



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

**OPENCONFERENCE:
APLICACIÓN WEB LIBRE PARA LA GESTIÓN DE CONGRESOS**

Antonio Falcón Aragón

14 de octubre de 2009



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERO TÉCNICO EN INFORMÁTICA DE SISTEMAS

OPENCONFERENCE: APLICACIÓN WEB LIBRE PARA LA GESTIÓN DE CONGRESOS

- Departamento: Lenguajes y Sistemas Informáticos
- Directores del proyecto: Manuel Palomo Duarte, Alejandro Álvarez Ayllón
- Autor del proyecto: Antonio Falcón Aragón

Cádiz, 14 de octubre de 2009

Fdo: Antonio Falcón Aragón

Agradecimientos

Quisiera dar las gracias especialmente a Manuel Palomo Duarte, director de este proyecto, por sacar siempre tiempo para ayudarme a resolver todos y cada uno de los problemas que me han surgido en la elaboración del mismo, gracias a mi novia Estefanía por su infinito apoyo y motivación, y a todos mis familiares por mostrar su ayuda para que siguiera adelante con este proyecto.

Licencia

Este documento ha sido liberado bajo Licencia GFDL 1.3 (GNU Free Documentation License). Se incluyen los términos de la licencia en inglés al final del mismo.

Copyright (c) 2009 Antonio Falcón Aragón.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Notación y formato

A continuación incluimos algunos aspectos relevantes sobre la notación y el formato empleado a lo largo del documento.

- Cuando nos refiramos a un programa en concreto, utilizaremos la notación: *Planner*.
- Cuando nos refiramos a un comando, o función de un lenguaje, usaremos la notación: `sort`.
- Cuando nos refiramos a un comando o expresión en consola, usaremos la notación:

```
sudo ./lampp start
```

- Cuando nos refiramos a trozos de código, usaremos la notación:

```
1  class Micontrolador extends Controllers
2  {
3      public function Index()
4      {
5          $this->cadena = "Hola Mundo";
6      }
7  }
```


Índice general

1. Introducción	19
1.1. Objetivos	19
1.2. Composición del documento	19
2. Conceptos Básicos	21
2.1. Visión Global del Producto	21
2.2. Antecedentes	22
2.3. Alcance	22
3. Planificación	23
3.1. Incrementos	23
3.1.1. Incremento 1: Requisitos básicos del sistema	23
3.1.2. Incremento 2: Panel de Administración	24
3.1.3. Incremento 3: Sistema de Noticias y Comentarios	25
3.1.4. Incremento 4: Sistema de Usuarios	27
3.1.5. Incremento 5: Sistema de Templates	27
3.1.6. Incremento 6: Sistema de configuración	29
3.1.7. Incremento 7: Sistema de Páginas y Navegación	30
3.1.8. Incremento 8: Panel lateral y enlaces	31
3.1.9. Incremento 9: Sistema de revisión por pares	31
3.1.10. Incremento 10: Sistema de creación de calendario	32
3.1.11. Incremento 11: Sistema de creación de certificados	32
3.1.12. Incremento 12: Sistema de Backup	33
3.1.13. Incremento 13: Sistema de Información y Estadísticas	33
3.1.14. Incremento 14: Template 1 (Emplode)	34
3.1.15. Incremento 15: Template 2 (Vector Lover)	35
3.1.16. Incremento 16: Template 3 (Colorvoid)	36
3.2. Diagrama de Gantt	36
3.3. Esfuerzo	39
4. Análisis	41
4.1. Metodología de Desarrollo	41
4.2. Análisis del sistema	42
4.3. Diagrama E-R	43
4.4. Definición de Actores	43
4.5. Definición de Procesos	43
4.6. Diagrama de Casos de Uso	44
4.6.1. Caso de Uso: Envío, modificación y comprobación de Papers por Autor	44
4.6.2. Caso de Uso: Revisión de Papers por Revisor	45

4.6.3.	Caso de Uso: Gestión de Proceso de Revisión de Artículos por Presidente	45
4.7.	Representación UML del Modelado Conceptual	46
4.7.1.	Caso Práctico: Sistema de Revisión de Artículos	47
4.8.	Funcionalidades de la Aplicación	48
4.8.1.	Panel de Administración	48
4.8.2.	Sistema de Noticias y Comentarios	48
4.8.3.	Sistema de control de Usuarios	49
4.8.4.	Sistema de Templates	49
4.8.5.	Sistema de Configuración	50
4.8.6.	Sistema de Páginas y Navegación	50
4.8.7.	Panel Lateral y Enlaces	50
4.8.8.	Sistema de Revisión por pares	51
4.8.9.	Sistema de Calendario	51
4.8.10.	Sistema de Certificación	52
4.8.11.	Sistema de Backup	52
4.8.12.	Sistema de Información y Estadísticas	52
5.	Diseño	53
5.1.	Administración	53
5.1.1.	Control	54
5.1.2.	Noticias	54
5.1.3.	Comentarios	54
5.1.4.	Usuarios	54
5.1.5.	Templates	54
5.1.6.	Enlaces	55
5.1.7.	Navegación	55
5.1.8.	Panel Lateral	55
5.1.9.	Páginas	55
5.1.10.	Settings	56
5.1.11.	Calendario	56
5.1.12.	Call For Papers	56
5.1.13.	Certificados	56
5.2.	Noticias	57
5.3.	Páginas	57
5.4.	Usuario	57
5.5.	Papers	58
5.6.	Características de los usuarios finales	58
6.	Implementación	59
6.1.	Desarrollo de OpenConference	60
6.1.1.	Panel de Administración	60
6.1.2.	Noticias	65
6.1.3.	Páginas	66
6.1.4.	Usuarios	66
6.2.	Desarrollo de Librerías	67
6.2.1.	Access_library	67
6.2.2.	Links_library	68
6.2.3.	Navigation_library	68
6.2.4.	Pages_library	69
6.2.5.	Sidebar_library	69

6.2.6.	System_library	70
7.	Pruebas	71
7.1.	Proceso de Prueba	71
7.1.1.	Incremento 1: Requisitos Básicos del sistema	71
7.1.2.	Incremento 2: Panel de Administración	72
7.1.3.	Incremento 3: Sistema de Noticias y Comentarios	72
7.1.4.	Incremento 4: Sistema de Usuarios	73
7.1.5.	Incremento 5: Sistema de Templates	73
7.1.6.	Incremento 6: Sistema de Configuración	73
7.1.7.	Incremento 7: Sistema de Páginas y Navegación	74
7.1.8.	Incremento 8: Panel Lateral y enlaces	74
7.1.9.	Incremento 9: Sistema de revisión por pares	75
7.1.10.	Incremento 10: Sistema de creación de calendario	75
7.1.11.	Incremento 11: Sistema de creación de certificados	76
7.1.12.	Incremento 12: Sistema de Backup	76
7.1.13.	Incremento 13: Sistema de Información y Estadísticas	76
8.	Conclusiones	77
8.1.	Posibles Ampliaciones	77
8.2.	Opinión personal	78
A.	Manual de Instalación	81
A.1.	Descargando OpenConference	81
A.2.	Ejecutando el instalador	81
A.3.	Borrando la carpeta install	86
B.	Manual de Usuario	87
B.1.	Ejecución	87
B.2.	Primeros Pasos	88
B.2.1.	Acceder al sistema	88
B.2.2.	Ver Perfil	89
B.2.3.	Editar Perfil	90
B.2.4.	Salir del sistema	91
B.3.	Administrar Posts y Comentarios	91
B.3.1.	Posts	91
B.3.2.	Comentarios	93
B.4.	Administrar Usuarios	93
B.5.	Administrar templates	94
B.6.	Administrar Páginas y Navegación	95
B.6.1.	Páginas	95
B.6.2.	Navegación	97
B.7.	Administrar CFP	98
B.8.	Crear Calendario	100
B.9.	Crear Certificado	101
B.10.	Cambiar configuración de la página	103
B.11.	Hacer copia de seguridad de los datos	103

C. Manual del Desarrollador	105
C.1. Estructura de Carpetas	105
C.2. Introducción de Nuevo Diseño en OpenConference	106
C.2.1. Paso 1: Creación de Directorios	106
C.2.2. Paso 2: Adaptar a OpenConference	106
C.2.3. Paso 3: Incluir Diseño en OpenConference	122
Bibliografía y referencias	125
GNU Free Documentation License	127
1. APPLICABILITY AND DEFINITIONS	127
2. VERBATIM COPYING	128
3. COPYING IN QUANTITY	128
4. MODIFICATIONS	129
5. COMBINING DOCUMENTS	130
6. COLLECTIONS OF DOCUMENTS	131
7. AGGREGATION WITH INDEPENDENT WORKS	131
8. TRANSLATION	131
9. TERMINATION	131
10. FUTURE REVISIONS OF THIS LICENSE	132
11. RELICENSING	132
ADDENDUM: How to use this License for your documents	132

Índice de figuras

3.1. Panel de Administración	24
3.2. Listado de posts	25
3.3. Creando nuevo post	26
3.4. Edición de Usuario	27
3.5. Elección de Template	28
3.6. Configuración global del sistema	29
3.7. Creación de nueva página	30
3.8. Administración del panel lateral	31
3.9. Etapa de revisión de un paper	32
3.10. Información relativa al sistema, autor, documentación...	33
3.11. Plantilla de diseño «Emplode»	34
3.12. Plantilla de diseño «Vector Lover»	35
3.13. Plantilla de diseño «Colorvoid»	36
3.14. 1ª Parte del Diagrama de Gantt	37
3.15. 2ª Parte del Diagrama de Gantt	38
4.1. Diagrama de Caso de Uso para Funcionalidades de Autor	44
4.2. Diagrama de Caso de Uso para Revisión de Papers	45
4.3. Diagrama de Caso de Uso para Gestión de Proceso de Revisión de Artículos	45
4.4. Diagrama de Clases	47
4.5. Captura de noticias publicadas en el sistema	49
4.6. Captura de pantalla mostrando enlaces de interés	51
A.1. Instalación de OpenConference	82
A.2. Paso 1 (Aviso) - Instalación de OpenConference	82
A.3. Paso 1 (Correcto) - Instalación de OpenConference	83
A.4. Paso 2 - Instalación de OpenConference	84
A.5. Paso 3 - Instalación de OpenConference	85
A.6. OpenConference instalado	86
B.1. Página de Inicio de OpenConference - Diseño Vector Lover	87
B.2. Página de Registro de OpenConference - Diseño Vector Lover	88
B.3. Página de Login de OpenConference - Diseño Vector Lover	89
B.4. Ver Perfil de Usuario en OpenConference- Diseño Vector Lover	90
B.5. Editar Perfil de Usuario OpenConference - Diseño Vector Lover	90
B.6. Administrar Posts en Openconference	91
B.7. Editar Post en OpenConference	92
B.8. Administrar Comentarios en OpenConference	93
B.9. Administrar Usuarios en OpenConference	94
B.10. Editar Usuario en OpenConference	94

B.11. Administrar Templates en OpenConference	95
B.12. Editar Página en OpenConference	96
B.13. Crear Página en OpenConference	96
B.14. Administrar Navegación en OpenConference	97
B.15. Editar Item de Navegación en OpenConference	97
B.16. Administrar Papers en OpenConference	98
B.17. Revisar Paper en OpenConference	99
B.18. Ver Valoración Paper en OpenConference	100
B.19. Administrar Calendario en OpenConference	100
B.20. Muestra de Calendario generado en OpenConference	101
B.21. Administrar Certificados en OpenConference	101
B.22. Ejemplo de Certificado generado en OpenConference	102
B.23. Configuración de OpenConference	103

Indice de tablas

3.1. Tabla de tareas resultantes en el Diagrama de Gantt	39
--	----

Capítulo 1

Introducción

1.1. Objetivos

Se pretende crear una aplicación web para gestionar todo el proceso que conlleva la organización de un congreso. Para ello el administrador u organizador del congreso se ayudará de un panel de control, desde dónde podrá gestionar tanto el proceso de inscripción de los usuarios, el envío y revisión de Papers (contribuciones) mediante el método CFP¹, la generación del calendario de ponencias, etc...incluso personalizar el diseño de la web.

Se desea que la aplicación sea lo más intuitiva y sencilla posible, para que cualquier persona (con o sin conocimientos informáticos) pueda organizar su propio congreso. Esto es importante y aporta un beneficio en tiempo y esfuerzo, ya que ahora el proceso no se realiza manualmente y por tanto no requiere muchos conocimientos en informática y desarrollo web.

1.2. Composición del documento

El presente documento se divide en 8 capítulos en los que iremos desglosando los detalles de cada una de las etapas por las que hemos pasado durante el proceso de desarrollo del proyecto.

En el capítulo introductorio, a parte de la sección en la cuál nos encontramos, hemos definido los objetivos que esperamos cumplir con este proyecto, una vez finalizado el mismo.

En el siguiente capítulo trataremos algunos conceptos básicos, explicando el tema a tratar y la tecnología usada para ello.

En los posteriores capítulos abordaremos con detalle cada una de las fases que se llevan a cabo durante la creación de cualquier sistema software. De esta manera, el tercer capítulo estará dedicado a la etapa de planificación, donde expondremos cada una de las tareas realizadas así como el tiempo dedicado a cada una de ellas. Para facilitarnos esta labor haremos uso de la herramienta *Planner*².

El cuarto capítulo se centrará en la etapa de análisis del sistema, aquí realizaremos un estudio para crear una base de datos adecuada a todas las necesidades de nuestra aplicación.

¹Call For Papers: es un método usado en el ámbito académico y otros contextos para recopilar actas, libros, artículos o presentaciones de conferencias. Un Paper se envía a las partes interesadas, describiendo el tema a tratar, y un resumen del mismo, que será sometido a evaluación en un plazo de tiempo especificado.

²«Planner» es una herramienta libre para la gestión de proyectos.

En el quinto capítulo profundizaremos en la fase de diseño, definiendo todos los detalles de nuestro sistema, y en el capítulo sexto se llevará a cabo la implementación del mismo, comentando las diferentes técnicas usadas así como las herramientas necesarias para su realización.

En el séptimo capítulo, y para finalizar el proceso de desarrollo, se comentarán las distintas pruebas realizadas para comprobar el correcto funcionamiento de todo el sistema. Seguidamente, en el octavo capítulo se extraerán las distintas conclusiones tras finalizar la realización del proyecto; el tiempo empleado, qué se ha aprendido con todo ello...

Y por último se incluirán una serie de apéndices donde podremos encontrar algunos manuales útiles tales como el manual de instalación, el manual de usuario y el manual de desarrollador (para posibles ampliaciones y/o modificaciones del sistema).

Capítulo 2

Conceptos Básicos

2.1. Visión Global del Producto

El presente proyecto nace con la idea de desarrollar un producto bajo licencia GPL que haga frente a la demanda de crear una aplicación que permita, íntegramente, gestionar todas las tareas a realizar cuando se organiza un congreso.

Cada vez que se organiza un congreso, normalmente, se llevan a cabo las siguientes tareas:

- Se crea la web del congreso con la información disponible en ese momento, al menos, temática, fecha de celebración, sede y petición de contribuciones (CFP) con un formato de artículo.
- Los investigadores que lo desean mandan artículos con el formato que se indicó en la web.
- Una vez llegada la fecha tope de recepción del CFP ya no se aceptan más artículos. A continuación estos pasan a ser evaluados.

Para evaluarlos hay un comité científico de expertos en la materia que lee los artículos bajo acuerdo de no divulgación. Cada uno de ellos emite una valoración sobre varios aspectos de los artículos (por ejemplo, originalidad, adecuación a las jornadas, uso del idioma, opinión, etc). También se les puede pedir que hagan un resumen del trabajo y que expliquen los puntos fuertes y débiles del trabajo.

- Una vez termina el proceso de revisión, el responsable del congreso decide (basándose en lo dicho por los expertos) qué artículos se aceptan y cuáles se rechazan. Esa decisión (y toda la información generada por los expertos excepto los nombres, porque todo es anónimo) se les hace llegar a los que enviaron trabajos por e-mail.
- Los autores cuyo trabajo ha sido aceptado tienen un período de tiempo para subir la versión definitiva (con las mejoras que les hayan sugerido los revisores) e inscribirse en el congreso.
- Se hace un calendario para que todo el mundo exponga sus trabajos. El calendario suele incluir una inauguración, una visita turística, una cena de gala, varias ponencias de científicos de prestigio invitados, etc.
- Se celebra el congreso y se entregan los certificados a todos los ponentes y asistentes.

En la actualidad existe una aplicación, casualmente llamada **OpenConference**[1], que permite llevar a cabo una parte de las tareas anteriormente citadas, pero creemos que es realmente compleja de utilizar si no se tienen los conocimientos informáticos adecuados.

2.2. Antecedentes

Realmente apenas existen aplicaciones destinadas a gestionar congresos, y de las existentes, la mayoría no permite llevar a cabo todo el proceso anterior desde una misma aplicación. Existen herramientas que sólo permiten gestionar el apartado dedicado a CFP, por tanto la creación de la web y demás tareas son responsabilidad del organizador del congreso, con lo cuál ya es necesario que dicha persona posea conocimientos informáticos, al menos un mínimo conocimiento en desarrollo web.

Como se ha dicho anteriormente, también existen aplicaciones con muchas funcionalidades, pero muy complejas de utilizar, razón por la cuál, en un primer momento, los usuarios sean reacios a utilizarlas.

Pos estos motivos, pensamos que se trata de un proyecto original, que se adecua perfectamente a la web 2.0, ya que está enfocado hacia usuarios finales que, sin conocimientos informáticos, pueden crear una web con multitud de funcionalidades, estableciendo un punto de encuentro entre todos los interesados en el congreso anunciado.

2.3. Alcance

Se pretende que la aplicación sea más sencilla de utilizar que las comentadas anteriormente, y que consiga hacer que los usuarios pierdan el miedo a organizar un congreso, ya no será una tarea tediosa de realizar y no será necesario poseer demasiados conocimientos informáticos para crear la web y ponerla a funcionar correctamente, y todo en cuestión de minutos, si se siguen las indicaciones expuestas más adelante, en este mismo documento. Creemos que la interfaz de OpenConference es agradable al usuario, y que el panel de administración desarrollado cumple con todas las expectativas.

En cuanto a la escalabilidad de la aplicación, decir que está desarrollada bajo el lenguaje de programación PHP, utilizando conocimientos básicos de HTML y CSS. Por tanto, al ser tecnologías muy populares actualmente, la mayor parte de los usuarios con conocimientos en desarrollo web conoce estas herramientas y podrá trabajar con ellas con mucha facilidad, incorporando nuevas funcionalidades y/o mejorando las existentes.

Capítulo 3

Planificación

3.1. Incrementos

Para desarrollar nuestro sistema hemos considerado que la mejor elección es hacer uso de un modelo incremental.

En una visión genérica, el proceso se divide en 4 partes: Análisis, Diseño, Implementación y Prueba. Sin embargo, para la producción del Software, se usa el principio de trabajo en cadena o "Pipeline", utilizado en muchas otras formas de programación.

Consideramos que esto es importante ya que estamos en constante contacto con los resultados obtenidos en cada incremento. Por tanto, se nos hace más sencillo incluir o desechar elementos al final de cada incremento, a fin de que el software se adapte mejor a nuestras necesidades reales. Este proceso se repite hasta que se elabore el producto completo. Y de esta forma también el tiempo de entrega se reduce considerablemente.

Cabe destacar que, al igual que otros métodos de modelado, el modelo incremental es de naturaleza interactiva, pero se diferencia de ellos en que al final de cada incremento se entrega un producto operacional.

En nuestro caso particular, en el primer incremento sólo consideraremos los requisitos básicos de la aplicación, y a partir de ahí ya iremos probando y añadiendo más funcionalidad al sistema.

3.1.1. Incremento 1: Requisitos básicos del sistema

En este incremento nos centramos en extraer toda la información posible para conocer cuáles son las funcionalidades que debe tener nuestro sistema. A raíz de eso, haremos un calendario con el tiempo disponible para realizar el proyecto, así como el tiempo que debemos dedicar en desarrollar cada una de esas funcionalidades.

También buscaremos todos los recursos necesarios para poder empezar a trabajar en la aplicación, véase: imágenes, fondos, iconos, etc...

Cabe destacar que la aplicación será desarrollada bajo el lenguaje de programación PHP, y haciendo uso del framework *CodeIgniter*¹. Estas herramientas eran totalmente desconocidas para mí, por tanto

¹«CodeIgniter» es un entorno de desarrollo abierto que permite crear webs dinámicas con PHP. Su principal objetivo es ayudar a que los desarrolladores puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero.

también tuve que dedicar un tiempo para aprender a usarlas correctamente. Lo mismo me ocurrió a la hora de elaborar este documento, ya que no conocía $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

3.1.2. Incremento 2: Panel de Administración

En este incremento nos centramos en crear toda la interfaz del panel de administración. Desde aquí se podrá controlar todo el sistema, por tanto podemos decir que es el núcleo de nuestra aplicación. En esta primera versión el usuario ya podrá acceder y visualizar todas las opciones del panel de administración pero, lógicamente, estas opciones aún no realizarán ninguna acción.

Primero hemos conseguido un estilo para nuestro panel de administración, extraído de la web **Arcsin**[2] que posee multitud de templates libres con los que aportar un buen aspecto a nuestra aplicación.

Seguidamente hemos realizado la misma operación para conseguir los iconos que formarán parte del panel, pero esta vez los hemos conseguido de la web **iconPot**[3] que también posee infinidad de iconos libres, listos para poder usarlos en nuestra web.

Aquí vemos el aspecto que tiene nuestro panel de administración.

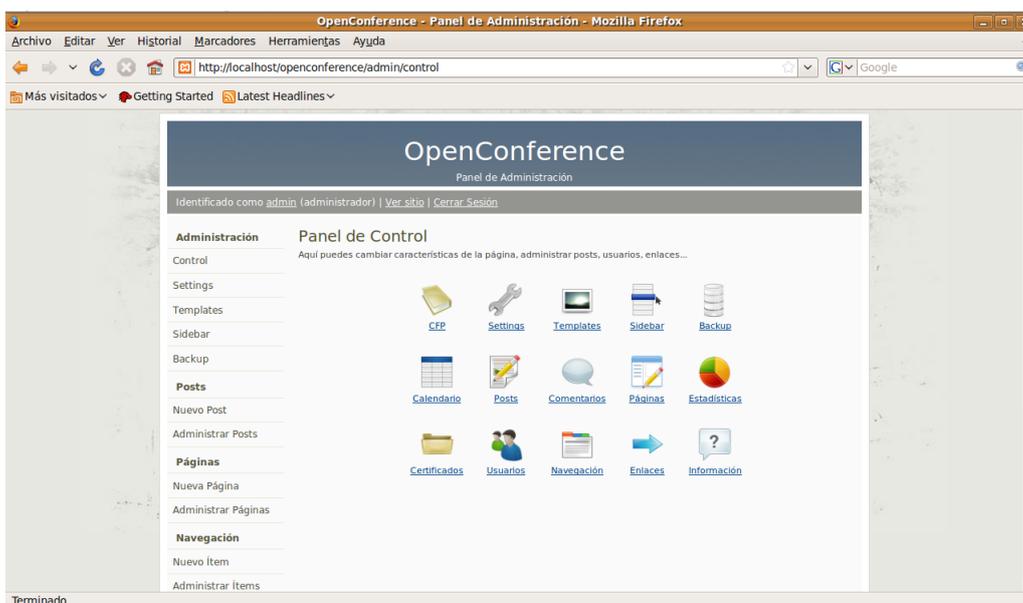


Figura 3.1: Panel de Administración

Durante la fase de pruebas simplemente nos aseguramos de que el panel cumple la función que debe realizar, que no es otra que aportar sencillez y claridad al usuario, que este sea capaz de gestionar todo un sistema completo con apenas unos clicks.

3.1.3. Incremento 3: Sistema de Noticias y Comentarios

Esta vez centraremos nuestra atención en el sistema para crear noticias y sus posibles comentarios. En el sistema únicamente podrá poner noticias el administrador, ya que lo que se pretende es que estas noticias se usen únicamente para informar a los demás usuarios sobre eventos, fechas o cualquier otro asunto importante.

Los comentarios sí podrán ser creados por cualquier usuario, siempre que el administrador permita que dicha noticia pueda recibir comentarios.

En el panel de administración, la sección para poder crear, editar o eliminar noticias recibe el nombre de Posts. De la misma forma la sección para administrar los comentarios, como ya se habrá podido pensar, recibe el nombre de Comentarios.

A continuación observamos la vista que tiene el administrador del sistema de todos los posts publicados.

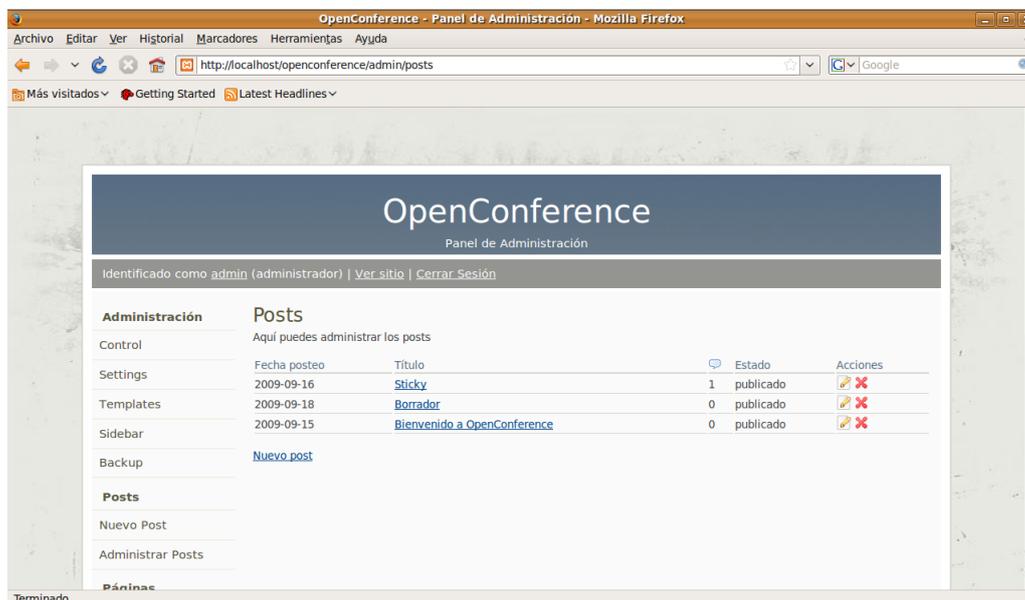


Figura 3.2: Listado de posts

Y en esta otra imagen vemos cómo se crea un nuevo post.

