => 'Varchar'
    );

    protected function cargarEdicionPrincipal() {

        $pestanasEditable = parent::cargarEdicionPrincipal();

        $campoNombre = new CampoEditable();
        $campoNombre->setNombre('Nombre');
        $campoNombre->setField(new TextField('Nombre', 'Nombre',
            $this->Nombre));
        $campoNombre->setValidador(new CampoRequeridoValidador());
        $pestanasEditable->add("Contenido", $campoNombre);

        $campoDescripcion = new CampoEditable();
        $campoDescripcion->setNombre('Descripci&oacute;n');
        $campoDescripcion->setField(new TextareaField('Descripcion',
            'Descripcion', 8, 3, $this->Descripcion));
        $campoDescripcion->setValidador(new CampoRequeridoValidador());
        $pestanasEditable->add('Contenido', $campoDescripcion);

        $campoSimboloSingular = new CampoEditable();
        $campoSimboloSingular->setNombre('Simbolo en singular');
        $campoSimboloSingular->setField(new TextField('SimboloSingular',
            'SimboloSingular', $this->SimboloSingular));
        $campoSimboloSingular->setValidador(
            newCampoRequeridoValidador());
        $pestanasEditable->add('Contenido', $campoSimboloSingular);

        $campoSimboloPlural = new CampoEditable();
        $campoSimboloPlural->setNombre('Simbolo en plural');
        $campoSimboloPlural->setField(new TextField('SimboloPlural',
            'SimboloPlural', $this->SimboloPlural));
        $campoSimboloPlural->setValidador(new CampoRequeridoValidador());
        $pestanasEditable->add('Contenido', $campoSimboloPlural);

        return $pestanasEditable;
    }
}

```

Figura 4.5: Código para crear y editar el TipoMoneda

Para cambiar de tema simplemente debemos indicar el que se va a utilizar dentro del fichero de configuración de SilverStripe (`_config.php`), tal como se puede ver en la Figura 4.8.

Todas las páginas en SilverStripe van a tener un tipo, de forma similar a como ocurre en la programación orientada a objetos, podríamos decir que cada página pertenece a una clase que puede heredar de otra. Pues bien, cuando se va a mostrar una página, SilverStripe busca dentro del tema definido una plantilla perteneciente al tipo de la página, en caso de no

encontrarlo sigue buscando en todos su tipos padres hasta encontrarlo, momento en el que utiliza esa plantilla para generar el contenido a mostrar.

Figura 4.6: Formulario de edición para el TipoMoneda

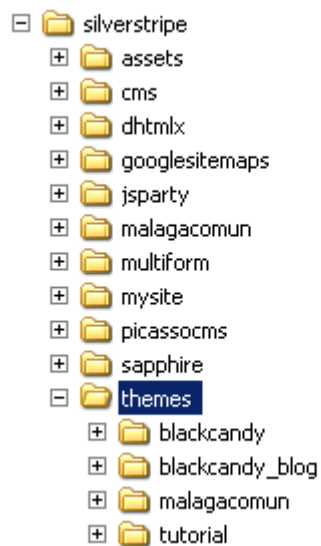


Figura 4.7: Carpetas con los diferentes temas instalados en la aplicación

```
SSViewer::set_theme('malagacomun');
```

Figura 4.8: código para seleccionar el tema para mostrar las páginas.

Para la aplicación se ha creado el tema *malagacomun*, aunque se podría utilizar uno de los muchos existentes y que se distribuyen de forma gratuita a través de la web del gestor de contenidos. Este tema simplemente consta de una plantilla, ya que como se indica en la Figura 4.1 la zona de contenidos es la única que va a variar de una página a otra, el resto mantiene la misma funcionalidad a lo largo de toda la aplicación.

Para mostrar la zona de contenidos, se han creado una serie de subplantillas (plantillas que se incluyen dentro de la plantilla principal) que se cargarán de forma dinámica y que ya no dependen del tema, sino que son estándar a toda la aplicación. Es decir, se indique el tema que se indique estas siempre se verán de igual forma.

Para mostrar la información de cada tabla definida en el modelo de datos existe una plantilla por cada una de sus clases, y se sobrescribe la función `CodigoVer` para indicar cual es la plantilla encargada de mostrar su contenido. En la Figura 4.9 podemos ver el código para mostrar la información del tipo de moneda.

Para más información sobre como crear un tema y su estructura se puede consultar la documentación online del sistema gestor de contenidos SilverStripe (<http://doc.silverstripe.org>).

```
class TipoMoneda extends ContenidoEditable {
    ...
    protected function CodigoVer() {
        return $this->renderWith('PlantillaVerTipoMoneda');
    }
    ...
}
```

Figura 4.9: Función para indicar la plantilla que va a mostrar los datos del TipoMoneda

4.2.3 El controlador

Todos los tipos de páginas que se vallan a mostrar a través de una plantilla necesitan una clase controladora. Esta clase se llama igual que la clase que define la página seguida de la palabra `Controller`.

El controlador se encarga de realizar las operaciones sobre las plantillas antes de mostrarlas por pantalla, y desde este se pueden acceder a los elementos de la clase que define la página. Es importante saber que solo deben tener controladores las clases que definan páginas, es decir que se vallan a mostrar por pantalla, una clase que simplemente defina una tabla en el modelo de datos y sea accedida por otra no tiene porque tener un controlador.

En la Figura 4.10 podemos ver una jerarquía de páginas típica de SilverStripe. En ella vemos que todas las páginas heredan de la clase `SiteTree` y su controlador se llama

Content_Controller. A partir de esta podemos ir creando nuevas clases de páginas junto con sus respectivos controladores.

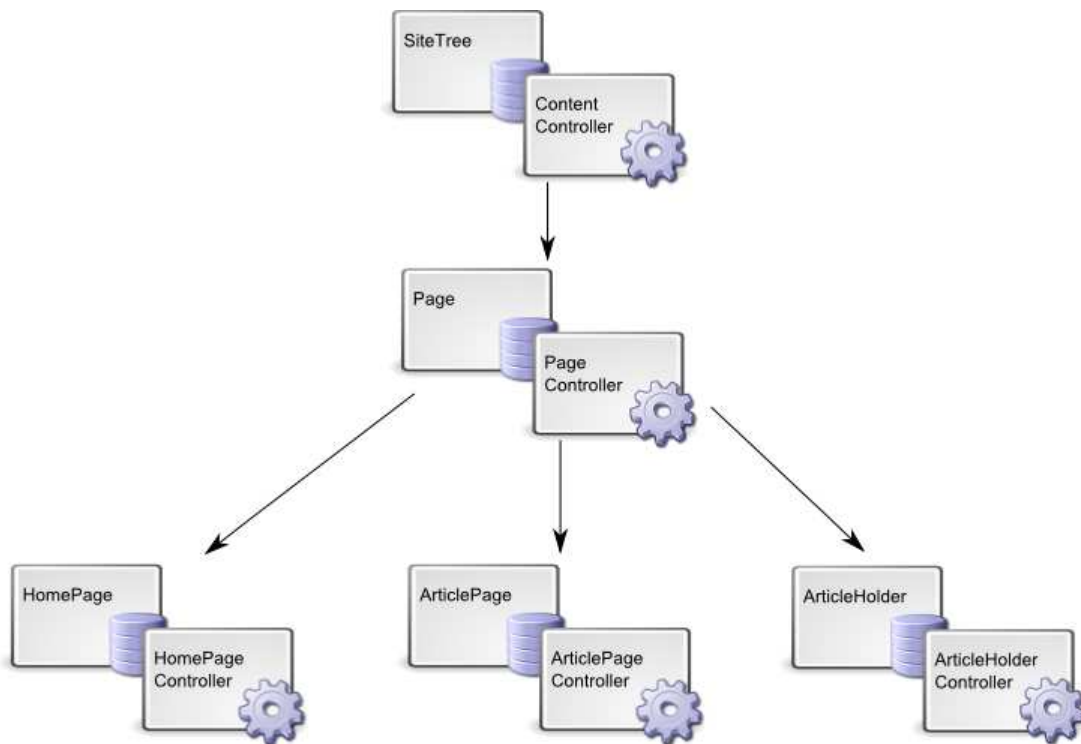


Figura 4.10: Jerarquía de páginas típica en SilverStripe

Los controladores son las clases que manejan el flujo de la aplicación, reciben los parámetros que se le envían y generan las respuestas. Tal como se definen en SilverStripe un controlador recibe una petición, la procesa y genera la respuesta. Esto a veces puede resultar poco flexible, sobre todo cuando se desea añadir nueva funcionalidad a una aplicación.

Para aumentar la flexibilidad el *framework* de desarrollo Picasso define un tipo de página llamada `Pagina_Picasso` que hereda de `SiteTree` y cuyo controlador hereda de `Content_Controller`. Este controlador basa las peticiones en eventos llamados acciones.

Una `Pagina_Picasso` consta de un conjunto de controles que se pueden añadir dinámicamente, de forma que cuando la página recibe una petición de una acción esta la propaga al resto de controles, posibilitando el que se puedan gestionar múltiples respuestas en cascada.

Un ejemplo típico de este comportamiento podría ser una página donde se define un control para la gestión de los formularios de edición y un control para la gestión de mensajes, de forma que cuando se produce una acción determinada, como puede ser la de crear un

nuevo componente, la página lo propaga por cada control de forma que cada uno realice las acciones que tiene definidas o bien la deje pasar (de forma similar al sistema de eventos de un sistema operativo).

A parte de este sistema de acciones, el controlador de la `Pagina_Picasso` se comporta de forma similar a cualquier otro controlador del sistema gestor de contenidos.

4.3 Estructura de la aplicación

Se pretende que la aplicación sea lo más genérica posible, de modo que cualquier ampliación o modificación implique poco impacto sobre lo que ya existe y funciona correctamente. Para ello se ha seguido una estructura en cuatro capas (Figura 4.11). Como núcleo de la aplicación tenemos al sistema gestor de contenidos SilverStripe. En una capa superior tenemos el *framework* de desarrollo Picasso. Sobre este se ha creado una aplicación genérica, con una serie de funciones comunes a cualquier otra aplicación web. Por último sobre la aplicación genérica se implementará la funcionalidad concreta de nuestra aplicación.

Para crear esta genericidad buscada se puede observar la división del espacio de las páginas (Figura 4.1), y vemos que sólo el espacio reservado para el contenido varía de una página otra e incluso de una web a otra, el resto se mantiene constante (a excepción del tema).

Por tanto, la funcionalidad se va a aportar mediante opciones de menú configurables, es decir, cada una de las opciones que aparecen en el menú lateral de la izquierda se pueden modificar de manera dinámica y se les puede indicar lo que van a mostrar en el área de contenidos. De forma que *a priori* no importa cual va a ser el funcionamiento de la aplicación, sino que la estructura va a ser igual para todas las aplicaciones y la funcionalidad genérica la marcará la acción a ejecutar por la opción de menú.



Figura 4.11: Capas de la aplicación

Para ello se ha creado un componente llamado `OpcionMenu` con los siguientes campos:

- **Nombre:** indica el nombre de la opción de menú.
- **Texto:** indica el texto que se va a mostrar en el menú.
- **Descripción:** indica una breve descripción de lo que realiza la opción de menú.
- **Nivel superior:** indica quien es el padre de la opción de menú, es decir, si es una subopción (opción contenida dentro de otra), aquí se indica quién es la opción superior.
- **Imagen:** indica la imagen asociada a la opción de menú.
- **Acción:** indica la acción a realizar. Es un componente del tipo `Accion` y es la clase que habría que implementar (en caso de no existir) para crear la funcionalidad propia de una aplicación concreta.

En la Figura 4.12 podemos ver un ejemplo de edición de una opción de menú para listar todos los tipos de monedas de la aplicación.

4.3.1 Acciones para las opciones de menú

Las acciones definen el comportamiento que van a tener las opciones de menú. Una acción es una clase que extiende de `Accion` y que debe sobrescribir el método `ejecutarAccion` indicando en él, el código a mostrar en el área de contenido de la aplicación.

En la Figura 4.13 vemos el código de una acción que muestra en el área de contenido de la aplicación el mensaje de texto que se indica en la edición del componente.

Las acciones son contenido editable del *framework* de desarrollo Picasso, por tanto también podemos definir campos propios para indicar parámetros en la acción. Es importante mencionar que los campos de edición deben indicarse en la función `cargarSubContenido` y no en la función `cargarContenidoPrincipal`, ya que el

contenido se cargará desde la edición de la opción de menú y no desde un formulario propio para la acción (apartado 4.2.1).

Si creamos una nueva opción de menú vemos como podemos seleccionar el nuevo tipo de acción (Figura 4.14).

Texto: Tipos de monedas

Descripción: Muestra un listado con todos los tipos de monedas definidos en la aplicación.

Nivel Superior: datossistema

Posición: 1 Posición actual

Imagen: Añadir contenido

Acción:

Nombre: AccionListarComponentesBtMalaga

Tipo de componente: TipoMonedaBtMalaga

Formulario de filtrado:

Informacion filtrado:

```
{
  "Campos": [
    { "Nombre": "ContenidoEditable", "Salida": "ContenidoEditable.ClassName" },
    { "Nombre": "ContenidoEditableConPermisos", "Salida": "ContenidoEditable" },
    { "Nombre": "TipoComponente", "Salida": "TipoMonedaBtMalaga.*", "Tipo": " " }
  ]
}
```

Figura 4.12: Edición de una opción de menú para listar los tipos de monedas

```
class AccionMostrarTexto extends Accion {
    static $db = array('Texto' => 'Text');

    public function ejecutarAccion($controlador, $recurso) {
        return $this->Texto;
    }

    protected function cargarSubContenido() {
        $camposEditable = parent::cargarSubContenido();

        $campoTexto = new CampoEditable();
        $campoTexto->setNombre('Texto');
        $campoTexto->setField(new TextField('Texto',
            'Texto', $this->Texto));
        $camposEditable->add($campoTexto);

        return $camposEditable;
    }
}
```

Figura 4.13: Acción para mostrar un texto

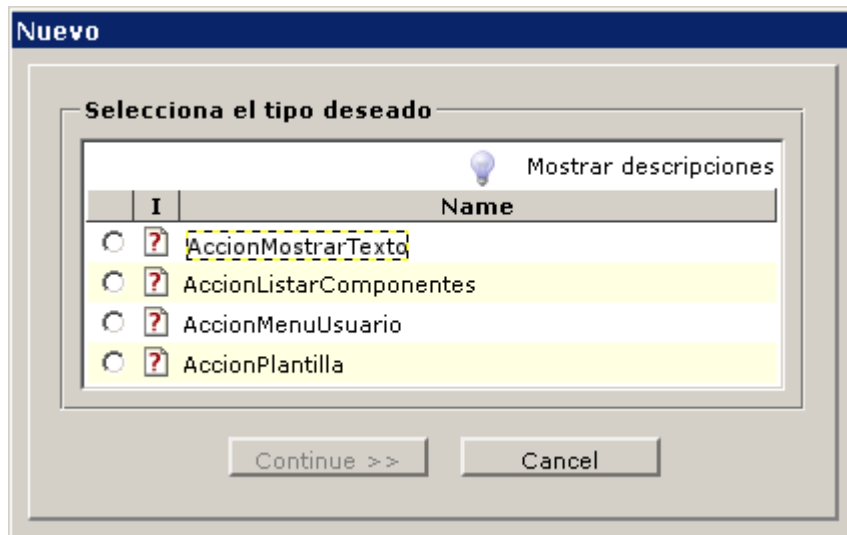


Figura 4.14: Selección de la acción AccionMostrarTexto desde la edición de una opción de menú

En la Figura 4.15 vemos como aparecen los campos de la acción seleccionada, si rellenamos todos los campos y guardamos la opción aparecerá en la web de la aplicación, y al pulsar sobre ella veremos el texto introducido en la edición, que es lo que se ha indicado en el código de la acción.

4.3.2 Permisos para las opciones

Para poder realizar una aplicación con opciones de configuración genéricas debe existir un control de permisos que indique que usuarios son los que pueden realizar las opciones, de lo contrario no se podría mantener la seguridad de la web, puesto que todo el mundo podría realizar cualquier acción.

Además del contenido editable el *framework* de desarrollo Picasso posee un tipo de contenidos llamado `ContenidoEditableConPermisos` con un gestor de permisos para el contenido que lo extienda. Los permisos se pueden definir a tres niveles diferentes: permisos para el autor del contenido, permisos para los usuarios pertenecientes a un grupo determinado, y permisos para cualquier otro usuario. A su vez estos permisos son:

- **Leer:** un usuario con este permiso puede ver el componente.
- **Escribir:** un usuario con este permiso puede crear y modificar un componente de este tipo.

- **Publicar:** un usuario con este permiso puede publicar y eliminarlo de la parte pública.
- **Cambiar permisos:** un usuario con este permiso puede modificar los permisos del componente.

The image shows a web form with two tabs: 'Contenido' and 'Permisos'. The 'Permisos' tab is active. The form contains the following fields:

- Nombre:** acción
- Texto:** acción
- Descripción:** Texto definido en la acción
- Nivel Superior:** Root
- Posición:** 5 Última posición
- Imagen:** Añadir contenido
- Acción:**
 - Nombre:
 - Texto: Texto definido en la acción

Figura 4.15: Edición de una opción de menú con la acción AccionMostrarTexto

La búsqueda de los permisos se realiza de la siguiente manera: en primer lugar se comprueba si el usuario conectado es el autor del componente sobre el que se están comprobando los permisos, de ser así se comprueba si el permiso se encuentra permitido o denegado, devolviendo el resultado. Si el usuario no es el autor, se comprueba si existe para algún grupo de los que el usuario pertenece, permisos definidos (tanto permitidos como denegados) en cuyo caso se devuelve el resultado. Por último, si no se encuentran permisos definidos para algún grupo del usuario se comprueba para la definición de “Otros” devolviendo el resultado en caso de existir.

Si en ninguno de los casos anteriores se encuentra definido se procede a buscar de forma similar en un nivel superior, atendiendo a la propiedad padre del componente.

En la Figura 4.16 podemos ver una imagen del sistema gestor de permisos dentro del formulario de edición de un componente.

Puesto que cuando un componente no tiene definido un permiso el gestor de permisos los busca en los componentes padre, para la aplicación genérica vamos a crear una serie de carpetas que sean los componentes padre para cada componente. Estas carpetas serán del tipo

Carpeta y tendrán los permisos por defecto para cada componente que se cree en el sistema.

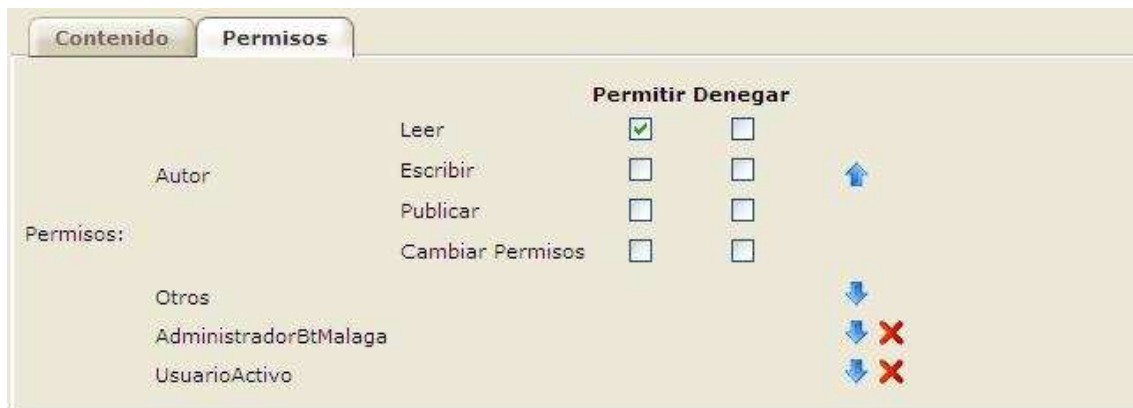


Figura 4.16: Gestión de permisos

Existe una opción en la configuración de la aplicación que permite listar y editar cualquier carpeta, de forma que se le cambien a cada componente los permisos que tienen por defecto. Así de esta manera conseguimos individualizar sobre un usuario todas las acciones que puede realizar. Por ejemplo, podemos permitir que un usuario publique su propio contenido indicándolo en los permisos de su carpeta padre. Una carpeta se tiene que llamar con la palabra `Contenedor` seguida del nombre del componente de cuyo tipo va a ser padre. En la Figura 4.17 podemos ver un listado de carpetas para los diferentes componentes de la aplicación.

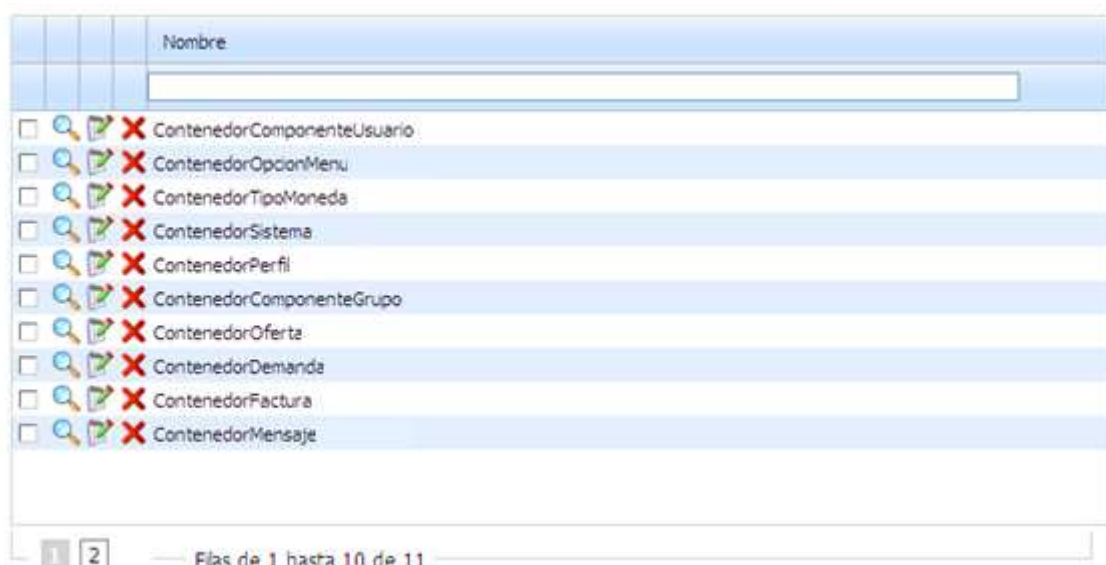


Figura 4.17: Listado de carpetas para los diferentes componentes de la aplicación

4.4 Módulo de la aplicación

SilverStripe es un sistema modular. Un módulo es un conjunto de clases, plantillas y cualquier otro tipo de recursos que se sitúan dentro de un directorio en la raíz del sistema gestor de contenidos. En SilverStripe existen diferentes tipos de módulos:

- **Aplicaciones:** funcionan de forma independiente al resto de aplicaciones y módulos instalados en el sistema.
- **CMS Add-ons:** una extensión del sistema es, por ejemplo algún tipo de página especial como un *blog* o un *foro*.
- **Widgets:** un módulo puede definir un *widget* de otra página, un widget es una funcionalidad añadida a una página. (véase la documentación del CMS [10]).
- **Herramientas de administración y desarrollo:** un módulo puede ser una aplicación aparte que se utilice para administrar alguna otra existente.

4.4.1 Estructura del módulo de la aplicación

La aplicación se ha desarrollado como un módulo de SilverStripe siguiendo la estructura de carpetas que podemos ver en la Figura 4.18.

El módulo se define como una carpeta llamada *malagacomun* que se sitúa en la raíz del sistema gestor de contenidos, y dentro de esta carpeta encontramos las carpetas:

- **code:** dentro de ella se definen todas las clases PHP de la aplicación. Fundamentalmente las clases de la aplicación pertenecen a tres tipos diferentes: tenemos las clases que representan páginas para la aplicación, clases para el modelo de datos, y clases que aportan alguna funcionalidad extra a las anteriores.
- **images:** almacena las imágenes utilizadas por la aplicación, no por el tema ni las hojas de estilo, sino sólo por las plantillas que no varían su aspecto a pesar de cambiar su tema.

- **javascript:** dentro se encuentra los ficheros de *scripts* propios utilizados por la aplicación, es decir, los que no se comparten con otras aplicaciones o han sido desarrollados con carácter genérico.
- **lang:** contiene ficheros de traducción para los diferentes idiomas de los que dispone la aplicación.
- **templates:** define las plantillas que van a permanecer inalterables independientemente del tema seleccionado para mostrar la página.

Es muy importante notar que para que SilverStripe reconozca las clases PHP dentro del módulo, en la raíz del módulo debe existir un fichero de configuración llamado “_config.php”.

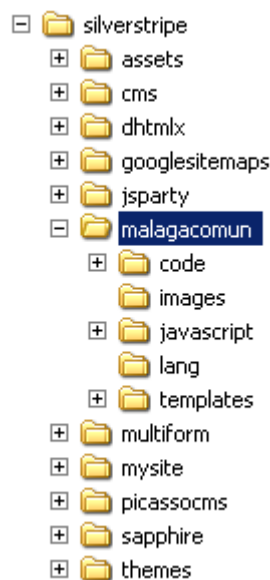


Figura 4.18: Estructura de directorios del modulo de la aplicación

4.4.2 Clases para el modelo de datos

Todos los elementos del modelo de datos van a compartir una serie de características comunes de la aplicación. Por tanto se ha definido una clase padre para los elementos del modelo de datos de la aplicación. Esta clase padre llamada `Componente` es heredada por el resto de elementos, y ella a su vez hereda de la clase

`ContenidoEditableConPermisos` del *framework* de desarrollo Picasso (apartado 4.2.1). Todas estas clases que representan al modelo de datos se encuentran dentro de la carpeta `componentes` del código del módulo y son las definidas en el análisis del modelo de datos (apartado 3.3.2).

4.4.3 Páginas y controles de la aplicación

Dentro de la carpeta del código del módulo encontramos también las clases que definen las páginas para la aplicación. En este caso y al igual que se ha realizado con los elementos del modelo de datos, también heredan de una clase padre que contiene todos los elementos comunes de la aplicación. Aquí la página padre se llama `WebMalagaComun` que hereda a su vez de `PaginaPicasso`, así como su controlador se llama `WebMalagaComun_Controller` y hereda de `PaginaPicasso_Controller` (apartado 4.2.3).

Para la aplicación se han definido dos tipos de páginas, una para mostrar el contenido estático de la web (lo que sería básicamente la parte pública), llamada `ContenidoInteriorWebMalagaComun` y otra para la aplicación en sí, llamada `ContenidoGestionWebMalagaComun` que contiene toda la lógica de la aplicación.

4.4.4 Instancias de páginas

Hay que recalcar que las clases definen a los tipos de páginas, sin embargo hay que crear las instancias concretas de ellas desde la administración de SilverStripe.

Para el correcto funcionamiento de la aplicación se ha creado la siguiente estructura de páginas en la administración del CMS:

- **webmalagacomun:** es la página de entrada a la aplicación, su tipo es `ContenidoInteriorMalagaComun` y todas las demás van a ser hijas de esta.
- **malagacomun:** es una página pública donde se define el sistema, se comenta su funcionamiento, etc. Es de tipo `ContenidoInteriorMalagaComun` y puede tener tantas páginas hijas como se deseen, todas del tipo `ContenidoInteriorMalagaComun`.

- **intercambios:** es una página del tipo `ContenidoInteriorMalagaComun` que define lo que son los intercambios y que a su vez tienen como hijas a dos páginas, una para mostrar las ofertas y otra para mostrar las demandas.
- **participantes:** es una página del tipo `ContenidoGestionMalagaComun`, con lo cual está es la página de la aplicación propiamente dicha, siendo el resto páginas estáticas que dan contenido a la web pública.
- **contactar:** es una página del tipo `ContactPage`, cuya diferencia con el tipo `ContenidoInteriorMalagaComun` es que además define una serie de campos para enviar un correo electrónico de contacto.
- **mapadelsitio:** define el mapa del sitio de la aplicación, y es del tipo `ContenidoInteriorMalagaComun`.
- **avisolegal:** define contenido estático para mostrar un aviso legal del sitio y es de tipo `ContenidoInteriorMalagaComun`.
- **protecciondatos:** contenido del tipo `ContenidoInteriorMalagaComun` para definir la política de protección de datos de la web.
- **registrarse:** página del tipo `ContenidoInteriorMalagaComun` utilizada para registrar a un nuevo usuario.

En la Figura 4.19 podemos ver como queda la estructura de páginas dentro del menú de administración del sistema gestor de contenidos. Esta estructura se puede modificar en todo momento, cambiando el orden, añadiendo, eliminando o modificando los datos de cada página.

4.4.4.1 Grupos y usuarios

Dentro de la carpeta `extensiones` del código, encontramos la carpeta `usuarios` en la que se encuentran las extensiones para los grupos, y usuarios de la aplicación (`GrupoMalagaComun` y `UsuarioMalagaComun` respectivamente). En ellas se extienden

el modelo de datos y la funcionalidad, para aportar características especiales de los usuarios. Como son los sistemas donde se encuentra registrado, su fotografía, el segundo apellido, o el teléfono móvil. Características que no se encuentran en las clases que aporta el sistema gestor de contenidos y el *framework* de desarrollo Picasso.

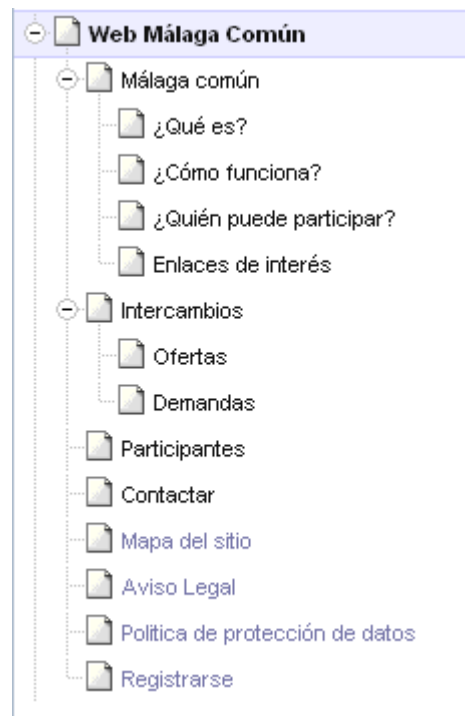


Figura 4.19: Estructura de páginas de la aplicación

4.4.4.2 Autenticación de los usuarios

El sistema gestor de contenidos SilverStripe posee un método dinámico para autenticar a los usuarios del sistema. Se pueden crear múltiples métodos de autenticación [10] y el gestor los irá recorriendo todos hasta que alguno sea válido, momento en el que dará por autenticado al usuario, o bien ninguno lo sea, momento en el que lo dará por un usuario no autorizado. De esta forma, un mismo formulario de autenticación en el sistema puede realizarse mediante usuario y contraseña, mediante un directorio LDAP (Lightweight Directory Access Protocol o Protocolo Ligero de Acceso a Directorios en español), etc.

En la carpeta `autenticadores` de las extensiones del sistema de nuestro módulo se encuentra un autenticador creado para nuestra aplicación, el cual comprueba que además de que el usuario existe dado su email y contraseña, también comprueba que sea del tipo `UsuarioMalagaComun`, de forma que si existe algún otro tipo de usuario registrado

en el sistema gestor de contenidos pero no es de nuestra aplicación en concreto tampoco pueda ser autenticado.

4.4.4.3 Formulario de registro de un nuevo usuario

Se ha creado un formulario de registro (en la carpeta `formularios` de las extensiones del gestor de contenidos) que pide todos los datos requeridos por la aplicación para poder dar de alta a un nuevo usuario.

El formulario pertenece a un tipo especial de formularios que existen en el sistema gestor de contenidos SilverStripe que constan de varias pantallas. Es decir, conforme se van introduciendo datos, se van generando diferentes páginas para pedir la introducción de nuevos datos. En concreto para nuestra aplicación se ha creado un formulario de registro en dos pasos o páginas, una primera donde se introduce la información personal del usuario y se indican los sistemas en los que desea participar. Y una segunda en la que con la información obtenida en la primera se indican las ofertas obligatorias que como mínimo hay que añadir para registrarse en cada sistema. Para obtener más información sobre como funcionan los formularios de múltiples páginas se puede ver la documentación de silverstripe (<http://doc.silverstripe.org>).

4.4.5 Controles para las acciones a realizar

Puesto que el sistema de acciones de las páginas del *framework* Picasso funcionan a través de controles (apartado 4.2.3) se han definido dos controles que podemos encontrar en la carpeta `controles` del código del módulo. Un control llamado `ComponenteMalagaComunControl` implementa todas las acciones para el tratamiento de los componentes de la aplicación, tales como la creación, modificación, publicación, etc. Mientras que otro control llamado `MensajeMalagaComunControl` implementa la gestión de mensajes, de forma que a determinadas acciones envía un mensaje de forma automática.

Aquí podemos apreciar la flexibilidad de este sistema para el tratamiento de acciones, de forma que si se desea realizar alguna otra operación como respuesta a alguna acción, como por ejemplo crear un determinado componente, no se debe modificar nada del código existente, sino que se podría añadir otro control que realizase alguna operación al recibir dicha acción.

4.4.6 Acciones para los menús de opciones

La mayoría de la funcionalidad con la que cuenta la aplicación se encuentra en el menú del lateral izquierdo. Este es un menú que se configura de forma dinámica (apartado 4.3.1). Cada una de estas opciones debe tener definida una acción a realizar. Para el desarrollo de nuestra aplicación se han definido tres acciones:

- **AccionPlantilla:** esta acción se utiliza para mostrar en la zona de contenidos de la página la plantilla que se le indica como parámetro.
- **AccionMenuUsuario:** esta acción se utiliza para mostrar el menú principal con los datos del usuario conectado actualmente. Se utiliza una acción específica en lugar de la acción para mostrar una plantilla genérica porque además carga una serie de datos necesarios sobre el usuario. Esta acción no tiene ningún parámetro adicional, ya que se encuentra específicamente creada para una opción concreta.
- **AccionListarComponentes:** esta acción se utiliza para cargar un listado del componente del modelo de datos que se le indica como parámetro. Además del parámetro que indica el componente a mostrar se le debe indicar como parámetros el formulario para filtrar los datos y el código para generar la consulta.

La Figura 4.20 muestra una captura de edición de una opción de menú que ejecuta la acción `AccionListarComponentes`, en ella podemos ver que la acción tiene tres campos de edición de datos.



Acción:

Nombre:	AccionListarComponentes
Tipo de componente:	TipoMoneda
Formulario de filtrado:	
Informacion filtrado:	<pre>{ "Campos": [{ "Nombre": "ContenidoEditable", "Salida": "ContenidoEditable.ClassName", "Tipo": "texto", "Campo": "", "Valor": "", "Tabla": "ContenidoEditable", "Relacion": "ContenidoEditable.ID=TipoMonedaBtMalaga.ID" }, { "Nombre": "ContenidoEditableConPermisos", "Salida": "ContenidoEditableConPermisos.", "Tipo": "texto", "Campo": "", "Valor": "", "Tabla": "ContenidoEditableConPermisos", "Relacion": "ContenidoEditableConPermisos.ID=ContenidoEditable.ID" }] }</pre>

Figura 4.20: Campos de edición de una acción del tipo `AccionListarComponentes`.

- **Nombre:** indica el nombre de la clase que define la acción.
- **Tipo de componente:** indica el componente del modelo de datos que se va a mostrar.
- **Formulario de filtrado:** indica el nombre de la función que va a devolver el formulario para filtrar los datos. Esta función debe estar en la clase que define el componente indicado en el campo “Tipo de componente”.
- **Información de filtrado:** indica la información para generar la consulta. Esta se encuentra codificada en formato JSON [12]. En la Figura 4.21 se puede ver como ejemplo la información utilizada para mostrar el listado de todos los sistemas disponibles en la aplicación.