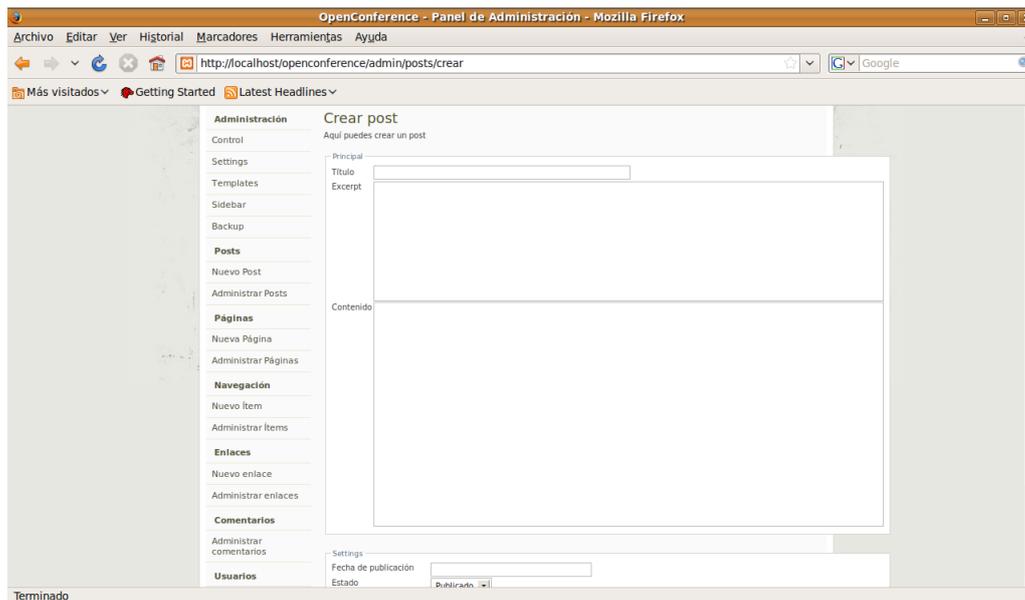


Figura 3.3: Creando nuevo post

En la fase de pruebas simplemente nos cercioramos de que tanto las noticias como los comentarios se crean correctamente, siendo sencilla la forma de introducirlos, que se editan con facilidad y de forma correcta, y que se eliminan de la misma forma.

3.1.4. Incremento 4: Sistema de Usuarios

En este incremento creamos la sección para administrar a los usuarios que se den de alta en nuestro sistema.

El administrador podrá eliminar a cualquier usuario (salvo a él mismo), también podrá editar sus datos, tal como el nombre de usuario, sus datos personales, el nivel de acceso al sistema (usuario registrado, revisor, administrador), etc...

En la siguiente imagen observamos la vista de edición de usuarios, desde el panel de administración.

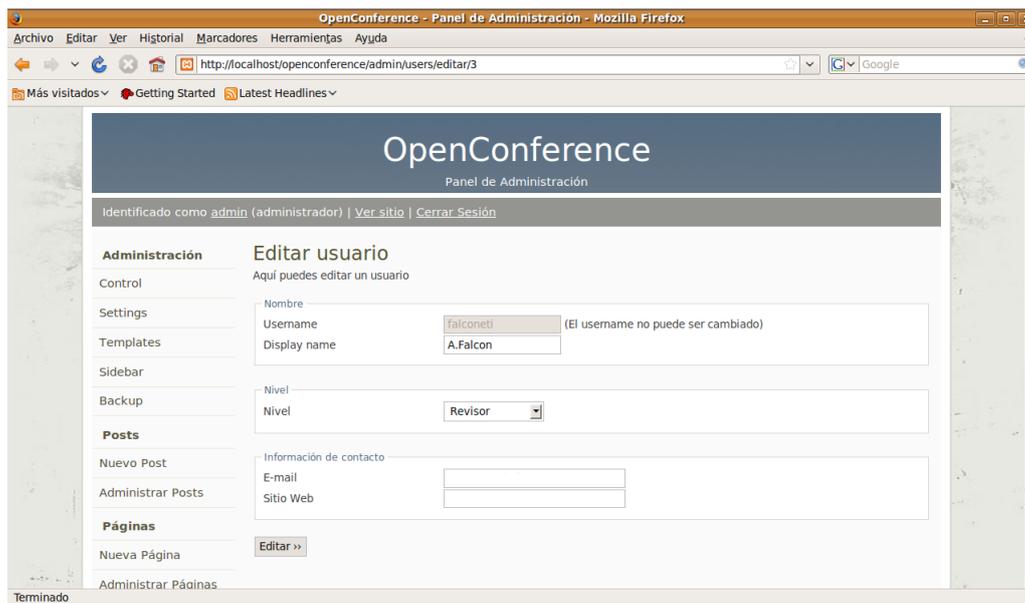


Figura 3.4: Edición de Usuario

Una vez en la fase de pruebas nos aseguramos de que todas las acciones funcionan correctamente, que se pueden eliminar usuarios del sistema de forma rápida y sencilla, y que se pueden editar sus datos y aportarles una categoría mayor para que puedan acceder a zonas inicialmente restringidas para ellos, como puede ser el caso del nivel Revisor, que tiene acceso a los CFP para revisarlos y valorarlos, o el nivel Administrador, que tiene acceso a todas las funciones del sistema.

3.1.5. Incremento 5: Sistema de Templates

En esta sección simplemente daremos una funcionalidad extra al sistema; la posibilidad de que el usuario elija entre diferentes templates puestos a su disposición, para personalizar así su página.

Para usuarios más avanzados que quieran aportar su propio diseño a la página, incorporamos en el apéndice de este documento un manual para desarrolladores, donde se explica cómo adaptar un diseño personal para que funcione con **OpenConference**.

Y aquí vemos el menú para elegir entre los distintos diseños.

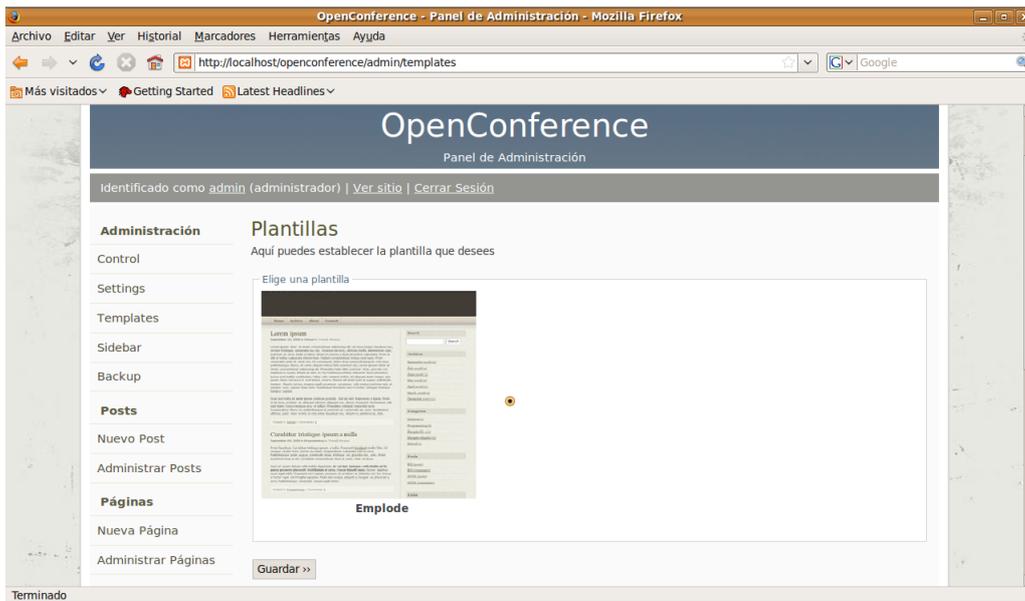


Figura 3.5: Elección de Template

Durante la fase de pruebas comprobamos que se puede cambiar con facilidad de diseño, y que estos se muestran de forma correcta en cualquier navegador web.

3.1.6. Incremento 6: Sistema de configuración

Esta vez nos centraremos en el sistema que nos permitirá cambiar la configuración que inicialmente le dimos a la aplicación. Podremos cambiar aspectos como, el título de la web, los datos del administrador, si están permitidos los registros, si la página está activa o no y su motivo, etc...

En el panel de administración de **OpenConference**, la sección dedicada a la configuración global recibe el nombre de Settings, y tiene la siguiente apariencia.

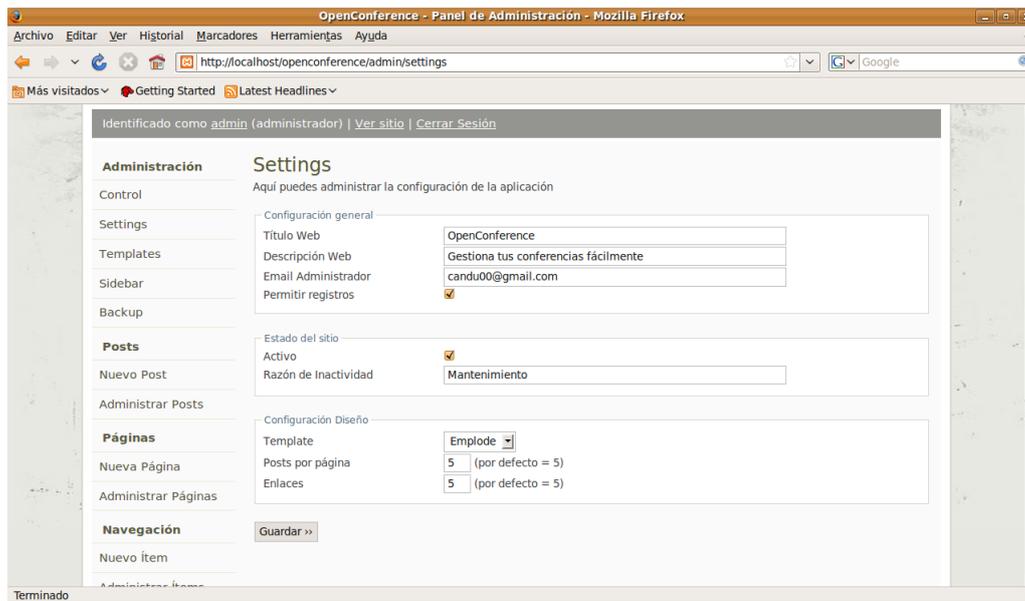


Figura 3.6: Configuración global del sistema

Como en los demás incrementos, durante la fase de prueba nos aseguraremos de que el sistema hace lo que debe, es decir, presenta correctamente los datos de la aplicación, y permite editarlos de forma sencilla.

3.1.7. Incremento 7: Sistema de Páginas y Navegación

Una de las funcionalidades más importantes de OpenConference es conseguir que el usuario pueda personalizar su página por completo, tanto en diseño como en contenido. De esta manera, hemos creado un sistema para permitir que el usuario añada cuantas páginas desee, y que automáticamente se añadan al menú de navegación de la página.

Aquí vemos una demostración de lo dicho anteriormente.

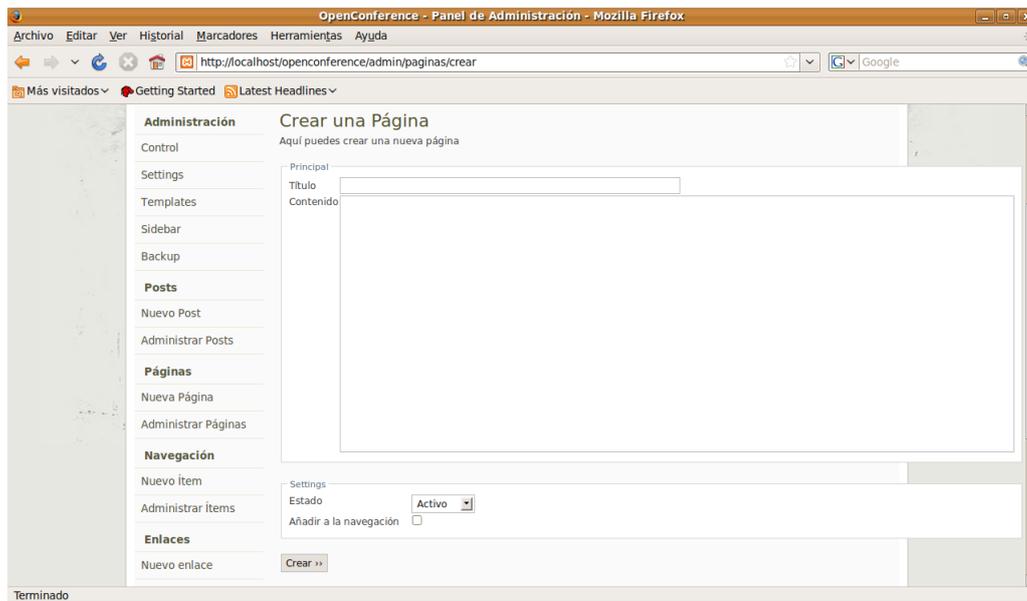


Figura 3.7: Creación de nueva página

En este incremento desarrollamos este sistema, y durante la fase de pruebas comprobamos que el usuario pueda añadir una nueva sección a su página con total facilidad, y que el contenido se refleje sobre la página correctamente, independientemente del diseño que en ese momento tenga la página.

3.1.8. Incremento 8: Panel lateral y enlaces

En este incremento creamos un sistema para visualizar un panel lateral, que podrá ser ocultado en cualquier momento por el administrador de la página.

En este panel lateral, existe una sección para acceder al panel de administración, así como para registrarse o loguearse en el sistema.

Y también disponemos de una sección donde el administrador podrá poner enlaces a las webs que considere oportunas, o incluso desactivar por completo el panel de enlaces, sin mostrar ninguno.



Figura 3.8: Administración del panel lateral

En la fase de pruebas comprobamos que los paneles se activan y desactivan correctamente, y que los enlaces se crean y funcionan bien.

3.1.9. Incremento 9: Sistema de revisión por pares

Este incremento es con el que más tiempo nos hemos llevado, ya que es uno de los pilares sobre los que se asienta **OpenConference**.

En este sistema concentramos todo el proceso de revisión y valoración de los CFP enviados por los usuarios. Es uno de los puntos más importantes a la hora de organizar conferencias, ya que en base a los artículos e investigaciones enviadas por los usuarios se creará el calendario de ponencias que compromerá el congreso.

La siguiente imagen muestra el proceso de revisión de un artículo.

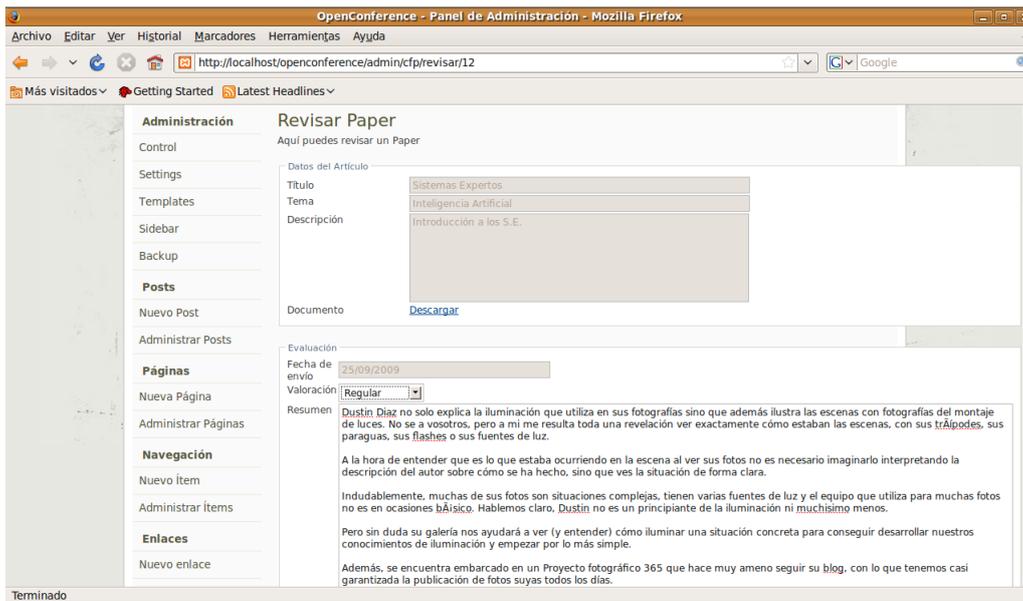


Figura 3.9: Etapa de revisión de un paper

Aquí, en la fase de pruebas, comprobamos que los usuarios pueden enviar correctamente los CFP, y que los revisores pueden acceder a ellos, revisarlos y dar una valoración sobre el mismo, para que luego sea el organizador del congreso el que, valorando esta información, los acepte o los rechace.

3.1.10. Incremento 10: Sistema de creación de calendario

En este incremento nos centramos en crear un sistema para que el usuario pueda crear el calendario de exposiciones del congreso de forma sencilla. En base a los artículos aceptados por el organizador para que sean expuestos en la conferencia, se le permitirá al administrador ir colocando cada ponencia en el día y hora que considere oportuno.

Durante la fase de pruebas comprobamos que todo funciona correctamente, y que no se hace demasiado pesado el proceso de elaboración del calendario.

3.1.11. Incremento 11: Sistema de creación de certificados

Normalmente, en la mayoría de los congresos que se organizan, cuando acaban las jornadas se concede a los participantes y a los ponentes un certificado que acredite su asistencia al mismo.

Desde OpenConference también será posible crear certificados personalizados al congreso en cuestión, que serán impresos y entregados a todos los participantes.

En este incremento nos dedicamos a crear el sistema que permita realizar esta tarea de forma rápida y sencilla, para que en cuestión de minutos tengamos generados todos los certificados de cada uno de los que participaron en el congreso.

Durante la fase de prueba nos cercioramos de que todo funcione a la perfección, y que el usuario sea capaz de crear su propio certificado sin mayores dificultades.

3.1.12. Incremento 12: Sistema de Backup

Hemos considerado conveniente crear un sistema para respaldar la información almacenada en nuestra base de datos ya que esta es crucial para que todo nuestro sistema funcione de forma adecuada.

En este incremento creamos el sistema de Backup, mediante el cuál, a través de un simple click de ratón, el usuario podrá generar y descargar una copia de seguridad de la base de datos. Esto es importante también si, por ejemplo, el administrador quiere realizar una migración a otro servidor sin perder la información que está contenida en la página.

En la fase de pruebas comprobamos que se genera correctamente el archivo .sql con la copia de seguridad, y que podemos descargarlo sin problemas a nuestro PC.

3.1.13. Incremento 13: Sistema de Información y Estadísticas

En este incremento se ha creado un sistema para generar información general sobre la aplicación, su creador, documentación del proyecto, etc...así como estadísticas varias.

La funcionalidad básica de este sistema es la de informar al usuario, aportar ayuda sobre cualquier aspecto relacionado con la aplicación.

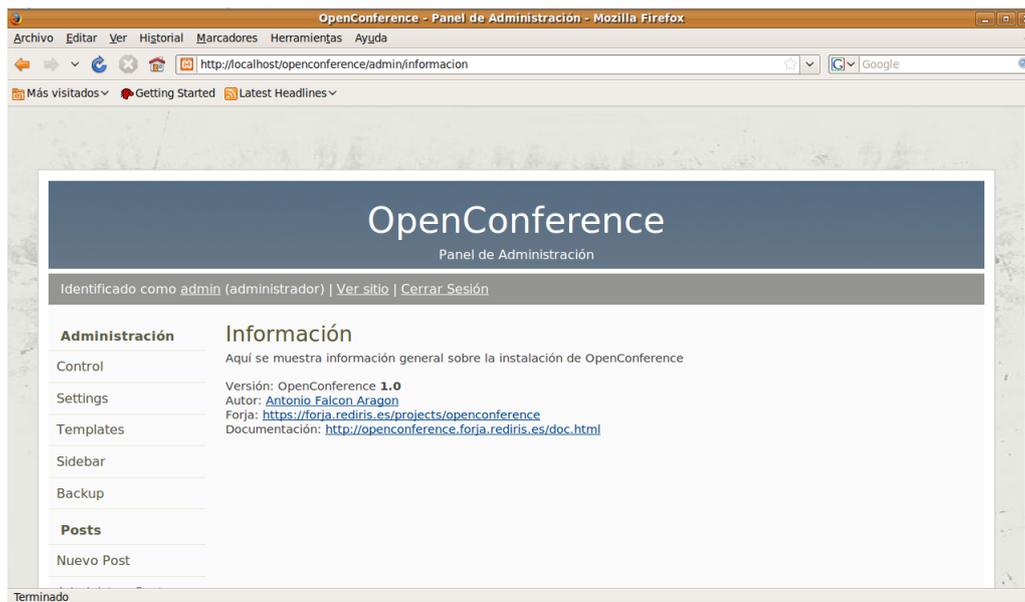


Figura 3.10: Información relativa al sistema, autor, documentación...

En la fase de pruebas nos aseguramos que el usuario puede acceder correctamente a esta información, así como a la documentación del proyecto, etc...

3.1.14. Incremento 14: Template 1 (Emplode)

Como ya dijimos anteriormente, **OpenConference** permite que el usuario seleccione entre distintas plantillas, para personalizar así la apariencia de su página.

A pesar de que cualquier usuario puede crearse su propio diseño, siguiendo las indicaciones del manual del desarrollador disponibles en este mismo documento, vamos a proporcionarle al usuario la posibilidad de elegir entre tres diseños iniciales (perfectamente ampliables en el futuro).

En este incremento, realizaremos las acciones necesarias para adaptar un primer diseño sacado de la web Arcsin[2], con el fin de que se acople perfectamente con **OpenConference**. El diseño es el siguiente.



Figura 3.11: Plantilla de diseño «Emplode»

Cabe destacar que tanto el presente diseño como los dos que vienen a continuación, y también el del panel de administración cumple con el estándar del W3C. Para validar el código hemos utilizado el servicio de validación (<http://jigsaw.w3.org/css-validator/>). El código HTML del resto de la aplicación ha sido validado a través de <http://validator.w3.org>.

3.1.15. Incremento 15: Template 2 (Vector Lover)

En este incremento realizaremos los mismos pasos que en el Incremento 14, pero esta vez escogiendo otro diseño, también de la web Arcsin.



Figura 3.12: Plantilla de diseño «Vector Lover»

3.1.16. Incremento 16: Template 3 (Colorvoid)

Y por último, adaptaremos un tercer diseño, para que se acople perfectamente con nuestra aplicación y aporte un nuevo diseño, una nueva forma de presentar los datos, ya que el contenido, la información, será siempre la misma, sea cual sea el diseño escogido.

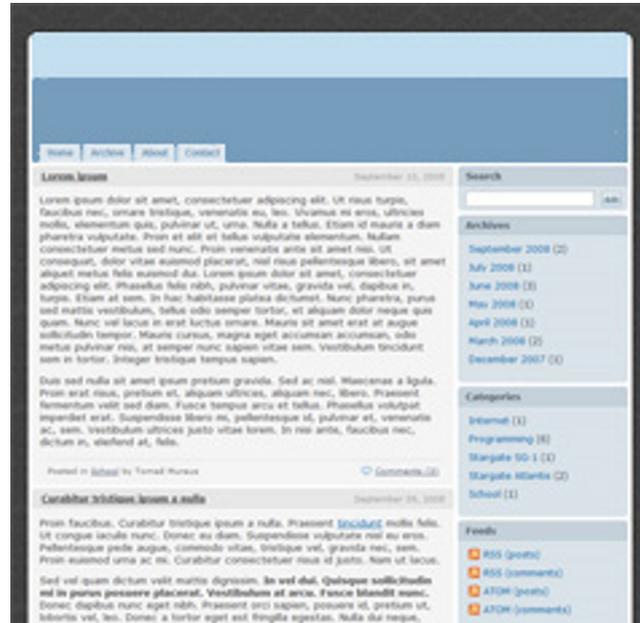


Figura 3.13: Plantilla de diseño «Colorvoid»

3.2. Diagrama de Gantt

A continuación mostraremos el diagrama de Gantt, generado con la herramienta *Planner*, donde se puede observar la planificación temporal del proyecto, el tiempo total de desarrollo del mismo, el tiempo empleado en cada incremento, así como las tareas que se han ido realizando en cada uno de esos incrementos.