```

{ "Campos": [
  { "Nombre": "ContenidoEditable",
    "Salida": "ContenidoEditable.ClassName",
    "Tipo": "texto",
    "Campo": "",
    "Valor": "",
    "Tabla": "ContenidoEditable",
    "Relacion": "ContenidoEditable.ID=TipoMonedaBtMalaga.ID" },
  { "Nombre": "ContenidoEditableConPermisos",
    "Salida": "ContenidoEditableConPermisos.*",
    "Tipo": "texto",
    "Campo": "", "Valor": "",
    "Tabla": "ContenidoEditableConPermisos",
    "Relacion": "ContenidoEditableConPermisos.ID=ContenidoEditable.ID" },
  { "Nombre": "TipoComponente",
    "Salida": "TipoMonedaBtMalaga.*",
    "Tipo": "texto", "Campo": "",
    "Valor": "",
    "Tabla": "TipoMonedaBtMalaga",
    "Relacion": "" }
] }

```

Figura 4.21: Código JSON para generar la consulta SQL para el listado de sistemas

Como se puede ver en la Figura 4.21 los campos de los que consta cada fila de la información de filtrado son:

- **Nombre:** indica el nombre con el que se va a referenciar toda la fila a la hora de generar la consulta SQL que devolverá los datos. Es importante saber que los campos del formulario de filtrado deben tener el mismo nombre que este campo.

- **Salida:** indica la salida que va a producir esta entrada. Es decir, es el valor que se va a poner en la parte `SELECT` del SQL al procesar esta entrada.
- **Tipo:** indica el tipo del valor y puede ser de cuatro tipos diferentes: texto para indicar que el valor del campo va a ser una cadena de caracteres y la comparación se generará utilizando el comparador “LIKE.” Numero, para indicar que el valor del campo es un número y por lo tanto el comparador será el signo “=”. Y los tipos “fechaantes” y “fechadespues” para indicar que los valores son de tipo fecha y los comparadores “<” y “>” respectivamente.
- **Campo:** indica el campo de la base de datos sobre el que se va a realizar la comparación con el valor introducido.
- **Valor:** si se indica un valor aquí este se asignará automáticamente sin tener que coger el valor del formulario del filtro. De lo contrario el valor para la comparación del Campo la obtendrá del campo del formulario del filtro cuyo nombre sea igual a la propiedad Nombre de la fila.
- **Tabla:** indica la tabla sobre la que se va a realizar la consulta.
- **Relación:** indica la relación existente entre la tabla sobre la que se va a realizar la consulta y la tabla principal del componente.

4.4.7 Javascript del módulo

Dentro de la carpeta javascript del módulo de la aplicación nos encontramos con cuatro ficheros donde se definen los diferentes scripts (apartado 2.4.6) utilizados por la aplicación. Estos scripts son cargados en determinadas páginas según las necesidades de la opción de menú que se valla a mostrar.

4.4.7.1 Fichero `alertamensajes.js`

Es cargado en todas las páginas de la aplicación y se utiliza para alertar de la existencia de los mensajes. Al cargarse comprueba si existen mensajes en la bandeja de entrada del usuario,

indicándolo mediante una pequeña alerta en la esquina inferior derecha de la pantalla en caso de existir.

4.4.7.2 Fichero `editarcomponente.js`

Este fichero es cargado cuando se edita un componente, y se utiliza para actualizar la información existente en el listado desde el cual se ha modificado o creado el componente sin necesidad de consultar la base de datos.

4.4.7.3 Fichero `listarcomponentes.js`

Este fichero se carga en las opciones generadas por la acción de menú `AccionListarComponentes` y tiene el código necesario para mostrar correctamente el listado. Además posee funciones para poder realizar todas las acciones permitidas por los controles (apartado 4.4.3) mediante AJAX.

4.4.8 Plantillas

Dentro del directorio `plantillas` que a su vez se encuentra en el directorio `templates` del módulo de la aplicación, se encuentran los ficheros encargados de mostrar por pantalla la información relativa a cada componente. Estos ficheros tienen la extensión “.ss” y se basan en el modelo de plantillas utilizado por el sistema gestor de contenidos SilverStripe [10]. A su vez cada componente del modelo de datos sobrescribe el método `codigoVer` de su clase para indicar cual será la plantilla encargada de mostrar sus datos.

En la Figura 4.22 podemos ver el la función que se sobrescribe para cargar la plantilla para mostrar los datos de un elemento de la clase `TipoMoneda`.

```
protected function codigoVer() {  
    return $this->renderWith('PlantillaVerTipoMoneda');  
}
```

Figura 4.22: Código para indicar la plantilla a utilizar a la hora de mostrar los datos de un `TipoMoneda`

4.5 Controles DHTMLX

Para mejorar el aspecto y la interacción con el usuario de la web se han incluido una serie de controles gratuitos realizados por la empresa DHTMLX (<http://www.dhtmlx.com>) en javascript.

La finalidad de estos controles son las de mejorar la interacción entre el usuario y la web evitando conexiones innecesarias al servidor, de forma que se ejecute la mayor parte del código posible en el ordenador del usuario. Con lo que se consigue mayor rapidez y eficiencia en la mayoría de las acciones.

Los controles utilizados se han almacenando como un módulo de SilverStripe en la carpeta raíz `dhtmlx` [13] y son:

- **Dhtmlxgrid**: este control define un *datagrid* o tabla de datos donde se muestran los datos de una consulta determinada. Ha sido utilizado en todas las acciones donde se ha requerido mostrar la información de un listado de componentes.
- **Dhtmlxscheduler**: este control define a un calendario y a sido utilizado tanto para el calendario que muestra la selección de fechas para la realización de un servicio, como en el calendario donde se indican las fechas de los servicios de un determinado usuario.
- **Dhtmlxwindows**: este control se utiliza para mostrar ventanas, y ha sido utilizado siempre que se ha necesitado una ventana emergente o *popup* en la aplicación.
- **Dhtmlxtoolbar**: este control sirve para añadir una barra de herramientas a la ventana generada con el control `dhtmlxwindows`. Y se utiliza para poner una barra de herramientas con las acciones a realizar en las ventanas que se abren para ver el contenido de un determinado componente de la aplicación.

CAPÍTULO 5

DESPLIEGUE Y USO DE LA APLICACIÓN

Nuestro proyecto entra en la etapa de definición, allí donde se lo presentamos al cliente o usuario final, sabiendo que funciona correctamente y responde a los requerimientos solicitados en su momento. Esta etapa es muy importante no sólo por representar la aceptación o no del proyecto por parte del cliente o usuario final sino por las múltiples dificultades que suele presentar en la práctica, alargándose excesivamente y provocando costos no previstos.

5.1 Instalación y paso a producción

La instalación del software [14] es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final. Constituye la etapa final en el desarrollo propiamente dicho del software. Luego de ésta el producto entrará en la fase de funcionamiento y producción, para el que fuera diseñado.

La instalación, dependiendo del sistema desarrollado, puede consistir en una simple copia al disco rígido destino (casos raros actualmente); o bien, más comúnmente, con una de complejidad intermedia en la que los distintos archivos componentes del software (ejecutables, bibliotecas, datos propios, etc.) son descomprimidos y copiados a lugares específicos preestablecidos del disco; incluso se crean vínculos con otros productos, además del propio sistema operativo. Este último caso, comúnmente es un proceso bastante automático que es creado y guiado con herramientas software específicas (empaquetado y distribución, instaladores).

En productos de mayor complejidad, la segunda alternativa es la utilizada, pero es realizada o guiada por especialistas; puede incluso requerirse la instalación en varios y distintos computadores (instalación distribuida).

También, en software de mediana y alta complejidad normalmente es requerido un proceso de configuración y chequeo, por el cual se asignan adecuados parámetros de funcionamiento y se testea la operatividad funcional del producto.

En productos de venta masiva las instalaciones completas, si son relativamente simples, suelen ser realizadas por los propios usuarios finales (tales como sistemas operativos, paquetes de oficina, utilitarios, etc.) con herramientas propias de instalación guiada; incluso la configuración suele ser automática. En productos de diseño específico o "a medida" la instalación queda restringida, normalmente, a personas especialistas involucradas en el desarrollo del software en cuestión.

Una vez realizada exitosamente la instalación del software, el mismo pasa a la fase de producción (operatividad), durante la cual cumple las funciones para las que fue desarrollado, es decir, es finalmente utilizado por el (o los) usuario final, produciendo los resultados esperados.

5.1.1 Despliegue en un servidor web

La aplicación Málaga Común es una aplicación web [15]. En la ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los webmails, wikis, weblogs, tiendas en línea, etc. que son ejemplos bien conocidos de aplicaciones web.

Es importante mencionar que una página web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

El proceso de instalación de la aplicación Málaga Común se detalla en el apartado 6.1 de este documento.

5.1.2 Problemas de integración

Al realizar la integración de la aplicación en un entorno web se han producido una serie de errores que se han tenido que solucionar para que la aplicación pueda funcionar correctamente.

- **Problemas de compatibilidad entre la codificación de las bases de datos:** el entorno de desarrollo ha sido Windows, mientras que la aplicación final se ha desplegado en un entorno Unix. Debido a esto hemos pasado de un entorno que no es sensible a mayúsculas y minúsculas, a otro que sí que lo es. Por defecto MySQL no es *case sensitive* en Windows y sí que lo es en Unix. Esto ha provocado un error en los nombres de las tablas en el fichero de exportación generado para la instalación, que ha debido de corregirse manualmente.
- **Problemas de codificación de caracteres entre los diferentes sistemas operativos:** por defecto en los sistemas Windows los ficheros de texto se guardan con las codificaciones Windows-1252 o CP-1252, mientras que en los sistemas Unix por defecto se guardan con la codificación UTF-8. Esta diferencia de codificaciones provoca que en algunos sitios no se vean correctamente los caracteres especiales. Para solucionarlo se ha tenido que repasar todos los lugares donde se utilizan y convertirlos todos a UTF-8.
- **Problemas con los tiempos de respetas en AJAX:** en un entorno local las peticiones a AJAX suelen tener tiempos de respuesta casi instantáneos. Sin embargo con el paso a un servidor remoto los tiempos aumentan hasta llegar a varios segundos. Ha sido necesario añadir un mecanismo de bloqueo de la página para evitar realizar otras operaciones mientras se está esperando la respuesta de una petición AJAX para algunas acciones.
- **Problemas con los envíos de correos electrónicos:** existen direcciones IP, sobre todo pertenecientes a servidores gratuitos, que se encuentran en la lista negra de algunos

proveedores de correo electrónicos como *gmail* o *hotmail*. Con lo cual los correos que se les envían los rechaza tratándolos como *spam*. La única manera de corregir este error es cambiar de servidor.

- **Problemas de compatibilidades con las versiones de los programas:** a veces las versiones con las que se ha desarrollado la aplicación no coinciden al 100% con las que nos encontramos en el servidor donde vamos a instalarla, en ese caso se pueden dar casos de incompatibilidad con alguna función. Para solucionarlo hay que detectar el error y actuar según se indique en el soporte técnico de la herramienta que provoca la incompatibilidad.

5.2 Pruebas de rendimiento y optimización

En la ingeniería del software [16], las pruebas de rendimiento [17] son las pruebas que se realizan, desde una perspectiva, para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo. También puede servir para validar y verificar otros atributos de la calidad del sistema, tales como la escalabilidad, fiabilidad y uso de los recursos. Las pruebas de rendimiento son un subconjunto de la ingeniería de pruebas, una práctica informática que se esfuerza por mejorar el rendimiento, englobándose en el diseño y la arquitectura de un sistema, antes incluso del esfuerzo inicial de la codificación.

Las pruebas de rendimiento pueden servir para diferentes propósitos. Pueden demostrar que el sistema cumple los criterios de rendimiento. Pueden comparar dos sistemas para encontrar cual de ellos funciona mejor. O pueden medir que partes del sistema o de carga de trabajo provocan que el conjunto rinda mal. Para su diagnóstico, los ingenieros de software utilizan herramientas como pueden ser monitorizaciones que midan qué partes de un dispositivo o software contribuyen más al mal rendimiento o para establecer niveles (y umbrales) del mismo que mantenga un tiempo de respuesta aceptable.

Es fundamental para alcanzar un buen nivel de rendimiento de un nuevo sistema, que los esfuerzos en estas pruebas comiencen en el inicio del proyecto de desarrollo y se amplíe durante su construcción. Cuanto más se tarde en detectar un defecto de rendimiento, mayor es el coste de la solución. Esto es cierto en el caso de las pruebas funcionales, pero mucho más en las pruebas de rendimiento, debido a que su ámbito de aplicación es de principio a fin.

En las pruebas de rendimiento, a menudo es crucial (y con frecuencia difícil de conseguir) que las condiciones de prueba sean similares a las esperadas en el uso real. Esto es, sin embargo, casi imposible en la práctica. La razón es que los sistemas en producción tienen un carácter aleatorio de la carga de trabajo y aunque en las pruebas se intente dar lo mejor de sí para imitar el volumen de trabajo que pueda tener el entorno de producción, es imposible reproducir exactamente la variabilidad de ese trabajo, salvo en el sistema más simple.

Los nuevos conceptos en la implementación de la arquitectura (por ejemplo: SOA – Service Oriented Architecture o Arquitectura Orientada a Objetos) han añadido complejidad adicional a las pruebas de rendimiento. Los servicios o recursos de la empresa (que comparten infraestructura o plataforma) requieren pruebas de rendimiento coordinadas (con la creación del volumen y carga de todos los sistemas que comparten la infraestructura o plataformas) para reproducir realmente el estado del entorno de producción. Debido a la complejidad, coste y tiempo necesario en torno a esta actividad, algunas organizaciones emplean herramientas que pueden mostrar y crear condiciones (también conocido como "ruido") en los entornos de pruebas de rendimiento para comprender la capacidad y las necesidades de recursos y verificar/validar los niveles de calidad.

5.2.1 Tipos de pruebas de rendimiento

A continuación podemos ver diferentes tipos de pruebas que se pueden realizar para analizar el rendimiento de una aplicación.

5.2.1.1 Pruebas de carga

Este es el tipo más sencillo de pruebas de rendimiento. Una prueba de carga se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada. Esta carga puede ser el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dura la carga. Esta prueba puede mostrar los tiempos de respuesta de todas las transacciones importantes de la aplicación. Si la base de datos, el servidor de aplicaciones, etc. también se monitorizan, entonces esta prueba puede mostrar el cuello de botella en la aplicación.

5.2.1.2 Prueba de estrés

Esta prueba se utiliza normalmente para romper la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

5.2.1.3 Prueba de estabilidad (*soak testing*)

Esta prueba normalmente se hace para determinar si la aplicación puede aguantar una carga esperada continuada. Generalmente esta prueba se realiza para determinar si hay alguna *fuga* de memoria en la aplicación.

5.2.1.4 Pruebas de picos (*spike testing*)

La prueba de picos, como el nombre sugiere, trata de observar el comportamiento del sistema variando el número de usuarios, tanto cuando bajan, como cuando tiene cambios drásticos en su carga. Esta prueba se recomienda que sea realizada con un software automatizado que permita realizar cambios en el número de usuarios mientras que los administradores llevan un registro de los valores a ser monitoreados.

5.2.2 Pruebas de carga para la aplicación Málaga Común

Antes de poner en producción nuestro programa debemos asegurarnos que va ha responder a las peticiones de los usuarios en un tiempo razonable.

Según el autor Jakob Nielsen, en el libro “Usability Engineering”[18] existen tres límites importantes en el tiempo de respuesta: [19, 20]

- **0,1 segundo:** es el límite en el cual el usuario siente que esta “manipulando” los objetos desde la interfaz de usuario.
- **1 segundo:** es el límite en el cual el usuario siente que está navegando libremente sin esperar demasiado una respuesta del servidor.

- **10 segundos:** es el límite en el cual se pierde la atención del usuario, si la respuesta tarda más de 10 segundos se deberá indicar algún mecanismo por el cual el usuario pueda interrumpir la operación.

5.2.2.1 Objetivo de las pruebas de carga

Para las pruebas de carga vamos a atacar a la página principal de un usuario conectado, ya que esta es la página que contiene mayor número de accesos a bases de datos, es decir, es la que se va a cargar más lenta de toda la aplicación. Para obtener los tiempos de carga vamos a simular 50 accesos simultáneos a la página, y calcularemos el tiempo medio, dándolo por válido sólo en el caso de que no supere los 10 minutos de media. De lo contrario deberemos optimizar el código para conseguir reducir los tiempos de carga de la página a los 10 minutos como máximo en condiciones extremas.

5.2.2.2 La herramienta utilizada para realizar las pruebas de carga

Para analizar el tiempo de respuesta del servidor se utilizó la herramienta Jmeter [21]. La versión utilizada de Jmeter durante este trabajo es la 2.4. Jmeter es una herramienta open source muy completa, implementada en Java que permite realizar test de comportamiento funcional y medir el rendimiento. También se puede utilizar para realizar pruebas de estrés, por ejemplo, en un servidor, y poner a prueba su rendimiento [22].

Para estas pruebas, se configuró un servidor Proxy que provee Jmeter, para poder construir un camino de navegación que implique el logado de un usuario y acceso a su página de usuario, y así simular la visita de un usuario registrado.

Para poder evaluar los resultados se utilizaron dos componentes provistos por la herramienta [23, 24].

1.- Aggregate Graph: Permite visualizar los resultados del test realizado, en una tabla. Los datos que presenta son:

- **#Muestras:** cantidad de peticiones realizadas al servidor.
- **Media:** tiempo promedio en milisegundos para un conjunto de resultados.

- **Mediana:** valor en tiempo del percentil 50.
- **Línea de 90%:** máximo tiempo utilizado por el 90% de la muestra, al resto de la misma le llevo más tiempo.
- **Min:** tiempo mínimo de la muestra de una determinada URL.
- **Max:** tiempo máximo de la muestra de una determinada URL.
- **%Error:** porcentaje de requerimientos con errores.
- **Rendimiento:** rendimiento medido en los requerimiento por segundo / minuto / hora.
- **KB/sec:** rendimiento medido en Kbytes por segundo.

2.- Grafico de Resultados: Esta componente permite visualizar gráficamente los siguientes datos: media, mediana, dispersión y el rendimiento (representado como el número actual de requerimientos/minutos que el servidor maneja).

5.2.2.3 Resultados

A continuación se van a mostrar los resultados obtenidos para la versión final que se va a instalar en producción sobre un servidor local. Hay que tener en cuenta que previamente a estos resultados se dieron resultados cuyos tiempos de carga superaban los 10 minutos, debiendo de optimarse. Para mejorar estos tiempos se ha optimizado el acceso a base de datos, reduciendo las consultas todo lo que se ha podido.

Los valores totales obtenidos por la componente *Aggregate Graph* con 50 usuarios simultáneos atacando a la página personal de un usuario registrado (Figura 6.16) se muestran en la Tabla 5.1.

Como puede verse en la Tabla 5.1 se han realizado 1750 peticiones al servidor con un tiempo promedio de 10,971 segundos para la instalación en desarrollo (entorno local). Tenemos que tener claro que en realidad se ha cargado 50 veces la página personal de cada usuario, sin embargo, la carga de esta página conlleva múltiples peticiones al servidor. En este

caso concreto cada página de usuario conlleva 35 peticiones al servidor, estas peticiones son además de la página principal, carga de imágenes, archivos de *scripts*, hojas de estilo, etc.

Tabla 5.1: Valores correspondientes a las pruebas de carga

	# Muestras	Media	Mediana	Línea de 90%	Min	Máx	% Error	Rendimiento	Kb/sec
Desarrollo	1750	10971	1313	21020	5	179490	0.00%	2.8/sec	50.2
Producción	1750	5826	1949	16003	190	103992	0.37%	4.9/sec	63.0

El tiempo total utilizado para los 50 usuarios en un entorno local (desarrollo) se puede calcular con la formula:

$$\text{Tiempo Total} = \# \text{Muestras} * \text{Media} = 1750 * 10971 = 19199250 \text{ milisegundos} \quad (5.1)$$

El tiempo promedio total requerido por cada usuario en el entorno local, se puede calcular:

$$\left(\frac{\text{Tiempo Total}}{1000} \right) / 60 / \text{cantidad de usuarios} = \left(\frac{19199250}{1000} \right) / 60 / 50 = 6,39975 \text{ minutos} \quad (5.2)$$

Tal como se han definido los requisitos de la aplicación para la gestión de moneda complementaria Málaga Común, la posibilidad de que existan 50 usuarios conectados simultáneamente es casi inexistente, ya que este es el número aproximado de usuarios que conforman el sistema. Sin embargo, y poniéndonos en el peor de los casos la aplicación va a responder bastante bien, ya que el tiempo de carga de la página que más va a tardar en cargarse, apenas pasa los 6 minutos, muy por debajo del umbral de los 10 minutos que nos habíamos marcado como máximo.

En la Figura 5.1 podemos ver una gráfica con los resultados del análisis realizado sobre la aplicación final en un entorno local (desarrollo).

El tiempo total utilizado para los 50 usuarios en el entorno final (producción) se puede calcular con la formula:

$$\text{Tiempo Total} = \# \text{Muestras} * \text{Media} = 1750 * 5826 = 10195500 \text{ milisegundos} \quad (5.3)$$

El tiempo promedio total requerido por cada usuario en el entorno de producción, se puede calcular:

$$\begin{aligned} & ((\text{Tiempo Total} / 1000) / 60) / \text{cantidad de usuarios} = \\ & ((10195500 / 1000) / 60) / 50 = 3,3985 \text{ minutos} \quad (5.4) \end{aligned}$$

Después de comprobar el tiempo de carga para 50 usuarios simultáneos nos da un tiempo de algo más de 3 minutos, que es mucho mejor de lo esperado. Esto es debido a que el entorno de producción, a pesar de encontrarse en red, se encuentra en un entorno donde todos los componentes, tanto software como hardware, están optimizados para la ejecución de este tipo de aplicaciones.

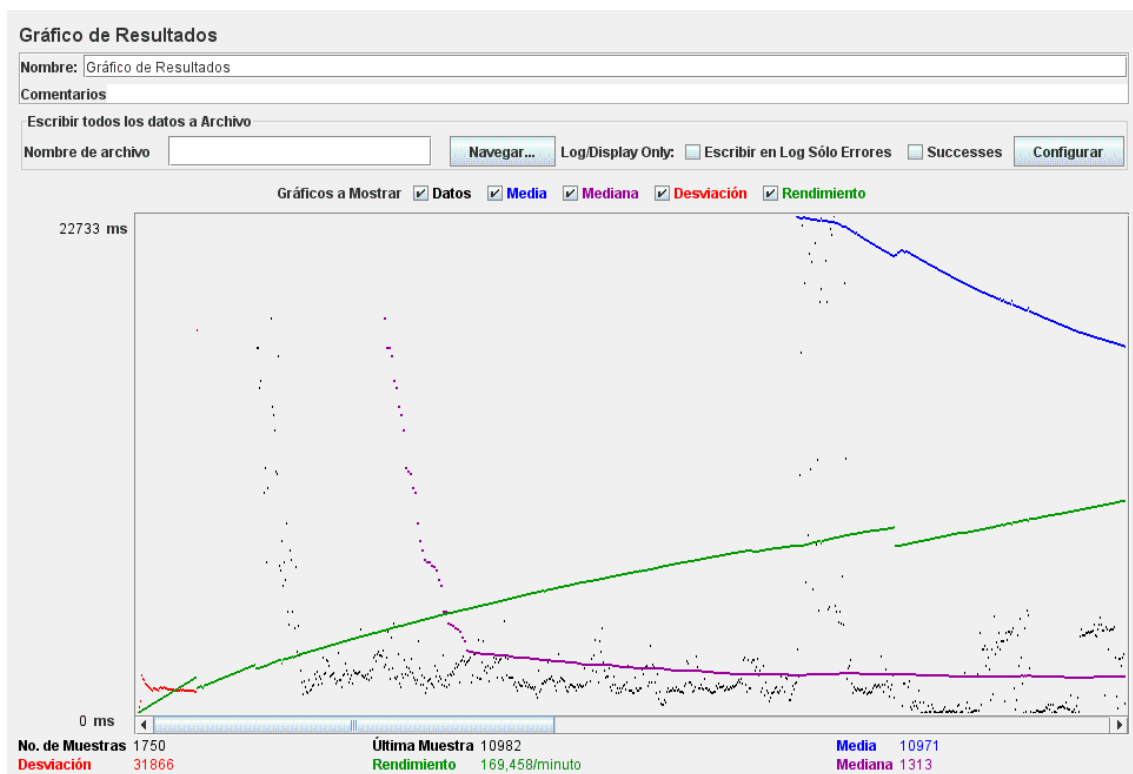


Figura 5.1: Resultados del análisis ejecutado en modo local

5.2.3 Análisis de carga de la aplicación Málaga Común

A la hora de diseñar la aplicación se ha tomado la decisión de que las consultas devuelvan todos los resultados existentes, realizando el paginado en caso de ser necesario en el navegador mediante javascript. La ventaja de esto es que se evitan peticiones al servidor, ya que al cambiar de página no se solicitan nuevos datos. Mientras que la desventaja es que se

deben descargar todos los datos, independientemente de los que se vayan a utilizar. Una manera de comprobar si esta decisión ha sido correcta es analizar los tiempos de carga de una página. Para realizar esta tarea vamos a utilizar la aplicación online WebPageTest (<http://www.webpagetest.org>).

WebPageTest es una herramienta originalmente desarrollada por AOL (<http://www.aol.com>) para uso interno cuyo código fue posteriormente liberado, con la que cualquier webmaster o usuario interesado puede testear un site para conocer su rendimiento.

WebPageTest mide el tiempo de carga, velocidad de acceso, velocidad de respuesta, además de un sinnfín de parámetros con los que conocer el rendimiento de un site y ayudarnos a optimizarlo para una mejor velocidad de carga.

El manejo de WebPageTest no presenta excesivos problemas en su análisis más básico, simplemente con introducir la URL y la localización (país o región) más cercana para, de forma automática, obtener un completo informe con múltiples gráficas que nos ayudarán a entender toda la información.

Analizando la carga de una página cualquiera de nuestra aplicación obtenemos la gráfica de la Figura 5.2, donde se observa que la página tarda 9 segundos aproximadamente en cargarse totalmente. Estos tiempos se encuentran repartidos en una primera fase de resolución del nombre del dominio (se obtiene la dirección IP correspondiente al dominio introducido), mostrado en azul verdoso. Una segunda etapa de inicio de la conexión, mostrado en la gráfica en naranja. Una tercera fase, que es la que más tarda, en la que la petición se encuentra encolada en el servidor a la espera de ser servida, esta se muestra de color verde en la gráfica. Y una última en la que se sirve el contenido solicitado, en azul celeste en la gráfica. Puesto que lo que tarda más tiempo con diferencia del resto de etapas, es la etapa en que la petición se encuentra en espera de ser servida, ha sido un acierto reducirla lo máximo posible evitando realizar peticiones innecesarias al servidor. Además de esta forma se recude la cola de espera, con lo que los tiempos del resto de peticiones también serán menores.

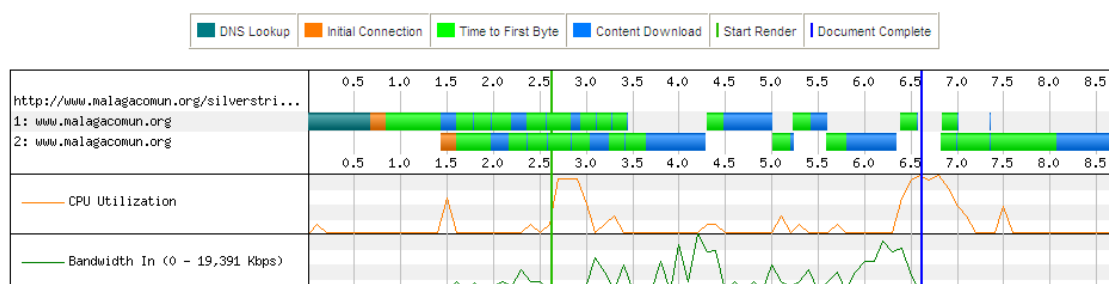


Figura 5.2: Gráfico con los tiempos de carga de una página.

5.3 Configuración de la aplicación

La aplicación ya se instala con gran parte de la configuración necesaria para el funcionamiento del sistema de moneda complementaria Málaga Común, sin embargo existen una serie de operaciones que se han tenido que llevar a cabo para dejarla totalmente operativa. En este apartado veremos algunas de las acciones que se han llevado a cabo para configurar tanto la aplicación como su entorno.

5.3.1 Creación de un administrador

Se ha creado un usuario administrador (`administrador@malagacomun.org`) con todos los permisos de administración sobre el sistema Málaga Común. Este usuario es el que va a recibir los impuestos procedentes de las operaciones en el sistema (en caso de existir tales impuestos), por tanto se recomienda que no se utilice para ninguna otra operación. Este usuario se ha creado con una dirección de correo electrónica que es una lista de distribución, de forma que se puedan añadir tantas redirecciones como usuarios se deseen que estén al tanto de lo que ocurre en el sistema. Es decir, todos los mensajes que le lleguen al administrador, le llegaran a los integrantes de su lista de distribución.

5.3.2 Traspasos de créditos

El grupo de sistema de moneda complementaria Málaga Común ya lleva un tiempo funcionando con otra aplicación. Por tanto, se debe realizar un traspaso de créditos de la anterior aplicación a nuestra aplicación. Para realizar ese traspaso se ha creado un nuevo usuario `traspasos@malagacomun.org` en el sistema, cuyo correo, al igual que para el administrador, también corresponde a una lista de distribución, para de esta forma mantener al tanto a los usuarios que se van a encargar de aceptar o rechazar las solicitudes de traspasos. Con este usuario se ha creado una nueva demanda, llamada “Traspaso de créditos” que debe ser solicitada por todos los usuarios que tengan un saldo positivo en el anterior sistema. Mientras que para los usuarios que tengan un saldo negativo, se ha creado, también con el usuario encargado de realizar los traspasos, la oferta “Traspaso de créditos”, que tendrán que solicitar los usuarios que quieran recibir su crédito. De esta forma los traspasos quedan

formalizados a través de sus correspondientes facturas, manteniendo la transparencia del sistema.

5.4 Mantenimiento

El mantenimiento de software [24] es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control y beta test. Esta fase es la última (antes de iterar, según el modelo empleado) que se aplica al ciclo de vida del desarrollo de software. La fase de mantenimiento es la que viene después de que el software está operativo y en producción.

De un buen diseño y documentación del desarrollo dependerá cómo será la fase de mantenimiento, tanto en costo temporal como monetario. Modificaciones realizadas a un software que fue elaborado con una documentación indebida o pobre y mal diseño puede llegar a ser tanto o más costosa que desarrollar el software desde el inicio. Por ello, es de fundamental importancia respetar debidamente todas las tareas de las fases del desarrollo y mantener adecuada y completa la documentación.

El período de la fase de mantenimiento es normalmente el mayor en todo el ciclo de vida [16]. Esta fase involucra también actualizaciones y evoluciones del software; no necesariamente implica que el sistema tuvo errores. Uno o más cambios en el software, por ejemplo de adaptación o evolutivos, puede llevar incluso a prever y adaptar desde parte de las primeras fases del desarrollo inicial, alterando todas las demás; dependiendo de cuán profundos sean los cambios. El modelo cascada común es particularmente costoso en mantenimiento, ya que su rigidez implica que cualquier cambio provoca regreso a fase inicial y fuertes alteraciones en las demás fases del ciclo de vida.

Durante el período de mantenimiento, es común que surjan nuevas revisiones y versiones del producto; que lo liberan más depurado, con mayor y mejor funcionalidad, mejor rendimiento, etc. Varias son las facetas que pueden ser alteradas para provocar cambios deseables, evolutivos, adaptaciones o ampliaciones y mejoras.

Básicamente se tienen los siguientes tipos de cambios:

- **Perfectivos:** aquellos que llevan a una mejora de la calidad interna del software en cualquier aspecto: reestructuración del código, definición más clara del sistema y su documentación; optimización del rendimiento y eficiencia.
- **Evolutivos:** agregados, modificaciones, incluso eliminaciones, necesarias en el software para cubrir su expansión o cambio, según las necesidades del usuario.
- **Adaptivos:** modificaciones que afectan a los entornos en los que el sistema opera, tales como: cambios de configuración del hardware (por actualización o mejora de componentes electrónicos), cambios en el software de base, en gestores de base de datos, en comunicaciones, etc.
- **Correctivos:** alteraciones necesarias para corregir errores de cualquier tipo en el producto software desarrollado.

5.4.1 Mantenimiento en la aplicación Málaga Común

La aplicación se ha puesto en producción a principios de Enero de 2011 y durante dos meses se ha realizado un mantenimiento de ella. Este mantenimiento ha englobado cambios:

- **Perfectivos:** tras realizar pruebas de carga se ha optimizado el código. La documentación también se ha mejorada tras ser leída por los usuarios.
- **Evolutivos:** se ha agregado funcionalidad pedida por los usuarios tras probar la aplicación.
- **Adaptativos:** para adaptar la aplicación al entorno de producción final.
- **Correctivos:** se han corregido diversos errores que han aparecido.

CAPÍTULO 6

Málaga Común:

MANUALES PARA LOS DISTINTOS TIPOS DE USUARIOS

Los distintos manuales de usuario de la aplicación Málaga Común se incluyen en este capítulo, que consta de estos 6 apartados:

1. **Instalación de la aplicación.** Proceso de instalación.
2. **Página web principal.** Funcionalidad de los usuarios que acceden a la aplicación sin registrarse, y que sólo pueden consultar la información sobre el sistema de moneda complementaria (las ofertas y las demandas publicadas).
3. **Sistema de gestión de la moneda complementaria.** Funcionalidad para los usuarios que se registran en la aplicación, y pueden participar en el sistema de moneda complementaria.
4. **Administración de la moneda complementaria.** Configuración de la aplicación por parte del administrador del sistema.
5. **Desarrollo de nuevas funcionalidades.** Configuración de la aplicación por parte de un desarrollador para ampliar su funcionalidad.
6. **Ejemplo de uso.** Ejemplo de configuración y utilización de la aplicación.

6.1 Instalación de la aplicación

Importante: la instalación de la aplicación la debe realizar una persona con conocimientos sobre sistemas webs. En estos pasos se da por sentado que la persona que los va a realizar conoce la terminología y conoce como instalar aplicaciones webs.

La instalación de la aplicación sobre un servidor web requiere los siguientes programas gratuitos que deben encontrarse instalados y funcionando correctamente:

- Servidor web Apache con el modulo `mod_rewrite` activado. Este programa es el que hace que el ordenador sea un servidor web. Se puede descargar de forma gratuita desde su web oficial en la dirección <http://apache.org>. Asimismo en esta misma dirección encontramos toda la información sobre su instalación y configuración.
- PHP5 versión 5.2.0 o superior. Este programa es el que nos va a poder permitir ejecutar páginas programadas en el lenguaje en que se encuentra desarrollada la aplicación de moneda complementaria. Es gratuito y se puede descargar desde la web <http://www.php.org>. En esta misma dirección se encuentra además toda la documentación necesaria para su instalación y configuración para poder ser ejecutado junto con el servidor web Apache.
- Base de datos MySQL. Este el sistema gestor de base de datos que utiliza la aplicación. Es en él donde se almacenan todos los datos. Es gratuito y se puede descargar desde su web oficial <http://www.mysql.org>. Para su instalación también podemos encontrar diferentes manuales en dicha web.

Normalmente al contratar un servicio de *hosting* (alojamiento web para nuestra aplicación) o bien utilizar uno gratuito, ya deberá tener todas estas aplicaciones instaladas. Sólo necesitamos instalarlas en caso de querer ejecutar la aplicación de moneda complementaria en un ordenador personal con acceso a Internet.

En el desarrollo del software se ha pretendido que la instalación sea lo más sencilla posible de modo que los pasos que hay que dar simplemente son:

1. Se debe copiar el directorio llamado *silverstripe* que se distribuye con la aplicación, en la raíz del servidor web (la que nos indique nuestro proveedor de alojamiento).
2. Se debe ejecutar el fichero de comandos SQL llamado *ss_malagacomum.sql* que se distribuye junto con la aplicación en el directorio raíz.
3. Se debe configurar el fichero *silverstripe/mysite/_config.php* para indicar los datos de conexión a la base de datos según se muestra en la Figura 6.1.