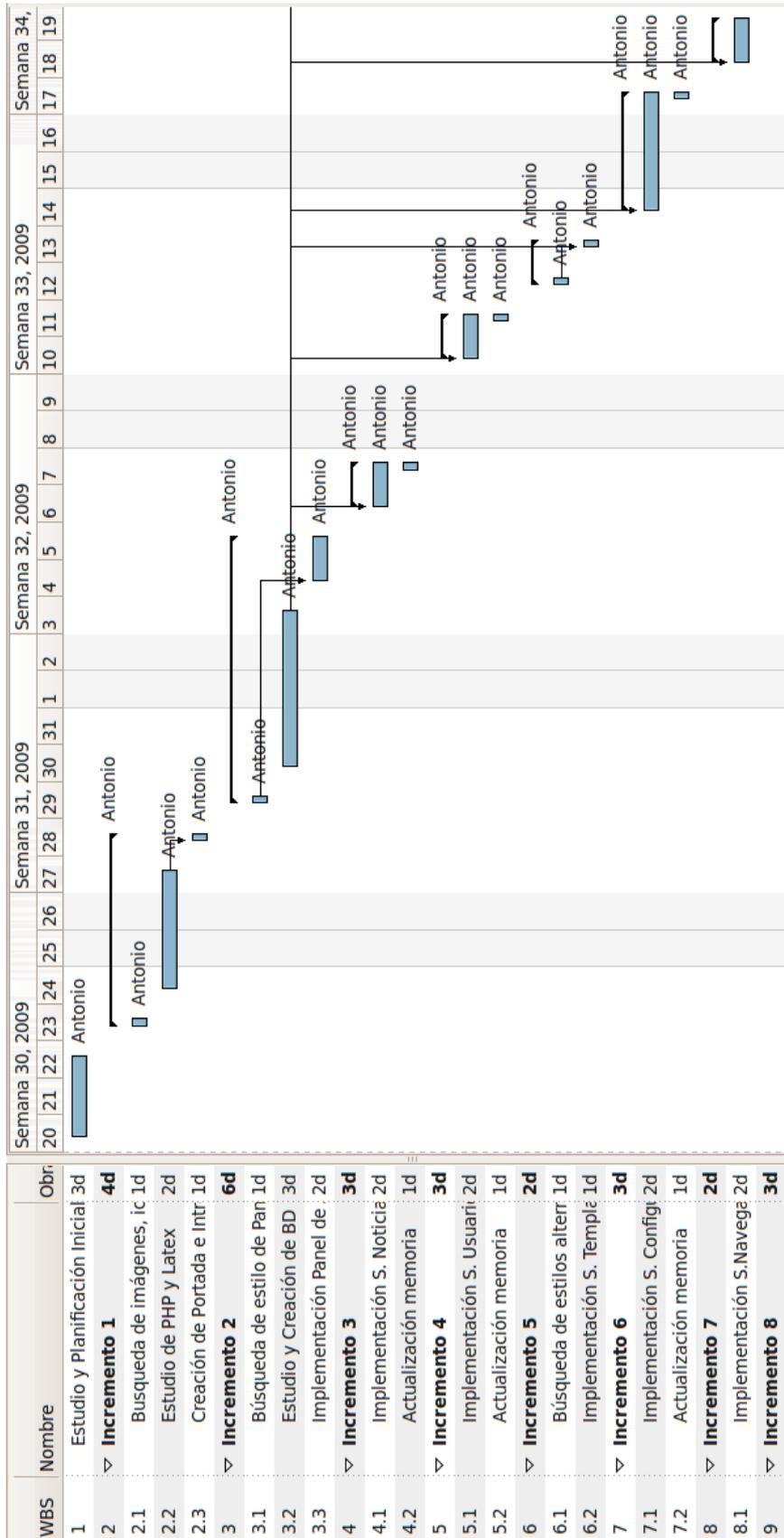


Figura 3.14: 1ª Parte del Diagrama de Gantt

1	Estudio y Planificación Inicial	jul 20	jul 22	3d	3d	55d	0	Antonio
2	Incremento 1	jul 23	jul 28	4d	4d	51d	0	Antonio
2.1	Busqueda de imágenes, iconos, etc...	jul 23	jul 23	1d	1d	54d	0	Antonio
2.2	Estudio de PHP y Latex	jul 24	jul 27	2d	2d	51d	0	Antonio
2.3	Creación de Portada e Introducción de memoria	jul 28	jul 28	1d	1d	51d	0	Antonio
3	Incremento 2	Jul 29	ago 5	6d	6d	45d	0	Antonio
3.1	Búsqueda de estilo de Panel de Adm.	jul 29	jul 29	1d	1d	48d	0	Antonio
3.2	Estudio y Creación de BD	jul 30	ago 3	3d	3d	44d	0	Antonio
3.3	Implementación Panel de Adm.	ago 4	ago 5	2d	2d	45d	0	Antonio
4	Incremento 3	ago 6	ago 7	3d	2d	43d	0	Antonio
4.1	Implementación S. Noticias	ago 6	ago 7	2d	2d	43d	0	Antonio
4.2	Actualización memoria	ago 7	ago 7	1d	1d	43d	0	Antonio
5	Incremento 4	ago 10	ago 11	3d	2d	41d	0	Antonio
5.1	Implementación S. Usuarios	ago 10	ago 11	2d	2d	41d	0	Antonio
5.2	Actualización memoria	ago 11	ago 11	1d	1d	41d	0	Antonio
6	Incremento 5	ago 12	ago 13	2d	2d	39d	0	Antonio
6.1	Búsqueda de estilos alternativos	ago 12	ago 12	1d	1d	39d	0	Antonio
6.2	Implementación S. Templates	ago 13	ago 13	1d	1d	39d	0	Antonio
7	Incremento 6	ago 14	ago 17	3d	2d	37d	0	Antonio
7.1	Implementación S. Configuración	ago 14	ago 17	2d	2d	37d	0	Antonio
7.2	Actualización memoria	ago 17	ago 17	1d	1d	37d	0	Antonio
8	Incremento 7	ago 18	ago 19	2d	2d	35d	0	Antonio
8.1	Implementación S.Navegación	ago 18	ago 19	2d	2d	35d	0	Antonio
9	Incremento 8	ago 20	ago 21	3d	2d	33d	0	Antonio
9.1	Implementación Panel Lateral	ago 20	ago 21	2d	2d	33d	0	Antonio
9.2	Actualización memoria	ago 21	ago 21	1d	1d	33d	0	Antonio
10	Incremento 9	ago 24	ago 27	4d	4d	29d	0	Antonio
10.1	Implementación S. Revisión Articulos	ago 24	ago 27	4d	4d	29d	0	Antonio
11	Incremento 10	ago 28	sep 2	4d	4d	25d	0	Antonio
11.1	Implementación S. Calendario	ago 28	sep 1	3d	3d	26d	0	Antonio
11.2	Actualización memoria	sep 2	sep 2	1d	1d	25d	0	Antonio
12	Incremento 11	sep 3	sep 7	3d	3d	22d	0	Antonio
12.1	Implementación S. Certificados	sep 3	sep 7	3d	3d	22d	0	Antonio
13	Incremento 12	sep 8	sep 9	2d	2d	20d	0	Antonio
13.1	Implementación S. Backup	sep 8	sep 9	2d	2d	20d	0	Antonio
14	Elaboración resto de memoria	sep 10	oct 7	20d	20d	0	0	Antonio
15	Depuración y Pruebas	sep 10	sep 23	10d	10d	10d	0	Antonio

Tabla 3.1: Tabla de tareas resultantes en el Diagrama de Gantt

3.3. Esfuerzo

Cabe destacar que gran parte del tiempo se ha empleado en aprender a usar las distintas herramientas con las que hemos desarrollado tanto la aplicación, como el presente documento. Sólo tenía conocimientos básicos de HTML y CSS, pero nunca había programado en PHP, con lo cuál tuve que buscar mucha información y aprender a programar en un lenguaje nuevo para mí. En esta tarea me ayudaron libros como “PHP, MySQL y Apache”[4] y “Desarrollo Web con PHP y MySQL”[5].

El mismo problema se me presentó con el framework *CodeIgniter*, ya que no conocía prácticamente nada de él. De gran ayuda ha sido la guía de usuario, presente en la web de CodeIgniter[6].

Tampoco había utilizado nunca el lenguaje $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ para preparar un documento, con lo cuál para usarlo correctamente me apoyé en libros como “El libro de Latex”[7], y en la wiki de Latex[8].

Y a parte de todo el proceso de aprendizaje, la tarea que más trabajo me costó realizar fue el sistema de revisión por pares, ya que inicialmente no tenía mucha idea de cómo se organizaba este proceso en la vida real, y tuve que realizar muchas modificaciones hasta conseguir adaptarlo a los requisitos iniciales de la aplicación.

Capítulo 4

Análisis

4.1. Metodología de Desarrollo

En este proyecto hemos seguido una metodología de desarrollo UWE UML (UML-Based Web Engineering).¹ Se trata de una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado.

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario.

Otras características relevantes del proceso y método de autoría de UWE son el uso del paradigma orientado a objetos, su orientación al usuario, la definición de un meta-modelo (modelo de referencia) que da soporte al método y el grado de formalismo que alcanza debido al soporte que proporciona para la definición de restricciones sobre los modelos.

Los principales aspectos en los que se fundamenta UWE son los siguientes:

- **Uso de una notación estándar**, para todos los modelos (UML: Lenguaje de modelado unificado).
- **Definición de métodos**: Definición de los pasos para la construcción de los diferentes modelos.
- **Especificación de Restricciones**: Se recomienda el uso de restricciones escritas (OCL: Lenguaje de restricciones de objetos) para aumentar la exactitud de los modelos.

Por lo que respecta al proceso de autoría de la aplicación, UWE hace un uso exclusivo de estándares reconocidos como UML y el lenguaje de especificación de restricciones asociado OCL. Para simplificar la captura de las necesidades de las aplicaciones web, UWE propone una extensión que se utiliza a lo largo del proceso de autoría. Este proceso de autoría está dividido en cuatro pasos o actividades:

- **Análisis de Requisitos**: Fija los requisitos funcionales de la aplicación Web para reflejarlos en un modelo de casos de uso.

¹«UWE UML» es una herramienta para modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización.

- **Diseño Conceptual:** Materializado en un modelo de dominio, considerando los requisitos reflejados en los casos de uso.
- **Diseño Navegacional:** Aquí podemos destacar; el Modelo del Espacio Navegacional y el Modelo de la Estructura de Navegación.
- **Diseño de Presentación:** Representa las vistas del interfaz del usuario mediante modelos estándares de interacción UML.

4.2. Análisis del sistema

La fase de análisis del sistema se lleva a cabo con el fin de crear un **Documento de Especificación de Requisitos**. En él se plasman los resultados de nuestro análisis a modo de contrato, con una serie de requisitos que deberá cumplir la aplicación.

Durante esta fase se han tenido en cuenta algunos requisitos que consideramos fundamentales, con el fin de incrementar la probabilidad de éxito de implantación de nuestra aplicación. Podemos estructurar dichos requisitos de la siguiente manera:

- **Portabilidad.** Debido a la dinamicidad del entorno tecnológico, a menudo es necesario implantar una misma aplicación en distintas plataformas, con distintas arquitecturas, con distintas tecnologías y/o atendiendo a distintos dispositivos de acceso, lo que obliga a desarrollar técnicas, modelos y herramientas que faciliten la reutilización e independicen, hasta donde sea posible, el desarrollo de la aplicación.
- **Inmediatez (Rapidez de Implantación).** El desarrollo de aplicaciones web requiere un período de implantación mucho más reducido, que influye en todo su ciclo de desarrollo.
- **Evolución Orgánica.** Es un aspecto fundamental en el ámbito de la web, donde tanto el contenido como los requisitos de las aplicaciones evolucionan a una velocidad vertiginosa. Esto es en parte debido a que los clientes de este tipo de aplicaciones suelen tener un conocimiento muy pobre de sus necesidades y de las posibilidades del sistema.
- **Seguridad en la comunicación.** Debido a que las aplicaciones web se encuentran disponibles a través de una red, es difícil limitar el grupo de usuarios finales que pueden acceder a ella. Es por ello que se hacen necesarios mecanismos para proteger información sensible y proporcionar modos seguros de transmisión de datos.
- **Calidad.** La permisividad mostrada por los usuarios ante los errores en aplicaciones Web es muy limitada; enlaces erróneos o información desactualizada provocan la pérdida de usuarios de la aplicación. Es por ello que en el desarrollo de este tipo de aplicaciones es primordial disponer de mecanismos exhaustivos de control de calidad que minimicen las posibilidades de fracaso de la aplicación.
- **Velocidad.** El uso intensivo de la red provoca que la elección de protocolos de comunicación y el mantenimiento de una velocidad de acceso adecuada sean una parte clave de diseño de dichas aplicaciones.
- **Importancia de la Interfaz.** La necesidad de implementar interfaces de usuario más intuitivas, capaces de capturar la atención del usuario y facilitar el acceso a la información a aquellos que poseen una habilidad limitada en el uso de aplicaciones informáticas.

En las secciones posteriores, y para hacer más comprensible el modelo de desarrollo que sigue UWE, vamos a mostrar el modelado del subsistema más importante de nuestra aplicación, el Sistema de Revisión de Artículos.

4.3. Diagrama E-R

4.4. Definición de Actores

El propósito del sistema es apoyar el proceso de evaluación y selección de papers para una conferencia. Los actores que encontramos son:

- **Presidente (Administrador):** Es el responsable de programar la conferencia, determinar las sesiones que tendrá y los temas, el comité del programa y decidir los artículos aceptados o rechazados, aconsejado por los revisores. El presidente también va a definir los plazos de conferencia: revisión, envío y notificación.
- **Revisor:** Es el responsable de revisar los artículos que se le asignan.
- **Autor:** Envía los artículos para la conferencia, que son registrados con diferentes identificadores.

Las reglas que controlan las acciones que pueden realizar cada uno de los actores son las siguientes:

- Todos los actores deben registrarse en el sistema.
- Los revisores son validados por el presidente, pero deben estar pre-registrados.
- Los actores deben registrarse si quieren enviar un artículo.
- Un autor puede registrar a un coautor, en cuyo caso el coautor debe registrarse también para tener acceso al sistema.

4.5. Definición de Procesos

- **Envío de Artículos:** Todos los autores registrados pueden enviar un artículo. Para enviarlo deben indicar el título, una pequeña descripción y un tema, escogido entre un conjunto previamente determinado por el presidente.
El sistema, una vez que verifica que el usuario está registrado, le asigna un identificador (ID) al artículo, y le permite modificarlo o cambiarlo por uno nuevo hasta la fecha tope de presentación.
- **Asignación de artículos a revisores:** El sistema asigna papers a un revisor, el cuál es responsable de la evaluación del artículo. Cada vez que un revisor termina el proceso de revisión el sistema avisa, mediante email, al miembro asociado.
- **Revisión:** Un miembro del comité o revisor, debe revisar los artículos que se le han asignado, cada revisión obtiene una calificación del 1 al 5. El presidente tiene acceso durante todo el proceso de revisión a todos los artículos.
- **Artículos Aceptados/Rechazados:** Una vez llegada la fecha tope de la revisión, se cierra dicho proceso. En ese momento, a todos los miembros se les permite ver los papers y su respectiva revisión.
Después de la revisión, el presidente escoge los papers que se aceptarán y se rechazarán.

4.6. Diagrama de Casos de Uso

Siguiendo el proceso de desarrollo de software unificado aplicado a la Ingeniería Web, se proponen los casos de uso necesarios para capturar los requisitos del sistema. Ésta es una técnica que fuerza a definir quienes son los actores (usuarios) de la aplicación y ofrece una manera intuitiva de representar la funcionalidad y aplicación en cada uno de los actores.

Basado en la descripción textual de los requisitos del Sistema de Revisión de Artículos, podemos ver que los usuarios pueden actuar en los papeles siguientes:

- Presidente
- Revisor
- Autor/Coautor

4.6.1. Caso de Uso: Envío, modificación y comprobación de Papers por Autor

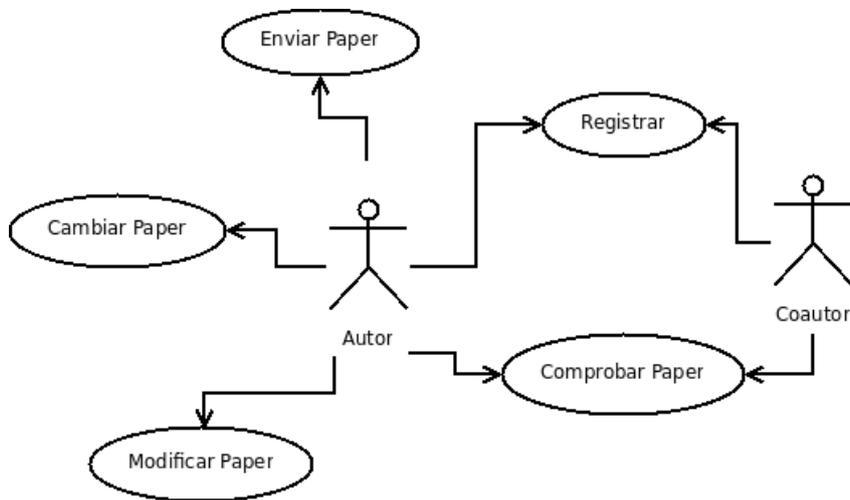


Figura 4.1: Diagrama de Caso de Uso para Funcionalidades de Autor

En el diagrama anterior podemos observar cómo el autor puede llevar a cabo cambios, modificaciones y comprobaciones sobre los artículos que ha enviado. También puede registrarse a sí mismo y compartir autoría con un coautor. El coautor debe registrarse también en el sistema.

4.6.2. Caso de Uso: Revisión de Papers por Revisor

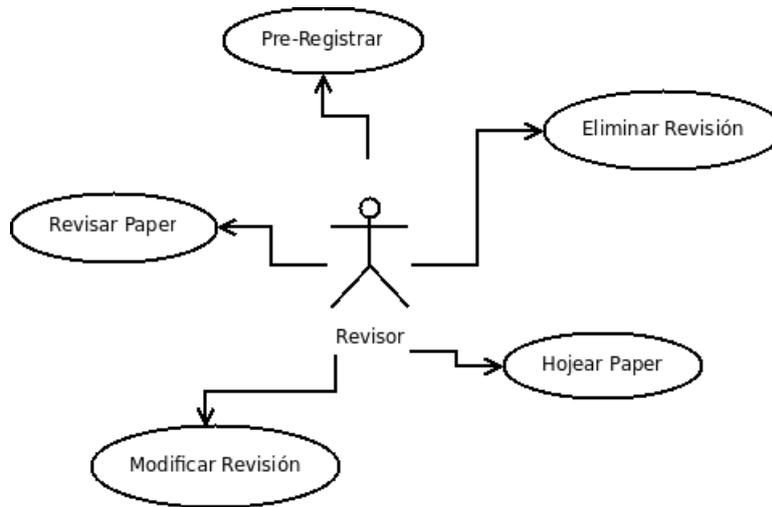


Figura 4.2: Diagrama de Caso de Uso para Revisión de Papers

En este otro diagrama se muestran las funciones que puede llevar a cabo el revisor de un artículo.

4.6.3. Caso de Uso: Gestión de Proceso de Revisión de Artículos por Presidente

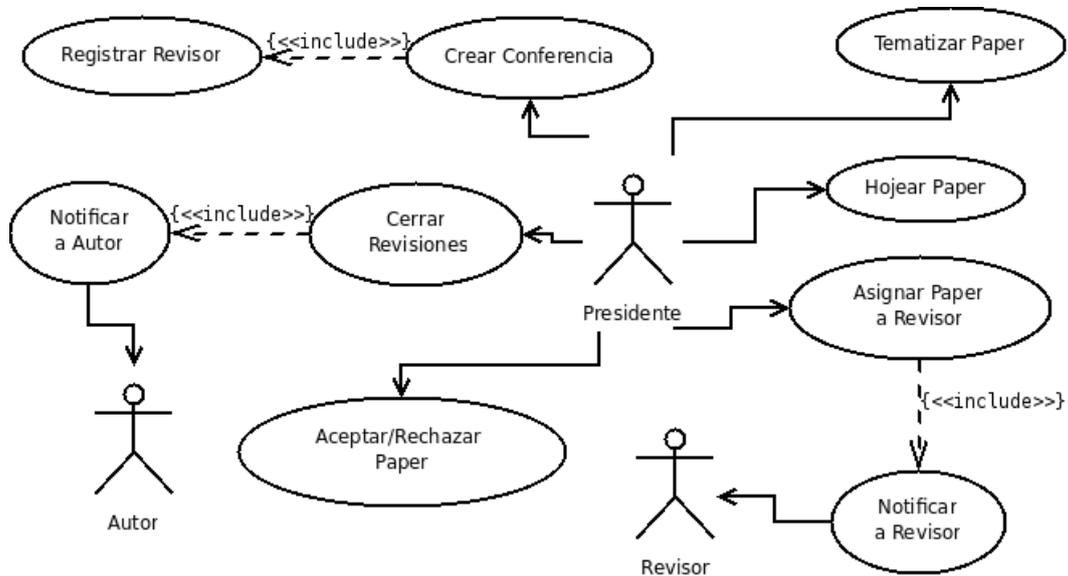


Figura 4.3: Diagrama de Caso de Uso para Gestión de Proceso de Revisión de Artículos

Por último, en este otro diagrama se nos muestra cómo el presidente gestiona todo el proceso que conlleva la revisión de artículos.

4.7. Representación UML del Modelado Conceptual

El diseño conceptual se base en el análisis de requisitos del pase anterior. Esto incluye los objetos involucrados entre los usuarios y la aplicación.

El modelo conceptual propone construir un modelo de clases con estos objetos, ignorando los aspectos de navegación, presentación e interacción. Los principales elementos de modelado son; las clases, asociaciones y paquetes.

En nuestro caso, vamos a seguir un proceso de modelado orientado a objetos, basado en UML. Los pasos que seguiremos son:

- Distinguir las clases, como Conferencia, Usuario, Paper y Revisión.
- Especificar los atributos más importantes y funcionamiento, como el título, ID, etc...
- Determinar las asociaciones entre las clases.
- Agregar las clases e identificar la composición de estas, como entre la clase Conferencia y TEma, y entre la clase Revisión y Evaluación...
- Definir las jerarquías de herencia, por ejemplo la jerarquía de roles de usuario.
- Definir las restricciones de los métodos.

4.7.1. Caso Práctico: Sistema de Revisión de Artículos

Baándonos en la descripción textual y en los casos de uso basados en el análisis de requisitos del paso anterior, identificaremos los objetos, relaciones y operaciones necesarias para construir el modelo conceptual del sistema.

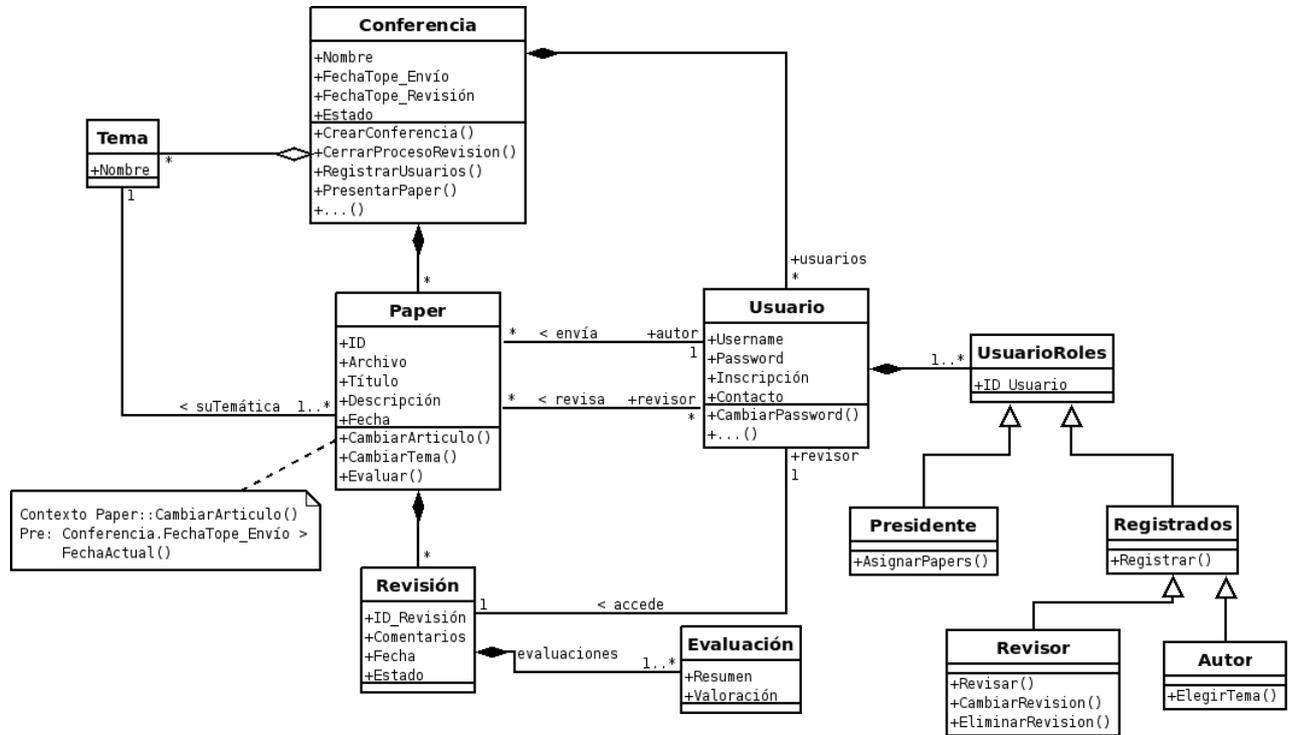


Figura 4.4: Diagrama de Clases

La clase conferencia contiene la información sobre el nombre de la misma y las fechas tope de la conferencia. El estado de la conferencia es un atributo cuyo valor cambia cuando se alcanza la fecha tope de inscripción o la fecha tope de revisión.

Un paper viene descrito por un título, un tema, una descripción y la fecha en la que se envió. Tiene asociado un archivo donde se almacena el artículo y una identificación generada por el sistema.

Además, tenemos una clase principal usuario, que va a gestionar los tipos de usuarios (presidente, revisor,...) y vamos a poderle asignar a cada tipo de usuario sus privilegios sobre el trato de los artículos.

4.8. Funcionalidades de la Aplicación

OpenConference está formado por muchos subsistemas, los cuáles realizan funciones muy concretas. Todos estos subsistemas se controlan desde el cerebro principal de la aplicación, el Panel de Administración. A continuación iremos describiendo en qué consiste cada uno de esos sistemas.

4.8.1. Panel de Administración

El pilar sobre el que se asienta OpenConference es su Panel de Administración. Su interfaz es sencilla y agradable, ya que la aplicación está dirigida a todo tipo de usuarios, y muchos de ellos no poseerán ningún conocimiento informático.

Cabe destacar que se ha elegido un diseño elegante y sin complejidades, manteniendo a la vista del usuario todas las funcionalidades más usuales e importantes de que dispone la aplicación. Desde aquí se gestiona toda la aplicación, por ello es importante que el usuario se sienta cómodo con el entorno y aprenda fácilmente a manejar todas sus funcionalidades.

4.8.2. Sistema de Noticias y Comentarios

Durante todo el período que dura el proceso de crear y administrar una conferencia, debemos destacar la importancia que tienen las noticias en este aspecto.

Por medio de las noticias, el presidente (administrador) del congreso avisará a todos los participantes sobre temas importantes durante el desarrollo de las jornadas. Como ejemplo, informar a los usuarios de fechas importantes, de cambios de última hora, etc...

Las noticias serán lo primero que el usuario vea al entrar en la aplicación. Tendrán un formato claro, ya que su función es informar con la mayor exactitud y rapidez posible.

Sólo el administrador puede poner noticias en el sistema, sin embargo, todos los usuarios podrán comentar cualquier noticia, siempre que el administrador haya habilitado los comentarios para la misma. El sistema permite postear tres tipos de noticias:

- Normales (Publicadas): Son las más comunes, se publican justo en el momento de ser posteada y ya pueden ser vistas por cualquier usuario.
- Borradores: El administrador escribe la noticia y selecciona una determinada fecha de publicación. La noticia no será visible hasta ese día concreto.
- Sticky: Son noticias muy importantes, casi de obligada lectura. Por tanto siempre estarán visibles en la primera posición. Este tipo de noticias aparecerán con una chincheta de color azul junto a su título.

Aquí vemos un ejemplo de lo comentado anteriormente, haciendo uso del diseño **Emplode** para representar todo el contenido.

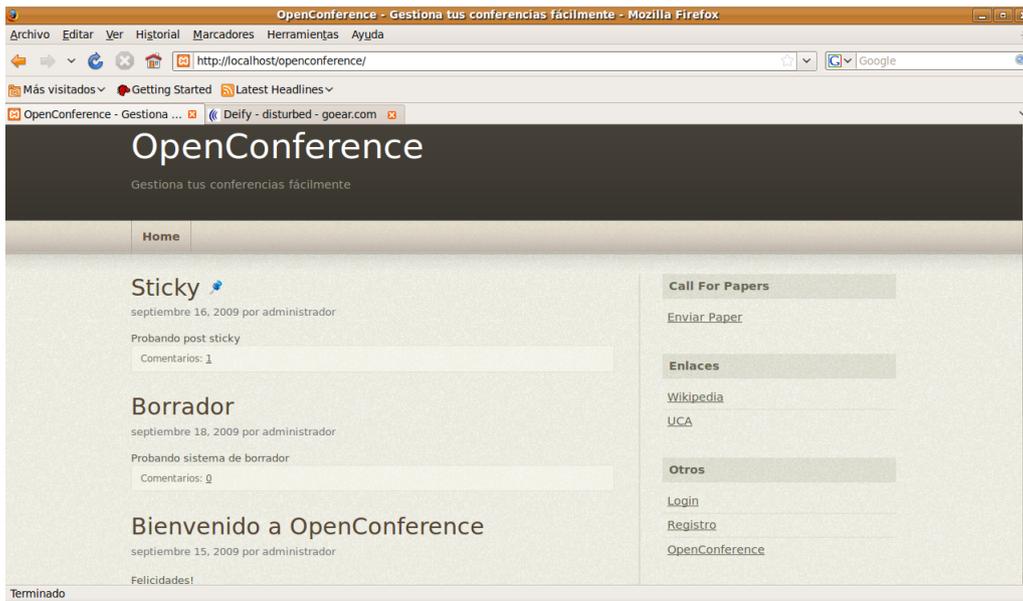


Figura 4.5: Captura de noticias publicadas en el sistema

4.8.3. Sistema de control de Usuarios

Es importante conocer de primera mano a todos los usuarios que forman parte de nuestra aplicación. Mediante este sistema obtendremos de forma inmediata una lista con todos ellos.

Algunas de las funciones que podremos realizar con los usuarios son las siguientes:

- Eliminar Usuario
- Cambiar información básica de usuario: username, password...
- Cambiar Nivel de Accesibilidad al sistema: Usuario, Revisor o Administrador. Esto es importante ya que desde aquí nombraremos a los revisores, que previamente han debido registrarse como usuarios en el sistema.

4.8.4. Sistema de Templates

OpenConference es una herramienta que ofrece muchas posibilidades de personalización. Entre ellas, es posible elegir entre varios diseños según las preferencias de estilo que tenga el usuario.

Inicialmente se ofrece al usuario tres diseños a elegir (ampliables en el futuro). Incluso el propio usuario puede aportar sus propios diseños, ya que OpenConference respeta todos los estándares. Por tanto el usuario puede crearse su propio diseño y, con unos sencillos cambios, adaptarlo a OpenConference.

El aspecto más importante de esta funcionalidad es el hecho de disponer de distintas representaciones de los datos, manteniendo siempre el mismo contenido, la misma información.

4.8.5. Sistema de Configuración

Cuando se instala la aplicación por primera vez es necesario aportar ciertos datos tales como el título de la web, una descripción, información sobre el administrador principal del sitio, la posibilidad de permitir o no registros en la web, etc...

Todas estas opciones estarán luego disponibles en la pestaña de Configuración del sistema. Desde aquí se podrá editar cualquiera de estas características, o incluso se podrá poner la Web como inactiva y dar algunos motivos, como por ejemplo, tareas de mantenimiento, etc... También se controlarán aspectos como el número de noticias por página, o el número de enlaces por panel.

Todos estos aspectos hacen que OpenConference sea totalmente configurable, incluso durante su instalación, y hacen que la administración de la Web se pueda realizar de forma sencilla y sin amplios conocimientos en la materia.

4.8.6. Sistema de Páginas y Navegación

OpenConference da la posibilidad al usuario de añadir cuantas pestañas desee, con el fin de que sea el propio usuario el que decida qué páginas tener y qué información representar en cada momento.

Con estos dos sistemas será posible realizar las tareas anteriores. No todas las Webs de los distintos congresos tienen las mismas páginas de información. Si bien es cierto que la mayoría del contenido es prácticamente similar, los administradores de cada Conferencia se encargan de añadir su toque personal, para así distinguirse de las demás.

Por esta razón es necesario disponer de un mecanismo para que el usuario añada cuantas páginas desee y ponga la información que considere oportuna. Con estos sistemas podrá generar un menú de páginas totalmente personalizado.

4.8.7. Panel Lateral y Enlaces

Siempre es útil poder poner todos los enlaces que se considere oportuno, con el fin de ayudar al usuario que visita la web a que amplie mucho más la información que ya se le da en nuestra web. Por ello, OpenConference pone a disposición del administrador un sistema para visualizar, en el Panel Lateral derecho de la web, una caja con enlaces a los sitios de interés que este haya considerado oportunos.

De nuevo sobresale el aspecto de la personalización de la web, uno de los principios más importantes de la aplicación, y que sin duda será agradecido por los usuarios que utilicen el sistema. A continuación vemos el ejemplo.

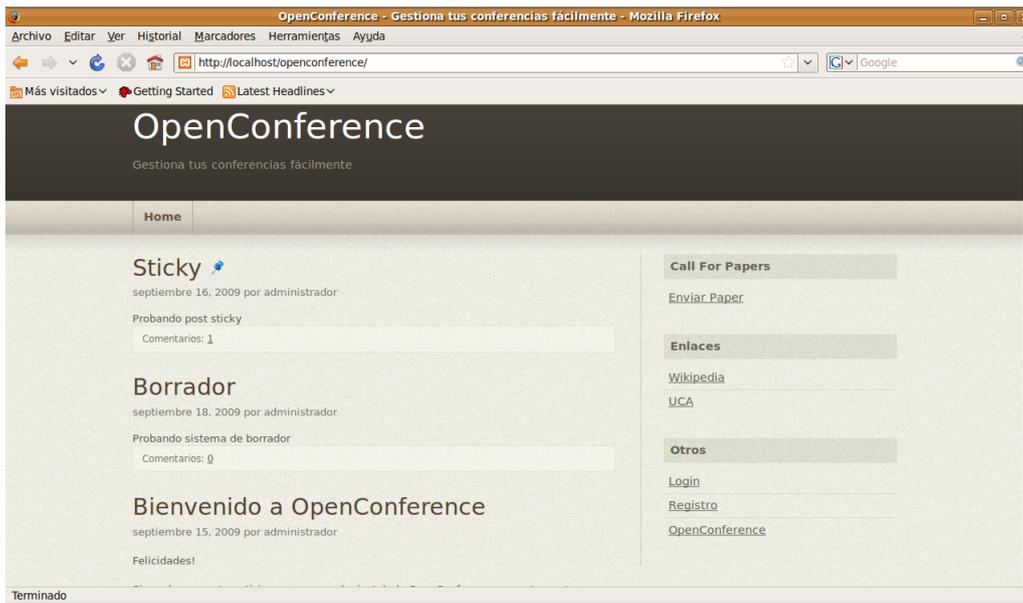


Figura 4.6: Captura de pantalla mostrando enlaces de interés

4.8.8. Sistema de Revisión por pares

Quizá sea este sistema uno de los más importantes de OpenConference, ya que la mayoría de los demás sistemas giran entorno a éste.

La función principal de este sistema es gestionar todo el proceso de revisión de papers. El proceso comienza cuando los autores mandan sus propuestas a la aplicación. El sistema acepta artículos hasta una determinada fecha tope, en la cuál comienza el período de revisión.

La revisión de los artículos la realizan unos usuarios destinados a tal fin, los revisores, que son nombrados por el presidente (administrador) del congreso.

Los revisores tienen acceso a los artículos enviados por los autores, se encargarán de leer los artículos y evaluarlos con una puntuación de 1 a 5. Además se les pedirá que aporten un resumen del trabajo leído, destacando los puntos fuertes y débiles del trabajo, originalidad, etc...

Una vez que termina este período de revisión, el administrador aceptará unos trabajos y rechazará otros. Los trabajos aceptados formarán parte del programa de la Conferencia.

4.8.9. Sistema de Calendario

Mediante este sistema se generará el calendario con las ponencias a realizar cada día del congreso, así como su hora determinada.

Este sistema permitirá al administrador ir colocando cada ponencia (habiendo aceptado previamente el paper de la misma) en un día y horario determinado.

De manera sencilla y siguiendo unos pocos pasos, en pocos minutos habrá generado el programa de la conferencia, donde aparecerá el título de la ponencia, quién la imparte, el día, la hora y el lugar de celebración.

4.8.10. Sistema de Certificación

Con este sistema el administrador podrá generar fácilmente todos los certificados, listos para imprimir, firmar y entregar a todos los participantes.

El usuario tan sólo tendrá que configurar algunos aspectos de presentación. Deberá añadir un título, un texto descriptivo del certificado, y si lo desea también puede añadir un fondo personalizado, que contenga un diseño del certificado. El resto de tareas las realizará la aplicación, que se encargará de generar un certificado por cada uno de los miembros que hayan asistido o participado en la conferencia.

Dichos certificados no tendrán validez si no se firman a mano por el responsable que se haya determinado, por tanto los certificados se generarán en formato .PDF para que puedan ser impresos con facilidad y sin ningún tipo de problemas.

4.8.11. Sistema de Backup

Este sistema otorga al administrador la posibilidad de generar copias de seguridad de todo el sistema, guardando toda la información que estuviera contenida en la aplicación hasta ese momento. Esto puede tener multitud de utilidades, por ejemplo, si el administrador desea migrar su aplicación a otro servidor le interesará mantener todos los datos que tenía hasta la fecha, por tanto, con este sistema podrá hacerlo sin ningún problema ya que la web se instalará en otro servidor cualquiera y seguirá manteniendo el mismo aspecto e información que anteriormente.

4.8.12. Sistema de Información y Estadísticas

Y por último, este es un pequeño sistema para generar información que puede ser de utilidad para el administrador del sistema. Se generan algunas estadísticas, como la noticia más comentada, etc...Y también información básica del sistema, y documentación del mismo para ayudar a solucionar cualquier tipo de problema que se le plantee durante su manejo o instalación.

Capítulo 5

Diseño

Durante la fase de Diseño nos centraremos en aplicar ciertas técnicas y principios con el propósito de definir nuestro Sistema con suficientes detalles como para permitir su interpretación e implementación.

Esta tarea de Diseño resulta sencilla, siempre que se tengan muy claros todos los requisitos extraídos en la etapa anterior de Análisis. El proceso de Diseño no es más que un conjunto de pasos repetitivos que nos van a permitir describir todos los aspectos del Sistema que estamos construyendo.

El diseño, fundamentalmente, debe implementar todos los requisitos explícitos contenidos en el modelo de análisis y debe acumular todos los requisitos implícitos que desea el cliente. Por tanto, debe ser una guía que puedan leer y entender los que construyen el código y los que prueban y mantienen el Software.

El diseño debe proporcionar una completa idea de lo que es el Software, enfocando los dominios de datos, funcional y comportamiento desde el punto de vista de la implementación.

No olvidemos que esta aplicación ha sido diseñada siguiendo el patrón **MVC (Modelo Vista Controlador)**¹. Al tratarse de una aplicación web, en nuestro caso, la **vista** es la página HTML y el código que provee de datos dinámicos a la página. El **modelo** es el Sistema de Gestión de Base de Datos, y el **controlador** es el responsable de recibir los eventos de entrada de la vista.

A continuación, y de forma breve, comentaremos las distintas funciones que componen cada uno de los **módulos controladores**, ya que son los que establecen el enlace entre las distintas capas del modelo, y por ello la importancia de describirlos.

5.1. Administración

En este apartado haremos referencia a todos aquellos módulos que forman parte del Panel de Administración. Cada uno de los módulos realiza una función diferente dentro del sistema de administración, pero todos se unen para el Panel desde donde se controla toda la aplicación. A continuación mostramos los detalles de cada uno.

¹«MVC» es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

5.1.1. Control

Módulo principal del Panel de Administración, desde aquí se hacen las llamadas a los demás módulos.

- **Control::index()**. Función que hace la llamada a la vista del Panel de Administración, donde se encuentran los enlaces a las demás herramientas.

5.1.2. Noticias

Módulo que contiene todas las funciones necesarias para el correcto funcionamiento del Sistema de Noticias.

- **Posts::index()**. Se encarga de llamar a la vista de Noticias, para obtener una lista de todas ellas.
- **Posts::crear()**. Presenta el formulario para introducir los datos de la noticia a crear.
- **Posts::editar(id)**. Muestra todos los datos de una noticia determinada (dada por su ID) para que puedan ser editados.
- **Posts::eliminar(id)**. Elimina de la lista una noticia determinada (dada por su ID).

5.1.3. Comentarios

Módulo con las funciones necesarias para el Sistema de Comentarios.

- **Comentarios::index()**. Llama a la vista de Comentarios, para obtener una lista de todos ellos.
- **Comentarios::editar(id)**. Muestra todos los datos de un comentario determinado para poder editarlos.
- **Comentarios::eliminar(id)**. Elimina el comentario dado de la lista de comentarios.

5.1.4. Usuarios

Módulo con las funciones necesarias para gestionar de manera adecuada a todos los usuarios dados de alta en el sistema.

- **Users::index()**. Llama a la vista de Usuarios para obtener una lista con todos los que están registrados en el sistema hasta la fecha.
- **Users::editar(id)**. Muestra todos los datos pertenecientes al usuario dado, para que puedan ser modificados.
- **Users::eliminar(id)**. Elimina el usuario dado de la lista de usuarios. Es como si se diera de baja al usuario.

5.1.5. Templates

En este módulo encontramos funciones para cambiar entre los distintos tipos de templates que hay disponibles.

- **Templates::index()**. Mediante esta función se selecciona, en vista de algunos parámetros prefijados, el tipo de plantilla que se va a elegir para el diseño de la Web.

5.1.6. Enlaces

Módulo que contiene todas las funciones necesarias para poner enlaces de interés en el panel lateral de la página.

- **Enlaces::index()**. Se encarga de llamar a la vista de Enlaces, para obtener una lista de todos ellos.
- **Enlaces::crear()**. Presenta el formulario para introducir los datos del enlace a crear.
- **Enlaces::editar(id)**. Muestra todos los datos de un enlace determinado (dado por su ID) para que puedan ser editados.
- **Enlaces::eliminar(id)**. Elimina de la lista un enlace determinado (dado por su ID).

5.1.7. Navegación

En este módulo nos encontramos con las funciones adecuadas para administrar el menú de navegación, añadiendo o quitando los items que se consideren oportuno.

- **Navegacion::index()**. Se encarga de llamar a la vista de Navegación, para obtener una lista de todos los items visibles.
- **Navegacion::crear()**. Presenta el formulario para introducir los datos del item de navegación a crear.
- **Navegacion::editar(id)**. Muestra todos los datos de un item de navegación determinado (dado por su ID) para que puedan ser editados.
- **Navegacion::eliminar(id)**. Elimina de la lista un item de navegación determinado (dado por su ID).

5.1.8. Panel Lateral

Módulo que contiene todas las funciones necesarias para poner enlaces de interés en el panel lateral de la página.

- **Sidebar::index()**. Se encarga de llamar a la vista del Panel Lateral, para obtener una lista con todos los elementos que forman parte del mismo.
- **Sidebar::activo(id)**. Activa el elemento dado, del Panel Lateral, para que sea visible a los usuarios.
- **Sidebar::inactivo(id)**. Desactiva el elemento dado, del Panel Lateral, para que esté oculto a los usuarios.
- **Sidebar::eliminar(id)**. Elimina de la lista un elemento determinado del Panel Lateral (dado por su ID).

5.1.9. Páginas

Módulo que contiene todas las funciones necesarias para crear páginas con contenido personalizado y agregarlas a la navegación.

- **Paginas::index()**. Se encarga de llamar a la vista de Páginas, para obtener una lista de todas ellas.

- **Paginas::crear()**. Presenta el formulario para introducir los datos y el contenido de la página a crear.
- **Paginas::editar(id)**. Muestra todos los datos de una página determinada (dada por su ID) para que puedan ser editados.
- **Paginas::eliminar(id)**. Elimina de la lista una página determinada (dada por su ID).

5.1.10. Settings

En este módulo encontramos funciones para mostrar todas las opciones de configuración que posee la aplicación.

- **Settings::index()**. Mediante esta función se muestra la página que contiene toda la información relativa a los parámetros de configuración más importantes de la aplicación.

5.1.11. Calendario

En este módulo nos encontramos con todas aquellas funciones necesarias para crear y generar el calendario de ponencias.

- **Calendario::index()**. Encargado de llamar a la vista de Calendario, para así obtener una lista de todas las ponencias que actualmente forman parte del programa, y un pequeño formulario donde ir añadiendo las demás ponencias.
- **Calendario::generar()**. Esta función es la encargada de recopilar toda la información de las distintas ponencias y crear el pdf con el horario.

5.1.12. Call For Papers

Módulo que contiene todas las funciones necesarias para administrar todo el proceso de revisión de papers.

- **Cfp::index()**. Se encarga de llamar a la vista de CFP, para obtener una lista de todos los artículos que se han enviado al sistema, independientemente del estado que tengan (recibido, en revisión, revisado).
- **Cfp::revisar(id)**. Presenta el formulario para que el revisor evalúe el paper seleccionado.
- **Cfp::eliminar(id)**. Elimina de la lista el paper seleccionado (dado por su ID).
- **Cfp::descargar(id)**. Permite descargar el contenido del artículo seleccionado (dado por su ID), para que el revisor lo vea y pueda valorarlo
- **Cfp::ver(id)**. Muestra los datos del paper seleccionado, así como la revisiones que se han hecho del mismo.

5.1.13. Certificados

En este módulo nos encontramos con todas las funciones necesarias para añadir plantillas de certificado y administrar las ya existentes.

- **Certificado::index()**. Esta función es la encargada de listar todos los certificados que hay disponibles en el sistema y mostrárselos al usuario para su elección. También presenta un formulario de envío de nuevos certificados.

- **Certificado::enviar()**. Mediante esta función podremos subir la plantilla con un nuevo certificado.
- **Certificado::crear(id)**. A partir de esta función se seleccionará la plantilla con el certificado por defecto y se generará el certificado personalizado de cada uno de los participantes.

5.2. Noticias

Dentro de este apartado haremos referencia a todas aquellas funciones necesarias para crear y mantener correctamente todo el contenido de la página principal de la Web.

- **Noticias::index()**. Esta función se encarga de listar todas las noticias, ordenadas por fecha de publicación.
- **Noticias::post(anno, mes, dia, urlTitulo)**. Muestra todos los datos de la noticia.
- **Noticias::comentario(id, url)**. Presenta un formulario para que los usuarios puedan escribir un comentario sobre la noticia seleccionada.

5.3. Páginas

En este otro apartado destacamos aquellas funciones que nos permiten añadir nuevos items al menú de navegación principal de la Web.

- **Pagina::pagina(urlTitulo)**. Mediante esta función lo que hacemos es conseguir la url de la página, para hacer que apunte a su contenido.

5.4. Usuario

Este vez presentamos todas las funciones que nos van a permitir tener un control de autenticidad de los usuarios que naveguen por nuestra web, permitiéndoles registrarse, identificarse, etc...En definitiva, manteniendo un control de acceso de usuarios.

- **Usuario::registro()**. Presenta el formulario de registro para que el usuario introduzca sus datos con el fin de darse de alta en la página.
- **Usuario::activacionCuenta()**. La función se encarga de mandar un mail al usuario con los datos del registro y un link de activación, que tendrá que visitar para que la cuenta sea operativa.
- **Usuario::passwordOlvidado()**. La función manda un mail al usuario recordándole sus datos de acceso a la aplicación.
- **Usuario::ver(nickname)**. Muestra los datos del usuario, su nick, su fecha de registro, su nivel de acceso, etc...
- **Usuario::perfil()**. Devuelve los datos del usuario para poder editarlos.
- **Usuario::login()**. La función verifica los datos del usuario y, según su nivel de acceso, le da entrada al sistema y le permite realizar unas tareas u otras.
- **Usuario::logout()**. La función cierra la sesión del usuario de manera segura, borrando todos los datos que haya podido utilizar el navegador.

5.5. Papers

En este apartado mostramos las funciones que nos van a permitir interactuar con el sistema y enviar los artículos que más adelante serán motivo de revisión para ver si son aceptados o no en la conferencia.

- **Cfp::enviar()**. Presenta el formulario para que el usuario envíe el artículo que quiere presentar en la conferencia. Deberá proporcionar algunos datos como el título del documento, una breve descripción, la temática, y adjuntar el documento pertinente.

5.6. Características de los usuarios finales

OpenConference ha sido diseñado con el objetivo de que cualquier usuario pueda usarlo con total facilidad, sin importar los conocimientos que este tenga. Se entiende que el usuario debe saber navegar por la web y manejar los periféricos del PC (teclado y ratón) de forma básica.

La aplicación va dirigida al público en general, ya que cualquier persona puede estar interesada en organizar alguna conferencia y desea hacerlo de forma rápida y sencilla, pero controlando perfectamente todas las etapas.

Quizá esto último es lo que más hace destacar a OpenConference, ya que existen otras aplicaciones que realizan labores parecidas, pero no con la misma sencillez, con lo cuál la mayoría de usuarios optan por no utilizar la aplicación y hacerlo todo manualmente. OpenConference se ha creado para solucionar este problema, entre otros, y para hacer que el proceso de organización de conferencias no sea tedioso y complicado.

Capítulo 6

Implementación

Durante la fase de implementación nos dedicaremos a codificar todo el sistema, analizado y diseñado en etapas anteriores, mediante un lenguaje de programación adecuado. En nuestro caso hemos elegido el lenguaje de programación PHP¹, y para facilitarnos el trabajo hemos hecho uso del framework *CodeIgniter*, que nos ayuda a desarrollar la aplicación siguiendo el patrón **MVC (Modelo Vista Controlador)**.

A parte de lo anterior, también hemos hecho uso de los lenguajes HTML (HyperText Markup Language)² y CSS (Cascading Style Sheets)³, sobre todo para la parte de la **Vista** de la aplicación.

El motivo de la elección de este lenguaje no es otro que su facilidad de aprendizaje y la extensa documentación que posee. Es un lenguaje que tuve que aprender por mi cuenta, ya que en la Universidad de Cádiz tan sólo se ve de forma básica en una asignatura optativa. Por una razón similar escogí el framework *CodeIgniter*, ya que posee una documentación excelente y, a parte, porque recibí un curso introductorio a esta herramienta, organizado por la Oficina de Software Libre de la Universidad de Cádiz (OSLUCA), donde pude descubrir el potencial de la misma.

A parte de los módulos que comentaremos a continuación, también se han desarrollado una serie de librerías que extienden las funcionalidades de las ya existentes en *CodeIgniter*, facilitándonos así todo el proceso de implementación. La razón por la cuál no se ha hecho el análisis y el diseño de dichas librerías es porque cuando pensé en cómo iba a desarrollar el sistema no sabía que me iban a hacer falta dichas librerías. Por tanto, han ido surgiendo durante la implementación del sistema, y por ello las comento en esta etapa.

¹«PHP» es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, es que permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

²«HTML» es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto.

³«CSS» es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

6.1. Desarrollo de OpenConference

Para la implementación de todo el sistema se ha utilizado el entorno *BlueFish*, el cuál es un editor HTML multiplataforma POSIX y con licencia GPL, lo que le convierte en software libre. Bluefish está dirigido a diseñadores web experimentados y programadores y se enfoca en la edición de páginas dinámicas e interactivas. Es capaz de reconocer diversos lenguajes de programación y de marcas.

Bluefish cuenta con características tales como rapidez, posibilidad de abrir múltiples archivos simultáneamente, soporte multiproyecto, soporte para archivos remotos mediante *gnome-vfs*, marcado de sintaxis personalizable basado en expresiones regulares compatibles con Perl, soporte para sub-patrones y patrones predefinidos (para HTML, PHP, Javascript, JSP, SQL, XML, Python, Perl, CSS, ColdFusion, Pascal, R, Octave/Matlab), diálogos para etiquetas HTML...[9].

A parte de lo anterior, necesitamos un servidor donde hacer correr el código PHP que vayamos creando. Para ello, utilizamos *XAMPP*, que va a realizar la función de servidor en nuestro propio equipo, sin necesidad de ningún acceso a Internet.

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP esta disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X.[10]

6.1.1. Panel de Administración

El Panel de Administración es el punto central de nuestra aplicación, desde donde se van a llevar a cabo todas las tareas necesarias para llevar un buen control de todo lo que ocurre en el sistema. Por esta razón, hemos escogido un diseño claro y sencillo, para que el usuario encuentre fácilmente todas las herramientas necesarias. Por ello, se ha cuidado mucho la estética y se ha dispuesto de un menú en el lateral izquierdo de la pantalla, que no es más que una lista desordenada de elementos.

En la parte central del Panel de Administración nos encontramos con una serie de iconos que nos desplazan hasta cada una de las herramientas disponibles en la aplicación. La forma de crear ese panel ha sido mediante una tabla de elementos, de 3 filas y 5 columnas. El resto de la pantalla está formado por una cabecera (header), un top (la barra gris que aparece justo debajo de la cabecera) y un pie (footer). Y todos los elementos están organizados ordenadamente y manteniendo un estilo impuesto por una hoja de estilo diseñada para tal fin.

El Panel de Administración no tiene ninguna lógica de control, su controlador tan sólo se encarga de comprobar si el usuario tiene permiso para acceder a la zona de administración (mediante la *access_library* creada por mí para tal fin), en cuyo caso llamará al método `load` de la *system_library* (ídem a la anterior) que se encargará de cargar el fichero con la vista de la página, con el contenido que hemos descrito anteriormente.

Noticias y Comentarios

En este apartado hemos englobado tanto el sistema de noticias como el de comentarios. La razón de dicha unión es porque ambos sistemas se comportan de la misma manera e incluso poseen los mismos métodos. Por tanto tan sólo describiremos uno de ellos, ya que su implementación es aplicable al otro también.

Pasamos a describir, por tanto, la implementación que hemos llevado a cabo para el sistema de noticias. Esta vez describiremos cómo hemos implementado el controlador de este módulo, así como el modelo, ya que la vista es realmente sencilla de construir, tan sólo haciendo uso de algunos elementos html así como de llamadas a las funciones contenidas en el controlador, que es el que recibe los eventos de la entrada.