```
$databaseConfig = array(  
    "type" => "MySQLDatabase",  
    "server" => "localhost",  
    "username" => "root",  
    "password" => "admin",  
    "database" => "SS_mysite",  
);
```

Figura 6.1: Código de configuración de la base de datos para la aplicación

6.2 Página Web Principal

En la Figura 6.2 vemos el portal de entrada a la parte pública del sistema. Desde ella se puede consultar información genérica sobre lo que es un sistema de moneda complementaria, ver las ofertas y demandas existentes, contactar con el administrador de la web, registrarse en el sistema, y acceder a la parte de gestión una vez que el usuario ya se encuentre registrado.

Desde el menú superior se puede acceder a los diferentes apartados de la aplicación, los cuales detallamos en las siguientes subsecciones. Igualmente también existen una serie de enlaces accesibles en todo momento en la parte inferior de la pantalla que también comentaremos en las siguientes subsecciones.

6.2.1 Málaga Común

Consta de una serie de apartados accesibles por cualquier usuario sin necesidad de estar registrado en el sistema, donde se definen diferentes aspectos de interés sobre los sistemas de moneda complementaria. Como se ve en la Figura 6.2, en el menú del lateral izquierdo se ven las secciones:

- **¿Qué es?.** Explica brevemente que es un sistema de moneda complementaria.
- **¿Como funciona?.** Explica como funcionan los diferentes sistemas de moneda complementaria.
- **¿Quién puede participar?.** Explica quien puede participar en este tipo de sistemas.
- **Enlaces de interés.** Donde se listan una serie de enlaces de interés.



Figura 6.2: Página de inicio de la aplicación

8.2.1 Intercambios

Muestra un listado de los intercambios existentes en el sistema, estos pueden ser de dos tipos: ofertas y demandas. Cada tipo se lista en un apartado diferente accesible a través del menú del lateral izquierdo. Un intercambio puede ser de un servicio, o de un objeto. Por ejemplo, un usuario que quiera vender productos de su huerto puede indicarlo con una oferta, en la que se detallará el precio de cada uno de los productos que vende. Por otro lado, un usuario que necesite recibir clases de inglés, puede solicitar una demanda indicándolo. Cada oferta o

demanda del sistema se encuentran categorizadas. Pulsando sobre la opción de menú *Ofertas* se abre una página con todas las categorías existentes (Figura 6.3). Al pulsar sobre la imagen que representa una categoría se abrirá otra página donde se listan todas las ofertas disponibles para dicha categoría (Figura 6.4). De igual forma podemos proceder para las demandas.



Figura 6.3: Categorías para las ofertas



Figura 6.4: Listado de ofertas pertenecientes a una categoría

Cabe destacar que simplemente se muestra un listado con las ofertas o demandas existentes, y de cada una sólo se indica el sistema al que pertenece (normalmente *malagacomun*), y el título de la oferta o demanda. Si se pulsa sobre la imagen de una pequeña lupa situada a la izquierda de cada oferta (Figura 6.4) se puede ver el detalle de cada oferta (igualmente para las demandas), siempre que el usuario se encuentre registrado y dado de alta en el sistema al que pertenece la oferta, de lo contrario se indicará mediante un error que no se tienen permisos para ver los detalles de esa oferta, junto con información relevante sobre cómo registrarse.

8.2.2 Participantes

Esta página (Figura 6.5) es el punto de entrada a la aplicación de gestión del sistema de moneda complementaria (apartado 6.3). Además desde aquí se pueden registrar los nuevos usuarios o recuperar sus contraseñas en caso de no recordarlas. Desde ella se debe introducir la dirección de correo electrónico y la contraseña del usuario que quiere acceder a la gestión del sistema de moneda complementaria. También existe la posibilidad, marcando el check que lo indica con el texto *Recordar la contraseña desde este equipo para no tener que introducirla cada vez (NOTA: Esto no debe usarse si el ordenador lo usan varias personas)*, de que la conexión se mantenga en futuros accesos a la web. Es decir, que no se tengan que volver a introducir las credenciales ya que el navegador (Internet Explorer, FireFox, etc.) los guardará en el ordenador del usuario y los introducirá automáticamente al acceder a la página. Es importante tener en cuenta que si varias personas van a utilizar el mismo ordenador para acceder a cuentas distintas, no se debería marcar esta opción, pues siempre entrarían con la cuenta del usuario que la ha marcado.

8.2.3 Contactar

Muestra un formulario desde el cual se puede contactar con el administrador de la web (Figura 6.6). Los datos que se deben rellenar son el nombre, el correo electrónico, y el asunto que se desea tratar. Una vez se pulse el botón enviar, la información introducida le llegará al administrador de la web vía correo electrónico.



Figura 6.5: Página de acceso a la gestión del sistema

6.2.6 Opciones del menú inferior

En la parte inferior derecha de la página (Figura 6.2), se pueden ver una serie de enlaces genéricos. Estos enlaces son:

- **Inicio.** Direcciona a la página principal de la aplicación.
- **Contactar.** Direcciona a la página de contacto (apartado 6.2.4).
- **Mapa web.** Muestra un listado de todas las páginas de la aplicación (Figura 6.7).
- **Aviso legal.** Este enlace sólo está accesible a través del mapa web. Muestra información sobre las reglas de participación en el sistema.
- **Política de protección de datos.** Muestra información sobre la política de protección de datos aplicada en la aplicación. La página sólo es accesible a través de la página del mapa web.



Figura 6.6: Página de contacto



Figura 6.7: Página con el mapa del sitio

6.3 Sistema de gestión de la moneda complementaria

Un sistema de moneda complementaria se basa en el intercambio de servicios u objetos entre los usuarios, estos servicios u objetos se anuncian mediante ofertas y demandas. Una oferta es el medio por el cual un usuario se ofrece para realizar un servicio u ofrecer un objeto, mientras que una demanda es el medio por el cual un usuario lo solicita. La aplicación para la gestión de monedas complementarias Málaga Común permite el uso de varios sistemas de moneda complementaria por un mismo usuario registrado.

En este apartado veremos la gestión de los datos de un usuario registrado y las opciones que tiene disponibles. El punto de entrada a la aplicación de gestión es la opción de menú *Participantes* (Figura 6.5) desde la que podremos introducir nuestras credenciales (correo electrónico y contraseña, como se explica en el apartado 6.3.3), registrarse en el sistema (apartado 6.3.1), o cambiar la contraseña en caso de no recordar la anterior, a través del correo electrónico (apartado 6.3.2).

6.3.1 Registro de un usuario en el sistema

Para registrarse en el sistema se debe acceder a la opción de menú *Participantes* (Figura 6.5). En esta página podemos ver el enlace *Registrarse* en el menú de la izquierda de la página. Si pulsamos sobre él accedemos al formulario de registro de un nuevo usuario. Este formulario se encuentra dividido en dos páginas. Una primera (Figura 6.8) donde se nos pedirán nuestros datos personales y debemos seleccionar los sistemas en los que nos queremos dar de alta (si sólo existe un sistema aparecerá seleccionado por defecto). Es importante recordar que la aplicación para la gestión de los sistemas de moneda complementaria permite el uso de varios sistemas de moneda complementaria por un mismo usuario registrado.

Los campos marcados con un asterisco (*) son obligatorios, además también se valida que el correo electrónico tenga un formato correcto, y no se esté utilizando ya por otro usuario.

Puesto que uno de los requisitos para que un sistema de moneda complementaria funcione correctamente es que todos los usuarios tengan algo que ofrecer, en la segunda página del formulario de registro (Figura 6.9) debemos introducir las ofertas mínimas para darse de alta en cada uno de los sistemas seleccionados en la página anterior. Este número de ofertas no es un valor fijo, sino que dependerá de cada sistema en concreto.

Participantes » Registrarse

Rellene el siguiente formulario para registrarse en el sistema:

Nombre*:

Primer Apellido*:

Segundo Apellido

Sexo

Teléfono fijo

Teléfono movil

Email*:

Password*:

Confirmar Contraseña

Sistemas disponibles*:

Nombre	Descripcion
<input checked="" type="checkbox"/> malagacomun	Sistema para Málaga Común

*Datos obligatorios

Siguiente

Figura 6.8: Formulario de registro de datos personales y sistemas disponibles

Participantes » Registrarse

Rellene el siguiente formulario para registrarse en el sistema:

Para registrarse en el sistema malagacomun hay que crear 1 ofertas iniciales

OFERTA 1 PARA MALAGACOMUN

Categoría:

Título*:

Descripción*:

Tipo del precio*:
 Por horas Fijo A convenir

Normalmente se piden 10 comunes por hora

Precio en comunes

*Datos obligatorios

Volver Enviar

Figura 6.9: Formulario de registro de ofertas obligatorias

Al igual que en la primera página del formulario (Figura 6.8), los campos marcados con un asterisco son obligatorios. En la fase de registro no se pueden añadir imágenes a las

ofertas, pero una vez registrado se pueden añadir tantas como se desee desde la opción de edición de ofertas (apartado 6.3.3.2.2).

Una vez completado el registro se retorna a la página de autenticación, desde la que deberemos introducir el correo electrónico y la contraseña con los que nos hemos registrado para poder acceder a nuestra área personal.

Al completarse el registro se enviará un correo electrónico a la dirección indicada con el siguiente mensaje: *“Gracias por registrarte en el sistema de moneda complementaria, en breve tus datos van a ser validados por un Administrador, mientras tanto la funcionalidad va a estar limitada”*. En el que se indica que hasta que un administrador del sistema no valide el registro sólo se podrán crear ofertas (apartado 6.3.3).

6.3.2 Restablecer una nueva contraseña

Desde la pantalla de autenticación también podemos restablecer una nueva contraseña. Para ello debemos acceder a la opción *He perdido mi contraseña* del menú de la izquierda de la Figura 6.5.



The image shows a web browser window displaying the 'MÁLAGA COMÚN' website. The header features the logo 'MÁLAGA COMÚN' and the text 'SISTEMA DE MONEDA COMPLEMENTARIA'. A navigation menu includes 'Málaga común', 'Intercambios', 'Participantes', and 'Contactar'. The main content area is titled 'Participantes » Contraseña Perdida' and contains the instruction: 'Introduzca su dirección de e-mail, y le enviaremos un enlace, con el cual podrá restaurar su contraseña'. Below this is a form with an 'E-mail' label and a text input field containing 'jagarrido@malagacomun.org'. A button at the bottom of the form reads 'Envíenme el enlace para restaurar la contraseña'. On the left side of the page, there is a vertical menu with three options: '» Iniciar sesión', '» Registrarse', and '» He perdido mi contraseña'.

Figura 6.10: Formulario de solicitud del enlace para renovar la contraseña

Las contraseñas se almacenan en el sistema de forma codificada, de modo que nadie pueda leerlas, por tanto recuperar la contraseña es imposible. Desde esta opción lo que se crea es una nueva contraseña para el usuario que lo solicite. Para ello debemos introducir nuestro correo electrónico (Figura 6.10), y el sistema nos enviará un correo con un enlace a una página desde la que podemos definir una nueva contraseña.

6.3.3 Acceso a los datos de un usuario registrado

Mediante el correo electrónico y la contraseña se puede acceder al área privada del usuario. Si dichas credenciales son correctas, se debe seleccionar el sistema en el que se desea entrar **sólo en el caso de que el usuario se encuentre registrado en más de un sistema** (Figura 6.11). Si el usuario sólo se encuentra registrado en un sistema se accederá a este de forma automática.

En un principio la funcionalidad de un nuevo usuario se encuentra limitada, pudiendo simplemente ver sus datos y los mensajes recibidos, y crear ofertas. De esto es informado a través de un mensaje enviado por el sistema en el momento de realizarse el registro (apartado 6.3.3.3). Al realizarse el registro de un nuevo usuario, un administrador del sistema es informado. Este deberá comprobar que los datos son correctos y validar al usuario. En el momento que el usuario sea validado por el administrador ya podrá interactuar de forma completa en el sistema. Al igual que con el registro, el usuario será informado mediante la notificación seleccionada (apartado 6.3.3.2.1) de que ha sido validado en el sistema y que ya puede participar en el. Si el mensaje fallara, la devolución de dicho mensaje será recibida por el administrador, de forma que éste sabrá que se trata de una cuenta de correo con errores (posiblemente falsa).



Figura 6.11: Ventana de selección de un sistema

Las opciones disponibles para el usuario se encuentran en el menú de la izquierda (Figura 6.12).

6.3.3.1 Funcionalidades genéricas de todas las opciones

En todas las opciones existen una serie de funcionalidades genéricas que veremos en este apartado para no tener que ir repitiéndolas a lo largo de cada opción.

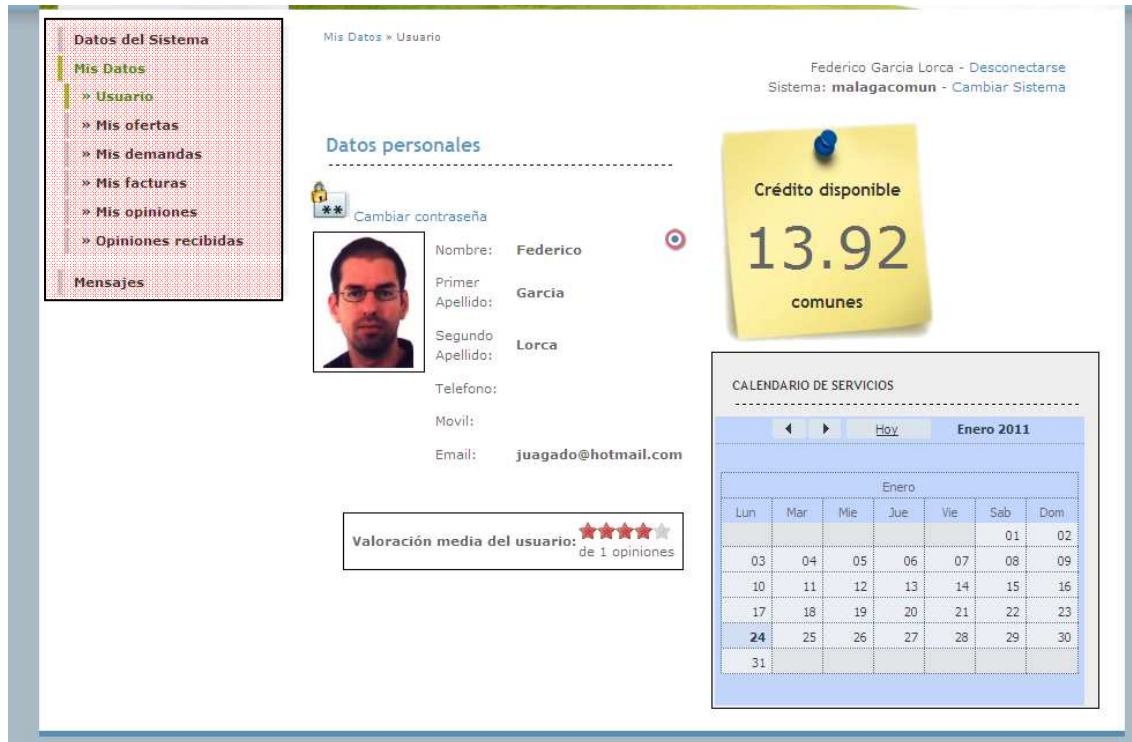


Figura 6.12: Menú lateral

6.3.3.1.1 Desconectarse

Dentro de cada opción del área privada del usuario se indica, en la parte superior derecha, el nombre y apellidos del usuario seguidos de un enlace para desconectarse del sistema (Figura 6.12). Este enlace desconecta al usuario del sistema. Es extremadamente recomendable cerrar cada sesión a través de este enlace y no cambiando directamente de dirección, ni cerrando el navegador, puesto que de lo contrario la conexión se mantiene activa, provocando que otras personas que utilicen el mismo ordenador sin cerrar el navegador puedan acceder a nuestros datos.









6.3.3.1.2 Cambiar de sistema

Una acción importante es la de poder conmutar entre los diferentes sistemas en los que se

encuentra registrado el usuario conectado, esto se puede hacer a través del enlace para cambiar de sistemas que aparece en el parte superior derecha del área de trabajo de la aplicación, junto con el nombre del sistema actual (Figura 6.12).

6.3.3.1.3 Borrado en cascada

En las opciones en las que aparezca un listado de componentes existe la posibilidad de borrar de una vez todos los elementos seleccionados (Figura 6.13) siempre que el usuario tenga permisos para ello, si existe algún componente de los seleccionados sobre los que no tiene permisos de borrado, este componente no se borrará. Debajo de cada listado existe un enlace para seleccionar todos los componentes, además de otro para desmarcarlos a todos. Una vez se pulse el enlace de borrado en cascada (icono con una X roja) se preguntará una sola vez si se desea borrar todos los elementos marcados.

	Asunto	Emisor
<input type="checkbox"/>		
<input checked="" type="checkbox"/>	 Bienvenido al sistema de moneda complementaria	Mensaje del sistema
<input checked="" type="checkbox"/>	 Servicio aceptado	Mensaje del sistema
<input checked="" type="checkbox"/>	 Servicio aceptado	Mensaje del sistema
<input checked="" type="checkbox"/>	 Servicio rechazado	Mensaje del sistema
<input checked="" type="checkbox"/>	 Servicio rechazado	Mensaje del sistema
<input checked="" type="checkbox"/>	 Servicio rechazado	Mensaje del sistema
<input checked="" type="checkbox"/>	 Oferta modificada	Mensaje del sistema
<input checked="" type="checkbox"/>	 Servicio aceptado	Mensaje del sistema

1 Filas de 1 a 8 de 8

[Marcar todos los elementos](#) / [Desmarcar todos los elementos](#)

Para todos los componentes marcados:  

Figura 6.13: Listado y sus opciones

Los datos se muestran agrupados en filas de 10, cuando existen más elementos se pagan, de forma que para cambiar de página hay que pulsar en los índices que aparecen en

la parte inferior derecha de la tabla. Donde se indica además el rango de filas mostradas y el total de filas existente (Figura 6.13).

Importante: la selección de una fila se mantiene a través del paginado, es decir, si se cambia de página los componentes seleccionados en otra se mantienen seleccionados.

6.3.3.1.4 Exportar datos a PDF

También existe la posibilidad de exportar la información de todos los elementos seleccionados a un documento PDF. Esto se consigue con el icono que hay a la derecha del icono de borrado, tal y como se ve en la Figura 6.13.

6.3.3.1.5 Operaciones de la tabla de un listado

En las opciones de listado los datos se pueden manipular para su visionado a través de una tabla (o rejilla) como la que aparece en la Figura 6.13. Las opciones son las siguientes:

- Pulsando sobre la cabecera de cada campo se puede ordenar el listado ascendente o descendientemente, de forma alternada.
- Debajo del nombre de cabecera existe un cuadro de texto por el que se pueden filtrar los resultados. Escribiendo en el cuadro algún carácter o conjunto de ellos se mostrarán los resultados que contengan esos caracteres en alguna posición.
- Cada fila de la tabla tendrá asociada una serie de acciones que se podrán realizar dependiendo de los permisos del usuario y el estado del componente. Las acciones están siempre a la izquierda de cada fila.
- Como se explicó más arriba, los elementos se encuentran paginados, cuando existen más de 10 elementos se pagan, de forma que para cambiar de página hay que pulsar en los índices que aparecen en la parte inferior derecha de la tabla. Donde se indica además el rango de filas mostradas y el total de filas existente (Figura 6.13).

6.3.3.1.6 Alerta de mensajes

Una alerta en la esquina inferior derecha de la pantalla (Figura 6.14) advierte de la existencia de mensajes en la bandeja de entrada (apartado 6.3.3.3).

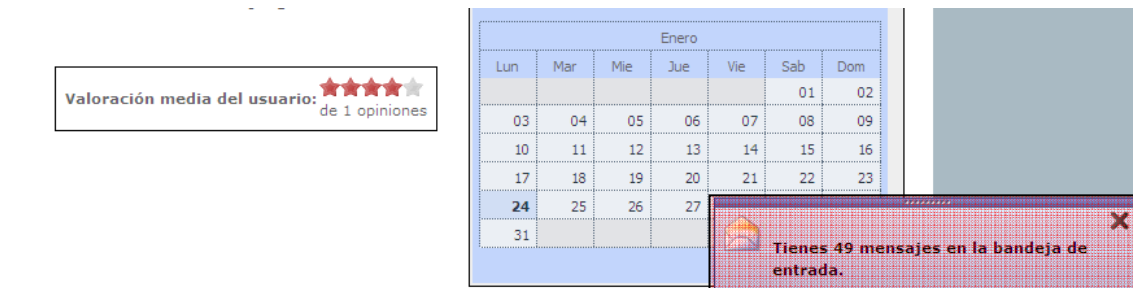


Figura 6.14: Alerta de mensajes en la bandeja de entrada

Esta alerta aparece mientras existan mensajes en la bandeja de entrada del servidor (no del correo electrónico). A los cinco segundos, si el usuario no la ha cerrado desaparecerá sola de la pantalla, y volverá a aparecer como recordatorio cada cinco minutos. Pinchando sobre ella se accede a la bandeja de entrada.

6.3.3.1.7 Acción ver datos del componente

En las opciones que implican un listado de componentes (ofertas, demandas, facturas, mensajes) una acción que aparece siempre es la de ver de forma detallada sus datos (representado por la imagen de una pequeña lupa – Figura 6.13). Esta acción abre una ventana emergente dividida en dos partes, una barra de herramientas superior (Figura 6.15) y el área con los datos del componente.

La barra de herramientas define una serie de acciones a realizar sobre el componente (siempre que el usuario tenga permisos, de lo contrario aparecerán las acciones como una imagen en blanco y negro sin posibilidad de pulsarse). Dos acciones que aparecerán en todos los componentes serán las de avanzar y retroceder, que permitirán la navegación entre diferentes componentes a través de enlaces dentro de la ventana que muestra el detalle de cada componente.

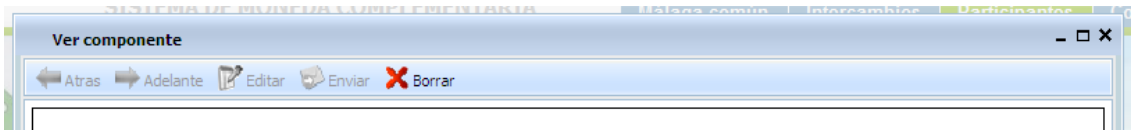


Figura 6.15: Barra de herramientas para el detalle de un mensaje de sistemas

6.3.3.2 Mis Datos

Muestra información sobre aspectos del usuario que se encuentra autenticado en el sistema. Sus opciones son las que se muestran en la Figura 6.16 y se explican en los siguientes subapartados.

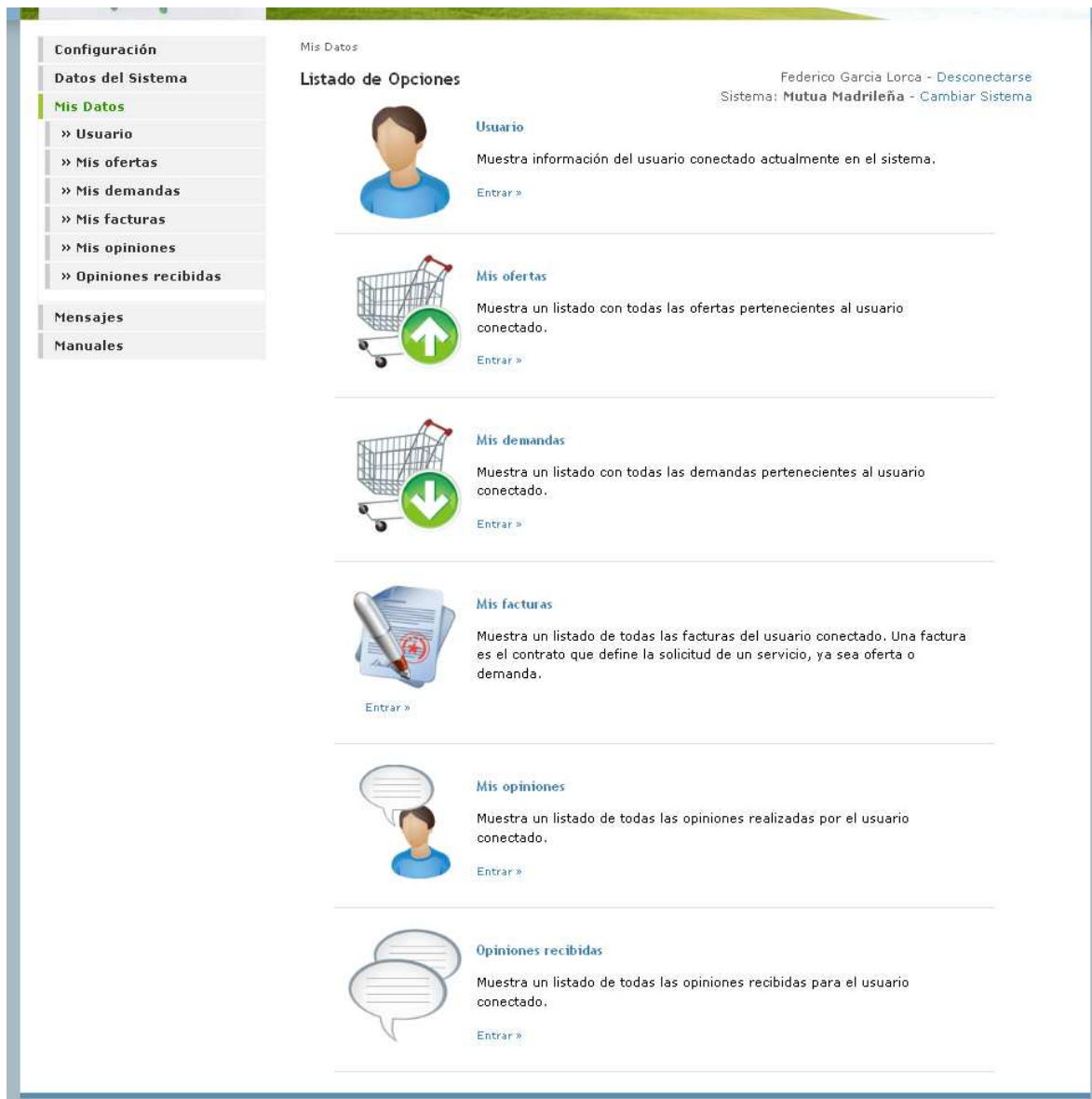


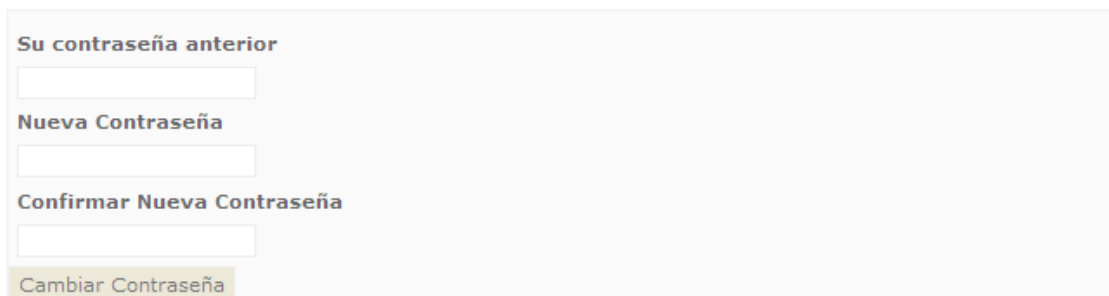
Figura 6.16: Apartados para la opción de menú Mis Datos

6.3.3.2.1 Usuario