Dentro del controlador nos encontramos con la clase Posts (que hereda de la clase **Controller** de *CodeIgniter*). En esta clase distinguimos por un lado su constructor, y por otro lado, los cuatro métodos siguientes: `index()`, `crear()`, `editar()` y `eliminar()`. Si no se especifica explícitamente a qué método llamar, se llama por defecto al método `index()`, que se encarga de establecer contacto con el modelo (que es el Sistema de Gestión de Base de Datos) para extraer información sobre las noticias publicadas en el sistema y listarlas ordenadamente, llamando para ello a la vista apropiada.

Los métodos `crear()` y `editar(id)` lo que hacen es presentar al usuario un formulario (creado mediante la clase **form** de *CodeIgniter*) para que introduzca datos o edite los ya existentes, respectivamente. Y posteriormente, cada método llama a la función correspondiente del modelo, que en este caso son las funciones `crear_post()` y `editar_post(id)`, para establecer contacto con la base de datos y llevar a cabo las tareas oportunas (ya sea de inserción, borrado, actualización...).

Por último, el método `eliminar(id)` lo que hace es llamar a la función `eliminar_post(id)` del modelo, para hacer un borrado de la base de datos de la noticia dada por su identificador (`id`).

Usuarios

Ahora hablaremos del sistema para gestionar usuarios. Cabe destacar que este sistema es distinto al Sistema de Autenticación de Usuarios, que se verá más adelante y que permite hacer las tareas cotidianas, tales como registro, login, recordatorio de password, etc...

El sistema de gestión de usuarios, que forma parte del módulo de administración, tan sólo se encarga de comprobar los usuarios que hay registrados en el sistema, pudiendo editar sus propiedades (cambiando por ejemplo el nivel de acceso a la aplicación) o eliminarlos del sistema.

Para realizar estas labores hemos creado lo de siempre, un controlador, un modelo y una vista asociada. La vista se encarga de llamar a las funciones del controlador, que son las que realmente realizan el trabajo, y de presentar la información al usuario, mediante html y una hoja de estilo. El controlador posee funciones como `editar(id)` o `eliminar(id)` y la ya conocida `index()`. Estas funciones a su vez llaman a los métodos que se encuentran en el modelo, que son los encargados de establecer contacto con la base de datos.

La funcionalidad de estos métodos es la misma que comentábamos en el apartado anterior de Noticias y Comentarios. Por ejemplo, la función `eliminar` llama al método `eliminar_usuario(id)` del modelo. Y este método, mediante las consultas adecuadas (ayudándonos para esta tarea de la clase **db** de *CodeIgniter*) busca el usuario (dado por su `id`) y lo elimina de la base de datos.

Templates

Para desarrollar el sistema de Templates se ha seguido el mismo procedimiento que en los anteriores. Esta vez tan sólo nos encontramos, dentro de la clase Templates del controlador, con la función `index()`. Esta función se va a encargar de presentar al usuario un formulario (que no son más que tres botones de tipo radio) con el fin de elegir entre tres posibles estilos para la aplicación.

Posteriormente se llama a la función `poner_template_defecto()` del modelo, que se va a encargar de actualizar la base de datos con el nuevo template por defecto, que será el que se cargue cada vez que se inicie la aplicación.

Y por último se llama a la vista de templates, para presentar toda esta información al usuario.

Configuración

El sistema de Configuración es muy parecido al sistema de templates comentado anteriormente. En este caso le presentamos al usuario un formulario con distintos campos que deberá rellenar de acuerdo a la configuración que quiere darle a la aplicación. El formulario está formado seis campos de tipo input, en los cuáles introducir texto. Dos campos de tipo checkbox para activar/desactivar cualquier aspecto relativo a la configuración, y un campo de tipo dropdown, que es como una lista desplegable en la que elegir una de las diferentes opciones posibles.

Todo esto se ha implementado ayudándonos de la clase **form** de *CodeIgniter*, que es muy potente y nos permite además validar los datos introducidos en cada campo, con lo cuál nos ahorra un gran trabajo de comprobación posterior. Respecto al controlador, también posee una única función llamada `index()`, que se encarga precisamente de comprobar que los datos del formulario son correctos y de llamar a la función `actualizar_settings()`, entre otras, del modelo, que se va a encargar de llamar a la base de datos e ir actualizando cada uno de los atributos de la tabla correspondiente, a partir de los datos enviados desde el formulario anterior.

Páginas y Navegación

En esta ocasión hemos unido estos dos sistemas ya que tienen funcionalidades muy parecidas y se implementan de la misma forma, por tanto bastará con comentar uno de ellos, ya que la implementación del otro se hace de igual forma. En concreto, trataremos el sistema de creación de páginas. Este sistema es realmente importante ya que nos ofrece una enorme capacidad para personalizar nuestra web. Podremos crear cuantas páginas deseemos, con el contenido que queramos.

Para implementar la clase **Paginas** del controlador hemos seguido el mismo proceso ya descrito anteriormente. Si se tienen las ideas claras, es realmente sencillo crear cualquier tipo de aplicación bajo *CodeIgniter*, ya que es muy intuitivo y casi siempre se sigue el mismo proceso, sea la aplicación que sea que estemos desarrollando.

Esta vez hemos necesitado, además de la función `index()` y del constructor propio de la clase, otros tres métodos que son los siguientes: `crear()`, `editar(id)`, y `eliminar(id)`. Sus nombres explican la funcionalidad de cada una de ellas.

Las funciones `crear()` y `editar()` lo que hacen es presentar al usuario un formulario creado con la clase **form**, y luego llaman a los métodos `crear_pagina()` y `editar_pagina()` de sus respectivos modelos. En el método `crear_pagina()` lo que hacemos es realizar una operación de inserción (función `insert` de la

clase **db**) en la base de datos, y en el método `editar_pagina(id)` lo que se hace es un `update` u operación de actualización (función `update` de la clase **db**) de datos. Para realizar todas estas operaciones nos ha sido de mucha utilidad la clase **db** incorporada en *CodeIgniter*, que nos abstrae de escribir toda la consulta, que en ocasiones son demasiado largas y resulta tedioso escribir tantas veces operaciones muy similares. Para la función `eliminar_pagina(id)` lo que se hace es una operación de borrado (función `delete` de la clase **db**).

Panel lateral

El sistema para administrar el panel lateral o sidebar es sencillo de implementar. Tan sólo debemos tener en cuenta que los elementos que aparezcan en este panel deben poder quitarse o ponerse según nuestras necesidades.

El controlador dispone de una clase llamada **sidebar** que contiene un constructor, encargado de cargar los módulos necesarios para la correcta ejecución, una función `index()` que llama al método `listar_sidebar()` del modelo, que accede a la base de datos y comprueba cuáles son los elementos que deben estar activos en el panel (si el atributo `enabled=1`) para listarlos. Seguidamente llama a la vista pasándole estos datos, para mostrárselos al usuario.

También poseemos tres funciones más, `activo(id)`, `inactivo(id)` y `eliminar(id)`. La función `activo(id)` llama al método `activo_sidebar_item(id)` que accede a la base de datos para actualizar (función `update` de la clase **db** de *CodeIgniter*) la información y poner el atributo `enabled`, del elemento seleccionado, a 1. La función `inactivo(id)` realiza la misma operación, pero esta vez poniendo el atributo `enabled` a 0, impidiendo que el elemento seleccionado del panel lateral no sea visible al usuario.

Por último, la función `eliminar(id)` hace lo mismo que en ocasiones anteriores. Se encarga de acceder a la base de datos (a la tabla `sidebar`) y realizar una consulta para borrar el dato oportuno (función `delete` de la clase **db**).

Revisión por Pares

Este sistema quizá sea el más complejo de implementar ya que es la parte más importante dentro del proceso de organización de una conferencia. El éxito del congreso depende en gran medida de las exposiciones que se hagan, y es con este sistema con el cuál se va a elegir entre unas y otras.

Como en las anteriores implementaciones, desarrollamos tanto el controlador, como el modelo y la vista. En la vista hacemos uso del lenguaje HTML para presentar los datos al usuario, ayudándonos de una cierta presentación dada por una hoja de estilo. Desde el modelo vamos a realizar todas las operaciones que necesiten establecer contacto con la base de datos, y todas estas operaciones las vamos a dirigir desde el controlador.

En este caso, dentro del controlador hemos creado una clase `cfp` que contiene, además de su constructor y la función `index()`, cuatro funciones que nos van a aportar varias funcionalidades. Las funciones son `revisar(id)`, `descargar(id)`, `ver(id)` y `eliminar(id)`. A todas las funciones le pasamos el identificador del paper sobre el que vamos a realizar la acción, ya que este identificador nos permite distinguirlos perfectamente.

La función `revisar(id)` se va a encargar de presentar al usuario un formulario donde podrá evaluar el artículo en cuestión. La evaluación es sencilla, tendrá que valorar el trabajo con una puntuación de 1 a 5, y además deberá aportar un resumen del artículo para ver qué ha entendido del mismo, destacando sus puntos fuertes y débiles, si es original, si es adecuado a las jornadas, etc...Esto se implementa de manera

muy sencilla; con un formulario que tenga un elemento de tipo dropdown (la lista desplegable para seleccionar la puntuación), y un elemento de tipo textarea (para escribir el resumen, la opinión personal...). Una vez que el usuario rellena el formulario, la función se encarga de llamar al método `revisar_cfp(id)` del modelo, que actualiza el paper añadiéndole los datos de la revisión.

La función `descargar(id)` se encarga de llamar al modelo para acceder a la base de datos y extraer el archivo seleccionado, para que el usuario pueda descargarlo a su computadora, con el fin de revisarlo.

La función `ver(id)` llama al método `ver_cfp(id)` del modelo, que realiza una consulta a la base de datos pero sin modificar nada, tan sólo los extrae (con la función `get` de la clase **db**) para mostrárselos posteriormente al usuario.

Y por último, la función `eliminar(id)` está implementada como hemos descrito ya en ocasiones anteriores.

Calendario

Este sistema es sencillo de implementar ya que tan sólo debemos generar un PDF con el calendario de todas las ponencias que se van a llevar a cabo durante las jornadas de conferencias.

Para facilitarnos el trabajo hemos hecho uso de la librería *FPDF*⁴.

Como en los demás sistemas desarrollados, implementaremos el controlador, el modelo y la vista. En la vista creamos la vista de usuario, cómo se presentarán los datos. En el modelo implementamos todas las funciones que nos van a permitir establecer contacto con la base de datos y extraer la información que nos sea necesaria.

En cuanto al controlador, vamos a describir cada una de las funciones que forman parte del mismo.

La función `index()` se va a encargar de listar todos las ponencias que se han incluido ya en el programa. Para ello llamará a los métodos `listar_ponencias()`, que extraerá los datos de todas las ponencias incluidas en el sistema. También llamará a `conseguir_nombre_ponente(user_id)` para realizar la acción que describe su nombre, y por último llamará al método `crear_calendario()` para generar el calendario interno en la aplicación, esto es, la ponencia con su ponente, el día y la hora cuando será impartida.

La función `generar()` lo que hará será llamar al método `crear_calendario_pdf()`, que a través de los métodos de la librería *FPDF* generará el calendario de ponencias en formato PDF para descargar al ordenador.

Certificados

En este sistema nos encargaremos de generar los certificados personalizados de cada participante. El sistema permite que el usuario envíe su propia plantilla en formato `.rtf` o `.pdf`, siempre que siga algunas instrucciones a la hora de crear dicha plantilla, como por ejemplo, poner marcas de sustitución para que el sistema las lea y las sustituya por el dato correspondiente extraído de la base de datos.

En el controlador nos encontramos con las siguientes funciones:

⁴«FPDF» es una biblioteca escrita en lenguaje de programación PHP que permite crear archivos en formato PDF sin ningún requerimiento adicional. Es gratuita, y su licencia permite que sea modificada libremente.

La función `index()` se encarga de listar todos los certificados que actualmente hay disponibles en el sistema. Para ello, dentro de la función `index()` tenemos una llamada al método `listar_certificados()` del modelo, que se va a dedicar a extraer información de la base de datos, como el nombre del diseño, el autor, etc...

La función `enviar()` mostrará un formulario desde el usuario podrá subir el fichero con la plantilla, e introducir algunos datos de interés como el nombre de la plantilla, quién la ha creado, etc...Para ello, una vez validado el formulario, llama al método `enviar_certificado(FILE)`, que realiza una inserción a la base de datos con la información proporcionada mediante el formulario.

Por último, la función `crear(id)` lo que hace es llamar al método `crear_certificado()`, que llamará a las funciones necesarias para inspeccionar la plantilla y sustituir las marcas de posición por los datos extraídos de la base de datos con respecto a los usuarios.

Backup

Este sistema es muy sencillo de implementar, ya que tan sólo hay que hacer uso de las funciones que nos ofrece *CodeIgniter*. En esta ocasión, para realizar una copia de seguridad de la base de datos hacemos uso de la función `backup`, que se encuentra dentro de la clase **dbutil**.

Una vez generado el backup, tan sólo hay que ponerle un nombre y "forzar" al navegador a que descargue el archivo. El nombre del archivo tendrá el formato siguiente: 'backup-'(Fecha Actual).sql.gz'. Y para forzar la descarga del backup hacemos uso de la función de PHP, `force_download()`, pasándole como parámetros tanto el nombre del archivo como el fichero con la copia de seguridad generada con **dbutil**.

6.1.2. Noticias

Este sistema es el que nos va a permitir visualizar las noticias de la web, escribir comentarios, etc...Este sistema es diferente al sistema de gestión de noticias que comentamos anteriormente. El sistema anterior formaba parte del panel de administración y tan sólo era accesible para el administrador, para escribir nuevas noticias o para editarlas o eliminarlas. En esta ocasión, el sistema es el que se encarga de presentar la noticia al usuario, de permitir que se escriban comentarios sobre la misma, de mostrar el contenido de otras páginas, etc...

Para la implementación se sigue el mismo método; creamos el controlador, el modelo y la vista. En esta ocasión, en el controlador hemos creado una clase Noticias que, además del constructor y la función `index(pagina)`, posee otras dos funciones que son `post(anno, mes, día, url_titulo)` y `comentario(id, url)`. La función `index()` se va a encargar de listar todos los posts que hay publicados en el sistema, pero primero los va a contar y los va a ir mostrando poco a poco (según hayamos elegido en la configuración del sistema). Para ello tendrá que hacer un pequeño cálculo, dividiendo el número total de noticias publicadas entre el número de noticias que queremos que aparezcan por cada página, para así saber cuántas páginas tiene que crear. Para esta tarea nos ayudaremos de la librería **pagination** disponible en *CodeIgniter*. De nuevo, la función `index()` llamará a los métodos del modelo que necesite, según los datos que le hagan falta. Las funciones del modelo se encargarán de acceder a la base de datos y luego proporcionar esos datos al controlador.

La función `post(anno, mes, día, url_titulo)` realiza una labor especial. Cuando se pulsa sobre el título de una noticia, el sistema nos manda a una página donde está todo el contenido de la misma. Para calcular la url donde nos debe enviar hace uso precisamente de la fecha de publicación de la noticia. Luego realiza llamadas a los métodos del modelo, para conseguir datos como el usuario que ha escrito la noticia, el

número de comentarios que se han hecho sobre la misma, etc...

Y por último, la función `comentario(id, url)` presenta al usuario un formulario para que pueda hacer un comentario sobre la noticia. Aquí comprobamos si el usuario está logueado al sistema, en cuyo caso el usuario sólo deberá proporcionar el comentario. O por el contrario si es un invitado o el usuario no está logueado, deberá proporcionar un nick, un email y el comentario en sí. Esto es así por seguridad, y para facilitar la administración de los comentarios desde el panel de administración. La función llama al método `crear_comentario(id)` del modelo, que se encarga de realizar la consulta apropiada (en este caso una inserción de datos, mediante la función `insert` de la clase **db**).

6.1.3. Páginas

En esta ocasión implementamos el sistema que nos permite visualizar las páginas que hemos creado nosotros mismos desde el panel de administración. La clase `Paginas` del controlador, además del constructor, posee una única función llamada `pagina(url_titulo)` que se va a encargar de establecer la url de la página que hemos creado dinámicamente, para que cuando pulsemos el enlace de la página se nos muestre el contenido asociado a la misma.

Para realizar esto, dentro de la función `pagina(url_titulo)` llamamos al método `conseguir_url_pagina(url_titulo)`, que va a realizar una consulta a la base de datos para extraer información de la página en cuestión. Así podremos comprobar si ya se ha creado el contenido de dicha página, o por el contrario aún no existe. Dependiendo de esto, se cargará la página indicada en el atributo `url_titulo`, o por el contrario se cargará una página de error, informando del problema existente.

6.1.4. Usuarios

Cabe recordar que este sistema es diferente al sistema de gestión de usuarios, que forma parte del panel de administración, y que ya hemos comentado anteriormente. En esta ocasión, el sistema nos aportará las funcionalidades típicas de cualquier sistema de usuarios, permitiendo registro, login, logout, recordar el password olvidado, actualizar el perfil de usuario, etc...

Para implementar este sistema hemos creado un controlador con una clase llama `Usuario`. Esta clase dispone de las siguientes funciones; `registro()`, `activacion_cuenta()`, `password_olvidado()`, `ver(nickname)`, `perfil()`, `login()`, `logout()`.

La función `registro()` tiene la misión de presentar al usuario un pequeño formulario de registro. Dicho formulario tendrá campos como el nick, el nombre, el password, la confirmación del password, dirección de correo, etc... Seguidamente la función realiza una llamada al método `crear_usuario()` del modelo, que es el que se encarga de hacer la consulta a la base de datos, en este caso una inserción (función `insert`) con la información del nuevo usuario registrado.

La función `activacion_cuenta()` realiza a su vez una llamada a la función `activa_cuenta(key, email)` del modelo. Esta función lo que hace es poner el estado del usuario a 1, es decir, le permite la entrada al sistema.

La función `password_olvidado()` presenta al usuario un pequeño formulario donde debe introducir su nick y su email, al cuál se le enviará la información necesaria para recuperar su password o actualizarlo por uno nuevo. Dentro de esta función se hace la llamada al método `password_olvidado()` del modelo, que ya se encarga de crear el email y mandárselo al usuario con las instrucciones pertinentes.

La función `ver()` es muy sencilla, tan sólo establece contacto con la base de datos para mostrar información del usuario. Y la función `perfil()` es muy parecida a la función `ver()`, pero en este caso nos permite editar los datos del perfil, pudiendo cambiar el nombre de usuario, el email, el password, etc...

La función `login()` se encarga de comprobar qué persona está accediendo al sistema, para darle más permisividad o no. Dependiendo del nivel que tenga el usuario, podrá acceder a unas zonas u otras de la aplicación. Para implementar este sistema hacemos uso de la **access_library**, creada por mí para tal fin. Y por último, la función `logout()` lo que hace es sacar al usuario del sistema borrando toda la información privada que se pudiera quedar en el navegador. Es una forma segura de abandonar la sesión del usuario. Para su implementación también hacemos uso de la **access_library**, la cuál comentaremos a continuación, junto con otras librerías creadas para facilitar el desarrollo de todas las funciones que hemos descrito anteriormente.

6.2. Desarrollo de Librerías

Durante la implementación de cada uno de los sistemas que forman parte de **OpenConference** me di cuenta que había muchos procesos que se podían abstraer y generalizar, para utilizarlos en cualquier tipo de proyecto. Por esta razón me decidí a crear una serie de librerías, algunas nuevas y otras que extienden la funcionalidad de las librerías que incorpora *CodeIgniter*.

Para crear una librería en **CodeIgniter** en primer lugar debemos nombrar el archivo con el siguiente formato: **Nombre_library.php**, siendo **Nombre** el que nosotros queramos. En cambio, si lo que queremos es extender la funcionalidad de una librería ya existente en *CodeIgniter*, lo único que debemos hacer es anteponer la palabra **MY** al nombre, de la siguiente manera: **MY_Nombre.php**.

Una vez visto el formato a seguir a la hora de crear o ampliar una librería, pasamos a describir cada una de ellas, destacando las funcionalidades que nos aportan.

6.2.1. Access_library

Esta librería la usamos para todo el trabajo que requiere la autenticación de usuarios, es decir, para controlar el acceso al sistema dependiendo del nivel que tenga el usuario. Según su nivel de acceso podrá entrar en unas zonas u otras de la aplicación. En **OpenConference** podemos distinguir tres niveles de acceso: usuario, revisor y administrador.

El usuario sólo podrá navegar por la página principal (no podrá acceder al panel de administración). El revisor si podrá acceder al panel de administración, pero sólo a la sección dedicada a la revisión de papers, ya que esa es precisamente su función. Y por último, el administrador tiene acceso a todo el sistema, es el encargado de administrar toda la aplicación por completo, por tanto tendrá el mayor nivel de acceso.

En cuanto a la implementación, poseemos una clase llamada de la misma manera que la librería (**Access_library**). Sin embargo, ahora dentro de las funciones no podremos hacer uso del constructor `$this`, ya que este sólo está disponible dentro de los controladores, los modelos o las vistas. Para solucionar este problema debemos llamar al método `get_instance()` y lo referenciamos a una variable. Esto es precisamente lo que se debe hacer dentro del constructor que se cree en cada una de las librerías, ya que esa variable creada se podrá utilizar como `$this`, permitiéndonos acceder a todos los recursos que nos ofrece *CodeIgniter*.

Las funciones que nos encontramos en esta clase son las siguientes: `is_logged_in()`, `is_admin()`, `is_reviser()`, `check_logged_in()`, `check_access()`.

La función `is_logged_in()` va a hacer uso del vector `userdata` contenido dentro de la clase **session** de *CodeIgniter*. Dicho vector contiene una serie de variables, perfectamente configurables por nosotros, a las que podemos acceder para comprobar el valor que tienen en un determinado momento. De esta manera, nosotros hemos añadido al vector `userdata` las siguientes variables: `logged_in` (valor `TRUE` o `FALSE`), y `nivel` (administrador, usuario, revisor).

En el caso que nos ocupa, la función `is_logged_in()`, comprobamos que la variable `logged_in` esté a `TRUE`, lo que quiere decir que el usuario ya está logueado en el sistema, y por tanto devolveremos `TRUE`, en caso contrario se devolverá `FALSE`.

La función `is_admin()` realiza una tarea similar a la anterior, pero esta vez comprueba que la variable `nivel` contenida en el vector sea igual al valor `'administrador'`, en cuyo caso devolveremos `TRUE`. La misma operación realiza la función `is_reviser()`, pero esta vez comprobando que la variable `nivel` tenga el valor `'revisor'`.

La función `check_logged_in()`, a diferencia de la función `is_logged_in()`, lo que va a hacer es comprobar si el usuario ya está logueado en el sistema, y si no lo está lo va a redireccionar a la pantalla que contiene el formulario de acceso al sistema.

Y por último, la función `check_access()` nos va a servir para restringir el acceso a determinadas zonas de la aplicación. Lo que hacemos es comprobar si la variable `nivel` tiene el valor `'usuario'`, en cuyo caso lo redireccionamos a la pantalla con el formulario de acceso al sistema.

6.2.2. Links_library

Debido a la capacidad que tiene el sistema para crear enlaces dinámicamente, sin necesidad de establecer ningún contacto con el código de la aplicación, se nos hace necesario poseer de ciertas herramientas que nos permitan averiguar los enlaces que se van creando.

De nuevo, creamos una clase con el mismo nombre que la librería (`Links_library`) y en el constructor de la misma utilizamos la función `get_instance()`, asignando su valor por referencia a una variable cualquiera. Esto nos va a permitir utilizar todos los recursos incorporados en *CodeIgniter*, como ya hemos dicho anteriormente.

Dentro de esta clase nos encontramos con una sola función llamada `get_links()`. Esta función va a establecer contacto con la base de datos para conocer en cada momento el número de enlaces que el usuario ha creado, así como la dirección a la que apunta cada uno de ellos.

Para conseguirlo realizamos una consulta a la base de datos extrayendo toda la información que necesitamos (función `get` de la clase **db**), limitando el resultado al número de enlaces que queremos tener en el panel lateral (configurable desde el menú `settings` del panel de administración).

6.2.3. Navigation_library

Esta librería se ha creado por el mismo motivo que la librería descrita anteriormente. Es decir, va a ser posible que el administrador pueda añadir los items que quiera al menú de navegación de la página, por tanto la aplicación tendrá que conocer cuántos y cuáles son los items que existen, para mostrarlos todos

automáticamente al cargar la aplicación.

En cuanto a la implementación, seguimos el mismo procedimiento que otras veces. Creamos la clase con el mismo nombre que la librería (`Navigation_library`), creamos su constructor añadiendo la variable que nos va a permitir utilizar todos los demás módulos y funciones disponibles en la aplicación.

Esta vez la clase dispone de una función llamada `get_navigation()` que realiza exactamente la misma operación que la función `get_links()`, solo que ahora obtenemos los items del menú de navegación, en vez de los enlaces del panel lateral.

6.2.4. `Pages_library`

Esta librería es igual que las dos anteriores, salvo que esta vez trabaja con páginas, y no con enlaces o items. Ahora tan sólo debemos tener una cosa en cuenta a la hora de realizar la consulta, y es comprobar que el estado de la página sea 'activa', ya que de lo contrario no se debería poder mostrar al usuario y no debería aparecer en el menú para su visita.

Esta función por tanto, va a extraer de la base de datos todas las páginas que el usuario ha creado y luego se las va a pasar a la vista, para que sea esta la que se encargue de ordenarlas y mostrárselas al usuario.

Estas tres últimas librerías descritas son muy sencillas de elaborar, pero nos aportan una gran funcionalidad al sistema ya que ofrecen al usuario un alto grado de personalización, que es otro de los objetivos que se quería conseguir con **OpenConference**.

6.2.5. `Sidebar_library`

Esta librería se comporta también como todas las anteriores, pero ahora nos va a permitir conocer el número de elementos que forman el panel lateral. La aplicación debe permitir activar o desactivar los elementos que queramos del panel lateral, salvo el elemento que contiene los enlaces para acceder al sistema o para registrarse, ya que si quitamos este panel la aplicación pierde toda su utilidad ya que tampoco se podría acceder al panel de administración.

Sin embargo existen otros dos paneles (se pueden ampliar) que pueden ser ocultados o no, según las preferencias del administrador de la página. Estos paneles son: el panel con los enlaces a otras webs de interés, y el panel de Papers, desde donde el usuario podrá enviar sus artículos.