Muestra los datos del usuario conectado (Figura 6.18). Esta página se encuentra dividida en cuatro apartados.

- **Crédito disponible:** en la parte superior derecha de la pantalla podemos ver el crédito del usuario conectado. Este puede ser tanto positivo como negativo.
- **Valoración media del usuario:** en la parte inferior izquierda podemos ver la valoración media del usuario. Esta se calcula según las opiniones recibidas para el usuario tal y como se explica en el apartado 6.3.3.2.4.
- **Calendario de servicios:** en la parte inferior derecha de la pantalla podemos ver un calendario donde se encuentran marcados los días en los que el usuario ha definido una factura (apartado 6.3.3.2.4). Al situar el ratón sobre uno de los días marcados aparece información sobre cada uno de los eventos definidos en dicho día. Si pulsamos en alguno se abre en una nueva ventana la información de la factura que lo formaliza.
- **Datos personales:** en la parte superior izquierda de la pantalla podemos ver los datos personales del usuario conectado. Desde esta zona podemos cambiar su contraseña pulsando sobre el enlace cuyo texto lo indica (*Cambiar contraseña*) se abrirá una página (Figura 6.17) con un formulario donde podemos indicar la nueva contraseña.


Puede cambiar su contraseña a continuación.




Formulario para cambiar la contraseña. El formulario contiene tres campos de texto etiquetados como 'Su contraseña anterior', 'Nueva Contraseña' y 'Confirmar Nueva Contraseña'. Debajo de los campos hay un botón que dice 'Cambiar Contraseña'.

Figura 6.17: Formulario para cambiar la contraseña

Datos personales

 Cambiar contraseña

 no photo

Nombre: **Federico**


Primer Apellido: **Garcia**

Segundo Apellido: **Lorca**

Telefono:

Movil:

Email: **juagado@hotmail.com**

Valoración media del usuario:  de 1 opiniones

Crédito disponible
11.96
comunes

CALENDARIO DE SERVICIOS

Hoy **Diciembre 2010**

Diciembre						
Lun	Mar	Mie	Jue	Vie	Sab	Dom
		01	02	03	04	05
06	07	08	09	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Figura 6.18: Visualización de los datos del usuario conectado

Pulsando sobre el botón con un círculo rojo y un punto azul en el centro (Figura 6.18), que se encuentra a la derecha de los datos personales se abrirá un formulario (Figura 6.19) donde podremos modificar los datos personales y añadir una imagen al perfil (o modificar una existente).

Contenido **Sistemas** **Configuración**

Nombre: Federico

Primer Apellido: Garcia

Segundo Apellido: Lorca

Sexo: Hombre

Telefono:

Movil:

Email: juagado@hotmail.com

Descripción: mi foto


Alternativa: mi foto

Foto:

Reemplace Imagen

Con una de su computadora

Con uno del gestor de archivos



Reemplace Imagen Elimine Imagen

Figura 6.19: Formulario para la edición de los datos personales

Este formulario se encuentra dividido en tres pestañas.

Pestaña Contenido: la pestaña contenido define los datos personales propiamente dichos. En ellos podemos modificar el nombre y apellidos, sexo, teléfono, teléfono móvil y el email del usuario. Sobre estos datos se validan que el nombre, primer apellido y email existan, además el email debe tener un formato correcto y no estar siendo utilizado por ningún otro usuario. También podemos añadir una imagen o modificarla en caso de existir. Para añadir una nueva imagen se debe pulsar sobre el botón con una cruz verde situado a la izquierda del campo *Imagen* (Figura 6.19). Al pulsar sobre este botón aparecerán los nuevos campos para añadir la imagen (Figura 6.20).

Los campos para una imagen son la descripción, que es un texto que aparecerá al pasar el ratón sobre ella. La alternativa, que es un texto que aparece si la imagen no puede ser cargada. Y la imagen propiamente dicha, que se puede cargar desde el ordenador local del usuario o bien utilizar una ya existente (subida con anterioridad). Para cargar la imagen debemos seleccionar la opción deseada y pulsar sobre el botón que lo indica (*Agregue imagen*).

Pestaña Sistemas: en la pestaña sistemas se definen los sistemas en los que puede participar el usuario. En todo momento en usuario puede seleccionar los sistemas en los que desea participar, el único requisito es que por lo menos debe seleccionar uno (Figura 6.21). Si un usuario elimina un sistema al que ya pertenecía simplemente dejará de tener acceso a el, sin embargo todos sus datos de mantienen y puede volver a recuperarlos en cualquier momento. Asimismo si selecciona un sistema nuevo debe ser ratificado por el administrador del sistema.

Figura 6.20: Campos para añadir una nueva imagen

Pestaña Configuración: la pestaña configuración esta pensada para albergar opciones de configuración genéricas para un determinado usuario. Actualmente sólo existe la opción para indicar el tipo de notificación. El tipo de notificación puede ser a través del correo electrónico

introducido por el usuario, a través del sistema de notificaciones de la aplicación (apartado 6.3.3.3), o ambos a la vez, que es el que se encuentra seleccionado por defecto.

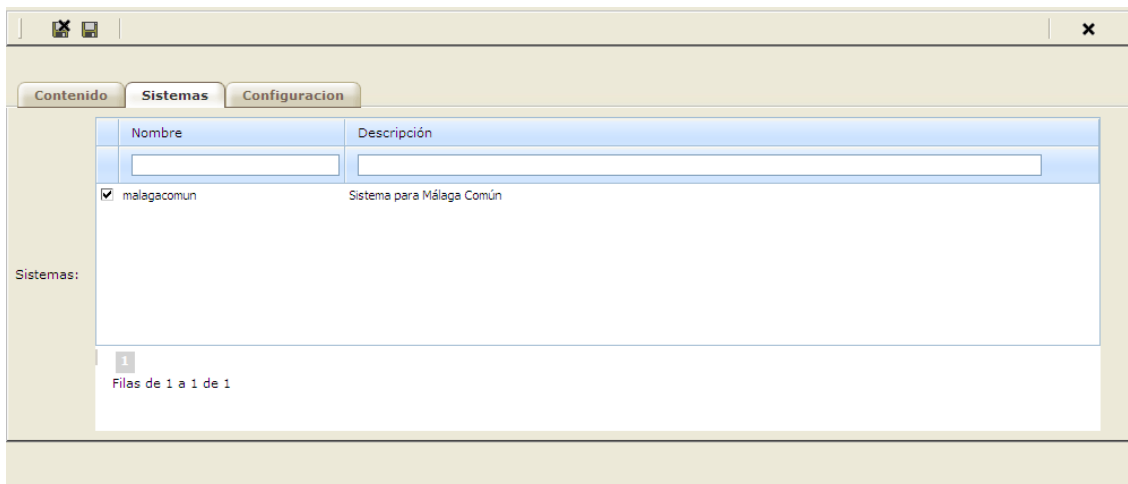


Figura 6.21: Selección de sistemas para un usuario

6.3.3.2.2 Mis Ofertas

Muestra un listado con todas las ofertas pertenecientes al usuario conectado. Por defecto veremos todas las ofertas, sin embargo podemos filtrar las ofertas mostradas atendiendo a una serie de parámetros (Figura 6.22).

Filtrado de datos

Categoría
Cuidado de personas y animales, compañía

Estado de publicación:
Publicada

Creada antes de:
01/02/2011

Creada después de:
03/02/2011

Mostrar resultados

Figura 6.22: Formulario de filtrado de ofertas

- **Categoría:** si el valor se encuentra en blanco buscará las ofertas pertenecientes a cualquier categoría. De lo contrario sólo mostrará las pertenecientes a la categoría indicada.

- **Estado de publicación:** si se deja en blanco devolverá todas las ofertas independientemente de su estado. De lo contrario se pueden seleccionar las ofertas que se encuentren publicadas o las que no.
- **Creada antes de:** si se introduce alguna fecha, sólo mostrará las ofertas que se han creado antes de esta. De lo contrario las mostrará todas.
- **Creada después de:** si se introduce alguna fecha, sólo mostrará las ofertas que se han creado después de la fecha indicada. De lo contrario se mostrarán todas.

Desde esta página además podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1). Pulsando la opción para ver el detalle podremos realizar todas las acciones definidas para las ofertas.

Para crear una nueva oferta debemos pulsar el botón con el texto *Añadir oferta*. Al pulsar sobre el botón para crear la oferta se nos abrirá en una nueva ventana el formulario para introducir los datos necesarios (Figura 6.23).

Pestaña Contenido: la pestaña contenido define los datos de la oferta.

- **Categoría:** indica la categoría para clasificar a la oferta y su selección es opcional.
- **Título:** es el título que va a tener la oferta, debe existir obligatoriamente.
- **Descripción:** en este campo es donde se describe la oferta, y es obligatorio.
- **Tipo del precio:** el tipo del precio puede ser un valor fijo, puede ser por horas, de forma que este se multiplique por cada hora de duración del servicio, o también se puede indicar que sea a convenir de forma que quede todo un poco más abierto a la negociación. Este valor es orientativo ya que en cualquier momento antes de realizar el pago de un servicio este valor puede ser modificado. Por defecto el valor será *A convenir*.

- **Precio:** indica el precio de la oferta. Es un valor orientativo ya que en cualquier momento antes de realizar el pago de un servicio este valor puede ser modificado. De hecho no es obligatorio poner valor alguno.
- **Imagen:** se pueden añadir a la oferta tantas imágenes como se deseen. Para añadir una nueva imagen se debe pulsar sobre el botón con una cruz verde situado a la derecha del campo (Figura 6.23). Al pulsar sobre este botón aparecerán los nuevos campos para añadir la imagen (Figura 6.20). Los campos para una imagen son la descripción, que es un texto que aparecerá al pasar el ratón sobre ella. La alternativa, que es un texto que aparece si la imagen no puede ser cargada. Y la imagen propiamente dicha, que se puede cargar desde el ordenador local del usuario o bien utilizar una ya existente (subida con anterioridad). Para cargar la imagen debemos seleccionar la opción deseada y pulsar el botón que lo indica (*Agregue imagen*).

Figura 6.23: Formulario para crear una nueva oferta

El formulario de edición y creación de una nueva oferta (Figura 6.23) posee en la parte superior los botones para guardar, guardar y salir, y salir, con los iconos de un disco, un disco con un aspa, y un aspa respectivamente. Al salir del formulario aparecen los detalles de la nueva oferta creada junto con las acciones que se pueden realizar en la parte superior de la ventana (Figura 6.24). Por defecto, al crear una nueva oferta aún no puede ser vista por el

resto de usuarios. Esto permite corregir posibles errores antes de hacerla pública. Para hacerla pública se debe pulsar el botón *Publicar* que se encuentra en la barra de opciones de la ventana de detalle de la oferta. Es importante notar que al crear una oferta se le notifica al administrador del sistema la creación de dicha oferta, de forma que pueda modificarla o incluso dejar de hacerla pública en caso de no cumplir con las normas del sistema. Además se le puede quitar al usuario la posibilidad de publicar sus ofertas si así lo estima el administrador.

Una vez que se ha publicado una oferta, en cualquier momento se puede dejar de hacerla visible públicamente. También se puede editar en cualquier momento, y hay que tener en cuenta que al editar una oferta, ésta deja de ser pública, y se notifica al administrador de que se han realizado cambios, para evitar ofertas que no se ajusten a las normas del sistema.











Figura 6.24: Detalle de una oferta

Las acciones disponibles para una oferta se pueden ver en la Figura 6.24 y son las definidas en la Tabla 6.1.

6.3.3.2.3 Mis demandas

Muestra un listado con todas las demandas pertenecientes al usuario conectado. Sus opciones son similares a las de la opción *Mis ofertas* (apartado anterior 6.3.3.2.2).

Tabla 6.1: Acciones disponibles para una oferta

	Ver: acción que permite ver los datos detallados de la oferta.
	Editar: permite modificar los datos de una oferta. Cada vez que se editan los datos de una oferta esta deja de estar visible públicamente, de forma que tenga que ser publicada nuevamente. Esta opción solo estará disponible mientras la oferta no tenga asociados servicios. Una vez que una oferta tiene asociado un servicio y para mantener la integridad de los datos de las facturas (apartado 6.3.3.2.4) las ofertas no se pueden modificar (pero sí clonar para copiar esa oferta creando una nueva).
	Borrar: elimina una oferta. Esta opción solo estará disponible mientras la oferta no tenga asociado algún servicio, de lo contrario y para mantener la integridad del sistema no se podrán borrar las ofertas.
	Publicar: publica una oferta para hacerla visible públicamente. Solo aparece en ofertas que aún no están publicadas.
	Solicitar servicio: esta acción es utilizada para que cualquier usuario que no sea el autor de la oferta y tenga crédito suficiente pueda solicitar la oferta.
	Ocultar: elimina una oferta del área pública de forma que solo sea visible por el autor y los administradores. Sólo aparece en ofertas ya publicadas.
	Clonar: puesto que una vez que se ha solicitado una oferta esta no se puede modificar, se permite clonar sus datos en otra nueva oferta que aún no ha sido solicitada y por tanto se pueden modificar sus datos. La nueva oferta no estará aún publicada.
	Borrar: una vez que una oferta tenga algún servicio solicitado, el borrado pasa a ser un borrado ficticio, en el que desaparece la oferta pero se mantienen sus datos en el sistema.

6.3.3.2.4 Mis facturas

Muestra un listado de todas las facturas del usuario conectado. Una factura es el contrato que define la solicitud de un servicio, ya sea oferta o demanda. Por defecto veremos todas las facturas del usuario, sin embargo podemos filtrar las facturas mostradas atendiendo a una serie de parámetros (Figura 6.25).

Filtrado de datos

Tipo de servicio:
Oferta

Creada antes de:
01/02/2011

Creada después de:
06/02/2011

Mostrar resultados

Figura 6.25: Formulario de filtrado de facturas

- **Tipo de servicio:** si el valor se encuentra en blanco buscará las facturas pertenecientes a cualquier tipo (oferta y demanda). De lo contrario sólo mostrará las pertenecientes al tipo seleccionado.
- **Creada antes de:** si se introduce alguna fecha, sólo mostrará las facturas que se han creado antes de esta, mientras que de lo contrario las mostrará todas.
- **Creada después de:** si se introduce alguna fecha, sólo mostrará las facturas que se han creado después de la fecha indicada. De lo contrario se mostrarán todas.

Ver componente

Factura núm: 945

Fecha de creación: 18/12/2010 2:14pm

Prestador del servicio: Federico Garcia Lanza

Receptor del servicio: Francisco Fernandez Sanchez

Servicio: Clase de informática

Sistema del servicio: malaga comun

Precio del servicio: 12 comunes por hora

Horas del servicio: 1

Total a pagar: 11,96 + 0,04 de impuestos

Estado: Servicio realizado

Horarios del servicio

Esta fecha me viene bien

Desde 19/12/2010 16:00 hasta 19/12/2010 17:00

Observaciones

Autor: Francisco Fernandez Sanchez

Fecha: 18/12/2010 2:14pm

Desear solicitar el servicio, pero he cambiado el precio por hora porque me parece un poco caro.

Figura 6.26: Detalle de una factura

Desde esta ventana además podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1). Pulsando la opción para ver el detalle de una factura (icono de una lupa) se podrán ver sus datos (Figura 6.26), y realizar las operaciones para los que el usuario tenga permisos (apartado 6.3.3.2.2).

Flujo de una factura

Al solicitar un servicio (apartado 6.3.3.2.2), ya sea una oferta o una demanda, la factura va a pasar por varios estados.

- Un usuario solicita un servicio, que puede ser una oferta o una demanda. El solicitante puede modificar tanto el precio, como indicar si este es fijo o por horas. Además puede indicar una fecha y algunas observaciones sobre el servicio si así lo desea. Una vez que se guarda esta información, se genera una factura. Esta factura se encuentra en estado de solicitud de aceptación por parte del otro usuario implicado en el servicio, que recibe una notificación indicando la solicitud del servicio. En este estado, el usuario que no ha generado la factura puede aceptarla, rechazarla o bien modificarla. El usuario que ha generado la factura puede modificarla mientras que el otro usuario no realice ninguna acción sobre ella.
- El usuario que no ha solicitado el servicio puede modificarlo, aceptarlo o rechazarlo. Si lo modifica, ya no podrá aceptarlo ni rechazarlo. Se le notifica al otro usuario implicado en el servicio que se han modificado las condiciones del servicio que estableció y es él ahora el usuario que debe aceptarlo o rechazarlo. Si el servicio se rechaza, sobre el servicio ya no se podrá realizar ninguna acción, se le notificará al otro usuario implicado y la factura se cierra. Por último si se acepta pasa a un nuevo estado pendiente de realización. Mientras los usuarios estén modificando las condiciones del servicio, se va alternando el usuario que debe aceptarlo.
- Una vez el servicio sea aceptado se puede modificar, rechazar (se pide confirmación) o pagar. Si se rechaza ya no se podrán hacer más acciones sobre él y quedará archivado. Si se modifica vuelve al estado anterior, en el que se encuentra pendiente de aceptación. Por último, si se paga, acción que sólo puede realizar el usuario que va

a dar el crédito, se indica que la transacción se ha efectuado con éxito y la factura pasa a estado *Servicio realizado*.

- Una factura en estado *Servicio realizado* ya no se puede modificar. Se encuentra realizada y el pago ya se ha efectuado. Sobre ella lo único que se puede hacer es opinar sobre el servicio realizado. Las opiniones las puede realizar cualquiera de los dos usuarios implicados en el servicio.

Crear una nueva opinión

Desde el detalle de una factura finalizada se puede crear una nueva opinión, o bien modificar una existente. Pulsando sobre el botón de opinar se nos abrirá el formulario para crear una nueva opinión (Figura 6.27).

Los campos de una opinión son la valoración, que debe ser un valor entre 0 y 10 e indica una puntuación media sobre el servicio realizado. Y la opinión en sí, que es un texto donde se pueden exponer las observaciones del servicio realizado.

Se puede valorar tanto al usuario que ha realizado un servicio, como al que lo ha recibido.

Por defecto, cuando se crea (o se graba) una nueva opinión aún no puede ser vista por el resto de usuarios. Para hacerla pública al resto de usuarios hay que publicar la opinión pulsando sobre el botón *Publicar* que aparece al ver el detalle de la opinión pulsando sobre al icono con la lupa de cada fila (Figura 6.13). Además, las opiniones se pueden editar siempre que se desee. Sin embargo, cada vez que se edite, la opinión dejará de estar publicada, de forma que se deba publicar explícitamente pulsando sobre el botón *Publicar*. De esa forma se evita que sean visibles opiniones sin finalizar.

Por otra parte, cada vez que una opinión se cree o modifique, se le notificará al administrador de forma que exista un control sobre las opiniones del sistema. Si el administrador así lo determina se le puede quitar a un usuario los permisos para publicar sus opiniones, siendo el administrador el único que pueda hacerlo.

Desde la opción de menú *Mis facturas*, pulsando el botón *Añadir factura*, se pueden crear facturas sin necesidad de que previamente se haya solicitado un servicio. Esta opción es útil cuando el servicio se apalabra previamente entre los usuarios sin necesidad de utilizar la aplicación, por ejemplo por teléfono. Al pulsar sobre el botón para crear la factura se nos

abrirá el formulario para introducir los datos necesarios (Figura 6.28). Estos se encuentran divididos en dos pestañas.



Figura 6.27: Formulario para crear una nueva opinión

Pestaña Contenido: la pestaña contenido define una serie de datos genéricos.

- **Servicio:** indica el servicio que se va a asociar a la factura. Si pulsamos sobre el botón con la cruz verde que hay al lado del campo (Figura 6.28) se abrirá en otra ventana del navegador una página donde podemos seleccionar el tipo del servicio (Figura 6.29), que puede ser a su vez una oferta o una demanda. Una vez seleccionado un tipo la ventana se cerrará automáticamente y en el formulario de inserción de datos (Figura 6.28) aparecerá un listado o bien de ofertas, o bien de demandas disponibles donde se tendrá que seleccionar una, que será el servicio al cual esté asociada la factura.
- **Tipo del precio:** el tipo del precio puede ser un valor fijo o un valor por horas. Si el valor es fijo, independientemente del tiempo tardado para realizar el servicio se cargará una cantidad fija a la factura. Mientras que si el valor es por horas, el precio que se introduzca se multiplicará por las horas en las que se ha realizado el servicio. Por defecto se indica el definido en la oferta, aunque para la realización del servicio puede ser modificado en cualquier momento. Cabe destacar que en la factura no tenemos el valor del precio *A convenir* puesto que aquí ya se debe indicar un valor real, aunque es posible modificarlo en todo momento por ambas partes.