Para conocer por tanto cuáles son los paneles que deben ser mostrados y cuáles no se ha creado esta librería, que posee una función llamada `get_sidebar_items()` que hace precisamente esta labor. Accede a la base de datos y comprueba cuáles son los paneles cuyo estado es 'enabled', para así pasárselos a la vista que es la encargada de mostrárselos al usuario.

6.2.6. System_library

En esta librería nos encontramos con algunas funciones que nos sirven para conocer la configuración que posee la aplicación. Y también tenemos un par de funciones que van a dedicarse a cargar la vista necesaria para presentar los datos, librándonos a nosotros de tener que estar llamando a la vista cada vez que queramos mostrar algún dato por pantalla.

Dentro de la clase `System_library` disponemos de las siguientes funciones: `get_default_template()`, `get_site_info()`, `check_site_status()`, `load(page, data, admin)`, `load_normal(page, data)`.

La función `get_default_template()` lo que va a hacer es devolvernos la ruta donde se encuentra almacenado el template escogido por defecto en la configuración del sistema. Para ello tan sólo realiza una consulta a la base de datos, extrayendo el atributo 'path' y comprobando que el atributo 'por_defecto' tenga el valor 1.

La función `get_site_info()` lo que va a hacer es darnos información del sistema, como por ejemplo el nombre del autor, el nombre de la aplicación, documentación, etc...Para ello hacemos una consulta a la base de datos para extraer toda esta información (funcion `get` de la clase **db**).

La función `check_site_status()` nos va a permitir comprobar el estado en el que se encuentra la página, ya que es posible hacer que la página se encuentre en estado inactivo, ya sea por motivos de mantenimiento o cualquier otra razón. Por tanto, mediante esta función vamos a poder conocer si la página se encuentra activa, y en el caso de que no se encuentre activa, nos mostrará un pequeño mensaje con la razón de su inactividad.

La función `load(page, data, admin)` lo que va a hacer es cargar la vista de la página dada (todo el contenedor) por el parámetro `page`, pasándole los datos necesarios para su correcta ejecución mediante el vector `data`. El parámetro `admin` nos permitirá distinguir entre la vista de la página contenida en el panel de administración y la vista de la página general. Esta función lo único que hace es cargar la vista mediante la función `view` de *CodeIgniter*.

Capítulo 7

Pruebas

En el presente apartado nos dedicaremos a describir cada una de las pruebas que le hemos ido realizando a cada una de las herramientas disponibles en **OpenConference**, con el fin de comprobar que el sistema se comporta tal y como lo especificamos inicialmente, y por tanto cumple todos los requisitos impuestos.

Cabe destacar que el sistema ha sido probado por terceras personas, para asegurarme que la aplicación es sencilla de utilizar y para comprobar realmente que el sistema se comporta adecuadamente ante cualquier solicitud del usuario.

7.1. Proceso de Prueba

El proceso de prueba comienza con la generación de un plan de pruebas en base a la documentación del proyecto y a la documentación del software a probar. A partir de dicho plan, se diseñan los casos de prueba, se ejecutan y los resultados obtenidos se comparan con los resultados esperados. Una vez evaluados los resultados de las pruebas, pueden realizarse dos actividades:

- Depurar los defectos
- Analizar los errores

La depuración puede corregir o no los defectos. Si no consigue localizarlos, puede ser necesario realizar pruebas adicionales para obtener más información. Si se corrige un defecto, se debe volver a probar el software para comprobar que el problema está resuelto.

Dado que no se pueden probar todas las posibilidades de funcionamiento del software, la idea fundamental para el diseño de casos de prueba consiste en elegir aquellas posibilidades que, por sus características, se consideran representativas del resto. De esta forma se asume que, si no se detectan defectos en el software al ejecutar dichos casos, podemos tener cierto nivel de confianza (que dependerá de la elección de los casos) en que el programa no tiene defectos. La dificultad de esta idea está en saber elegir los casos que se deben ejecutar. [11]

7.1.1. Incremento 1: Requisitos Básicos del sistema

¿La aplicación funciona correctamente en cualquier navegador web? ¿El diseño y las imágenes se muestran correctamente?

Todas las pruebas realizadas, y las que vienen a continuación, se han llevado a cabo bajo distintos navegadores web tales como: Mozilla Firefox, Internet Explorer, Konqueror, Opera, Google Chrome.

En todos ellos las respuestas han sido las esperadas, comportándose el sistema adecuadamente ante todas las solicitudes realizadas. De la misma manera, el diseño o aspecto gráfico de la aplicación se visualiza de forma correcta como era de esperar ya que **OpenConference** respeta todos los estándares y, por tanto, cualquier navegador que también los respete (como los mencionados anteriormente) no debe tener problema en ejecutar correctamente la aplicación.

7.1.2. Incremento 2: Panel de Administración

¿El Panel de Administración se muestra correctamente? ¿Funcionan adecuadamente todos los enlaces que hay en él?

Como hemos dicho anteriormente, durante las pruebas realizadas bajo diferentes navegadores se aprecia claramente que todo el diseño, los fondos, las imágenes, etc...se visualizan correctamente. Todos los iconos aparecen centrados y con el tamaño impuesto, el menú lateral se muestra ordenado y homogéneo. En cuanto a los enlaces, todos funcionan perfectamente, tanto los del menú lateral como los iconos del panel central. Todos nos llevan a la página que deben, no hay ningún enlace inconsistente y la estratégica disposición que adoptan en el panel de administración nos ayuda a conseguir nuestro cometido: sencillez y facilidad de uso.

7.1.3. Incremento 3: Sistema de Noticias y Comentarios

¿El sistema muestra todas las noticias y comentarios que hay publicados en el sistema? ¿Los datos asociados a cada noticia y comentario son correctos? ¿Se crean las nuevas noticias satisfactoriamente? ¿Se puede editar una noticia o comentario con facilidad? ¿Y eliminar?

Al acceder al apartado de Noticias o al de Comentarios se puede observar que todos los que se han publicado en el sistema aparecen en una lista detallada y perfectamente ordenada por fecha de publicación. El sistema es capaz de listar tanto las noticias que están publicadas como las que se encuentran en estado 'draft' o 'borrador' a espera de su publicación.

En la propia lista, tanto de Noticias como de Comentarios, aparece información detallada como la fecha de publicación, el autor, el estado, el título y un conjunto de acciones que se pueden realizar sobre el elemento. Toda la descripción que aparece es la correcta, a cada noticia y comentario se le asocia bien el autor que lo ha escrito, así como la fecha en que lo ha hecho y demás información.

A la hora de crear una nueva noticia, el sistema se comporta adecuadamente. El formulario para introducir los datos de la noticia se visualiza correctamente y si intentamos meter datos erróneos el sistema nos va a avisar y va a cancelar la publicación de la noticia. Además, una vez publicada la noticia se observa que todos los datos corresponden con los que hemos insertado y no se presenta ninguna inconsistencia.

Las acciones que se pueden llevar a cabo con una Noticia o Comentario son: Editar y Eliminar. Ambas acciones se comportan de forma esperada y funcionan a la perfección, sin fallos en su ejecución.

7.1.4. Incremento 4: Sistema de Usuarios

¿El sistema muestra todos los usuarios registrados en el sistema? ¿Aporta información detallada del usuario y lo hace adecuadamente? ¿Permite editar la información del usuario? ¿Se puede eliminar fácilmente a cualquier usuario?

Cuando entramos en el apartado dedicado a la gestión de usuarios observamos claramente cómo se nos muestra una lista con todos los miembros registrados en el sistema. Se muestran todos los usuarios adecuadamente, ni más ni menos. Además el sistema nos aporta información detallada de cada usuario como por ejemplo, su username, su email, la fecha del último acceso, el nivel de acceso al sistema...Se ha comprobado que toda esta información es la correcta y pertenece a cada uno de los usuarios.

En cuanto a las acciones que se pueden realizar con ellos, tanto la edición como la eliminación se ejecutan correctamente. Es decir, tras eliminar a un usuario se observa cómo toda su información relativa se borra del sistema y no aparece en la lista. En cuanto a la edición, se observa que se lleva a cabo de forma correcta y que tras editar los datos se muestran todos los cambios realizados.

7.1.5. Incremento 5: Sistema de Templates

¿Se muestran todos los templates disponibles? ¿El sistema nos permite elegir cualquier template de la lista? ¿Se aplica correctamente el template elegido?

El sistema de Templates se comporta bien en todos los aspectos. La aplicación nos muestra de forma precisa cada uno de los templates disponibles en el sistema, asegurándose que efectivamente el template se puede aplicar. Además el sistema nos muestra el formulario de elección correctamente y al seleccionar uno u otro template lo realiza satisfactoriamente.

En cuanto a los cambios aplicados tras elegir el template podemos decir que el sistema se comporta como se esperaba, ya que cambia por completo la apariencia de la Web, según el template escogido, y además toda la información se sigue mostrando adecuadamente y el sistema mantiene su funcionalidad por completo.

7.1.6. Incremento 6: Sistema de Configuración

¿El sistema muestra la configuración actual de la aplicación? ¿El sistema permite cambiar dicha configuración? ¿Se aplican correctamente los cambios cuando variamos la configuración?

El sistema de configuración muestra en todo momento el estado de cada uno de los parámetros que se pueden configurar en la aplicación. Los datos mostrados son correctos y si cambiamos alguno de ellos, las variaciones se aplican correctamente.

Al cambiar cualquiera de los parámetros de configuración, automáticamente se aplican los cambios a la aplicación y todo esto se lleva a cabo satisfactoriamente. No se observa ningún fallo en la ejecución y cuando se intenta dar un valor erróneo a alguno de los parámetros de configuración, el sistema muestra el aviso oportuno y cancela los cambios.

7.1.7. Incremento 7: Sistema de Páginas y Navegación

¿El sistema reconoce y muestra todas las páginas que forman parte de la aplicación? ¿Y los items del menú de navegación? ¿El sistema permite crear nuevas páginas con su contenido? ¿Y nuevos items para el menu de navegación? ¿El sistema permite editar con facilidad tanto las páginas como los items de navegación? ¿Y eliminarlos?

Dentro del sistema de gestión de páginas o de navegación vemos claramente una lista con todas las páginas o items que existen en ese momento en la aplicación. Si no existe ninguno, la aplicación muestra un mensaje informativo.

A la hora de crear nuevas páginas o items de navegación el sistema se comporta satisfactoriamente ya que los formularios para introducir los datos se visualizan correctamente y, como en el caso anterior, si intentamos introducir datos no válidos, la aplicación nos avisa y cancela el proceso. Una vez creada la página o item de navegación observamos como estos aparecen en la lista y además lo hacen de forma correcta, mostrando todos los datos tal como los hemos introducido.

Cuando se crea una nueva página o item de navegación los cambios se aplican correctamente, ya que aparece un nuevo item en el menú y las páginas se crean adecuadamente y son accesibles al usuario desde el primer momento.

La funciones Editar y Eliminar también se comportan bien con respecto a lo esperado, ya que tanto al editar como al eliminar se aplican correctamente todos los cambios al instante y sin fallos en la ejecución.

7.1.8. Incremento 8: Panel Lateral y enlaces

¿Se muestran correctamente todos los elementos del panel lateral? ¿Y los enlaces de interés? ¿El sistema crea nuevos enlaces correctamente? ¿Podemos activar/desactivar los elementos del panel lateral? ¿El sistema permite editar y eliminar los enlaces?

Tanto el sistema de administración del panel lateral como el sistema de administración de enlaces muestran una lista con todos los elementos existentes en la aplicación, y los muestran sin fallos y con información coherente.

El sistema nos permite crear nuevos enlaces de interés en cualquier momento y lo hace adecuadamente, ya que nos presenta el formulario de forma precisa y nos avisa si intentamos introducir datos no válidos, cancelando automáticamente todo el proceso. Una vez creado el enlace se puede observar en la página que aparece tal y como lo hemos creado, y que el enlace funciona y nos lleva a la página especificada por nosotros.

El sistema también nos permite activar o desactivar los elementos que aparecen en el panel lateral y lo hace sin fallos, ya que si desactivamos cualquier elemento vemos como en la página dicho elemento ya no aparece, y en el caso contrario se produce el mismo efecto, con lo cuál el sistema funciona sin fallos.

Por último, las tareas de edición y eliminación de enlaces se llevan a cabo correctamente y se observa como se aplican los cambios sin ningun tipo de problema en la ejecución de la tarea.

7.1.9. Incremento 9: Sistema de revisión por pares

¿El sistema muestra todos los artículos que han enviado los usuarios? ¿La información de cada artículo es correcta? ¿El proceso de revisión del artículo se lleva a cabo satisfactoriamente? ¿Se puede ver la revisión hecha sobre un artículo? ¿Se puede editar la revisión hecha sobre el artículo? ¿Se puede eliminar cualquier artículo de la lista?

Este sistema quizá es el más complejo y el que más funcionalidades presenta, por tanto hay que realizar una gran variedad de pruebas para comprobar que, efectivamente, funciona tal y como se especificó inicialmente.

La aplicación muestra perfectamente todos los artículos que se han enviado al sistema, además la información relativa a cada artículo es correcta y no se observan inconsistencias en la información.

El proceso de revisión es correcto, ya que el sistema se comporta adecuadamente cuando solicitamos descargar el documento para su revisión. Además el formulario de revisión es sencillo y también es validado antes de ser enviado, con lo cuál el sistema nos informará ante cualquier tipo de error. Una vez revisado el artículo, se observa claramente como el estado de dicho artículo pasa del estado 'recibido' al estado 'revisado'.

La revisión del artículo puede ser editada (por el mismo revisor que la hizo) y se realiza de forma correcta, aplicándose satisfactoriamente todos los cambios. Al acceder a la valoración que se ha hecho sobre el artículo, el sistema también se comporta de acuerdo a lo esperado, ya que nos muestra exactamente la información dada por el revisor, y la muestra de forma clara.

Por último, se observa que los artículos solo pueden ser eliminados por el administrador de la página y que el proceso de eliminación se lleva a cabo sin fallos en la ejecución.

7.1.10. Incremento 10: Sistema de creación de calendario

¿El sistema muestra un listado con todas las ponencias incluidas en el programa? ¿El sistema permite modificar las ponencias que ya se han incluido en el programa? ¿Y añadir nuevas? ¿El sistema genera el archivo PDF adecuadamente?

El sistema de creación de calendario se comporta de acuerdo a lo pactado en la especificación del mismo. Con respecto a la primera pregunta, podemos decir que desde el primer momento el sistema nos muestra la lista ordenada con el programa de ponencias, los autores de cada una de ellas, los días y horas en que se van a impartir, etc...

El sistema también nos permite añadir nuevas entradas a la lista con tan sólo seleccionar la ponencia de la lista desplegable, así como el día y la hora en que se llevará a cabo, y de igual manera nos permite actualizar los datos de cualquier ponencia que aparezca en la lista.

Por último, una vez creado el calendario, el sistema genera adecuadamente el archivo PDF con respecto a lo esperado. Se visualiza correctamente la tabla, los acentos y demás símbolos aparecen correctamente y las ponencias aparecen ordenadas primero por días y luego por horas, generando automáticamente el programa que se llevará a cabo durante las jornadas de conferencias.

7.1.11. Incremento 11: Sistema de creación de certificados

¿El sistema permite seleccionar cualquier plantilla de la lista y la establece como predeterminada? ¿El sistema nos permite enviar nuestras propias plantillas? ¿Se genera correctamente el certificado con los datos correctos de cada usuario?

El sistema de certificados se comporta adecuadamente ante todas las pruebas realizadas. Nos permite seleccionar cualquier plantilla de la lista y automáticamente la establece como predeterminada, con lo cual todos los certificados generados se crearán a partir de dicha plantilla. También nos permite enviar nuestras propias plantillas (formato .rtf o .pdf) y seguidamente nos la muestra en la lista para poder seleccionarla. Y por último, el sistema genera correctamente el certificado, con el nombre de cada usuario participante en las conferencias.

7.1.12. Incremento 12: Sistema de Backup

¿Se genera adecuadamente la copia de seguridad de la base de datos? ¿Se puede descargar dicho archivo correctamente?

El sistema de backup se comporta adecuadamente ante cualquier solicitud. Al usar la herramienta para generar el backup, el sistema automáticamente genera el archivo .sql con toda la información contenida en la base de datos, y lo pone a disposición del usuario para su descarga. Todo este proceso lo lleva a cabo correctamente y sin ningún fallo de ejecución.

7.1.13. Incremento 13: Sistema de Información y Estadísticas

¿Se muestra correctamente la información del sistema? ¿Y las estadísticas?

Este sistema da información al usuario y aporta documentación sobre el uso de la aplicación. Es un sistema sencillo que realiza su trabajo adecuadamente, presentando la información correctamente y sin fallos, así como las estadísticas de diferentes aspectos, como el número de noticias del sistema, el número de comentarios, etc...

Capítulo 8

Conclusiones

En este apartado hablaremos sobre las distintas funcionalidades que se le podrían añadir a la aplicación para convertirla así en un sistema mucho más completo, y además ofreceremos una opinión personal del proyecto realizado, elaborando un pequeño resumen de todo lo aprendido y de la experiencia vivida durante el desarrollo de **OpenConference**.

8.1. Posibles Ampliaciones

En versiones posteriores la aplicación podría extender buena parte de la funcionalidad que ofrece, realizando algunos cambios como los siguientes:

- Internacionalización. Se podría modificar el sistema para que pueda adecuarse a múltiples idiomas y convenciones culturales. Esto haría llegar al producto a mercados internacionales, lo que implicaría un aumento enorme del número de posibles usuarios.
- Inscripción con Pago Online. Podría crearse un sistema electrónico de pago por internet mediante tarjeta de crédito o e-mail. Esto es realmente importante, sobretodo para la comodidad del usuario, que no tendría que desplazarse al banco a pagar la inscripción de la conferencia.
- Inclusión de más estilos. Se podría ampliar la oferta de templates para que el usuario tenga más variedad donde elegir y no tenga que recurrir a crearse (o buscar) su propio diseño.
- Documentación de librerías. En el proyecto se han documentado brevemente las librerías (por falta de tiempo), pero en futuras versiones se podría crear una documentación aparte y más completa, para que cualquiera pueda utilizar dichas librerías en la elaboración de otros proyectos.
- Creación de portal web. Se podría crear una web donde presentar el producto, donde publicar las futuras versiones del mismo, donde más desarrolladores puedan colaborar y ayudar a que la aplicación vaya creciendo, etc...

8.2. Opinión personal

Una vez completada la aplicación y la presente memoria, procedemos a valorar todo el trabajo y esfuerzo realizados para sacar adelante este proyecto.

En primer lugar, hay que destacar que el proyecto aún puede sufrir numerosos cambios (véase el apartado anterior de Posibles Ampliaciones) que lo conviertan en un producto mucho más completo, capaz de llegar a un número más amplio de usuarios. Por tanto, toda persona que quiera colaborar en dicha ampliación tendrá mi total apoyo y motivación, ya que este proyecto es realmente gratificante y rápidamente se obtienen los frutos de tantas horas de dedicación y entrega a la elaboración del mismo.

En cuanto a los resultados obtenidos después del desarrollo de **OpenConference**, he quedado realmente satisfecho ya que creo que la calidad del software es bastante buena, y si le añadimos que es mi primera aplicación elaborada con PHP, lenguaje con el que partía de cero, mi satisfacción es aún mayor, ya que un proyecto que inicialmente se me antojaba complicado al final se ha hecho realidad después de tanto esfuerzo.

A parte de lo mencionado anteriormente, me encuentro también satisfecho con todo lo aprendido y con todas las herramientas que desconocía y he llegado a aprender con el fin de elaborar el proyecto con la mayor calidad posible.

En este sentido puedo empezar hablando del entorno de programación que me ha acompañado durante la implementación del sistema, hablo de *Eclipse* y del plugin *PhpEclipse*, que juntos forman un excelente IDE para programar en PHP, ofreciéndonos la posibilidad de tener controlados todos los ficheros que forman parte de la aplicación, ayudándonos en la creación de código repetitivo (como bucles, sentencias condicionales), etc...

Otra de las herramientas utilizadas ha sido el framework *CodeIgniter*, que nos ha ayudado a construir la web siguiendo el patrón MVC, comentado también en el presente documento. La razón por la cuál hemos escogido este framework entre todos los disponibles ha sido principalmente por la excelente documentación que presenta, que nos ha ayudado a solucionar todas las dudas que nos han surgido, además de por su rapidez y facilidad de uso.

Hemos utilizado también la herramienta XAMPP (nos ofrece Apache, MySQL, PHP y Perl) para crear el servidor local en nuestro sistema donde hemos ido probando la aplicación.

Y algunas herramientas más que hemos utilizado son; *Subversion*, para llevar un control de versiones de la aplicación y subir el contenido a la forja donde se encuentra el proyecto alojado. *Doxygen*, para documentar todo el código fuente, y *Planner*, que nos ha ayudado a planificar todo el desarrollo de **OpenConference**. Esto último es realmente importante ya que en el presente proyecto hemos aprendido a organizar eficientemente todo el tiempo del que disponemos, y por tanto, herramientas como *Planner* nos ha sido más fácil e intuitivo ver todas las tareas realizadas y todas las tareas pendientes de realizar, así como el tiempo que hay disponible para llevarlas a cabo.

Usar una de las metodologías de desarrollo de la Ingeniería de Software es realmente importante, ya que nos ayuda a llevar un control de cada una de las etapas que debemos realizar a la hora de crear cualquier tipo de software, y además nos ofrece cierta garantía de que siguiendo una correcta metodología en el desarrollo del mismo vamos a conseguir un resultado satisfactorio.

Considero que hemos seguido al pie de la letra cada una de las indicaciones que nos ofrece dicha metodología y por tanto hemos cumplido con los objetivos inicialmente propuestos, con lo cuál podemos decir que la aplicación cumple en toda regla cada una de las especificaciones y se comporta de acuerdo a lo esperado.

Por último, otro de los aspectos a destacar es el hecho de saber escuchar a los demás, de conocer las necesidades del cliente para poder llevar a la práctica todas sus necesidades. Este es un aspecto muy importante, ya que si no sabemos escuchar opiniones, críticas o sugerencias, jamás podremos elaborar un sistema que cubra todas las necesidades a las que iba dirigido. Esto último es lo que más nos ha ayudado a avanzar con el proyecto, y lo que más nos ha hecho aprender.

Apéndice A

Manual de Instalación

A.1. Descargando OpenConference

En primer lugar tienes que descargarte la última versión de OpenConference. Para hacerlo ingresa en la forja donde se encuentra alojada la aplicación [12], o si lo prefieres puedes obtenerlo mediante subversion, ingresando en consola el siguiente comando:

```
svn checkout https://forja.rediris.es/svn/openconference
```

Una vez que hayas descargado la última versión de Openconference, tendrás que subir los archivos al servidor:

1. Descomprimir los archivos usando el programa que más le guste (si no conoces ninguno, te recomendamos *Winrar*).
2. Subir los archivos al servidor web (puedes utilizar *FileZilla* o el cliente que más le guste).

A.2. Ejecutando el instalador

Una vez que los archivos se han cargado correctamente en el servidor, necesitas ejecutar el instalador de OpenConference, el cuál hará el trabajo duro por ti.

1. Visita la web donde tu has cargado los archivos - por ejemplo:
`http://www.tusitio.com/install/index.php`
Lo resaltado en negrita es lo que debes añadir a tu url para que se ejecute el instalador de OpenConference.
2. En la pantalla que aparece, haz click en el enlace **Instalar OpenConference**.
3. Una vez hecho el paso anterior, verás una página como la siguiente:



Figura A.1: Instalación de OpenConference

4. A continuación, presiona el botón "**Comenzar Instalación**", y deberías ver una página como la siguiente:



Figura A.2: Paso 1 (Aviso) - Instalación de OpenConference

5. Ahora necesitas cambiar los permisos de la carpeta config/ para que todos puedan escribir sobre ella (chmod 777). Una vez que hayas cambiado los permisos, haz click en el botón "**Actualizar**", y entonces el símbolo amarillo de advertencia cambiará por un stick verde:



Figura A.3: Paso 1 (Correcto) - Instalación de OpenConference

6. Haz click en el botón **Paso 2** y aparecerá lo siguiente:

The screenshot shows the 'OpenConference' installation interface, specifically 'Paso 2 / 3'. The form is divided into several sections:

- Configuración Base de Datos:** Fields for Database hostname (localhost), Database username, Database password, and Database name.
- Configuración Básica:** URL (http://localhost/opencon/), and a checked checkbox for 'Enable SEO URLs' with a note: '(seleccionalo si tu servidor tiene activado el módulo mod_rewrite)'. There is also a 'Paso 1' button.
- Configuración Web:** Fields for Titulo, Descripción, and Email Administrador. A checked checkbox for 'Permitir Registros'. Two numeric input fields for 'Noticias por página' and 'Enlaces por caja', both set to 5, with '(por defecto = 5)' next to them.
- Detalles Administrador:** Fields for Username Administrador, Display Name, Password, and Email.

At the bottom right is an 'Instalar >>' button. The footer says 'Powered by OpenConference'.

Figura A.4: Paso 2 - Instalación de OpenConference

Necesitarás rellenar el formulario anterior. Aquí ofrecemos una explicación de cada campo:

Configuración Base de Datos:

- **Database hostname** - el hostname de tu base de datos (comúnmente no tendrás que cambiar que cambiar el valor que viene por defecto (**localhost**)).
- **Database username** - el username de tu base de datos.
- **Database password** - el password para tu username.
- **Database name** - nombre de la base de datos donde se almacenarán las tablas de OpenConference.

Configuración Básica:

- **URL** - URL a la ruta de tu página, por ejemplo:
http://www.tusitio.com/openconference/ (es obligatorio colocar la última barra).

- **Enable SEO URLs** - elige esta opción si tu servidor web tiene activado el módulo mod_rewrite (el instalador debería detectar automáticamente esta propiedad y seleccionar/deseleccionar el checkbox por ti).

Configuración Web:

- **Título** - el título de tu página (por ejemplo: "**Jornadas de Software Libre**").
- **Descripción** - la descripción de la página o fecha del evento (por ejemplo: "Del 25 al 28 de Marzo en Cádiz").
- **Email Administración** - dirección que aparecerá en los emails que envíes.
- **Permitir Registros** - si quieres que los usuarios puedan registrarse en la página, deja el stick seleccionado.
- **Noticias por página** - número de noticias que se mostraran cada vez en una página.
- **Enlaces por caja** - número de enlaces que se mostrarán en la caja del panel lateral destinada a enlaces de interés.