- **Precio:** indica el precio del servicio, que puede ser un valor fijo si el tipo es fijo, o bien el precio de cada hora, si el tipo es por horas. Por defecto aparece el definido en la oferta, aunque puede ser modificado en cualquier momento.

- **Horas del servicio:** En este campo se definen las horas en las que se realiza un servicio, hay que tener en cuenta que este campo sólo es necesario si el tipo del precio es por horas, de lo contrario su valor es indiferente. Como se ve en la Figura 6.28, se puede seleccionar un valor automático o manual. En el primer caso las horas se obtienen a través de las imputaciones de horarios introducidas en el calendario, como se explica a continuación. En el segundo se introduce el valor de forma manual, debiendo ser este un valor numérico.

- **Observaciones:** al crear una factura (se solicita un servicio automáticamente) se pueden añadir tantas observaciones como se deseen. Para añadir una observación hay que pulsar en el botón con una cruz verde que hay a la izquierda del campo *Observaciones* (Figura 6.28). Esto hará aparecer una serie de campos nuevos (Figura 6.30). Para borrar una observación, se debe pulsar el botón con el aspa roja que aparece junto a los nuevos campos.

- **Receptor de la factura:** Indica el usuario que va a recibir la factura. Por ejemplo, si en el servicio hemos seleccionado una oferta, este campo representa al usuario que va a solicitar esa oferta (el que paga el servicio). Pero si hemos seleccionado una demanda el receptor de la factura será el encargado de realizar el servicio (el que lo cobra). Por defecto aparece el usuario que crea la factura, aunque puede ser cualquier otro. Esto es así ya que, a veces, el propio usuario que debe cobrar el servicio decide crear la factura, de este modo selecciona su oferta y pone al usuario que le ha realizado el servicio como receptor de la factura.

Los campos para una observación los podemos ver en la Figura 6.30 y son:

- **Autor:** indica cual es el autor de la observación y es un texto fijo que no se puede modificar. Una observación puede ser realizada tanto por la persona que va a recibir el servicio, como por la persona que lo va a realizar.

- **Creada:** indica la fecha en la que se ha creado la observación. Es un valor fijo que no se puede modificar.
- **Texto:** indica el texto de la observación, y es el único campo editable de ella.

Figura 6.28: Formulario de creación de una factura

Figura 6.29: Selector para indicar el tipo de una factura manual

Figura 6.30: Campo de observaciones de una factura

Pestaña Calendario: la pestaña calendario muestra un calendario donde se pueden indicar las fechas y horas en las que se va a realizar el servicio ofertado (Figura 6.31). Aunque no es obligatorio definir una fecha al principio, sí será obligatorio definir antes de realizar el pago si este es por horas y se ha marcado la forma automática. En caso contrario el servicio valdrá cero, pues no hay datos en el calendario de donde sacar las horas que se imputarán.

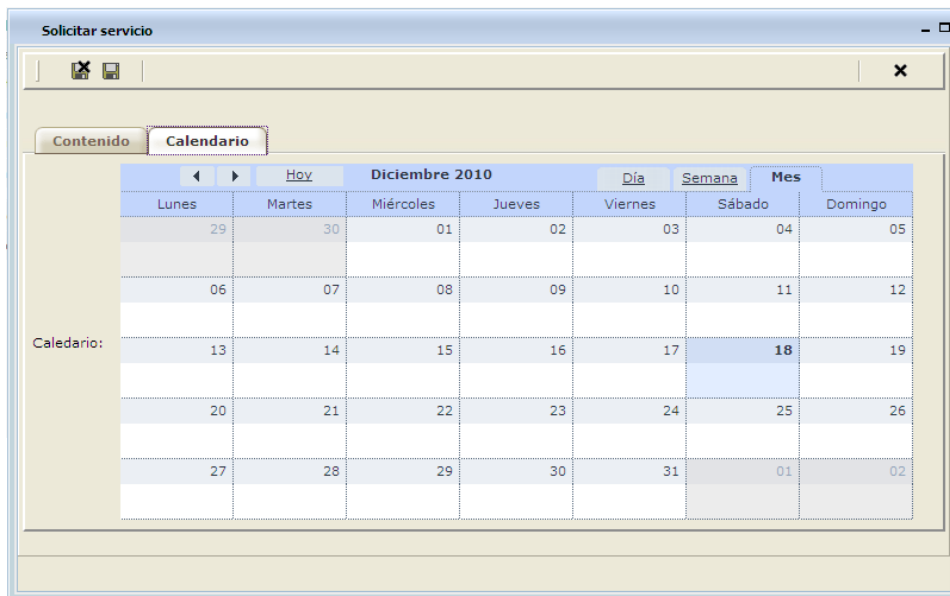


Figura 6.31: Pestaña Calendario para solicitar una oferta

El calendario definido en la pestaña *Calendario* posee las vistas de *Día*, *Semana* y *Mes* que es la que aparece por defecto. En cualquiera de ellas, para definir una nueva entrada se debe pulsar dos veces sobre una de las celdas (días en la vista por mes, horas en la vista por día, y ambos en la vista por semana). Al hacer doble clic se abre una nueva ventana donde se puede indicar una descripción y el período de tiempo que durará el servicio (Figura 6.32).










Figura 6.32: Nueva entrada en el calendario

En la parte inferior de esta ventana (Figura 6.32) aparecen los botones para guardar los cambios de la entrada del calendario, cancelarlos o bien eliminar la entrada del calendario.

Al guardar la solicitud del servicio se creará una nueva factura (apartado 6.3.3.2.4).

Las acciones disponibles para una factura, como puede verse en la Figura 6.26, se pueden ver en la Tabla 6.2.

Tabla 6.2: Acciones disponibles para una factura

	Ver: acción que permite ver todos los datos detallados de la factura.
	Editar: permite modificar los datos de una factura. Cada vez que se editan los datos de una factura esta pasa a estar pendiente de aceptación por el otro usuario implicado en la realización del servicio.
	Aceptar: acepta el servicio pasando la factura al estado “Pendiente de aceptación por el ofertante”.
	Rechazar: rechaza el servicio pasando la factura al estado “Rechazada”.
	Servicio realizado (Pagar): indica que el servicio se ha realizado generándose el pago automáticamente.
	Opinar: permite crear una nueva opinión sobre un servicio realizado.
	Modificar opinión: modifica una opinión ya existente sobre el servicio realizado.

6.3.3.2.5 Mis opiniones






Muestra un listado de todas las opiniones realizadas por el usuario conectado. Por defecto veremos todas las opiniones del usuario. Sin embargo podemos filtrar las opiniones mostradas atendiendo a una serie de parámetros (Figura 6.33):

- **Tipo de servicio:** si el valor se encuentra en blanco buscará las opiniones pertenecientes a cualquier tipo (oferta y demanda). De lo contrario sólo mostrará las pertenecientes al tipo seleccionado.
- **Estado de publicación:** si se deja en blanco devolverá todas las opiniones independientemente de su estado, mientras que de lo contrario se pueden seleccionar o bien las opiniones que se encuentren publicadas, o las que no lo están.

- **Opinión para (Email):** si se introduce un email sólo se mostrarán las opiniones para el usuario a cuyo email le corresponda el introducido. De lo contrario se mostrarán las opiniones realizadas por el usuario conectado para cualquier usuario.
- **Creada antes de:** si se introduce alguna fecha, sólo mostrará las facturas que se han creado antes de esta. De lo contrario las mostrará todas.
- **Creada después de:** si se introduce alguna fecha, sólo mostrará las facturas que se han creado después de la fecha indicada. De lo contrario se mostrarán todas.

Desde esta página, además, podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1). Pulsando la opción para ver el detalle de una opinión se podrán ver sus datos, y realizar las operaciones que se pueden realizar sobre una opinión (Tabla 6.3).

Tabla 6.3: Acciones disponibles para una opción

	Ver: acción que permite ver todos los datos de la opinión.
	Editar: permite modificar los datos de una opinión. Cada vez que se editan los datos de una opinión ésta deja de ser pública, de forma que tenga que ser publicada nuevamente para que sea visible a todos los usuarios.
	Borrar: elimina una opinión, pidiendo confirmación antes de borrarla definitivamente.
	Publicar: publica una opinión para hacerla visible a todos los usuarios. Sólo aparece en opiniones que aún no están publicadas.
	Ocultar: elimina del área pública una opinión de forma que solo sea visible por el autor y los administradores. Sólo aparece en opiniones ya publicadas.

6.3.3.2.6 Opiniones recibidas

Muestra un listado de todas las opiniones recibidas para el usuario conectado. Por defecto veremos todas las opiniones recibidas para el usuario. Sin embargo podemos filtrar las opiniones mostradas atendiendo a una serie de parámetros, como se muestra en la Figura 6.34.

Filtrado de datos

Tipo de servicio:
Oferta

Estado de publicación:
Publicada

Opinión para (Email):
jagarrido@malagacomun.org

Creada antes de:
01/02/2011

Creada después de:
06/02/2011

Mostrar resultados

Figura 6.33: Formulario de filtrado de opiniones realizadas

Filtrado de datos

Tipo de servicio:
Oferta

Estado de publicación:
Publicada

Autor de la opinión (Email):
jagarrido@malagacomun.org

Creada antes de:
01/02/2011

Creada después de:
06/02/2011

Mostrar resultados

Figura 6.34: Formulario de filtrado de opiniones recibidas

- **Tipo de servicio:** si el valor se encuentra en blanco buscará las opiniones pertenecientes a cualquier tipo (oferta y demanda), mientras que de lo contrario sólo mostrará las pertenecientes al tipo seleccionado.
- **Estado de publicación:** si se deja en blanco devolverá todas las opiniones independientemente de su estado. De lo contrario se puede seleccionar o bien las opiniones que se encuentren publicadas o las que no.

- **Autor de la opinión (Email):** si se introduce un email sólo se mostrarán las opiniones a cuyo autor le corresponda dicho email. En caso contrario se mostrarán las opiniones de todos los usuarios.
- **Creada antes de:** si se introduce alguna fecha, sólo mostrará las facturas que se han creado antes de esta. De lo contrario las mostrará todas.
- **Creada después de:** si se introduce alguna fecha, sólo mostrará las facturas que se han creado después de la fecha indicada. De lo contrario se mostrarán todas.

Desde esta ventana, además, podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1). Pulsando la opción para ver el detalle de una opinión se podrán ver sus datos, y realizar las operaciones definidas para una opinión recibida, que son las mostradas en la Tabla 6.3.

6.3.3.3 Datos del Sistema

Esta opción esta disponible en el menú lateral. Muestra información sobre aspectos del sistema en el que se encuentra autenticado el usuario y sólo aparece cuando el usuario ha sido validado por el administrador tras su registro (apartado 6.3.1). Sus opciones son las que se muestran en la Figura 6.35 y se explican en los siguientes subapartados.

6.3.3.3.1 Usuarios

Esta opción del menú de la Figura 6.35 muestra un listado de todos los usuarios que pueden interactuar en el sistema. Desde esta ventana además podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1), ver el detalle de cada usuario y modificarlo si se tiene permisos para ello (apartado 6.3.3.2.2.1).

6.3.3.3.2 Ofertas

Con esta opción del menú de la Figura 6.35 se muestra un listado con todas las ofertas que se pueden solicitar en el sistema, es importante saber que pueden existir otras ofertas que por su estado no puedan solicitarse, y por tanto no aparecerán en este listado (apartado 6.3.3.2.2).

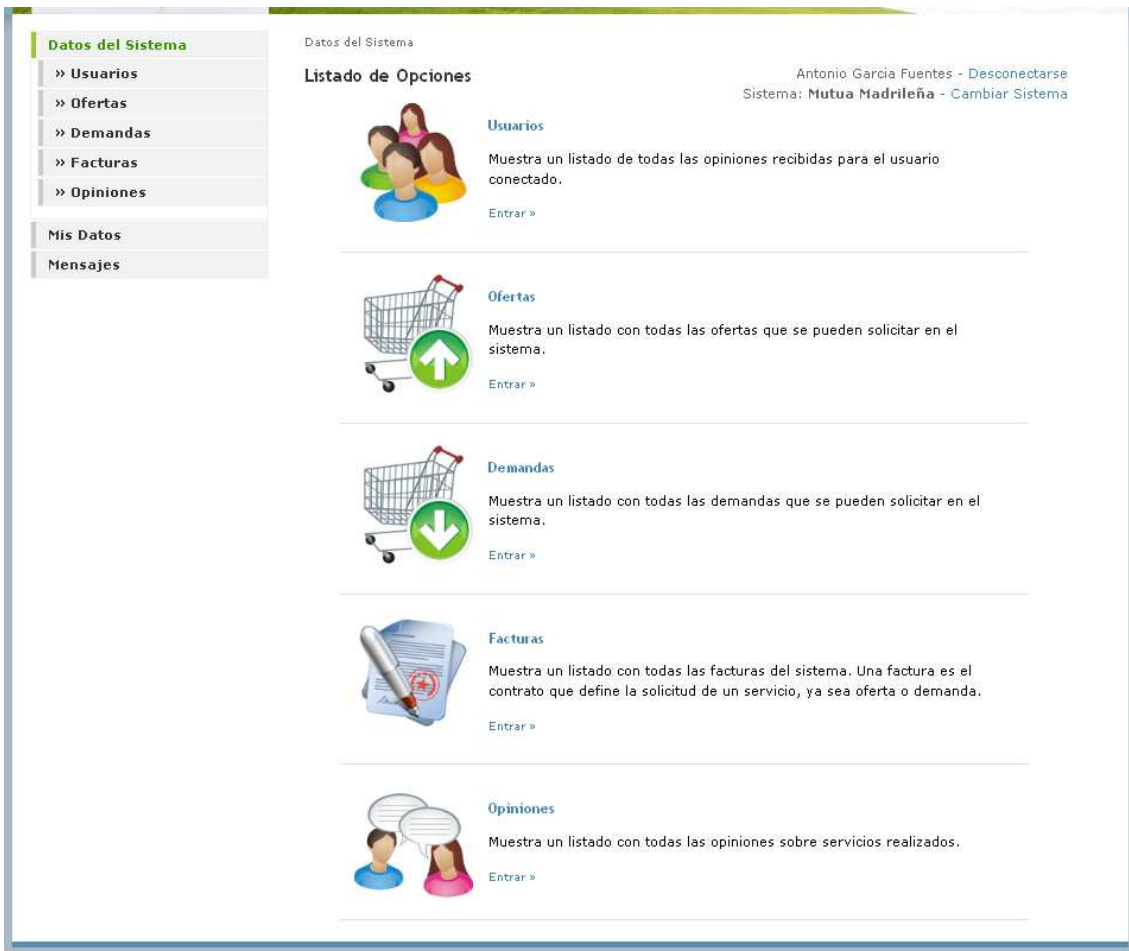


Figura 6.35: Opción de menú Datos del Sistema

Por defecto veremos todas las ofertas que se pueden solicitar del sistema. Sin embargo podemos filtrar las ofertas mostradas atendiendo a una serie de parámetros (Figura 6.36).

Filtrado de datos

Categoria:
Cuidado de personas y animales, compañía

Creada antes de:
01/02/2011

Creada después de:
20/02/2011

Autor de la oferta (Email):
jagarrido@malagacomun.org

Mostrar resultados

Figura 6.36: Formulario de filtrado de ofertas

- **Categoría:** si el valor se encuentra en blanco buscará las ofertas pertenecientes a cualquier categoría. En caso contrario sólo mostrará las pertenecientes a la categoría indicada.
- **Creada antes de:** si se introduce alguna fecha, sólo mostrará las ofertas que se han creado antes de esta. De lo contrario las mostrará todas.
- **Creada después de:** si se introduce alguna fecha, sólo mostrará las ofertas que se han creado después de la fecha indicada, mientras que de lo contrario se mostrarán todas.
- **Autor de la oferta (Email):** si se introduce un email sólo se mostrarán las ofertas a cuyo autor le corresponda dicho email, pero de lo contrario se mostrarán las ofertas de todos los autores.

Desde esta página además, podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1). Pulsando la opción para ver el detalle de una oferta existirán una serie de opciones según quien sea el autor de la oferta. Si el autor de la oferta es el usuario que está viendo sus detalles podrá realizar todas las acciones definidas para éste (apartado 6.3.3.2.2). Por otro lado, si el autor de la oferta es diferente del usuario que está viendo el detalle, se habilitará el botón para solicitar la oferta (Figura 6.37).

Al pulsar sobre el botón para solicitar la oferta se nos abrirá el formulario para introducir los datos necesarios (Figura 6.38). Estos se encuentran divididos en dos pestañas:

Pestaña Contenido: la pestaña contenido define una serie de datos genéricos.

- **Servicio:** indica el nombre del servicio que se va a solicitar. Este campo es un texto fijo, ya que el servicio es desde donde se ha llamado a la factura y ya está definido.
- **Tipo del precio:** el tipo del precio puede ser un valor fijo o un valor por horas. Si el valor es fijo, independientemente del tiempo tardado para realizar el servicio se cargará una cantidad fija a la factura. Mientras que si el valor es por horas, el precio que se introduzca se multiplicará por las horas en las que se ha realizado el servicio.

Por defecto se indica el definido en la oferta, aunque para la realización del servicio puede ser modificado en cualquier momento.

- **Precio:** indica el precio del servicio, que puede ser un valor fijo si el tipo es fijo, o bien el precio de cada hora, si el tipo es por horas. Por defecto aparece el definido en la oferta, aunque puede ser modificado en cualquier momento.

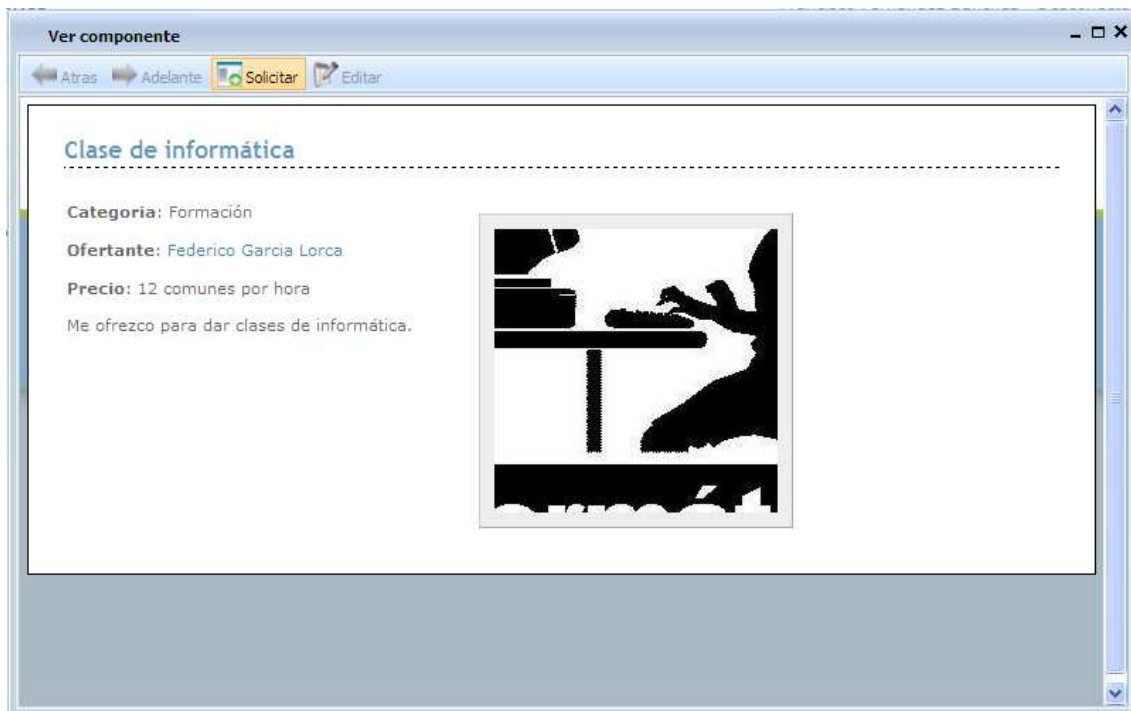


Figura 6.37: Botón para solicitar una oferta en la parte superior, que sólo aparece cuando la oferta es de otro usuario

- **Observaciones:** al solicitar un servicio se pueden añadir tantas observaciones como se deseen. Para añadir una observación hay que pulsar en el botón con una cruz verde que hay a la izquierda del campo *Observaciones* (Figura 6.28). Esto hará aparecer una serie de campos nuevos. Para borrar una observación, se debe pulsar el botón con el aspa roja que aparece junto a los nuevos campos (Figura 6.30).
- **Autor:** indica cual es el autor de la observación y es un texto fijo que no se puede modificar. Una observación puede ser realizada tanto por la persona que va a recibir el servicio, como por la persona que lo va a realizar.

- **Creada:** indica la fecha en la que se ha creado la observación. Es un valor fijo que no se puede modificar.
- **Texto:** indica el texto de la observación, y es el único campo editable de ella, si el usuario actual es el autor de la observación. En caso contrario no será editable.

The screenshot shows a web application window titled "Solicitar servicio". It has two tabs: "Contenido" (active) and "Calendario". The "Contenido" tab contains the following fields and controls:

- Servicio:** Clase de informática
- Tipo del precio:** Radio buttons for "PorHoras" (selected) and "Fijo".
- Precio (comunes):** Text input field containing "12".
- Autor:** Francisco Fernandez Sanchez
- Creada:** 2010-12-18 14:24:59
- Observaciones:** A text area containing "Deseo solicitar el servicio, pero he cambiado el precio por hora porque me parece un poco caro". To the right of this field are a green plus sign and a red minus sign.
- Texto:** An empty text input field below the observations.

Figura 6.38: Pestaña Contenido para solicitar una oferta

Pestaña Calendario: la pestaña calendario muestra un calendario donde se pueden indicar las fechas en las que se va a realizar el servicio ofertado (Figura 6.31). Aunque no es obligatorio definir una fecha al principio, sí es obligatorio definirla antes de realizar el pago si este es por horas y es automático el cálculo, ya que de lo contrario el servicio valdrá cero.

El calendario definido en la pestaña Calendario posee las vistas de Día, Semana y Mes que es la que aparece por defecto. En cualquiera de ellas, para definir una nueva entrada se debe pulsar dos veces sobre una de las celdas (días en la vista por mes, horas en la vista por día, y ambos en la vista por semana). Al hacer doble clic se abre una nueva ventana donde se puede indicar una descripción y el período de tiempo que durará el servicio (Figura 6.32).

Al guardar la solicitud del servicio con el botón que representa a un disco (guardar) o un disco con un aspa (guardar y salir) se creará una nueva factura (apartado 6.3.3.2.4).

6.3.3.3 Demandas

Con esta opción del menú de la Figura 6.35 se muestra un listado con todas las demandas que se pueden solicitar en el sistema (es importante saber que pueden existir otras demandas que por su estado no puedan solicitarse, y por tanto no aparecerán en este listado). Su funcionamiento es idéntico al definido para las Ofertas (apartado 6.3.3.3.2).

6.3.3.4 Facturas

Con esta opción del menú de la Figura 6.35 se muestra un listado con todas las facturas del sistema (no solo las relacionadas con el usuario conectado).

Una factura es el contrato que define la solicitud de un servicio, ya sea oferta o demanda. Por defecto veremos todas las facturas del sistema, sin embargo podemos filtrar las facturas mostradas atendiendo a una serie de parámetros como se muestra en la Figura 6.39.



The image shows a web-based filtering form titled "Filtrado de datos". It contains several input fields: a dropdown menu for "Tipo de servicio" with "Oferta" selected; text input fields for "Autor del servicio (Email)" containing "jagarrido@malagacomun.org" and "Solicitante del servicio (Email)" containing "fgarcia@malagacomun.org"; date pickers for "Creada antes de" (01/02/2011) and "Creada después de" (06/02/2011); and a "Mostrar resultados" button at the bottom.

Figura 6.39: Formulario de filtrado de facturas

- **Tipo de servicio:** si el valor se encuentra en blanco buscará las facturas pertenecientes a cualquier tipo (oferta y demanda). En otro caso sólo mostrará las pertenecientes al tipo seleccionado.
- **Autor del servicio (Email):** si se introduce un email sólo se mostrarán los servicios (peticiones de ofertas o demandas) a cuyo autor (usuario que lo solicita) le corresponda dicho email. En otro caso se mostrarán las facturas de todos los servicios.

- **Solicitante del servicio (Email):** si se introduce un email sólo se mostrarán las facturas solicitadas por el usuario a cuyo email le corresponda el introducido, y de lo contrario se mostrarán las facturas solicitadas por cualquier usuario.
- **Creada antes de:** si se introduce alguna fecha, sólo mostrará las facturas que se han creado antes de esta, mientras que de lo contrario las mostrará todas.
- **Creada después de:** si se introduce alguna fecha, sólo mostrará las facturas que se han creado después de la fecha indicada. De lo contrario se mostrarán todas.

Desde esta página, además, podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1). Pulsando la opción para ver el detalle de una factura se podrán ver sus datos (Figura 6.23), y si se tienen los permisos suficientes, realizar las acciones que tiene definidas (apartado 6.3.3.2.4), como son los mostrados en la Tabla 6.2.

6.3.3.3.5 Opiniones

Con esta opción del menú de la Figura 6.35 se muestra un listado con todas las opiniones sobre servicios realizados. Por defecto veremos todas las opiniones del sistema, sin embargo podemos filtrar las opiniones mostradas atendiendo a una serie de parámetros como se observa en la Figura 6.40.

- **Tipo de servicio:** si el valor se encuentra en blanco buscará las opiniones pertenecientes a cualquier tipo de servicio (oferta y demanda). De lo contrario sólo mostrará las pertenecientes al tipo seleccionado.
- **Creada antes de:** si se introduce alguna fecha, sólo mostrará las opiniones que se han creado antes de esta. De lo contrario las mostrará todas.
- **Creada después de:** si se introduce alguna fecha, sólo mostrará las opiniones que se han creado después de la fecha indicada. De lo contrario se mostrarán todas.

- **Autor de la opinión (Email):** si se introduce un email sólo se mostrarán las opiniones cuyo autor sea el propietario de dicho email. De lo contrario se mostrarán las opiniones de todos los usuarios.
- **Opinión para (Email):** si se introduce un email sólo se mostrarán las opiniones para el usuario a cuyo email le corresponda el introducido, de lo contrario se mostrarán las opiniones realizadas para cualquier usuario.