Detalles Administrador

- **Username Administrador** - username que usarás para loguearte en el sistema.
- **Display name (opcional)** - nombre que se mostrará a los demás usuarios.
- **Password** - password que usarás para loguearte en el sistema (elígelo prudentemente).
- **Email** - tu dirección de Email.

Cuando hayas rellenado todos los campos, presiona el botón "**Instalar**".

7. Si todo fue bien, OpenConference se instaló correctamente y saldrá una página como la siguiente:



Figura A.5: Paso 3 - Instalación de OpenConference

A.3. Borrando la carpeta install

Por último, lo único que tiene que hacer es eliminar la carpeta **install**. Cuando lo hagas, dirígete a la dirección de tu página y verás el post de bienvenida de Openconference:



Figura A.6: OpenConference instalado

¡Felicidades, la aplicación fue correctamente instalada!

Ahora puedes acceder al sistema con el **username** y **password** que facilitaste en el **Paso 2** de la instalación, y comenzar a gestionar tus propias conferencias.

Apéndice B

Manual de Usuario

En este manual damos detalle de cada una de las instrucciones y pasos que el usuario debe realizar para sacar provecho a cada una de las funcionalidades que ofrece el sistema OpenConference. Con este manual el usuario aprenderá a usar correctamente la aplicación, y resolverá cualquier duda que se le plantee durante el uso de la misma.

B.1. Ejecución

Una vez que la aplicación ha sido correctamente instalada en el servidor web, para ejecutarla tan sólo debe abrir el navegador web que más le agrade, e introducir la dirección donde se aloja la página. Una vez hecho esto, le aparecerá la página de inicio de la aplicación, cuya apariencia dependerá del estilo que usted elija para la misma (más adelante se explicará cómo hacerlo). A continuación mostramos un ejemplo, escogiendo el diseño **Vector Lover**¹ como estilo predeterminado.



Figura B.1: Página de Inicio de OpenConference - Diseño Vector Lover

¹En todos los ejemplos del manual se ha utilizado este estilo como apariencia predeterminada.

B.2. Primeros Pasos

Si acaba de instalar la aplicación y es la primera vez que accede a la web, le aparecerá un post de bienvenida al sistema y la única pestaña que aparecerá en el menú de Navegación será **Home**. A la derecha podrá visualizar el panel lateral, que dependiendo de los elementos que haya activos tendrá una apariencia u otra. Como norma general, inicialmente le aparecerá la caja desde la cuál podrá enviar Papers, la caja de enlaces de interés, y la caja desde la cuál podrá registrarse o loguearse en el sistema.

B.2.1. Acceder al sistema

Para acceder a todas las funcionalidades que nos ofrece el sistema es necesario estar registrado previamente. Para hacerlo tan sólo debemos pulsar en el enlace **Registro**, situado en la caja **Otros** del panel lateral. Una vez dentro nos aparecerá una página como la siguiente:

Registro

Por favor, rellena el formulario y pulsa el botón Registrar

Datos Personales

Username *

Display Name

Nombre *

Apellidos *

Password

Password *

Repite password *

Contacto

Email *

Website

Registrar >>

Papers

Enviar Paper

Enlaces

Wikipedia

UCA

Otros

Login

Registro

OpenConference

Figura B.2: Página de Registro de OpenConference - Diseño Vector Lover

En ella tendrás que introducir algunos datos como el nick (username), display name (nombre que verán los demás en el sistema), nombre y apellidos reales (a partir de estos datos se generarán los certificados y el calendario de ponencias en su caso), password, email y website. Los campos que poseen un asterisco de color rojo a su derecha indican que son obligatorios.

Una vez rellenado el formulario, pulsamos el botón **Registrar**. Si todo ha salido bien recibirás un correo, a la dirección email que proporcionaste, con los datos del registro y un enlace que deberás visitar para que la cuenta se active. Una vez hecho esto, ya podrás loguearte en el sistema y empezar a usar todas las funcionalidades.

Para loguearte en el sistema tan sólo debes pulsar el enlace **Login**, situado en la caja **Otros** del panel lateral. Aparecerá una pantalla como la siguiente:

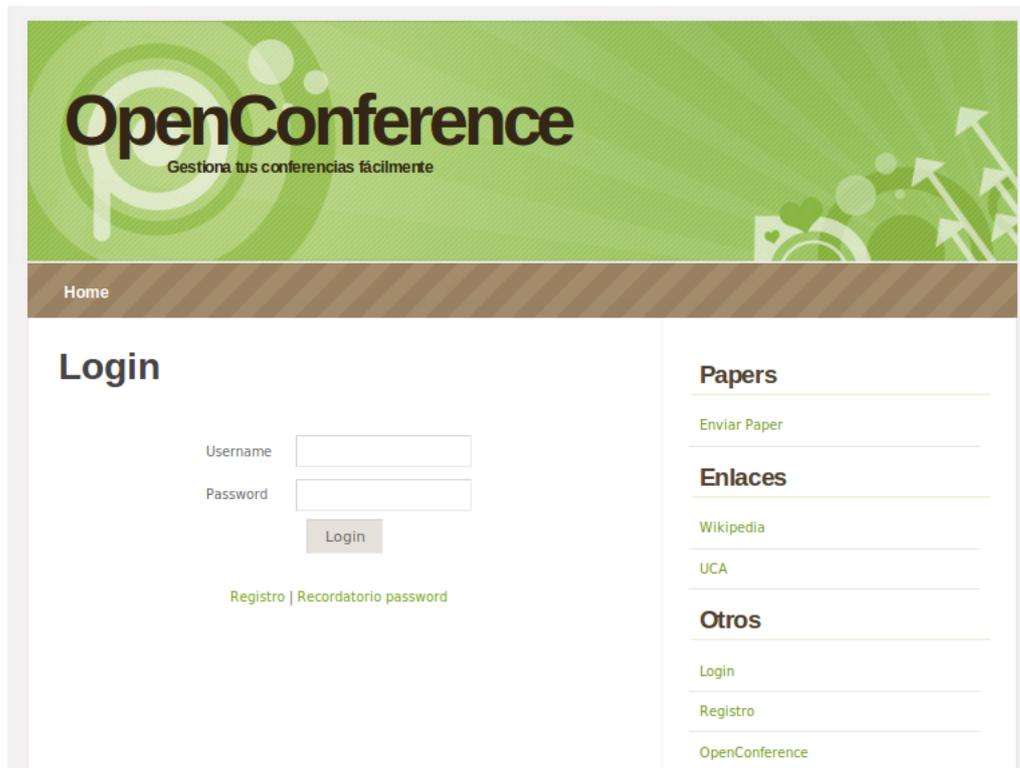


Figura B.3: Página de Login de OpenConference - Diseño Vector Lover

Aquí tendrá que introducir el username y el password que proporcionó al registrarse. Si por alguna razón no recuerda su password, podrá solicitar uno nuevo pulsando en el enlace **Recordatorio Password** situado justo abajo del botón **Login**. También podrá registrarse si aún no lo ha hecho.

Si usted posee un nivel de acceso superior al del usuario cotidiano, es decir, si usted es administrador o revisor, cuando se loguee en el sistema le aparecerá un nuevo enlace en la caja **Otros** del panel lateral. Este enlace es **Panel de Administración** y da acceso al núcleo principal de la aplicación, desde donde se controla todo el sistema.

Además del nuevo enlace anterior, también aparecerán otros tres nuevos (independientemente del nivel de acceso que posea) enlaces, **Ver Perfil**, **Editar Perfil** y **Cerrar Sesión**, que los vemos a continuación.

B.2.2. Ver Perfil

Si pulsamos el enlace **Ver Perfil** accederemos a una página donde se nos mostrará información sobre los datos de usuario, tales como Username, Nombre, Apellidos, Fecha de Registro, Website...El ejemplo, a continuación:



Figura B.4: Ver Perfil de Usuario en OpenConference- Diseño Vector Lover

B.2.3. Editar Perfil

Al acceder a esta página, tendremos permiso para modificar los datos de nuestro perfil y aportar nuevos valores, es decir, podremos cambiar nuestro nick, nuestro password, nombre o apellidos, etc...Aquí tenemos un ejemplo:

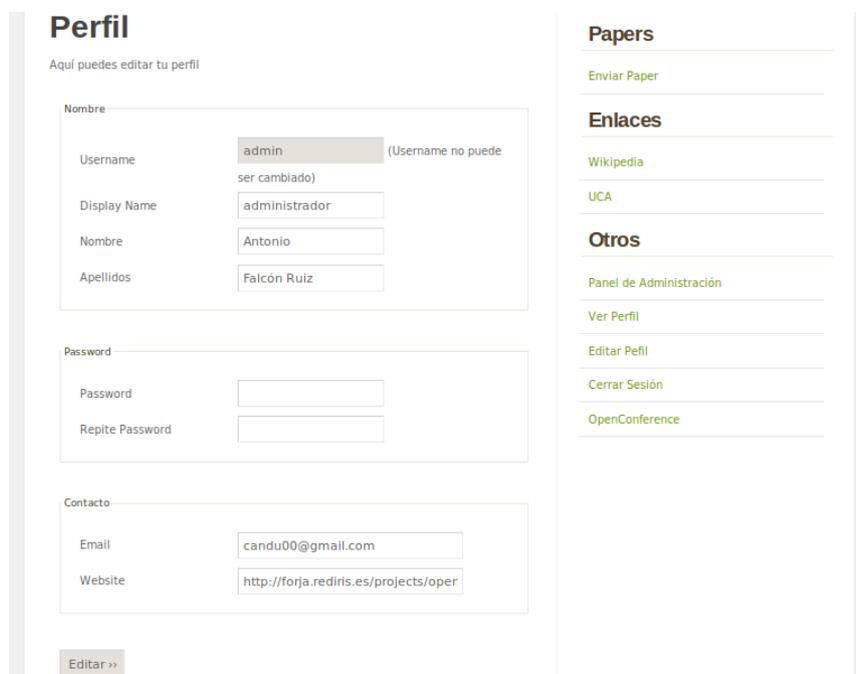


Figura B.5: Editar Perfil de Usuario OpenConference - Diseño Vector Lover

B.2.4. Salir del sistema

Si pulsamos sobre el enlace **Cerrar Sesión**, saldremos de manera segura del sistema. Es conveniente siempre cerrar sesión cada vez que salgamos de la aplicación, ya que así el sistema borrará todos los datos que hayan quedado grabados mientras navegábamos por la Web.

B.3. Administrar Posts y Comentarios

A partir de este apartado en adelante nos centramos en las funcionalidades que ofrecen los diferentes sistemas que se incluyen dentro del Panel de Administración, al cuál solo tienen acceso los revisores y los propios administradores.

En el presente apartado explicamos cómo crear nuevos posts, cómo borrarlos, cómo editarlos y también aprenderemos a borrar y editar los comentarios que los usuarios realizan sobre las noticias.

B.3.1. Posts

Para administrar las noticias que se publican en el sistema, accederemos al apartado **Posts** a través del icono de la consola central o a través del enlace **Administrar Posts**, situado en el menú de navegación de la izquierda. Una vez dentro, nos aparecerá una pantalla como la siguiente:



The screenshot shows the OpenConference administration interface. At the top, it says 'OpenConference Panel de Administración'. Below that, it indicates the user is logged in as 'admin (administrador)' with links for 'Ver sitio' and 'Cerrar Sesión'. The main content area is divided into two columns. The left column is a sidebar menu with 'Administración' selected, containing links for 'Control', 'Settings', 'Templates', 'Sidebar', 'Backup', 'Posts', 'Nuevo Post', and 'Administrar Posts'. The right column is titled 'Posts' and contains the text 'Aquí puedes administrar los posts'. Below this is a table with columns for 'Fecha posteo', 'Título', 'Estado', and 'Acciones'. The table lists three posts: 'Sticky' (published 2009-09-16, 3 comments), 'Borrador' (published 2009-09-18, 0 comments), and 'Bienvenido a OpenConference' (published 2009-09-15, 0 comments). Each row has edit and delete icons. A 'Nuevo post' link is also visible.

Fecha posteo	Título	Estado	Acciones
2009-09-16	Sticky	3 publicado	
2009-09-18	Borrador	0 publicado	
2009-09-15	Bienvenido a OpenConference	0 publicado	

Figura B.6: Administrar Posts en Openconference

Ahí vemos información relativa a cada noticia, como la fecha de publicación, el título, los comentarios que se han hecho sobre la misma, el estado en el que se encuentra (publicado o borrador), y también las acciones que podemos realizar sobre ella. Entre las acciones están: Editar y Eliminar.

todas las noticias, y nos aparecerá una página idéntica a la de **Editar Post**, salvo que esta vez no habrá ningún dato que editar, sino que los tendremos que proporcionar nosotros.

B.3.2. Comentarios

De la misma manera podemos administrar los comentarios que los usuarios van haciendo sobre las diferentes noticias que se publican en el sistema. Si accedemos al apartado **Comentarios** desde el Panel de Administración, nos aparecerá una página con información detallada de todos los comentarios que se han hecho, como por ejemplo, la fecha en que se escribió el comentario, el título del post al cuál se le hizo el comentario, el usuario que comentó dicho post, el contenido del comentario, y además nos aparecerán las acciones que podemos llevar a cabo sobre un comentario: Editar y Eliminar. La página con el listado de comentarios es la siguiente:

Administración	Comentarios				
Control	Aquí puedes administrar comentarios				
Settings	Fecha posteo	Título post	Autor	Comentario	Acciones
Templates	06-10-2009	Sticky	administrador	Me parece bien. Es muy buena noticia...	 
Sidebar	30-09-2009	Sticky	juan	Muy buena noticia.	 
Backup	19-09-2009	Sticky	falcon	Buena noticia.	 

Figura B.8: Administrar Comentarios en OpenConference

Tanto la edición como la eliminación se llevan a cabo de la misma manera que en el sistema de noticias (Posts).

B.4. Administrar Usuarios

Para administrar los usuarios registrados en el sistema tan sólo tendremos que acceder al apartado **Usuarios**, ya sea desde el icono de la consola central del Panel de Administración, o desde el enlace **Administrar Usuarios** situado en el menú de navegación de la izquierda.

Una vez que accedemos, nos aparece una página con un listado de todos los usuarios que hay registrados en el sistema, donde se nos muestra información de cada uno de ellos, como la fecha de registro, los datos personales, el nivel de acceso a la aplicación...y un conjunto de acciones que podemos llevar a cabo sobre ellos: Editar, Eliminar y Generar Certificado.

Administración		Usuarios				
Control	Aquí puedes administrar a los usuarios					
Settings	Fecha de registro	Username	E-mail	Nivel	Nombre	Acciones
Templates	15-09-2009	falconeti	morenita_03@msn.com	revisor	Antonio Falcón Aragón	  
Sidebar	14-09-2009	admin	candu00@gmail.com	administrador	Antonio Falcón Ruiz	
Backup	26-09-2009	anesfa	estefaniacorbeni@hotmail.com	usuario	Estefanía Cordero Benítez	  

Figura B.9: Administrar Usuarios en OpenConference

Si queremos **editar** los datos de un usuario pulsamos sobre el icono de lápiz y papel, y nos aparecerá una página donde podremos modificar los datos e introducir unos nuevos. Aquí tenemos un ejemplo:

Administración		Editar usuario	
Control	Aquí puedes editar un usuario		
Settings	Nombre		
Templates	Username	falconeti	(El username no puede ser cambiado)
Sidebar	Display name	A.Falcon	
Backup	Nombre	Antonio	
Posts	Apellidos	Falcón Aragón	
Nuevo Post	Nivel		
Administrar Posts	Nivel	Revisor	
Páginas	Información de contacto		
Nueva Página	E-mail		
Administrar Páginas	Sitio Web		
	Editar >>		

Figura B.10: Editar Usuario en OpenConference

Para **eliminar** un usuario no hay más que pulsar en el símbolo de las dos aspas en color rojo, y nos aparecerá un mensaje de advertencia preguntándonos si queremos verdaderamente eliminar a dicho usuario.

Por último, el sistema también nos permite generar certificados de asistencia de cada uno de los participantes. Para **generar el certificado** sólo hay que pulsar en el símbolo de PDF, que hay junto al símbolo de **eliminar**, y nos aparecerá una ventana en el navegador para descargar el certificado a nuestro ordenador. El diseño del certificado generado dependerá del que hayamos seleccionado previamente (se verá posteriormente en el apartado de Crear Certificado).

B.5. Administrar templates

En esta ocasión nos centramos en el sistema para configurar el diseño que queremos darle a la página principal de la Web. Inicialmente disponemos de tres diseños a elegir, aunque el usuario puede perfec-

tamente crearse su propio diseño e integrarlo en OpenConference (se mostrará posteriormente, en el manual para desarrolladores). Una vez que accedemos al apartado de **Templates** nos aparece una página como la siguiente, donde podremos escoger entre los diferentes diseños existentes:

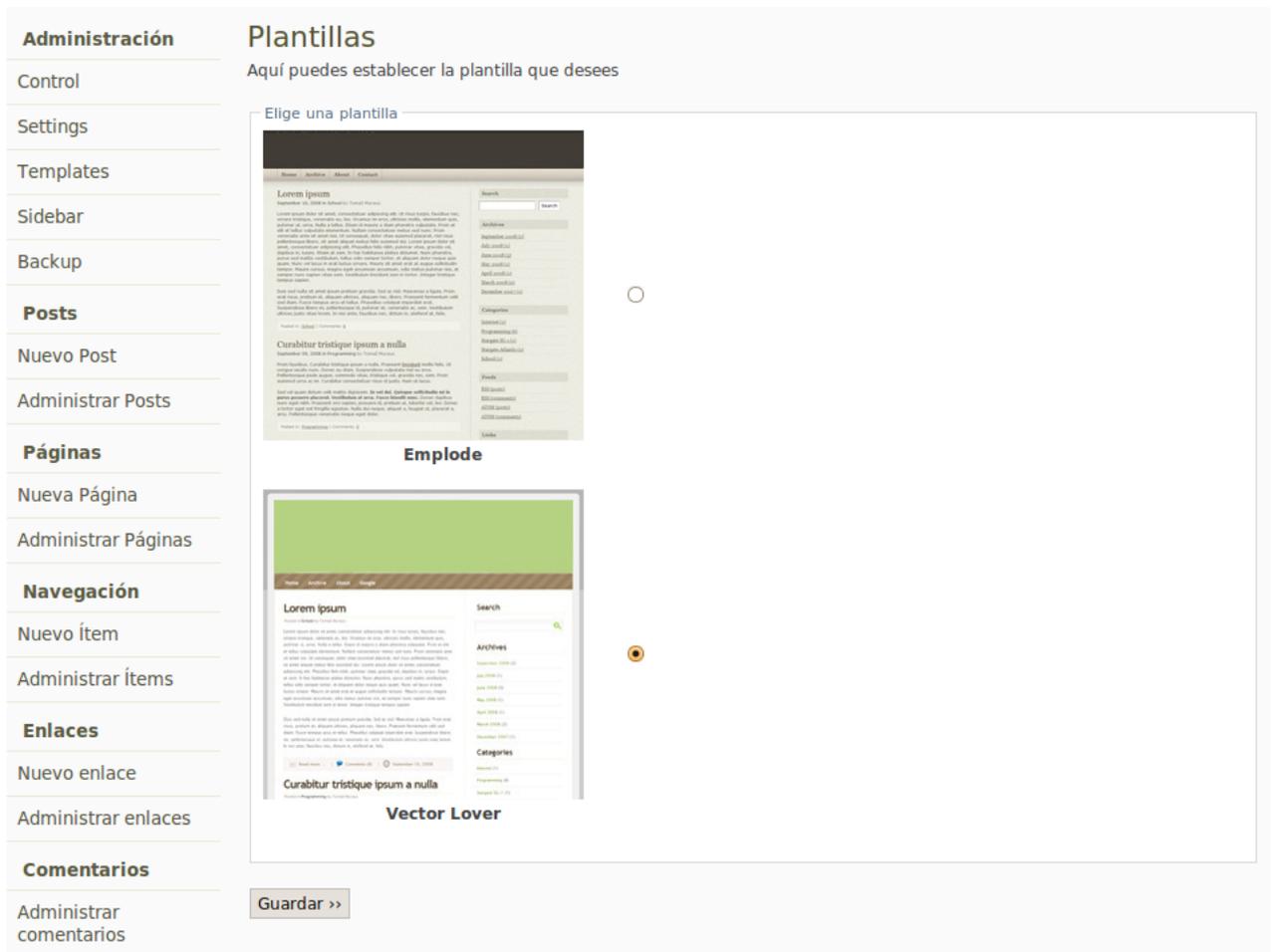


Figura B.11: Administrar Templates en OpenConference

B.6. Administrar Páginas y Navegación

En el presente apartado explicamos cómo crear nuevas páginas, cómo borrarlas, cómo editarlas y también aprenderemos a crear, borrar y editar los items del menú de navegación de la página principal.

B.6.1. Páginas

Si accedemos al apartado para administrar **Páginas**, veremos una lista con todas las páginas que hemos creado y obtendremos información sobre cada una de ellas, como la fecha de creación, el nombre de la página, el estado, y las acciones que podemos realizar sobre ellas: Editar y Eliminar.

Si queremos **Editar** una página sólo tenemos que pulsar en el icono de lápiz y papel y nos aparecerá una pantalla donde modificaremos la información ahí presentada. Aquí tenemos un ejemplo:

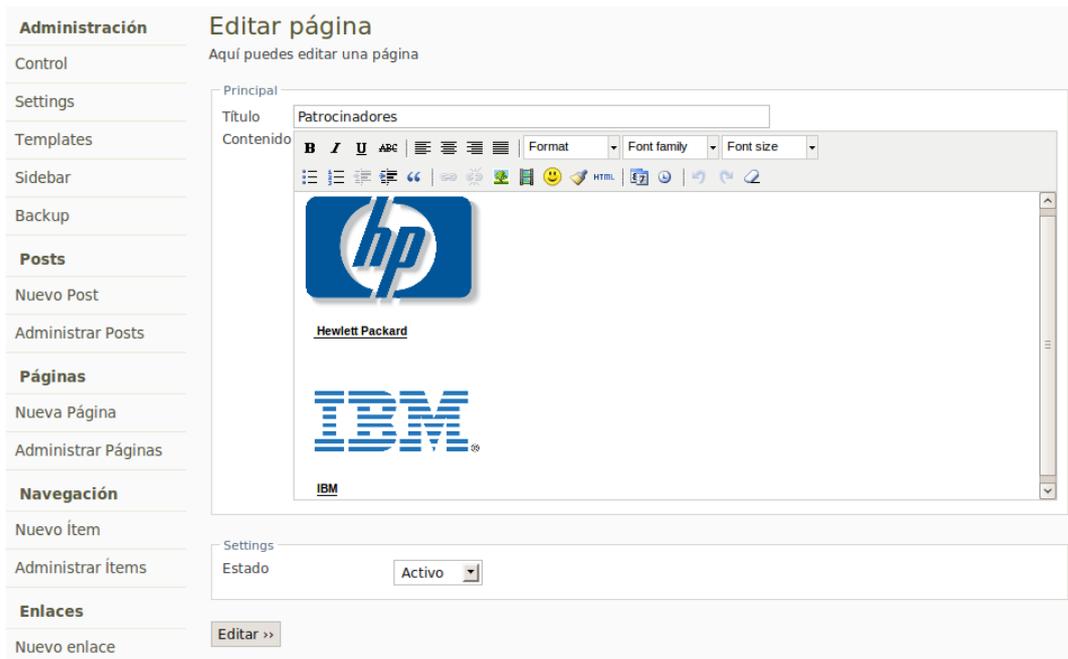


Figura B.12: Editar Página en OpenConference

Para eliminar una página, seguiremos los mismos pasos que anteriores ocasiones.

Si lo que queremos es **crear una nueva página** pulsaremos en el enlace que nos aparece justo debajo de la lista con todas las páginas, y nos aparecerá una pantalla como la siguiente:

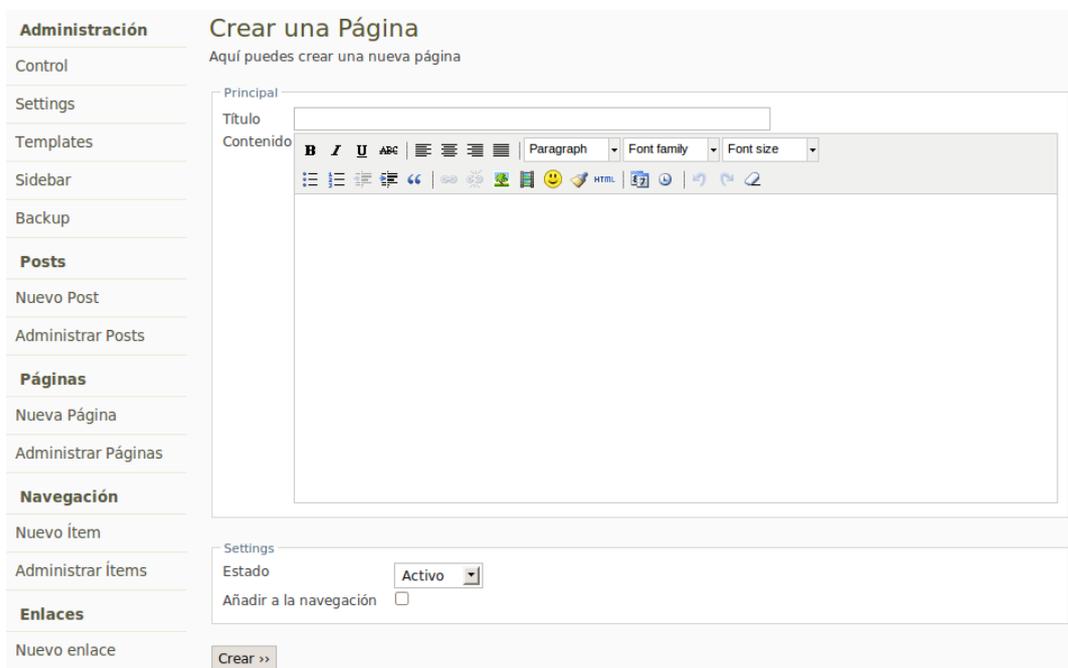


Figura B.13: Crear Página en OpenConference

Aquí proporcionaremos datos como el título de la nueva página, el contenido de la misma, el estado y podremos activar una opción para que la página se añada automáticamente al menú de navegación, que vemos a continuación.