Figura 6.40: Formulario de filtrado de opiniones

Por otra parte, desde esta página podremos realizar todas las operaciones genéricas de cada listado (véase apartado 6.3.3.1). Pulsando la opción para ver el detalle de una opinión se podrán ver sus datos (Figura 6.41), y si se tienen los permisos suficientes, realizar las acciones definidas para esta (apartado 6.3.3.2.4), como se indica en la Tabla 6.3.

Figura 6.41: Detalle de una opinión



Figura 6.42: Opción de menú Mensajes

6.3.3.4 Mensajes

Con esta opción del menú de la Figura 6.42 se muestra información sobre las notificaciones del usuario que se encuentra autenticado en el sistema. Las notificaciones se pueden recibir por email, mediante alertas, o de las dos formas, que es como se encuentra definido por defecto (véase apartado 6.3.3.2.1). Sus opciones se explican en los siguientes apartados.

6.3.3.4.1 Bandeja de entrada

Con esta opción del menú de la Figura 6.42 se muestra un listado de los mensajes recibidos por el usuario conectado. Por defecto veremos todos los mensajes recibidos para el usuario, pero podemos filtrar los mensajes mostrados atendiendo a dos parámetros (Figura 6.45).

- **Enviado antes de:** si se introduce alguna fecha, sólo mostrará los mensajes que se han enviado antes de esta fecha. De lo contrario los mostrará todos.

- **Enviado después de:** si se introduce alguna fecha, sólo mostrará los mensajes que se han creado después de la fecha indicada. De lo contrario se mostrarán todos.



El formulario, titulado "Filtrado de datos", contiene un icono de lupa a la izquierda. A continuación, se encuentran dos campos de texto con iconos de calendario a la derecha. El primer campo, etiquetado "Enviado antes de:", contiene la fecha "01/02/2011". El segundo campo, etiquetado "Enviado después de:", contiene la fecha "06/02/2011". Debajo de estos campos hay un botón rectangular con el texto "Mostrar resultados".

Figura 6.45: Formulario de filtrado de mensajes en la bandeja de entrada

Desde esta página además podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1). Pulsando la opción para ver el detalle de un mensaje se podrán ver sus datos (Figura 6.46), y realizar las operaciones definidas para los mensajes, que son las que se ven en la Tabla 6.4.

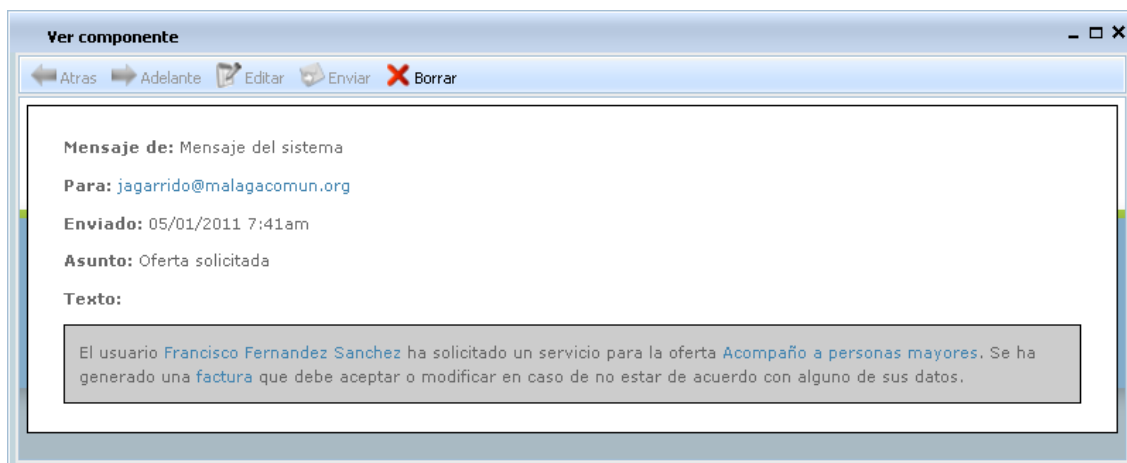


Figura 6.46: Detalle de un mensaje

6.3.3.4.2 Borrador

Muestra un listado de los mensajes escritos y aún no enviados por el usuario conectado. Por defecto veremos todos los mensajes recibidos para el usuario. Sin embargo podemos filtrar los mensajes mostrados atendiendo a dos parámetros (Figura 6.45).

- **Enviado antes de:** si se introduce alguna fecha, sólo mostrará los mensajes que se han enviado antes de esta. De lo contrario los mostrará todos.
- **Enviado después de:** si se introduce alguna fecha, sólo mostrará los mensajes que se han creado después de la fecha indicada, de lo contrario se mostrarán todos.

Desde esta ventana también podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1). Pulsando la opción para ver el detalle de un mensaje se podrán ver sus datos (Figura 6.46), y realizar las operaciones definidas para los mensajes, que son las que se ven en la Tabla 6.4.

Para crear un nuevo mensaje debemos pulsar el botón con el texto *Añadir mensaje*. Al pulsar sobre el botón para crear el mensaje se nos abrirá el formulario para introducir los datos necesarios (Figura 6.47), el cual tiene dos pestañas en la parte superior:

Figura 6.47: Formulario para crear un nuevo mensaje

Pestaña Contenido: la pestaña contenido define los datos del mensaje, ambos obligatorios.

- **Asunto:** indica el título del mensaje.
- **Texto:** indica el cuerpo del mensaje que se desea enviar.

Pestaña Receptor: indica el receptor del mensaje (Figura 6.48). Sólo puede ser uno y es un campo obligatorio.

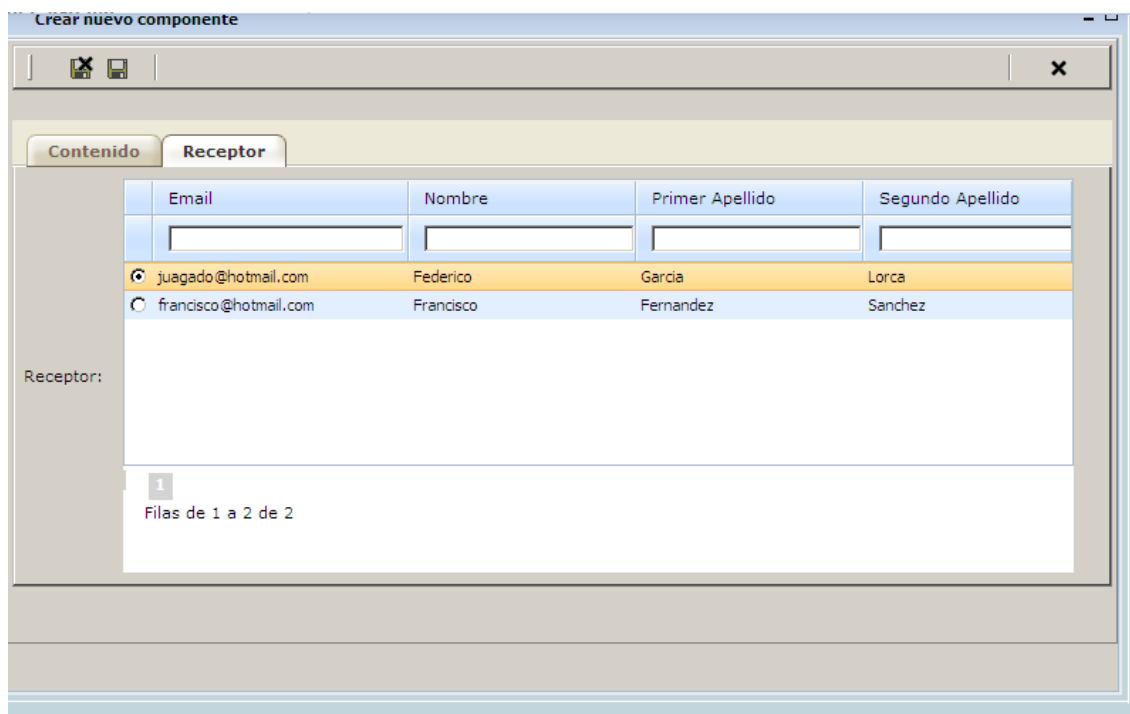






Figura 6.48: Pestaña Receptor del formulario para crear un nuevo mensaje

El formulario de edición y creación de un nuevo mensaje posee en la parte superior tres botones para guardar, guardar y salir, y salir, señalados con los iconos de un disco, un disco con un aspa, y un aspa respectivamente. Al salir del formulario aparecen los detalles del nuevo mensaje creado junto con las acciones que se pueden realizar en la parte superior de la ventana (Figura 6.46).

Tabla 6.4: Acciones disponibles para un mensaje

	Ver: acción que permite ver todos los datos detallados de un mensaje.
	Editar: permite editar los datos de un mensaje. Esta opción sólo estará disponible mientras el mensaje aún no esté enviado.
	Borrar: elimina un mensaje (pasa a la lista de mensajes borrados).
	Enviar mensaje: envía el mensaje al destinatario seleccionado previamente. Esta opción sólo estará disponible mientras el mensaje aún no esté enviado.

Es importante saber que hasta que no se pulse el botón *Enviar* no se enviará el mensaje. Cuando se envía un mensaje éste se envía según las opciones de notificación del usuario receptor. Es decir, puede ser bien mediante una alerta del sistema, un correo electrónico, o ambos.

Las acciones disponibles para un mensaje se pueden ver en la Tabla 6.4.

6.3.3.4.3 Mensajes enviados

Muestra un listado de los mensajes enviados por el usuario conectado. Por defecto veremos todos los mensajes recibidos para el usuario, pero podemos filtrar los mensajes mostrados atendiendo a dos parámetros (Figura 6.45):

- **Enviado antes de:** si se introduce alguna fecha, sólo mostrará los mensajes que se han enviado antes de esta. De lo contrario los mostrará todos.
- **Enviado después de:** si se introduce alguna fecha, sólo mostrará los mensajes que se han creado después de la fecha indicada. De lo contrario se mostrarán todos.

Desde esta ventana además podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1). Pulsando la opción para ver el detalle de un mensaje se podrán ver sus datos (Figura 6.46), y realizar las operaciones para los que el usuario tenga permisos, que son las que se ven en la Tabla 6.4.

6.3.3.4.4 Mensajes borrados

Muestra un listado de los mensajes borrados por el usuario conectado. Por defecto veremos todos los mensajes borrados del usuario, y también podemos filtrar los mensajes mostrados atendiendo a dos parámetros (Figura 6.45):

- **Enviado antes de:** si se introduce alguna fecha, sólo mostrará los mensajes que se han enviado antes de esta. De lo contrario los mostrará todos.
- **Enviado después de:** si se introduce alguna fecha, sólo mostrará los mensajes que se han creado después de la fecha indicada. De lo contrario se mostrarán todos.

Desde esta ventana además podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1). Pulsando la opción para ver el detalle de un mensaje se podrán ver sus datos (Figura 6.46), y realizar las operaciones para los que el usuario tenga permisos (Tabla 6.4).

6.3.3.5 Manuales

Con esta opción del menú de la Figura 6.49 se muestran los distintos manuales de los que se compone la aplicación. Cada usuario podrá ver los manuales para los que tenga permisos.



Figura 6.49: Opción de menú para mostrar los manuales de un usuario registrado

6.3.3.4.1 Manual de usuario

Muestra un documento web donde se explica la funcionalidad para los usuarios que se registran en la aplicación y pueden participar en el sistema de moneda complementaria.

6.4 Administración de la moneda complementaria

En este apartado veremos la administración de la moneda complementaria. Esta sólo se puede realizar por los usuarios pertenecientes al grupo de administradores, y se lleva a cabo a través de una serie de opciones que sólo estarán disponibles para este grupo.

6.4.1 Datos del sistema

Dentro de este menú (Figura 6.59) también accesible por cualquier usuario registrado existen una serie de opciones que sólo estarán disponibles para los usuarios pertenecientes al grupo de

administradores. Muchas de las opciones también disponibles para los usuarios activos contienen funcionalidades adicionales.

6.4.1.1 Tipos de monedas

Muestra un listado con todos los tipos de monedas definidos en la aplicación. Desde esta ventana además podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1). Pulsando la opción para ver el detalle podremos realizar todas las acciones definidas para el autor, como son las definidas en la Tabla 6.5.

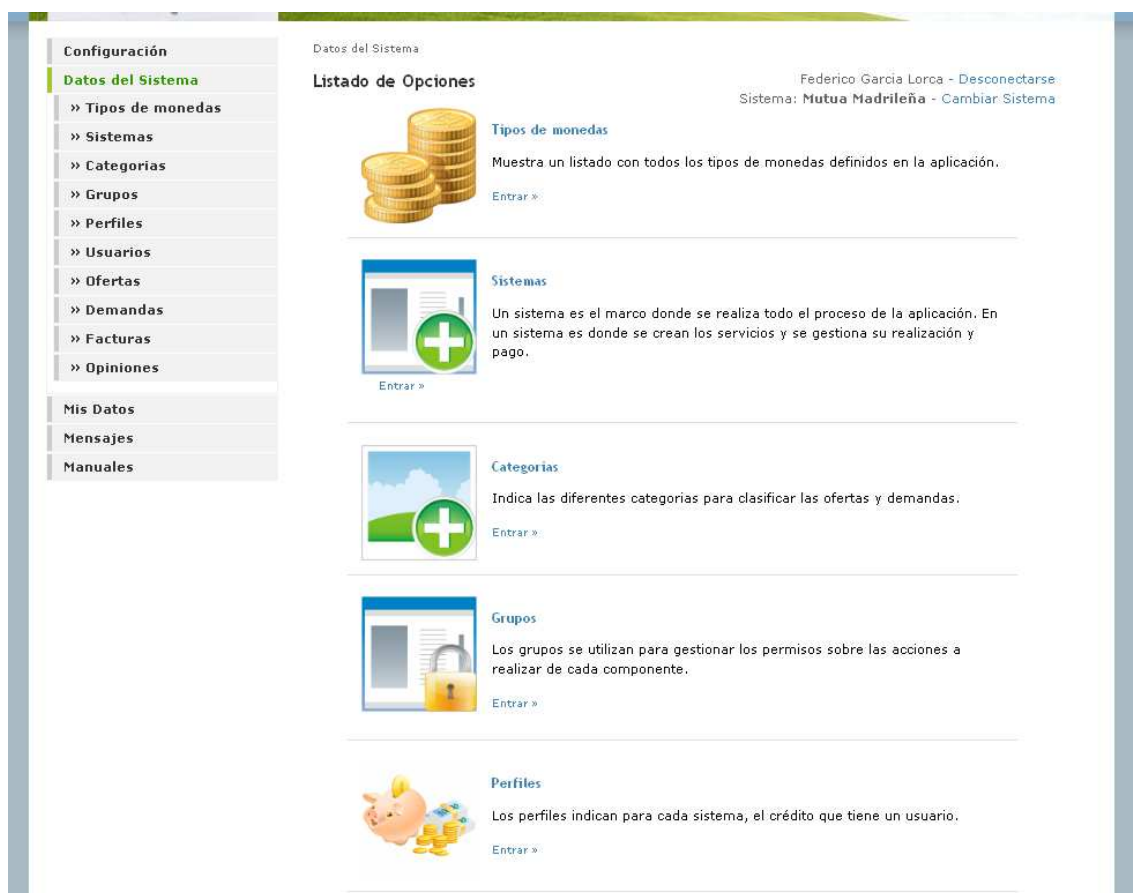


Figura 6.50: Opción de menú para administrar el sistema

Para crear un nuevo tipo de moneda debemos pulsar el botón con el texto *Añadir tipo de moneda*. Al pulsar sobre el botón para crear el tipo de moneda se nos abrirá el formulario para introducir los datos necesarios (Figura 6.51), el cual tiene dos pestañas:

Figura 6.51: Formulario para crear un nuevo tipo de moneda

Pestaña Contenido: la pestaña contenido define los datos de la opción de menú.

- **Nombre:** texto obligatorio con el que se definirá la denominación de la moneda creada.
- **Descripción:** texto obligatorio que detalla la finalidad del tipo de moneda.
- **Símbolo en singular:** símbolo utilizado para representar el tipo de moneda en caso de existir un único elemento de esta moneda.
- **Símbolo en plural:** símbolo utilizado para representar el tipo de moneda en caso de existir varios elementos de esta moneda.

Pestaña Permisos: esta pestaña se define en el apartado 6.4.3.

El formulario de edición y creación de un nuevo tipo de moneda posee en la parte superior tres botones para, en este orden: guardar, guardar y salir, y salir sin guardar. Al salir del formulario aparecen los detalles del nuevo tipo de moneda creada junto con las acciones que se pueden realizar en la parte superior de la ventana (Figura 6.52), y son las definidas en la Tabla 6.5.

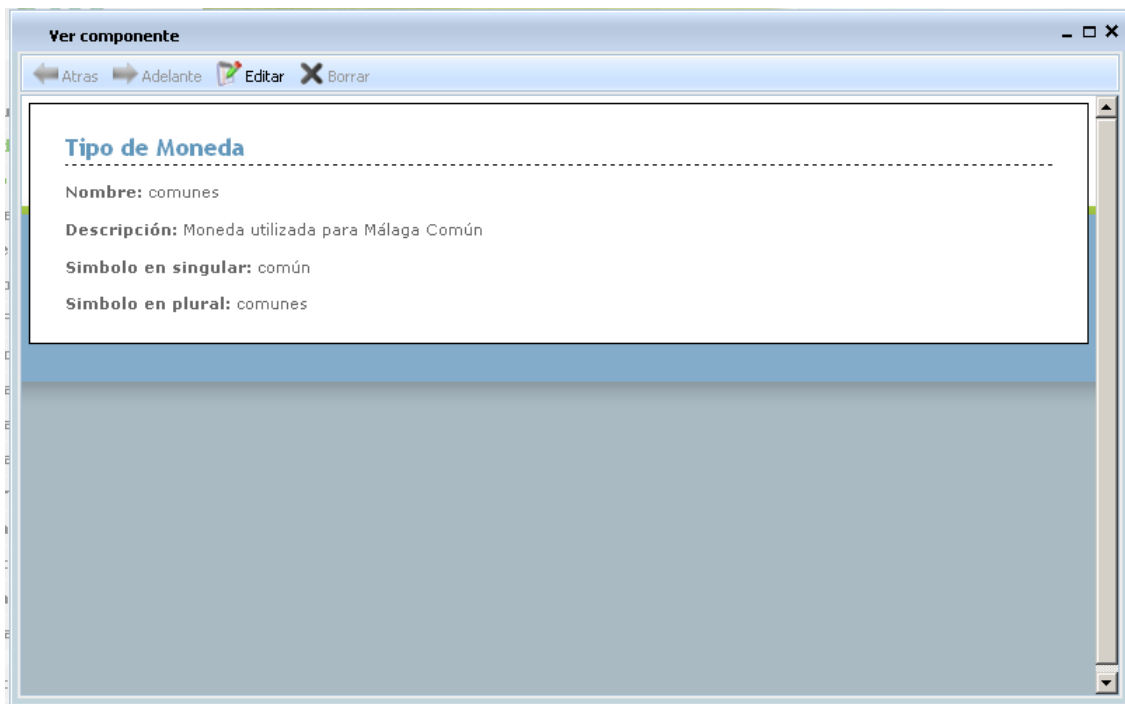





Figura 6.52: Detalle de un tipo de moneda

Tabla 6.5: Acciones disponibles para un tipo de moneda

	Ver: acción que permite ver todos los datos detallados de un tipo de moneda.
	Editar: permite editar los datos de un tipo de moneda.
	Borrar: elimina un tipo de moneda. Solo se pueden eliminar los tipos de monedas cuando no están asociados a algún sistema. Se pide confirmación.

6.4.1.2 Sistemas

Un sistema es el marco donde se realiza todo el proceso de la aplicación. En un sistema es donde se crean los servicios y se gestiona su realización y pago. Por defecto veremos todos los sistemas existentes. Sin embargo podemos filtrar los sistemas mostrados atendiendo a dos parámetros (Figura 6.53).

- **Tipo de moneda:** si el valor se encuentra en blanco buscará los sistemas con cualquier tipo de moneda. De lo contrario sólo mostrará los que tengan asociado el tipo de moneda indicado.
- **Borrado:** si está en blanco mostrará todos los sistemas. De lo contrario se puede seleccionar si se desea ver los sistemas que están borrados o los que no. Un sistema

borrado se puede volver a restaurar, manteniendo los datos que tenía en el momento en que se borró.

El formulario de filtrado de datos tiene un icono de lupa a la izquierda del título "Filtrado de datos". Incluye un menú desplegable "Tipo de moneda" con "comunes" seleccionado, un menú desplegable "Borrado" con "No Borrado" seleccionado, y un botón "Mostrar resultados" en un recuadro amarillo.

Figura 6.53: Formulario de filtrado de sistemas

Desde esta ventana también podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1). Pulsando la opción para ver el detalle podremos realizar todas las acciones definidas para el autor, como las mostradas en la Tabla 6.6.

Para crear un nuevo sistema debemos pulsar el botón con el texto *Añadir sistema*. Al pulsar sobre el botón para crear el sistema se nos abrirá el formulario de la Figura 6.54 para introducir los datos necesarios, el cual tiene estas pestañas:

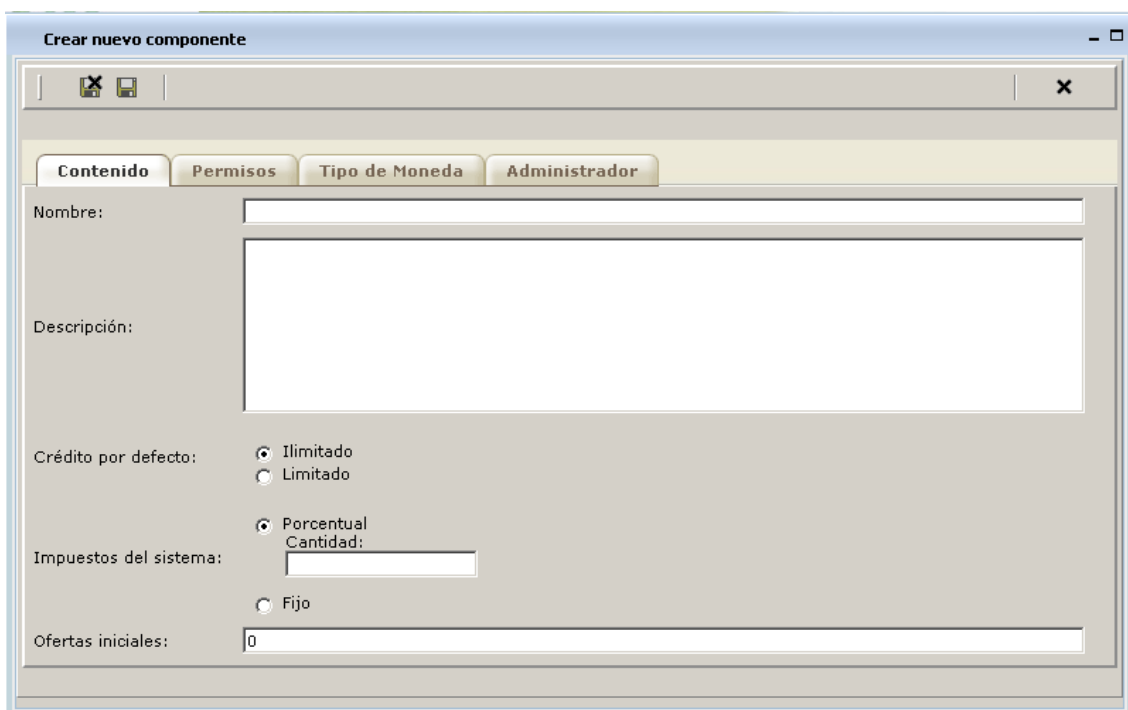
El formulario "Crear nuevo componente" tiene pestañas para "Contenido", "Permisos", "Tipo de Moneda" y "Administrador". El panel "Contenido" muestra campos para "Nombre:", "Descripción:", "Crédito por defecto:" (con opciones "Ilimitado" y "Limitado"), "Impuestos del sistema:" (con opciones "Porcentual" y "Fijo", y un campo "Cantidad:"), y "Ofertas iniciales:" con el valor "0".

Figura 6.54: Formulario de edición de sistemas

Pestaña Contenido: define los datos del sistema.

- **Nombre:** texto obligatorio con el que se denominará el sistema creado.
- **Descripción:** texto obligatorio que detalla la finalidad del sistema.
- **Crédito por defecto:** indica si el crédito del sistema se encuentra ilimitado o bien limitado por algún valor. Si el crédito es ilimitado los usuarios que no tengan un perfil asociado (apartado 6.4.4) podrán acumular tanto crédito negativo como deseen. Por el contrario, si se limita el crédito a un valor, los usuarios no podrán sobrepasar nunca ese crédito negativo. Se aconseja poner un crédito limitador.
- **Impuestos del sistema:** El sistema puede tener un impuesto, cuyo valor puede ser fijo, en este caso se le restará a cada operación una cantidad fija. O bien un tanto por ciento del valor del servicio.
- **Ofertas iniciales:** Indica las ofertas iniciales obligatorias para poder registrarse en el sistema.




Pestaña Permisos: esta pestaña se define en el apartado 6.4.3.

Pestaña Tipo de moneda: permite seleccionar el tipo de moneda del sistema.

Pestaña Administrador: permite seleccionar al usuario administrador del sistema. Este usuario es el que recibirá los impuestos y las notificaciones.

Las acciones disponibles para un sistema se definen en la Tabla 6.6.

Tabla 6.6: Acciones disponibles para un sistema

	Ver: acción que permite ver todos los datos detallados de un sistema.
	Editar: permite modificar los datos de un sistema.
	Borrar: elimina un sistema. Solo se pueden eliminar los sistemas cuando no están asociados a algún componente (usuario, perfil, etc.).

6.4.1.3 Categorías

Esta opción de la Figura 6.50, muestra un listado con todas las categorías o temáticas en las que se pueden clasificar las ofertas y demandas.

Desde esta ventana además podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1). Pulsando la opción para ver el detalle podremos realizar todas las acciones definidas para las categorías.

Para crear una nueva categoría debemos pulsar el botón con el texto *Añadir categoría*. Al pulsar sobre el botón para crear la categoría se nos abrirá en una nueva ventana el formulario para introducir los datos necesarios (Figura 6.55), el cual tiene dos pestañas:

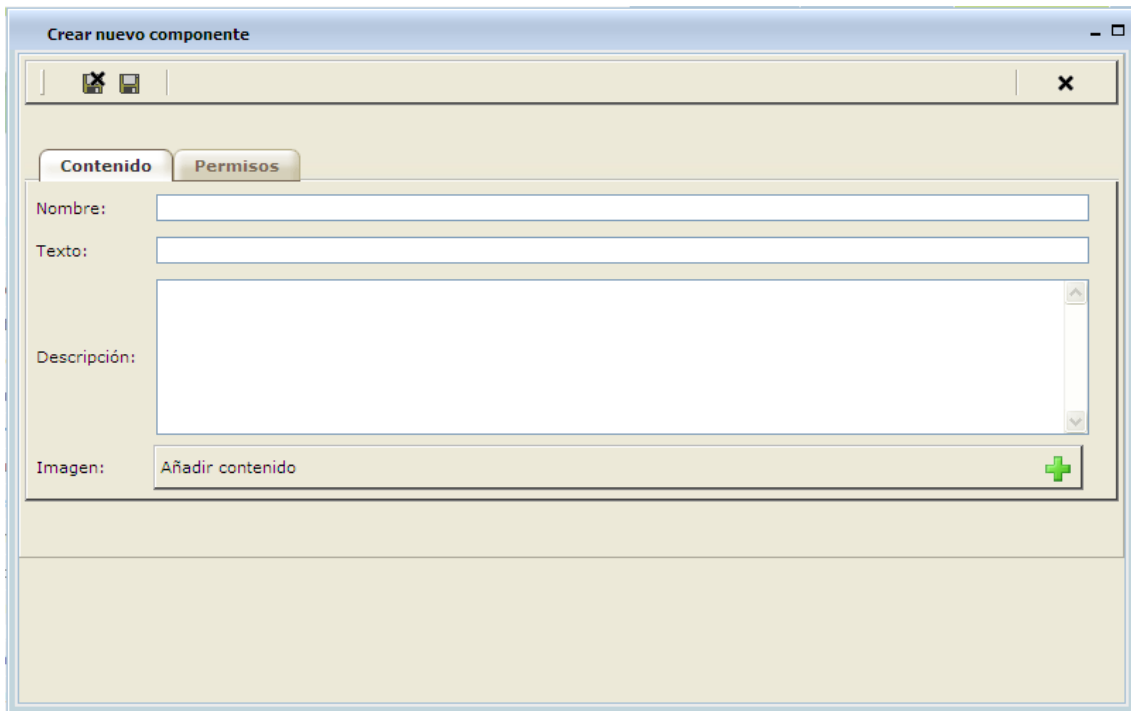
The image shows a software window titled "Crear nuevo componente". At the top, there are standard window controls (minimize, maximize, close) and a toolbar with icons for deleting and saving. Below the toolbar, there are two tabs: "Contenido" (selected) and "Permisos". The "Contenido" tab contains four input fields: "Nombre:" (a single-line text box), "Texto:" (a single-line text box), "Descripción:" (a multi-line text area with scrollbars), and "Imagen:" (a button labeled "Añadir contenido" with a green plus icon to its right). The window has a light beige background and a blue title bar.

Figura 6.55: Formulario para crear una nueva categoría

Pestaña Contenido: define los datos de la categoría.




- **Nombre:** indica el nombre de la categoría, se utiliza de forma interna para referenciar a la categoría.
- **Texto:** representa el texto que se va a escribir como título de la categoría.

- **Descripción:** en este campo es donde se describe la categoría.
- **Imagen:** se pueden añadir a la categoría tantas imágenes como se deseen. Para añadir una nueva imagen se debe pulsar sobre el botón con una cruz verde situado a la derecha del campo (Figura 6.54). Al pulsar sobre este botón aparecerán los nuevos campos para añadir la imagen (Figura 5.20). Los campos para una imagen son la descripción, que es un texto que aparecerá al pasar el ratón sobre ella. La alternativa, que es un texto que aparece si la imagen no puede ser cargada. Y la imagen propiamente dicha, que se puede cargar desde el ordenador local del usuario o bien utilizar una ya existente (subida con anterioridad). Para cargar la imagen debemos seleccionar la opción deseada y pulsar el botón que lo indica (*Agregue imagen*). Esta imagen será lo que se muestra a la hora de listar las categorías en una oferta o una demanda (Figura 6.3).

Pestaña Permisos: esta pestaña se define en el apartado 6.4.3.

Las acciones disponibles para una categoría son las definidas en la Tabla 6.7.

Tabla 6.7: Acciones disponibles para un sistema

	Ver: acción que permite ver los datos detallados de la categoría.
	Editar: permite modificar los datos de una categoría.
	Borrar: elimina una categoría previa confirmación.

6.4.2.4 Grupos

Los grupos se utilizan para gestionar los permisos sobre las acciones a realizar de cada usuario. Desde esta ventana además podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1). Pulsando la opción para ver el detalle podremos realizar todas las acciones definidas para el autor, que son las definidas en la Tabla 6.7.

Para crear un nuevo grupo debemos pulsar el botón con el texto *Añadir grupo*. Al pulsar sobre el botón para crear el grupo se nos abrirá el formulario (Figura 6.56) para introducir los datos necesarios, el cual tiene dos pestañas:

Pestaña Contenido: define los datos del grupo.

- **Nombre:** texto obligatorio con el que se denominará el grupo creado.
- **Descripción:** texto obligatorio que detalla la finalidad del grupo.

Pestaña Permisos: esta pestaña se define en el apartado 6.4.3.

Las acciones disponibles para un grupo se definen en la Tabla 6.8.

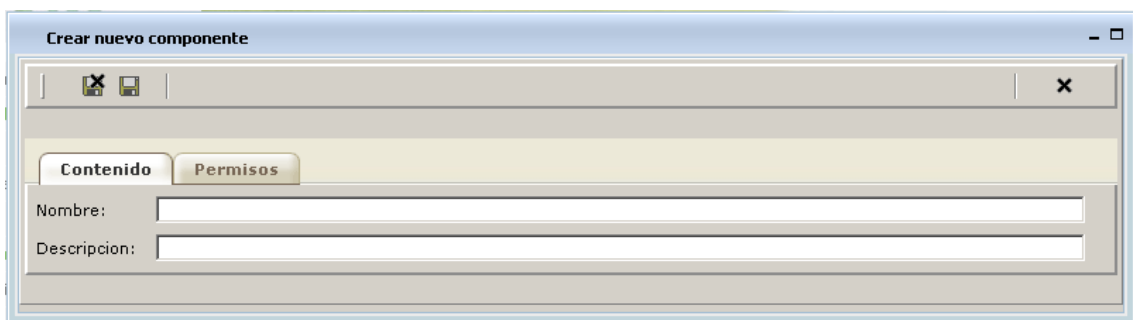





Figura 6.56: Formulario para crear un nuevo grupo

Tabla 6.8: Acciones disponibles para un grupo

	Ver: acción que permite ver todos los datos detallados de un grupo.
	Editar: permite modificar los datos de un grupo.
	Borrar: elimina un grupo. Solo se pueden eliminar los grupos cuando no están asociados a ningún usuario. Antes de borrarlo pide confirmación.

Por defecto existen cuatro grupos de usuarios en la aplicación:

- **Administrador:** a este grupo pertenecerán todos los usuarios que sean administradores de la aplicación.
- **UsuarioActivo:** a este grupo pertenecen todos los usuarios que han sido activados por un administrador. Al pulsar el botón *activar* para un usuario, se le añade el grupo automáticamente.

- **Difusores:** los usuarios que pertenecen a este grupo pueden ver la opción para enviar mensajes a todos los usuarios del sistema (mensajes de difusión).
- **Desarrollador:** los usuarios que pertenecen a este grupo pueden ver las acciones para configurar la aplicación (apartado 6.5).

6.4.2.5 Perfiles

- Los perfiles indican para cada sistema, el crédito que tiene un usuario. Por defecto un usuario tiene asociado el crédito definido en el sistema, que puede ser un valor ilimitado o bien definido por el administrador del sistema. Sin embargo, a veces es deseable que un usuario tenga un crédito diferente al del resto de usuarios. A un perfil se le indica el crédito, y el perfil se le puede asociar a un usuario determinado. Por defecto veremos todos los perfiles, pero podemos filtrar los perfiles mostrados atendiendo al parámetro **Sistema**: si el valor se encuentra en blanco buscará los perfiles de cualquier sistema, pero de lo contrario sólo se mostrarán los perfiles del sistema seleccionado.

Desde esta ventana también podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1). Pulsando la opción para ver el detalle podremos realizar todas las acciones definidas para el autor, como son las mostradas en la Tabla 6.7.

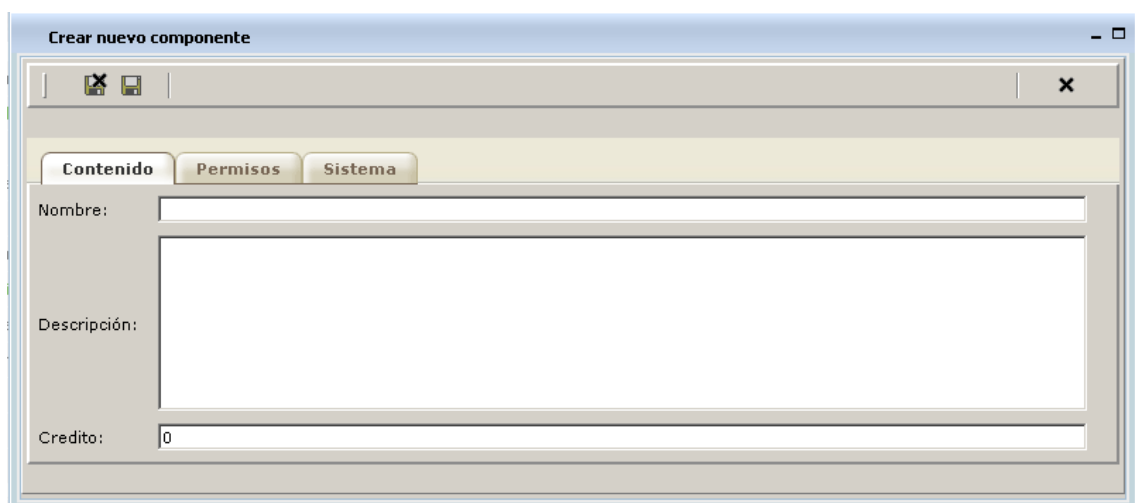


Figura 6.57: Formulario para crear un nuevo perfil

Para crear un nuevo sistema debemos pulsar el botón con el texto *Añadir perfil*. Al pulsar sobre el botón para crear el perfil se nos abrirá el formulario (Figura 6.57) para introducir los datos necesarios, el cual tiene tres pestañas:

Pestaña Contenido: define los datos del perfil.

- **Nombre:** texto obligatorio con el que se denominará el perfil creado.
- **Descripción:** texto obligatorio que detalla la finalidad del perfil.
- **Crédito:** indica el crédito del perfil. Cualquier usuario con este perfil nunca podrá sobrepasar esta cantidad en saldo negativo.




Pestaña Permisos: esta pestaña se define en el apartado 6.4.3.

Pestaña Sistema: permite seleccionar el sistema sobre el que se define el perfil.

- **Sistema:** indica el sistema sobre el que se define el perfil.

Las acciones disponibles para un perfil son las definidas en la Tabla 6.9.

Tabla 6.9: Acciones disponibles para un perfil

	Ver: acción que permite ver todos los datos detallados de un perfil.
	Editar: permite modificar los datos de un perfil.
	Borrar: elimina un perfil. Solo se pueden eliminar los perfiles que no estén asociados a ningún usuario. Antes de borrar se pide confirmación.

Si un usuario no tiene asociado un perfil para un sistema determinado, por defecto se le aplicará el crédito de dicho sistema.

6.4.2.6 Usuarios

Esta opción es similar a la opción de menú con el mismo nombre vista para los usuarios activos del sistema (apartado 6.3.3.3.1), con la diferencia de que en este caso también

podemos ver los usuarios que aún no han sido activados, y los que han sido desactivados mediante un formulario de filtrado (Figura 6.58).

El formulario de filtrado de usuarios muestra un icono de lupa a la izquierda del título "Filtrado de datos" con un ícono de cerrar. Debajo, se encuentra el campo "Estado del usuario:" con un menú desplegable que muestra "Pendiente de activar". En la parte inferior del formulario hay un botón que dice "Mostrar resultados".

Figura 6.58: Formulario de filtrado de usuarios

6.4.2.7 Ofertas

El funcionamiento de esta opción de menú es similar a la de la opción *Ofertas* para un usuario activo (apartado 6.3.3.3.2), con la diferencia de que ahora también podemos filtrar según el estado de publicación de una oferta, de modo que también podemos ver las ofertas que no se encuentran publicadas para todos los usuarios.

Desde esta opción también podemos crear nuevas ofertas o editar las existentes, de forma similar a como se realiza para un usuario que no sea administrador, pero con la funcionalidad añadida de que aparece una nueva pestaña, la pestaña Ofertante (Figura 6.59), donde se debe indicar el usuario que realiza la oferta, de modo que un administrador pueda editar y crear ofertas para cualquier otro usuario.

No debe olvidarse de GUARDAR la oferta creada o modificada con el botón cuya imagen representa un disco o un disco con un aspa, o bien cancelar pulsando sobre el botón cuya imagen representa un aspa (Figura 6.59).

6.4.2.8 Demandas

El funcionamiento de esta opción de menú es similar a la de la opción *Demandas* para un usuario activo (apartado 6.3.3.3.3), con la diferencia de que ahora también podemos filtrar según el estado de publicación de una demanda, de modo que también podemos ver las demandas que no se encuentran publicadas para todos los usuarios.

Desde esta opción también podemos crear nuevas demandas o editar las existentes, de forma similar a como se realiza para un usuario que no sea administrador, pero con la funcionalidad añadida de que aparece una nueva pestaña, Demandante (Figura 6.60) donde se debe indicar el usuario que realiza la demanda, de modo que un administrador pueda editar y crear demandas para cualquier otro usuario.

No debe olvidarse de GUARDAR la oferta creada o modificada con el botón cuya imagen representa un disco o un disco con un aspa, o bien cancelar pulsando sobre el botón cuya imagen representa un aspa (Figura 6.60).

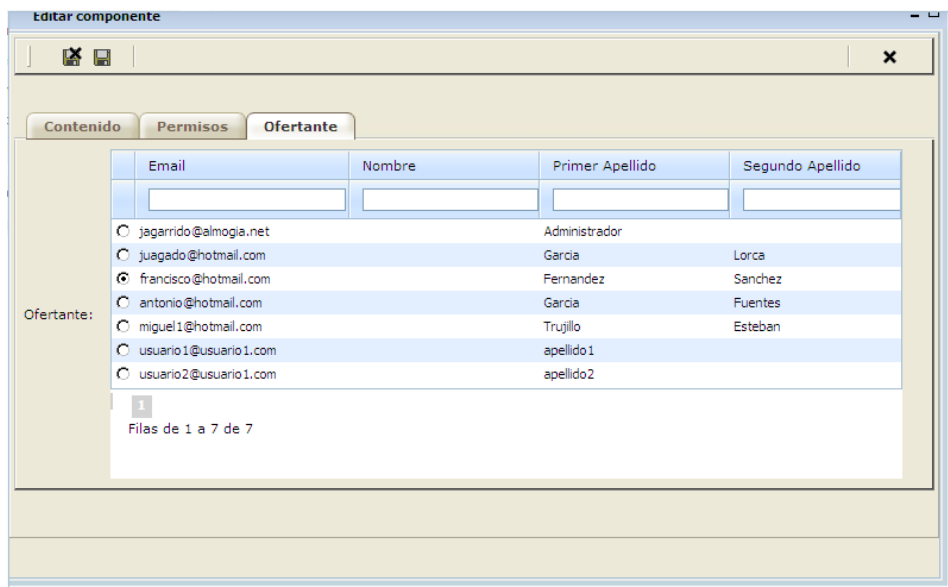


Figura 6.59: Pestaña Ofertante de una oferta, sólo accesible para el administrador

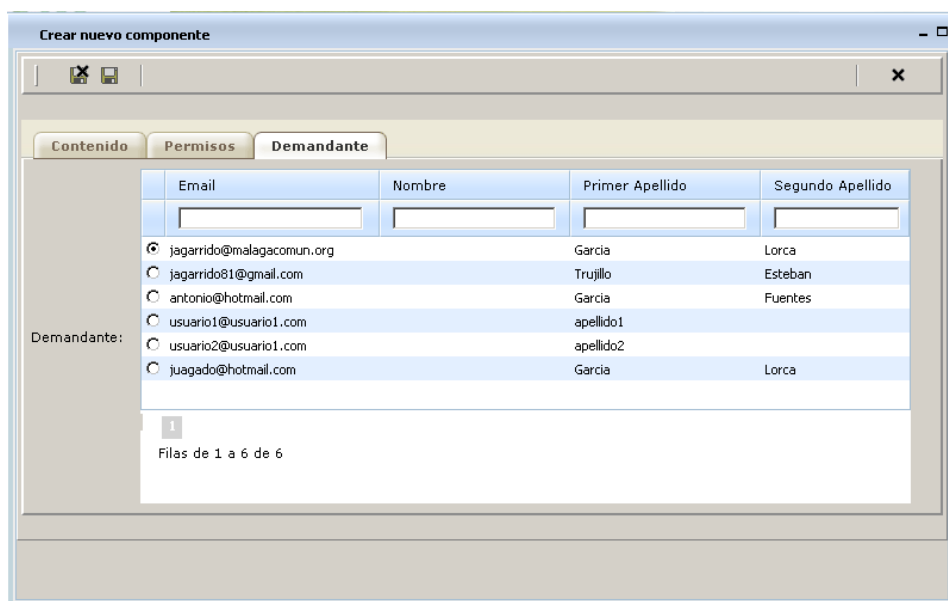


Figura 6.60: Pestaña Demandante de una demanda, sólo accesible para el administrador

6.4.2.9 Facturas

El funcionamiento de esta opción de menú es similar a la de la opción *Facturas* para un usuario activo (apartado 6.3.3.3.4).

Un usuario perteneciente al grupo de *Administrador* puede editar cualquier factura, para modificar sus datos ((apartado 6.3.3.2.4).

6.4.2.10 Opiniones

Muestra un listado con todas las opiniones sobre servicios realizados. Esta opción es similar a la de los usuarios activos (apartado 6.3.3.2.5) con la diferencia de que un administrador puede modificar una opinión existente de otro usuario, y crear nuevas asignándoles también el usuario que se considerará como el autor. En este caso se puede indicar la factura donde irá asociada y el usuario al que se le asigna.

6.4.3 Gestión de permisos

Como se ha mostrado anteriormente, todos los componentes del sistema tienen la opción de gestionar sus permisos en una pestaña como la de la Figura 6.61. Los permisos que puede tener un componente son:

- **Leer:** si un usuario tiene un permiso para leer un componente podrá ver sus datos. De lo contrario, este siempre se mostrará oculto al usuario, ya sea en los listados como en la vista detallada.
- **Escribir:** un usuario que tenga permisos para escribir en un componente podrá tanto crear nuevos componentes como editar uno existente.
- **Publicar:** este permiso da la posibilidad de publicar el recurso. El usuario que tenga este permiso podrá tanto publicarlo como eliminarlo de la parte pública
- **Cambiar permisos:** con este permiso se podrán cambiar los permisos del componente. Cabe destacar que la opción de gestión de los permisos sólo

aparecerá en la edición de un componente siempre que el usuario tenga este permiso.

Estos permisos se pueden aplicar a:

- **Autor:** los permisos se aplican al autor del componente.
- **Grupos:** los permisos se pueden aplicar directamente a un grupo concreto.
- **Otros:** cuando no se aplican ni al autor ni a un grupo.

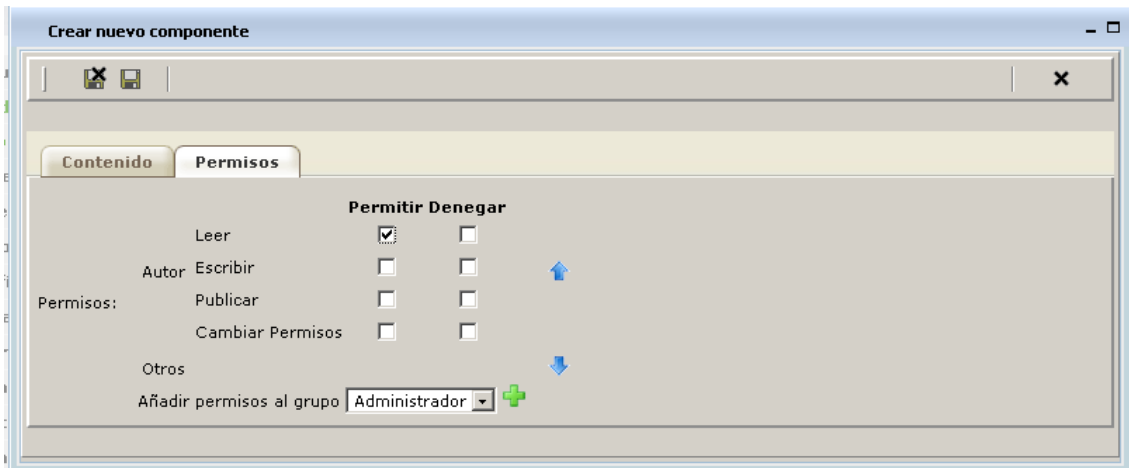


Figura 6.61: Gestión de permisos

La búsqueda de los permisos se realiza de la siguiente manera: en primer lugar se comprueba si el usuario conectado es el autor del componente, de ser así se comprueba si el permiso se encuentra permitido o denegado, devolviendo el resultado. Si el usuario no es el autor, se comprueba si existe para algún grupo del usuario, permisos definidos (tanto permitidos como denegados) en cuyo caso se devuelve el resultado. Por último, si no se encuentran permisos definidos para algún grupo del usuario se comprueba para la definición de *Otros* devolviendo el resultado en caso de existir.

Si en ninguno de los casos anteriores se encuentra definido se procede a buscar en un nivel superior. En este caso en las carpetas definidas en la aplicación para cada componente (apartado 6.4.1.2).

6.4.4 Mensajes de difusión

Esta opción sólo estará disponible para los usuarios pertenecientes al grupo *Difusores* (apartado 6.4.2.4). Es importante saber que un usuario puede pertenecer a varios grupos, de forma que puede ser *Administrador* y *Difusor* a la vez. Desde esta opción se envían mensajes a todos los usuarios del sistema. Los mensajes les llegarán según la configuración de cada usuario, es decir, mediante un correo electrónico, mediante una alerta de la aplicación, o mediante ambos. En la opción (Figura 6.62) aparece un formulario con un campo para indicar el asunto, y otro el mensaje a difundir. Como remitente del mensaje aparecerá el correo electrónico del usuario que ha enviado el mensaje de difusión.

The image shows a web application interface for sending broadcast messages. On the left is a navigation menu with categories like 'Configuración', 'Datos del Sistema', 'Mis Datos', and 'Mensajes'. Under 'Mensajes', 'Mensajes de difusión' is selected. The main area shows the form for sending a message. It includes a breadcrumb trail 'Mensajes > Mensajes de difusión' and user information 'Administrador Administrador - Desconectarse Sistema: malagacomun - Cambiar Sistema'. The form has two required fields: 'Asunto*' (Subject) with the text 'Nueva aplicación' and 'Mensaje*' (Message) with the text 'Ya está disponible la nueva aplicación de Málaga Común.'. Below the message field is a note '*Campos obligatorios' and an 'Enviar a todos' button.

Figura 6.62: Formulario de un mensaje de difusión

6.5.1 Manuales

Con esta opción del menú de la Figura 6.63 se muestran los distintos manuales de los que se compone la aplicación. Cada usuario podrá ver los manuales para los que tenga permisos. A diferencia de la opción del apartado 6.3.3.5, los administradores también pueden ver el manual de administración de la aplicación.

6.5.5.1 Manual de usuario

Muestra un documento web donde se explica la funcionalidad para los usuarios que se registran en la aplicación y pueden participar en el sistema de moneda complementaria. Es similar al del apartado 6.3.3.4.1.

6.5.5.2 Manual de administración

Muestra un documento web donde se explica la configuración de la aplicación por parte del administrador del sistema.



Figura 6.63: Manuales disponibles para un administrador

6.5 Desarrollo de nuevas funcionalidades

En este apartado veremos las opciones para manipular la funcionalidad. Esta sólo se puede realizar por los usuarios pertenecientes al grupo de desarrolladores, y se lleva a cabo a través de una serie de opciones que sólo estarán disponibles para este grupo, y que pueden verse en la Figura 6.64.



Figura 6.64: Opción de menú para configurar la aplicación

6.5.1 Configuración

Muestra información sobre la configuración de la aplicación, **no del sistema de gestión de la moneda complementaria**. Las opciones dentro de este menú sólo deben ser manipuladas por usuarios expertos.

6.5.1.1 Opciones del menú

Las opciones del menú representan a cada una de las opciones del menú lateral de la izquierda. Desde esta ventana podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1). Pulsando la opción para ver el detalle podremos realizar todas las acciones definidas para las opciones de menú.

Para crear una nueva opción de menú debemos pulsar el botón con el texto *Añadir opción de menú*. Al pulsar sobre el botón para crear la opción de menú se nos abrirá el formulario para introducir los datos necesarios (Figura 6.65), el cual tiene dos pestañas:

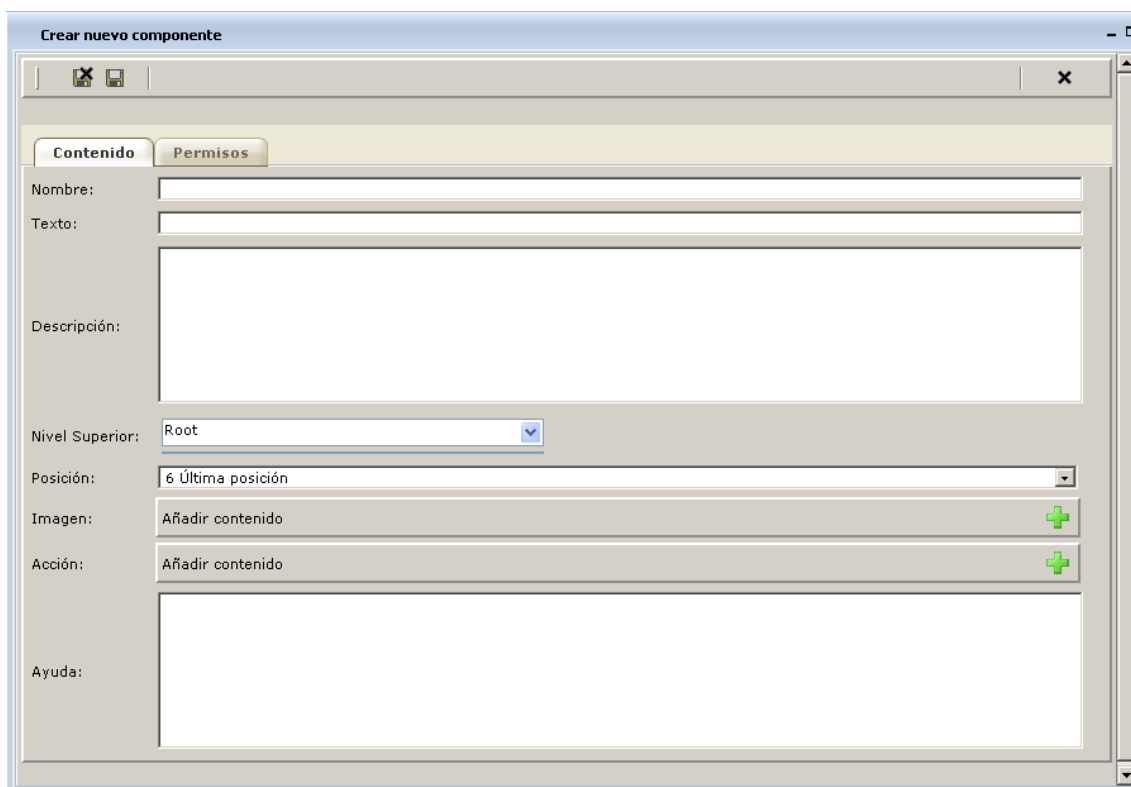


Figura 6.65: Formulario para crear una nueva opción de menú

Pestaña Contenido: define los datos de la opción de menú.

- **Nombre:** indica el nombre utilizado para identificar a la opción de menú.
- **Texto:** indica el texto que se va a pintar en pantalla a la hora de representar a la opción.
- **Descripción:** en este campo es donde se describe la opción de menú.
- **Nivel superior:** indica el nivel de la opción de menú. Si el nivel superior es *Root* (raíz), indica que la opción de menú es una opción principal, que no se encuentra contenida sobre ninguna otra. En caso contrario, la opción se encontrará contenida sobre la indicada en este campo.
- **Posición:** indica la posición de la opción de menú dentro de su jerarquía (el nivel definido en el campo *Nivel superior*).
- **Imagen:** la opción de menú puede tener una imagen asociada si se desea. Para añadir una nueva imagen se debe pulsar sobre el botón con una cruz verde situado a la derecha del campo *Imagen* (Figura 6.65). Al pulsar sobre este botón aparecerán los nuevos campos para añadir la imagen (Figura 6.20). Los campos para una imagen son la descripción, que es un texto que aparecerá al pasar el ratón sobre ella. La alternativa, que es un texto que aparece si la imagen no puede ser cargada. Y la imagen propiamente dicha, se puede cargar desde el ordenador local del usuario o bien utilizar una ya existente (subida con anterioridad). Para cargar la imagen debemos seleccionar la opción deseada y pulsar el botón que lo indica (Agregue imagen). Esta imagen se usará en la definición de la opción cuando se muestra desde la opción de listado de opciones (Figura 6.64).
- **Acción:** las opciones de menú deben definir las acciones que van a realizar. Este campo es el que las define. Para definir una nueva acción se debe pulsar sobre el botón con la cruz verde que se encuentra a la derecha del campo. Al pulsarlo se nos abrirá una nueva ventana donde se nos permitirá seleccionar una acción de las disponibles.

Según cual sea la acción seleccionada aparecerán nuevos campos de edición para la opción de menú.

- **Ayuda:** las opciones de menú pueden tener una pequeña ayuda. Si se introduce texto en este campo, se mostrará como ayuda básica. Se puede introducir código HTML.

Pestaña Permisos: esta pestaña se define en el apartado 6.4.3.




El formulario de edición y creación de una nueva opción de menú posee en la parte superior los botones para guardar, guardar y salir, y salir. Al salir del formulario aparecen los detalles de la nueva opción de menú creada junto con las acciones que se pueden realizar en la parte superior de la ventana (Figura 5.45).



Figura 6.67: Detalle da una opción de menú

Las acciones disponibles para una opción de menú se pueden ver en la Tabla 6.10.

Tabla 6.10: Acciones disponibles para una opción de menú

	Ver: acción que permite ver todos los datos detallados de una opción de menú.
	Editar: permite modificar los datos de una opción de menú.
	Borrar: elimina una opción de menú, previa confirmación.

6.5.1.2 Carpetas

Las carpetas representan a los contenedores padre de cada componente del sistema. Desde éstas se pueden modificar los permisos por defecto para éstos. Desde esta ventana además podremos realizar todas las operaciones genéricas de cada listado (apartado 6.3.3.1). Pulsando la opción para ver el detalle podremos realizar todas las acciones definidas para el autor, que se encuentran definidas en la Tabla 6.9.

Para crear una nueva carpeta debemos pulsar el botón con el texto *Añadir carpeta*. Al pulsar sobre el botón para crear la carpeta se nos abrirá el formulario para introducir los datos necesarios (Figura 6.68) la cual tiene dos pestañas. La segunda pestañas:

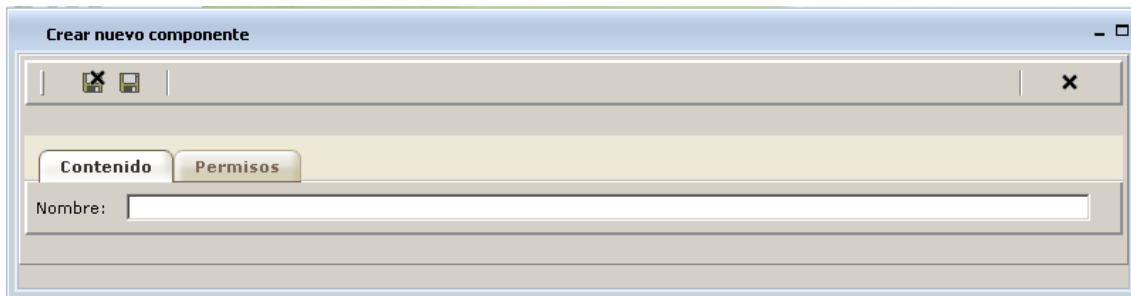


Figura 6.68: Formulario para crear una nueva carpeta o editar una ya existente

Pestaña Contenido: define los datos de la opción de menú.

- **Nombre:** texto obligatorio con el que se definirá la carpeta creada, para su correcto funcionamiento el nombre debe ser *Contenedor* seguido del nombre del componente del que se desea que sea la carpeta padre.

Pestaña Permisos: esta pestaña se define en el apartado 6.4.3.

El formulario de edición y creación de una nueva opción de menú posee en la parte superior los botones para guardar, guardar y salir, y salir. Al salir del formulario aparecen los detalles de la nueva carpeta creada junto con las acciones que se pueden realizar en la parte superior de la ventana (Figura 6.69).

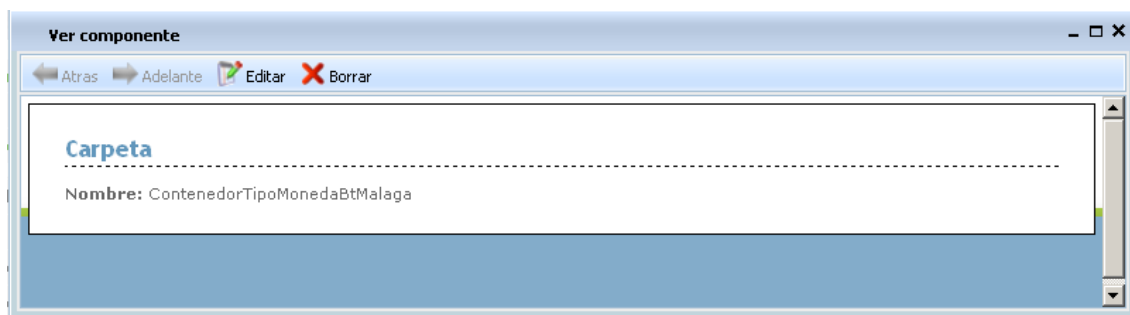





Figura 6.69: Detalle da una carpeta

Las acciones disponibles para una carpeta se muestran en la Tabla 6.11.

Tabla 6.11: Acciones disponibles para una carpeta

	Ver: acción que permite ver todos los datos detallados de una carpeta.
	Editar: permite modificar los datos de una carpeta.
	Borrar: elimina una carpeta, previa confirmación.

6.5.2 Administración del contenido estático de la aplicación

La aplicación de moneda complementaria se encuentra desarrollada bajo el sistema gestor de contenidos *SilverStripe* (<http://www.silverstripe.org>). Este sistema gestor de contenidos, CMS (Content Manager System) a partir de ahora, posee un área privada de administración que se accede a través de la dirección <http://www.malagacomun.org/silverstripe/admin>, es importante mencionar que esta dirección no se encuentra referenciada por ninguna parte de la aplicación por cuestiones de seguridad. Mantenerla oculta significa que habrá menos usuarios que intenten entrar sin tener las credenciales necesarias. Al acceder a la dirección se nos pedirán las credenciales mediante nombre de usuario y contraseña (Figura 6.5), a través de un formulario igual al utilizado para autenticarse en el sistema de moneda complementaria (apartado 6.2.3).

El usuario administrador del gestor de contenidos no tiene porque ser el mismo que en el sistema de moneda complementaria, de hecho no tiene ni siquiera porque estar registrado en él. Una vez se accede con las credenciales vemos en el área de administración (Figura 6.70) un menú de opciones en la parte izquierda de la pantalla.



Figura 6.70: Pantalla de acceso a la administrador del CMS

En el menú de la izquierda podemos observar la imagen de un pequeño globo terráqueo seguido de las palabras *Contenido del sitio*. Debajo de este enlace y siguiendo una jerarquía de padres e hijos se encuentran las diferentes páginas que conforman la aplicación. Algunas son contenido genérico que se crea junto con la instalación del CMS, a nosotros las que nos interesan son las páginas dentro de la jerarquía de la página llamada *Málaga Común*. A través de los enlaces para desplegar o replegar cada jerarquía de páginas (símbolos + y –) podemos navegar hasta la página que nos interese y pinchando sobre ella en el área principal de la pantalla aparecerán una serie de campos para modificar su contenido (Figura 6.71).



Figura 6.71: Edición de una página de la aplicación

Una vez estamos de acuerdo con el contenido editado en los campos pulsamos el botón *Guarda y Publicar* que se encuentra situado en la esquina inferior derecha de la pantalla y ya estarán disponibles los cambios de contenidos.

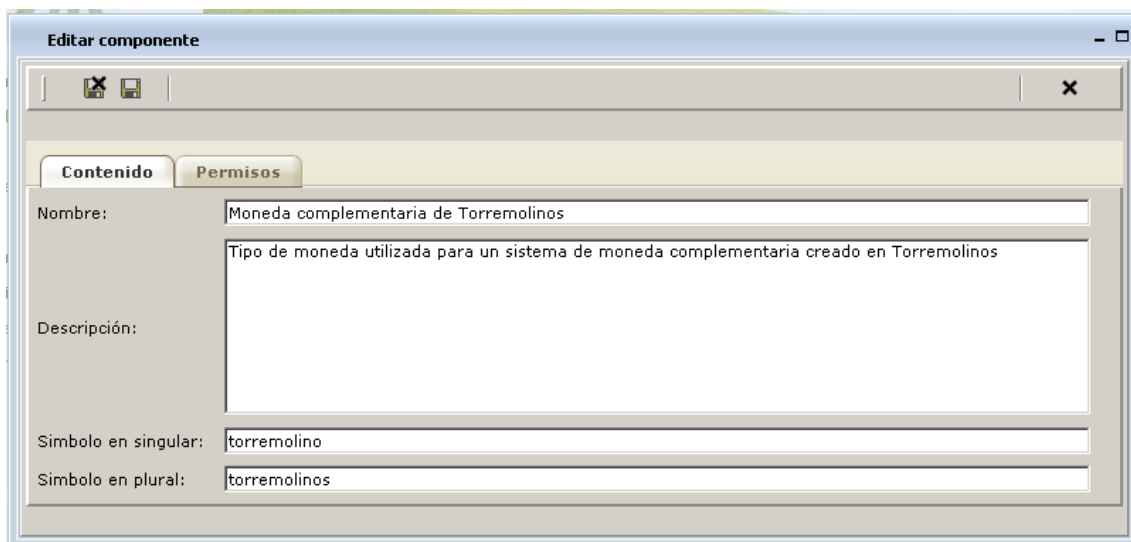
Nota: Para tener más información sobre el resto de opciones de la administración del gestor de contenido SilverStripe, vea la documentación que se encuentra en la web <http://www.silverstripe.org>.

6.6 Ejemplo de uso

A continuación se va a mostrar un ejemplo de uso de configuración y utilización de la aplicación. Para ello vamos a crear un sistema propio junto con dos usuarios, también registrados explícitamente para ilustrar el uso de la aplicación.

6.6.1 Ejemplo de creación de un nuevo sistema

Para crear un nuevo sistema lo primero que se debe hacer es conectarse a la aplicación con un usuario con permisos de administración (Figura 6.5).



The screenshot shows a web application window titled "Editar componente". It has a toolbar with icons for back, save, and close. Below the toolbar are two tabs: "Contenido" (selected) and "Permisos". The form contains the following fields:

Nombre:	Moneda complementaria de Torremolinos
Descripción:	Tipo de moneda utilizada para un sistema de moneda complementaria creado en Torremolinos
Simbolo en singular:	torremolino
Simbolo en plural:	torremolinos

Figura 6.72: Ejemplo de nuevo tipo de moneda en modo edición

Una vez conectado como administrador debemos crear el tipo de moneda sobre el que se va a basar nuestro sistema. Para ello se debe ir a la opción de menú *Tipos de monedas* de la sección *Datos del sistema* y pulsamos sobre el botón *Añadir tipo de moneda*. De esta forma

nos aparecerá un formulario para insertar los datos de nuestro nuevo tipo de moneda (Figura 6.72). Para nuestro ejemplo estos datos serán los que se muestran en la Figura 6.72.

Nota: en todos los casos del ejemplo vamos a ignorar la pestaña *Permisos* puesto que vamos a utilizar siempre los permisos por defecto de la carpeta padre.

Pulsamos el icono representado por la imagen de un disco con un aspa para guardar y salir, con lo que nos aparecerá en modo vista el nuevo tipo de moneda creado (Figura 6.73).

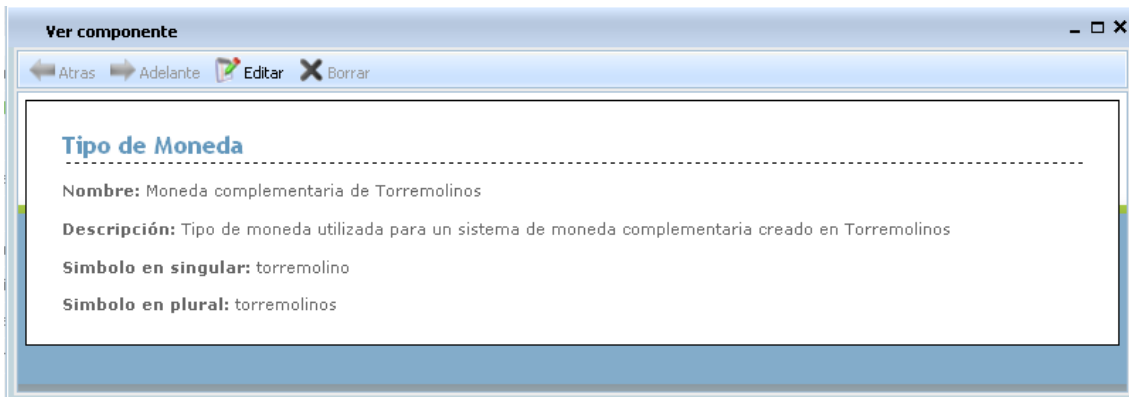


Figura 6.73: Ejemplo de nuevo tipo de moneda en modo vista

Cerramos esta ventana y podemos ver como en el listado de tipos de monedas existentes ha aparecido nuestro nuevo tipo de moneda (Figura 6.73).

Ahora vamos a la opción de menú *Sistemas* de la sección *Datos del sistema* para crear el nuevo sistema. De forma similar a como hemos hecho con el tipo de moneda, pulsamos sobre el botón *Añadir sistema* he introducimos los siguientes datos:

- **Nombre:** Moneda complementaria de Torremolinos.
- **Descripción:** Sistema para la moneda complementaria de Torremolinos.
- **Crédito por defecto:** lo mantenemos en el valor ilimitado.
- **Impuestos del sistema:** le ponemos un valor fijo de 0.3.
- **Ofertas iniciales:** le ponemos un valor de 1.

- **Tipo de moneda:** Moneda complementaria de Torremolinos (nota: se debe cambiar la pestaña y seleccionar el tipo a través del listado).
- **Administrador:** Seleccionamos al usuario *Administrador*.

Pulsamos guardar y salir, con lo cual se abre la ventana de en modo vista del nuevo sistema. Le damos a cerrar y observamos que efectivamente el nuevo sistema aparece en el listado de sistemas existentes (Figura 6.74).

	Nombre	Descripción
<input type="checkbox"/>	comunes	Moneda utilizada para Málaga Común
<input type="checkbox"/>	Moneda complementaria de Torremolinos	Tipo de moneda utilizada para un sistema de moneda complementaria creado en Torri

1 Filas de 1 a 2 de 2

Figura 6.74: Ejemplo de listado con el nuevo tipo de moneda

6.6.2 Ejemplo de registro de un nuevo usuario con sus ofertas mínimas

Ahora vamos a registrar un nuevo usuario. Para ello vamos a la opción definida en el apartado 6.3.1 e introducimos los datos requeridos en el formulario que aparecerá (Figura 6.8).

Pulsamos el botón *Siguiente*. Tal como se indicó en la creación del sistema sobre el que nos queremos registrar. Para que un usuario se registre en el sistema se debe dar de alta una oferta inicial.

Para finalizar el registro vamos a dar de alta una oferta con las siguientes características en la ventana que se muestra en la Figura 6.9 :

- **Categoría:** Formación
- **Título:** Clases de matemáticas
- **Descripción:** Me ofrezco para dar clases de matemáticas a niños de secundaria y bachillerato.
- **Tipo del precio:** por horas.
- **Precio:** 10.

Quando se pulse el botón *Registrarse* se habrá enviado un correo electrónico a la dirección de correo que hemos indicado en el registro indicándonos que nos hemos registrado en el sistema. Ahora ya podemos conectarnos con los datos de email y contraseña introducidos. Una vez conectado aparece una alerta de mensaje en la esquina inferior derecha. Si pulsamos sobre ella se abrirá la bandeja de entrada con un mensaje de bienvenida donde se explica la necesidad de que el usuario sea activado por un administrador del sistema para poder acceder a toda la funcionalidad proporcionada por la aplicación. Borrarnos el mensaje una vez lo hayamos leído y nos desconectamos. En el momento en que se active la cuenta se nos notificará mediante un correo electrónico o alerta.



Figura 6.75: Mensaje de registro de un nuevo usuario

Para activar al usuario nos conectamos como un administrador, y observamos que nos ha llegado un nuevo mensaje indicándonos el registro de un nuevo usuario y la necesidad de

activarlo. Para activar al usuario existen varias opciones, desde aquí veremos la que requiere menos pasos:

1.- Abrimos el mensaje que nos indica el registro del usuario (Figura 6.75)

2.- Pulsamos en el nombre del nuevo usuario registrado, con lo que se abrirá la página con sus datos (Figura 6.76), pulsamos sobre el botón *Activar* para activar al usuario. Automáticamente se asigna el grupo de *UsuarioActivo* y vemos que el botón cambia para pasar a ser ahora un botón de desactivar. Por último pulsamos el botón *Atrás* para volver atrás y borramos el mensaje, de forma que ya no nos vuelvan a aparecer las alertas por este mensaje.

Ver componente

Atrás Adelante Editar Activar

Datos personales

Nombre: Antonio
Primer Apellido: Garcia
Segundo Apellido: Fuentes
Telefono: 952430322
Movil: 666032343
Email: antonio@hotmail.com

no photo

Crédito disponible
0
comunes

Valoración media del usuario: ★★★★★ de 0 opiniones

CALENDARIO DE SERVICIOS

Hoy Diciembre 2010

Lun	Mar	Mie	Jue	Vie	Sab	Dom
		01	02	03	04	05
06	07	08	09	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Figura 6.76: Datos del usuario

Ahora que ya se ha activado el usuario, podemos entrar con sus credenciales y podremos ver todas las opciones de la aplicación.

Importante: para seguir con el ejemplo vamos a registrar un nuevo usuario siguiendo el mismo proceso que hemos seguido para el primer usuario y creamos una oferta con los datos:

- **Categoría:** Artículos de casa
- **Título:** Jarrón chino
- **Descripción:** Vendo jarrón chino de color blanco y negro.
- **Tipo del precio:** fijo.
- **Precio:** 30.

6.6.3 Ejemplo de creación, solicitud y realización de una oferta

Una vez que el segundo usuario se encuentra registrado y activo en el sistema, entramos con sus credenciales. Seleccionamos la opción de menú *Mis ofertas* de la sección *Mis datos* (Figura 6.16). En esta sección veremos que tenemos una oferta, que es la oferta que hemos creado al registrarnos en el sistema. Pulsamos sobre el icono de la lupa para ver sus detalles (la que se ve en la Figura 6.13) y la editamos para insertar una imagen. A continuación guardamos los cambios y publicamos la oferta para hacerla visible a todos los usuarios. Por último nos desconectamos del sistema.

Manteniéndonos desconectados vamos a la sección *Intercambios*. Seleccionamos *Ofertas* dentro del menú lateral y elegimos la categoría *Artículos de casa*. Observamos que se encuentra la oferta publicada por el usuario. Pulsamos para ver sus detalles y se nos indicará que nos autentiquemos en el sistema. Lo hacemos con el primer usuario que hemos creado.

Ahora podemos ver los detalles de la oferta. Si pulsamos sobre la acción *Solicitar servicio*, aparecerá un formulario con los datos necesarios para formalizar la factura del servicio, que en este caso únicamente es la selección de las fechas en que se realizará el servicio (Figura 6.31).

Pulsando dos veces sobre una fecha podemos insertar información sobre la realización del servicio. De modo que pinchamos dos veces sobre una fecha cualquiera e insertamos una breve descripción del servicio. Ahora pulsamos guardar y salir y nos aparecerán los datos de la factura que se ha creado, donde cabe destacar que se encuentra en el estado de ser aceptada

por el usuario que va a realizar el servicio. Si vamos a la sección *Mis facturas* veremos que existe una nueva factura pendiente de aceptación por el ofertante del servicio (Figura 6.77).

	Num. Factura	Precio	Estado
<input type="checkbox"/>	1026	-30	Pendiente de aceptación por el ofertante

1 Filas de 1 a 1 de 1

Figura 6.77: Listado de facturas del usuario conectado

Nos conectamos con el segundo usuario que hemos registrado, ya que es el que debe aceptar la factura (la solicitud del servicio). Al conectarnos una alerta nos informa de la petición del servicio, podemos abrir el mensaje y pulsar sobre el enlace de la factura para ver sus datos. Una vez en los datos de la factura pulsamos *Aceptar* para aceptarla con lo cual ésta pasa a estar en el estado *Pendiente de realizar* (también podríamos haberla cancelado o editado). Por último pulsamos el botón de atrás para volver a la vista del mensaje y lo borramos para evitar futuras alertas por este mensaje.

Si se modifica una factura aceptada vuelve al estado de pendiente de aceptación. También puede ser finalizada sólo por el usuario que va a realizar el pago, que en este ejemplo es el que ha solicitado la oferta, es decir, el primer usuario que hemos registrado para el ejemplo. Por tanto nos conectamos con este usuario y comprobamos que hay una alerta que nos indica que se ha aceptado el servicio. En este caso y aunque podemos indicar la terminación de este directamente desde el mensaje lo normal sería borrar la alerta y esperar a la realización del servicio para indicarlo. Para ello una vez que el servicio se ha realizado correctamente nos vamos a la opción de menú *Mis facturas* de la sección *Mis datos* y en el listado de todas las facturas (similar al de la Figura 6.77) seleccionamos la que queremos

cerrar, abrimos su detalle pulsando en el icono de un pequeña lupa, y en su detalle pulsamos el botón *Indicar servicio realizado*, momento en el que se realiza el pago del servicio.

Una vez realizado un servicio los dos usuarios implicados podrán opinar sobre él. Para ello, en las opciones de las facturas realizadas aparece un botón con la imagen de un bocadillo (Tabla 6.2). Si pulsamos sobre él podemos insertar la opinión. Una opinión consta de una valoración, que es una puntuación de 0 a 10 (que se mostrará de forma grafica en los datos del usuario) y una descripción sobre el servicio realizado o las incidencias habidas. Una vez realizada una opinión esta debe ser publicada, de igual manera que se publican las ofertas.

Si ahora vemos los datos de los usuarios veremos como se han actualizado sus créditos y valoraciones. Y las opiniones pueden ser vistas por cualquier usuario.

6.6.4 Ejemplo de creación, solicitud y realización de una demanda

Las demandas son iguales que las ofertas. Para crear una nueva demanda nos vamos a conectar con el segundo usuario que hemos creado para el ejemplo. Seleccionamos la opción de menú *Mis demandas* de la sección *Mis datos*, pulsamos sobre el botón *Añadir demanda* e introducimos los siguientes datos en el formulario:

- **Categoría:** Formación
- **Título:** Clases de inglés
- **Descripción:** Me gustaría recibir clases de inglés para mejorar mi gramática.
- **Tipo de precio:** Por horas.
- **Precio:** 0

Pulsamos el botón guardar y salir, y podremos ver los datos de la demanda que se ha creado. Esta demanda aún no está publicada, la publicamos pulsando el botón publicar.

Ahora vamos a solicitar la demanda con el primer usuario que hemos registrado, para ello nos conectamos con ese usuario y nos vamos a la opción de menú *Demandas* de la sección *Datos del sistema*. Desde esta opción podemos ver un listado con todas las demandas disponibles. Seleccionamos la deseada (en nuestro caso la que hemos creado previamente) y

pulsamos sobre la acción *Solicitar servicio* con lo que nos aparecerá un formulario con los datos necesarios para formalizar la factura del servicio, que en este caso además de la selección de las fechas en que se realizará el servicio, debemos indicar el precio que va a tener la realización del servicio. Es importante notar que si el usuario no tiene el crédito necesario para pagarlo por haber sobrepasado el límite de su perfil o del sistema, se mostrará un error indicando que no se puede realizar dicho servicio tal como se ve en la Figura 6.78.



Figura 6.78: Error de validación del crédito de una factura de demanda

Rellenamos los datos correctamente y pulsamos el icono con un disco y un aspa para guardar y salir y nos aparecerán los datos de la factura que se ha creado, donde cabe destacar que se encuentra en el estado de ser aceptada por el usuario que va a realizar el servicio.

Ahora nos conectamos con el segundo usuario que hemos registrado para el ejemplo, que es el que debe aceptar la factura (la solicitud del servicio). Al conectarnos una alerta nos informa de la petición del servicio, podemos abrir el mensaje y pulsar sobre el enlace de la factura para ver sus datos. Una vez en los datos de la factura pulsamos *Aceptar* para aceptarla con lo cual esta pasa a estar en el estado *Pendiente de realizar* (también podríamos haberla cancelado o editado para cambiar el precio, por ejemplo). Por último pulsamos el botón de atrás para volver a la vista del mensaje y lo borramos para evitar futuras alertas por este mensaje. En el campo observaciones también se puede poner el teléfono de forma que ambos usuarios pueden ponerse por teléfono de acuerdo en precio, horarios, etc.

Para pagar una factura aceptada el usuario que debe realizar la acción es el que va a realizar el pago, en este ejemplo será el segundo usuario que hemos registrado, por tanto ya estamos conectados con este usuario. Así, una vez que el servicio se ha realizado correctamente nos vamos a la opción de menú *Mis facturas* de la sección *Mis datos* (Figura 6.16) y en el listado de todas las facturas (Figura 6.77) seleccionamos la que queremos cerrar, pulsando para ello el botón *Indicar servicio realizado* momento en el que se realiza el pago del servicio.

Una vez realizado un servicio sus dos implicados podrán opinar sobre él, tal y como hemos visto en el apartado anterior.

Si ahora vemos los datos de los usuarios veremos como se han actualizado sus créditos y valoraciones.

6.6.5 Ejemplo de envío de un mensaje

En este ejemplo el primer usuario que hemos registrado va a enviar un mensaje al segundo usuario, para ello lo primero que debemos hacer es conectarnos con el primer usuario, e ir a la opción de menú *Borrador* de la sección *Mensajes* (Figura 6.16). Desde ahí creamos un nuevo mensaje indicando en el receptor al usuario al que le vamos a enviar el mensaje.

Una vez creado el mensaje simplemente debemos pulsar en el botón enviar.

Conclusiones y Líneas Futuras

Se ha desarrollado un software atendiendo a tres principios básicos fundamentalmente. En primer lugar, como proyecto de fin de carrera que es, se ha pretendido que con su desarrollo se aprendan una serie de conceptos y técnicas para el desarrollo de software. Este objetivo se ha cumplido y se culmina con la realización de esta memoria, en la que se detallan aspectos tan importantes en el desarrollo de software como pueden ser la toma de requisitos, el análisis del sistema, el desarrollo, y la puesta en producción del producto final. Además, se han evaluado las principales herramientas disponibles actualmente, escogiendo la que mejor se adaptase a los requisitos del sistema, y se ha innovado en muchos otros aspectos, buscando las mejoras que marquen la diferencia entre este software y otros desarrollados con la misma finalidad.

Como segunda finalidad se ha pretendido hacer un software útil, que pueda ser utilizado por el colectivo de moneda complementaria Málaga Común para la gestión de sus actividades a través de la web. Esto implica que el software ha de ser sencillo de utilizar, rápido, flexible, fácil de mantener y barato de implantar. A diferencia de otros proyectos que no tienen más que la finalidad académica, este requisito ha implicado entrar de lleno en la fase de implantación del software. Aquí se han tenido que comparar las diferentes alternativas en cuanto a alojamientos webs, instalar la aplicación para que funcione realmente, abrir un periodo de mantenimiento, optimizar los accesos a base de datos, someter la aplicación al testeo de los usuarios que lo van a utilizar, y abrir una vía de comunicación para notificar errores y posibles mejoras.

Por último, y como tercera finalidad, se ha propuesto crear un software lo más genérico posible, de forma que a partir de una estructura común, se puedan desarrollar programas con características similares en un tiempo breve. Para ello se ha creado un sistema de acciones, que implementen la funcionalidad específica de la aplicación, manteniendo de forma genérica al resto, con lo que aproximadamente se mantiene el 70% del código si deseamos cambiar la funcionalidad para gestionar otras aplicaciones similares como una tienda online, una aplicación de gestión de datos, etc.

A partir de estas finalidades se abren varias líneas futuras de desarrollo. Por un lado la aplicación de gestión de la moneda complementaria Málaga Común, entra en un periodo de

explotación. En el cual irán surgiendo futuras necesidades que podrán ser tratadas como evolutivos en algún momento dado. Entre estas necesidades, y una vez el software ha empezado a utilizarse por parte de los usuarios finales han surgido una serie de peticiones y mejoras, estas son:

- Traducción de la aplicación a otros idiomas como el inglés.
- Posibilidad de indicar en las ofertas y demandas URL a imágenes ya existentes en alguna dirección de Internet.
- Marcado de las alertas ya leídas, para de esta forma poder identificarlas fácilmente.

Esto es algo que se ha mantenido presente durante todo el proceso de desarrollo, de forma que la aplicación final es escalable y fácil de modificar.

En cuanto a su genéricidad se abre una línea de futuro muy interesante. Con una correcta separación entre el código genérico y el específico se podría utilizar el primero para el desarrollo eficiente de otras aplicaciones como pueden ser tiendas online, gestión de videoclubs, talleres, etc.

REFERENCIAS

- [1] José Galindo, "Salvemos Nuestro Planeta". Editorial Lulu.com, 2008 (www.resumelibros.tk).
- [2] Wikipedia, la enciclopedia libre - <http://www.wikipedia.es> (Accedido en Julio 2009).
- [3] The LETSystem Design Manual – Michael Linton, Angus Soltar. (1999). <http://www.gmlets.u-net.com>
- [4] Portal de la red de banco del tiempo – <http://www.red-bdt.org> (Accedido en Julio 2009).
- [5] Wikipedia: <http://es.wikipedia.org> (Accedido en Mayo 2010).
- [6] Ellen Siever, Stephen Figgins, Robert Love y Arnold Robbins. Linux. Ed. Anaya Multimedia, 2008.
- [7] Galindo, J., Urrutia, A., Piattini, M., Fuzzy Databases: Modeling, Design and Implementation. Idea Group Publishing Hershey, USA, 2006.
- [8] Universidad Politécnica de Valencia, “Manual de estilo para la web de la universidad politécnica de Valencia” - http://www.upv.es/entidades/ASIC/manuales/guia_estilos_upv.pdf (Accedido en Agosto 2010).
- [9] Información sobre desarrollo web - <http://www.desarrolloweb.com> (Accedido en Agosto 2010).
- [10] Documentación online de SilverStripe - <http://doc.silverstripe.org> (Accedido desde Mayo 2010 hasta Enero 2011).
- [11] Accesibilidad web – <http://www.w3c.es> (Accedido Mayo 2010).
- [12] Introducing JSON – <http://www.json.org> (Accedido Mayo 2010).

- [13] Documentación controles DHTML – <http://www.dhtmlx.com> (Accedido desde Julio 2009 hasta Enero 2011).
- [14] Artículo sobre la instalación del software – http://es.wikipedia.org/wiki/Instalaci3n_de_software (Accedido en Enero 2011).
- [15] Definición de aplicación web - http://es.wikipedia.org/wiki/Aplicaci3n_web (Accedido en Enero 2011).
- [16] Definición de ingeniería del software – http://es.wikipedia.org/wiki/Ingenier3a_de_software (Accedido en Enero 2011).
- [17] Definición de pruebas de rendimiento – http://es.wikipedia.org/wiki/Pruebas_de_rendimiento_del_software (Accedido en Enero 2011).
- [18] Jakob Nielsen, “Usability Engineering”. Publicado por: Morgan Kaufmann, San Francisco, 1994.ISBN 0125184069.
- [19] The Need for Speed - <http://www.useit.com/alertbox/9703a.html> (Accedido en Enero 2011).
- [20] Response Times: The 3 Important Limits - <http://www.useit.com/papers/responsetime.html> (Accedido en Enero 2011).
- [21] Web del proyecto Apache jmeter - <http://jakarta.apache.org/jmeter/> (Accedido en Enero 2011).
- [22] Tutorial sobre jmeter - <http://www.osmosislatina.com/jmeter/basico.htm> (Accedido en Enero 2011).
- [23] Manual Apache jmeter – http://jakarta.apache.org/jmeter/usermanual/component_reference.html (Accedido en Enero 2011).
- [24] Definición de mantenimiento del software – http://es.wikipedia.org/wiki/Mantenimiento_de_software (Accedido en Enero 2011).