B.6.2. Navegación

En este apartado podremos administrar los items que aparecerán en el menú de navegación de nuestra página. Debemos tener en cuenta que si al crear una nueva página, seleccionamos la opción **Añadir a la Navegación**, nos aparecerá automáticamente en la lista de items, como vemos a continuación:

Administración	Navegación		
Control	Aquí puedes administrar la navegación		
Settings	Título	Descripción	Acciones
Templates	Home	Index	
Sidebar	Patrocinadores	Patrocinadores	
Backup	Nuevo ítem		

Figura B.14: Administrar Navegación en OpenConference

En dicha lista aparecen todos los items que actualmente forman parte del menú de navegación, ofreciéndonos información como el título, una breve descripción y las acciones que podemos realizar sobre ellos: Editar y Eliminar.

Si queremos **editar** un item de navegación pulsaremos en el icono correspondiente y nos aparecerá una página como la siguiente:

Administración	Editar Ítem	
Control	Aquí puedes editar un ítem	
Settings	Editar Ítem	
Templates	Título	<input type="text" value="Patrocinadores"/>
Sidebar	URL	<input type="text" value="paginas/Patrocinadores/"/>
Backup	Descripción	<input type="text" value="Patrocinadores"/>
	<input type="button" value="Editar >>"/>	

Figura B.15: Editar Item de Navegación en OpenConference

Ahí observamos datos como el título, la url (ya sea interna o externa), y una breve descripción.

Para **eliminar** un item seguimos los mismos pasos que hasta ahora, y para **crear un nuevo ítem** sólo debemos pulsar el enlace situado justo debajo de la lista con todos los items, y nos aparecerá una página idéntica a la de **editar ítem**, pero esta vez sin ningún dato. Debemos destacar que si queremos añadir una página **interna** a la navegación, en la creación de dicha página no debe haberse seleccionado la opción **Añadir a la Navegación**. Por otro lado, el formato que debe tener una url interna es el siguiente: **paginas/nombrepagina/** . Siendo **nombrepagina**, el título que le dimos a la página al crearla (respetando mayúsculas, acentos y demás símbolos).

También podemos crear un nuevo ítem que apunte a una página externa, para lo cuál, la url debe tener el siguiente formato: <http://www.ejemplo.com>.

B.7. Administrar CFP

En este apartado aprenderemos a administrar los papers que los usuarios van enviando a la web, para que las personas responsables los revisen y evalúen con el fin de poder participar en las conferencias. Una vez que accedemos al apartado **CFP** desde el Panel de Administración, aparece una página con un listado de todos los papers enviados al sistema, e información como la fecha del envío, el autor del paper, el título, la temática, el estado en el que se encuentra (recibido, en revisión, revisado), y la evaluación que se le ha hecho (una nota del 1 al 5). Además vemos un conjunto de acciones que podemos realizar sobre cada paper: Revisar, Ver y Eliminar. Aquí mostramos un ejemplo:

Administración		CFP				
Control	Aquí puedes administrar los CFP					
Settings	Fecha de envío	Título	Temática	Estado	Valoración	Acciones
Templates	25-09-2009	Paralelismo	Arquitectura de Computadores	revisado	5	  
Sidebar	25-09-2009	Guernica	Arte	revisado	1	  
Backup	25-09-2009	Oracle	Base de Datos	revisado	1	  
	25-09-2009	MySQL	Bases de Datos	revisado	4	  
	25-09-2009	Sistemas Expertos	Inteligencia Artificial	revisado	3	  
Posts	06-10-2009	OpenConference	Desarrollo Web	revisado	3	  
Nuevo Post	10-10-2009	Tecnología Avanzada de Base de Datos	Base de Datos	recibido	-	  
Administrar Posts	10-10-2009	Redes Inalámbricas	Redes y Comunicación	recibido	-	  
	11-10-2009	Sistemas Operativos	S.O	recibido	-	  

Figura B.16: Administrar Papers en OpenConference

Para ver la revisión que han hecho sobre un paper determinado pulsamos sobre el icono de la lupa, y nos aparecerá una pantalla con la valoración que los revisores han hecho sobre el paper. Se muestra a continuación:

Figura B.18: Ver Valoración Paper en OpenConference

Por último, para eliminar un paper sólo hay que pulsar sobre el icono con las dos aspas en rojo, y nos aparecerá el mensaje de advertencia preguntándonos si realmente deseamos eliminar dicho paper.

B.8. Crear Calendario

Para crear el calendario con la fecha y hora de cada ponencia, tan sólo debemos entrar en el apartado **Calendario** del Panel de Administración. Una vez dentro obtendremos una pantalla como la siguiente:

Ponencia	Ponente	Día	Hora
MySQL	Antonio Falcón Ruiz	1	15:00
Paralelismo	Antonio Falcón Ruiz	1	10:30
Guernica	Antonio Falcón Ruiz	1	11:00
Oracle	Antonio Falcón Ruiz	1	12:00
Sistemas Expertos	Antonio Falcón Ruiz	2	10:00
OpenConference	Antonio Falcón Ruiz	2	13:00
Tecnología Avanzada de Base de Datos	Antonio Falcón Ruiz	3	09:30
Redes Inalámbricas	Antonio Falcón Ruiz	3	10:30
Sistemas Operativos	Antonio Falcón Aragón	3	12:30

Figura B.19: Administrar Calendario en OpenConference

Ahí observamos en primer lugar una lista con las ponencias. Esas son las ponencias que ya han sido agregadas al calendario, que pueden ser editadas simplemente seleccionándola en la lista desplegable

que aparece más abajo y cambiando el día y la fecha en la que se llevará a cabo la ponencia. Para agregar más ponencias al calendario sólo debemos buscarla en la lista desplegable que aparece en la parte inferior y seleccionar un día y una hora. No hace falta hacerlo ordenadamente, el sistema ordenará el calendario según el día y la hora.

Una vez que se han agregado todas las ponencias al calendario es hora de generarlo. Para ello pulsamos en el enlace **Crear Calendario**, que aparece en la parte inferior de la página, y automáticamente se nos abrirá un archivo en formato .PDF para descargarlo. El archivo tendrá una apariencia similar a esta:

Día	Hora	Ponencia	Ponente
1	10:30	Paralelismo	Antonio Falcón Ruiz
1	11:00	Guernica	Antonio Falcón Ruiz
1	12:00	Oracle	Antonio Falcón Ruiz
1	15:00	MySQL	Antonio Falcón Ruiz
2	10:00	Sistemas Expertos	Antonio Falcón Ruiz
2	13:00	OpenConference	Antonio Falcón Ruiz
3	09:30	Tecnología Avanzada de Base de Datos	Antonio Falcón Ruiz
3	10:30	Redes Inalámbricas	Antonio Falcón Ruiz
3	12:30	Sistemas Operativos	Antonio Falcón Aragón

Figura B.20: Muestra de Calendario generado en OpenConference

B.9. Crear Certificado

Este sistema nos permitirá crear nuestro propio diseño de un certificado, para posteriormente generar los certificados de todos los participantes.

Al acceder a este apartado, nos aparecerá una página como la siguiente:

Figura B.21: Administrar Certificados en OpenConference

En dicha página podemos seleccionar la plantilla de certificado que deseemos, y a partir de la cuál se generarán los certificados. Para seleccionar la plantilla no hay más que buscarla en la lista desplegable y luego pulsar el botón **Editar**, con lo que automáticamente se establecerá como plantilla por defecto.

Si queremos añadir nuestra propia plantilla de certificado sólo tenemos que pulsar sobre el enlace **Inserir Nueva Plantilla** y nos aparecerá una página en la que podremos subir nuestro diseño.

La plantilla de certificado debe estar en formato .rtf o .pdf. Generar la plantilla en formato .rtf es muy sencillo, podremos utilizar para ello *Writer* de **OpenOffice**, por ejemplo. Generar el .pdf requiere más trabajo, pero no deja de ser sencillo. Podemos crear la plantilla en formato .odt con *Writer* y posteriormente imprimirla en formato .pdf con alguna herramienta que nos permita editar las propiedades de impresión. El archivo se debe guardar en formato ASCII y es necesario desactivar la compresión. Una vez configuradas estas propiedades ya podemos generar el pdf.

A la hora de crear el diseño, tan sólo hay que tener en cuenta la parte dinámica del certificado, es decir, lo que va a ser diferente en cada uno de ellos. Concretamente, en nuestro caso, lo único que debemos hacer dinámico es el nombre del usuario que va a recibir el certificado. Para ello basta con poner el marcador de posición «**Name**» donde queramos que aparezca el nombre. El resto del diseño es estático.

A continuación mostramos un ejemplo de certificado generado con OpenConference. La plantilla .pdf ha sido extraída del material que viene con el libro *Desarrollo Web con PHP y Mysql*[5].



Figura B.22: Ejemplo de Certificado generado en OpenConference

B.10. Cambiar configuración de la página

Para cambiar la configuración de la página, sólo hay que acceder al apartado **Settings** del Panel de Administración. Una vez dentro, observaremos una página como la siguiente:

Administración	
Control	
Settings	
Templates	
Sidebar	
Backup	
Posts	
Nuevo Post	
Administrar Posts	
Páginas	
Nueva Página	
Administrar Páginas	
Navegación	

Settings

Aquí puedes administrar la configuración de la aplicación

Configuración general

Título Web	OpenConference
Descripción Web	Gestiona tus conferencias fácilmente
Email Administrador	candu00@gmail.com
Permitir registros	<input checked="" type="checkbox"/>

Estado del sitio

Activo	<input checked="" type="checkbox"/>
Razón de Inactividad	Mantenimiento

Configuración Diseño

Template	Vector Lover
Posts por página	5 (por defecto = 5)
Enlaces	5 (por defecto = 5)

Guardar >>

Figura B.23: Configuración de OpenConference

Aquí encontraremos características como el título de la página web, la descripción, el email del administrador, si es posible registrarse en el sistema, si el sitio está activo o inactivo y la razón de inactividad, el template por defecto que cambiará la apariencia de la página, el número de noticias por página, etc...

B.11. Hacer copia de seguridad de los datos

Hacer un backup de la información contenida en nuestra base de datos es realmente sencillo en OpenConference. Sólo debemos pulsar en el icono **Backup** de la consola central del panel o en el enlace con el mismo nombre, del menú de navegación de la izquierda. En cualquier caso, al pulsar se generará automáticamente un archivo .sql con la información de la base de datos. El navegador abrirá una ventana desde la cuál podremos descargar dicho archivo a nuestro ordenador.

Apéndice C

Manual del Desarrollador

Antes de introducirnos en el desarrollo de este manual debemos destacar que el mismo está dirigido a personas con cierto nivel en el lenguaje de programación PHP y que tenga conocimientos en CSS y HTML, para que así puedan crear sus propios diseños y adaptarlos a **OpenConference**, como veremos a continuación.

C.1. Estructura de Carpetas

OpenConference posee una jerarquía muy clara de carpetas, para facilitar la ampliación o modificación de alguna funcionalidad a todo aquel desarrollador que lo desee. La aplicación posee cuatro carpetas principales, cada una de ellas con un contenido específico, que pasamos a comentar a continuación:

application En esta carpeta almacenamos todo el código de **OpenConference**. Tanto librerías, controladores, módulos, vistas...se incluyen dentro de este directorio. Aquí se incluyen otras siete carpetas:

config En esta carpeta encontramos todos los archivos de configuración de la aplicación.

errors Aquí se incluyen los archivos que se mostrarán cuando se produzca algún tipo de error en la aplicación.

helpers En esta carpeta incluimos el código que va a ser utilizado frecuentemente, funciones de ayuda, etc...

language En caso de internacionalizar, debemos crear una carpeta por cada idioma, donde alojamos las distintas traducciones.

libraries Aquí se almacenan tanto las librerías de propia creación como las que trae *CodeIgniter* por defecto.

modules Esta carpeta aloja en su interior los distintos módulos que componen **OpenConference**. Dentro de cada módulo tenemos dos carpetas más, que son: *Controllers* y *Models*. Que respectivamente alojan tanto los controladores como los módulos del sistema en cuestión.

views Esta carpeta también se divide en módulos, debido a la naturaleza del sistema, y dentro de cada módulos nos encontramos con los archivos necesarios para implementar la vista de la aplicación, lo que se mostrará al usuario.

doc En esta carpeta se incluye toda la documentación del sistema, e incluso la presente memoria y todos los manuales.

system En esta carpeta están alojados todos los archivos para configurar correctamente el framework *CodeIgniter*.

En el presente manual nos centraremos en desarrollar nuestros propios diseños, para incluirlos en *OpenConference*, para ello nos situaremos en el directorio *application/views/templates/*.

C.2. Introducción de Nuevo Diseño en OpenConference

Para crear un nuevo diseño para nuestra aplicación tan sólo debemos respetar algunas normas. La principal es establecer una jerarquía de carpetas bien definidas. Si accedemos al directorio *application/views/templates/* nos encontraremos con tres carpetas: *explode*, *vector_lover* y *colorvoid*. Esas carpetas representan a los tres diseños que ya están incluidos en **OpenConference**.

A continuación veremos los pasos a seguir para lograr crear e incorporar nuestro propio estilo a **OpenConference**.

C.2.1. Paso 1: Creación de Directorios

Si queremos incluir un nuevo diseño, no hay más que crear un directorio en el mismo nivel en el que se encuentran las tres carpetas mencionadas anteriormente. Una vez creado el directorio cuyo nombre sea, por ejemplo, **nuevo_estilo**, accederemos a él. Por tanto ahora nos encontraremos en el directorio *application/views/templates/nuevo_estilo/*.

Una vez dentro de este directorio, tenemos que crear dos nuevas carpetas con los siguientes nombres: **layout** y **static**.

Dentro de la carpeta **static** incluiremos la hoja de estilo determinada, las imágenes, iconos y demás recursos que formen parte de nuestro nuevo diseño. Así, dentro del directorio **static**, crearemos otros dos con los siguientes nombres: **images** y **style**.

Dentro de **images** se alojarán todas las imágenes, iconos y todos los gráficos que necesitemos para crear nuestro diseño. Y dentro del directorio **style** incluiremos la/s hoja/s de estilo que nos hagan falta para maquetar y crear la apariencia que más nos guste.

Una vez creado el contenido de todas estas carpetas nos dirigimos al directorio *application/views/templates/nuevo_estilo/layout/*. Aquí dentro tendremos que crear dos nuevos directorios, **menu** y **pages** y tres nuevos archivos, **container.php**, **header.php** y **footer.php**. En el siguiente paso veremos el contenido de cada uno de ellos.

C.2.2. Paso 2: Adaptar a OpenConference

Aquí veremos el contenido de cada archivo y cada directorio que se encuentra dentro de *application/views/templates/nuevo_estilo/layout/*. El contenido de los archivos que mostraremos a continuación forman parte del estilo **explode**. El contenido que debemos crear nosotros es similar al que aquí se muestra, lo único que debemos modificar serán los atributos de los elementos html, para adaptarlos a las restricciones de nuestra/s hoja/s de estilo.

Archivo: container.php

```
1 <?php $this->system_library->check_site_status(); ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
   w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr">
4
5 <head>
6     <meta http-equiv="content-type" content="text/html; charset=UTF-8"
   />
7     <meta name="description" content="<?php echo $this->system_library
   ->settings['descripcion_pagina']; ?>" />
8     <link rel="stylesheet" type="text/css" href="<?php echo base_url();
   ?>application/views/templates/emplode/static/style/main.css"
   media="screen" />
9     <title>
10         <?php if (isset($post['titulo'])): ?>
11             <?php echo $post['titulo']; ?>
12         <?php elseif (isset($page_data['titulo'])): ?>
13             <?php echo $page_data['titulo']; ?>
14         <?php else: ?>
15             <?php echo $this->system_library->settings['
   titulo_pagina']; ?> - <?php echo $this->
   system_library->settings['descripcion_pagina'];
   ?>
16         <?php endif; ?>
17     </title>
18 </head>
19
20 <body>
21
22 <div id="header">
23     <div class="center_wrapper">
24         <?php $this->load->view('templates/emplode/layout/header');
   ?>
25     </div>
26 </div>
27
28 <div id="navigation">
29     <div class="center_wrapper">
30         <?php $this->load->view('templates/emplode/layout/menu/
   navigation'); ?>
31     </div>
32 </div>
33
34 <div id="main_wrapper_outer">
35     <div id="main_wrapper_inner">
36         <div class="center_wrapper">
37
38             <div class="left" id="main">
39                 <div id="main_content">
40                     <?php $this->load->view('templates/
   emplode/layout/pages/' . $pagina
   ); ?>
41                 </div>
42             </div>
43         </div>
44     </div>
45 </div>
```

```

42         </div>
43
44         <div class="right" id="sidebar">
45             <div id="sidebar_content">
46                 <?php $this->load->view('templates/
                    emplode/layout/menu/sidebar');
                    ?>
47             </div>
48         </div>
49
50         <div class="clearer">&nbsp;</div>
51
52     </div>
53 </div>
54 </div>
55
56 <div id="footer">
57     <div class="center_wrapper">
58         <?php $this->load->view('templates/emplode/layout/footer');
            ?>
59     </div>
60 </div>
61
62 </body>
63 </html>

```

Archivo: header.php

```

1 <div id="site_title">
2     <h1><a href="<?php echo base_url(); ?>"><?php echo $this->
        system_library->settings['titulo_pagina']; ?></a></h1>
3     <p><?php echo $this->system_library->settings['descripcion_pagina'
        ]; ?></p>
4 </div>

```

Archivo: footer.php

```

1 <div class="left"><?php echo "Powered by"; ?> <a href="http://forja.rediris
    .es/projects/openconference" target="_blank">OpenConference</a></div>
2 <div class="right"><?php echo "Template by"; ?> <a href="http://arcsin.se/"
    target="_blank">Arcsin</a></div>
3 <div class="clearer">&nbsp;</div>

```

Archivo: menu/cfp.php

```

1 <div class="box">
2     <div class="box_title"><?php echo "Papers"; ?></div>
3     <div class="box_content">
4         <ul>

```

```

5         <li><a href="<?php echo site_url('cfp/enviar'); ?>"
6             ><?php echo "Enviar Paper"; ?></a></li>
7     </ul>
8 </div>

```

Archivo: menu/links.php

```

1 <div class="box">
2     <div class="box_title"><?php echo "Enlaces"; ?></div>
3     <div class="box_content">
4         <ul>
5             <?php if (($links = $this->links_library->get_links
6                 ()): ?>
7                 <?php foreach ($links as $link): ?>
8                     <li><a href="<?php echo $link['url
9                         ']; ?>" title="<?php echo $link
10                        ['descripcion']; ?>" target="<?
11                        php echo $link['target']; ?>"><?
12                        php echo $link['nombre']; ?></a
13                    ></li>
14                <?php endforeach; ?>
15            <?php endif; ?>
16        </ul>
17    </div>
18</div>

```

Archivo: menu/navigation.php

```

1 <?php if (($navegacion = $this->navigation_library->get_navigation()): ?>
2     <ul>
3         <?php foreach ($navegacion as $navegacion_item): ?>
4             <li><a href="<?php echo ($navegacion_item['externo
5                 ' ] == 0) ? site_url($navegacion_item['url']) :
6                 $navegacion_item['url']; ?>" title="<?php echo
7                 $navegacion_item['descripcion']; ?>"><span><?php
8                 echo $navegacion_item['titulo']; ?></span></a
9             ></li>
10        <?php endforeach; ?>
11    </ul>
12<?php endif; ?>
13<div class="clearer">&nbsp;</div>

```

Archivo: menu/other.php

```

1 <div class="box">
2     <div class="box_title"><?php echo "Otros"; ?></div>
3     <div class="box_content">
4         <ul>

```

```

5      <?php if ($this->session->userdata('logged_in') == false):
6          ?>
7          <li><a href="<?php echo site_url('usuario/login');
8              ?>"><?php echo "Login"; ?></a></li>
9          <li><a href="<?php echo site_url('usuario/registro
10             '); ?>"><?php echo "Registro"; ?></a></li>
11      <?php else: ?>
12      <?php if ($this->session->userdata('nivel') == "
13          administrador" || $this->session->userdata('nivel')==
14          "revisor"): ?>
15          <li><a href="<?php echo site_url('admin/control');
16              ?>"><?php echo "Panel de Administraci&oacute;n";
17              ?></a></li>
18      <?php endif; ?>
19      <li><a href="<?php echo site_url('usuario/ver/' .
20          $this->session->userdata('username')); ?>"><?php
21          echo "Ver Perfil"; ?></a></li>
22      <li><a href="<?php echo site_url('usuario/perfil');
23          ?>"><?php echo "Editar Perfil"; ?></a></li>
24      <li><a href="<?php echo site_url('usuario/logout');
25          ?>"><?php echo "Cerrar Sesi&oacute;n"; ?></a></
26          li>
27      <?php endif; ?>
28      <li><a href="http://forja.rediris.es/projects/
29          openconference" target="_blank">OpenConference</
30          a></li>
31
32      </ul>
33  </div>
34 </div>

```

Archivo: menu/sidebar.php

```

1 <?php foreach ($this->sidebar_library->sidebar as $item): ?>
2     <?php $this->load->view('templates/emplode/layout/menu/' . $item['
3         file']); ?>
4 <?php endforeach; ?>

```

Archivo: pages/cfp/send.php

```

1 <h2><?php echo "Env&iacute;o de Paper"; ?></h2>
2
3 <p><?php echo "Por favor, rellene el formulario y pulse el bot&oacute;n
4     Enviar"; ?></p>
5
6 <?php if($this->session->flashdata('message')): ?>
7     <div class="message">
8         <?php echo $this->session->flashdata('message');?>
9     </div>
10 <?php endif; ?>
11
12 <?php if (validation_errors()): ?>

```

```

12     <div class="error">
13         <?php echo validation_errors(); ?>
14     </div>
15 <?php endif; ?>
16
17     <table width="100%">
18     <tr>
19         <td colspan="2">
20             <fieldset id="name" class="profile">
21                 <legend class="profile"><?php echo "Datos"; ?></
                legend>
22                 <table>
23                     <tr>
24                         <td colspan="2">
25                             <form action="<?php echo site_url('cfp/
                                enviar'); ?>" method="post" enctype="
                                multipart/form-data">
26                             <td width="150px"><p><?php echo "T&iacute;
                                tulo"; ?></p></td>
27                             <td><input name="nombre" id="nombre" type
                                ="text" size="45" class="styled" /></
                                td>
28                         </tr>
29                         <tr>
30                             <td width="150px"><p><?php echo "Tema";
                                ?></p></td>
31                             <td><input name="tema" id="tema" type="
                                text" size="45" class="styled" /></td>
32                         </tr>
33                         <tr>
34                             <td width="150px"><?php echo "Descripci&
                                oacute;n"; ?></td>
35                             <td><textarea name="descripcion" id="
                                descripcion" cols="46" rows="10" class
                                ="styled"></textarea></td>
36                         </tr>
37                     </table>
38                 </fieldset>
39             </td>
40         </tr>
41     <tr>
42         <td colspan="2">&nbsp;</td>
43     </tr>
44     <tr>
45         <td colspan="2">
46             <fieldset id="password" class="profile">
47                 <legend class="profile"><?php echo "Documento";
                ?></legend>
48                 <table>
49                     <tr>
50                         <td width="150px"><?php echo "Documento";
                                ?></td>
51                         <td><input name="archivo" id="archivo"
                                type="file" class="styled" /></td>
52                     </tr>
53                 </table>
                </fieldset>

```



```

24
25     <?php if ($pagina_actual > 1): ?>
26         <?php if ($pagina_anterior == 1): ?>
27             <div class="right"><a href="<?php echo
                site_url(); ?>"><?php echo "Entradas
                nuevas &raquo;"; ?></a></div>
28         <?php else: ?>
29             <div class="right"><a href="<?php echo
                site_url('noticias/pagina/' .
                $pagina_anterior); ?>"><?php echo "
                Entradas nuevas &raquo;"; ?></a></div>
30         <?php endif; ?>
31     <?php endif; ?>
32 </p>
33 <?php endif; ?>

```

Archivo: pages/noticias/single_post.php

```

1 <div class="post">
2     <div class="post_title"><h2><?php echo $post['titulo']; ?></h2></div>
3     <div class="post_date"><?php echo strftime('%B %d, %Y', strtotime(
4         $post['fecha_posteo'])); ?> <?php echo "por"; ?> <?php echo
5         $post['display_name']; ?></div>
6     <div class="post_body">
7         <?php echo $post['excerpt']; ?>
8         <?php if ($post['contenido']): ?>
9             <p><?php echo $post['contenido']; ?></p>
10        <?php endif; ?>
11    </div>
12
13    <p class="right"><a href="#respond"><?php echo "Deja un comentario"; ?>
14        &#187;</a></p>
15    <div class="clearer">&nbsp;</div>
16
17    <?php if ($post['cuenta_comentarios'] > 0): ?>
18        <h3 class="left" id="comments"><?php echo "Respuestas (".$post['
19            cuenta_comentarios'].") a " ?> "<?php echo $post['titulo']; ?>"</h3>
20        <div class="clearer">&nbsp;</div>
21
22        <div class="comment_list">
23            <?php $i = 0; ?>
24            <?php foreach ($comentarios as $comentario): ?>
25                <?php if ($i % 2 == 0): ?>
26                    <div class="comment alt" id="comment-<?php echo
27                        $comentario['id']; ?>">
28

```

```

29         <div class="comment_gravatar left"><img alt="" src=
30             "<?php echo base_url(); ?>application/views/
31             templates/emplode/static/images/default_avatar.
32             jpg" height="32" width="32" /></div>
33
34         <div class="comment_author left">
35             <?php echo $comentario['autor']; ?>
36         <div class="comment_date"><a href="#comment
37             -<?php echo $comentario['id']; ?>"><?php
38             echo strftime(' %B %d, %Y ' . 'a la' . '
39             %H:%M:%S', strtotime($comentario['fecha
40             '])); ?></a></div>
41     </div>
42     <div class="clearer">&nbsp;</div>
43
44     <div class="comment_body">
45         <p><?php echo $comentario['contenido'];
46         ?></p>
47     </div>
48     <?php $i++; ?>
49 <?php endforeach; ?>
50 </div>
51 <?php endif; ?>
52
53 <a name="respond"></a>
54 <?php if ($post['permitir_comentarios'] == 1): ?>
55     <fieldset>
56         <div class="legend"><h3><?php echo "Deja un comentario"; ?></h3></
57         div>
58
59         <?php if (validation_errors()): ?>
60             <br /><div class="error">
61                 <?php echo validation_errors(); ?>
62             </div>
63         <?php endif; ?>
64
65         <form action="<?php echo post_url($post['url_titulo'], $post['
66             fecha_posteo']); ?>" method="post">
67             <?php if ($this->session->userdata('logged_in') == false):
68                 ?>
69                 <div class="form_row">
70                     <div class="form_property form_required"><?php echo "
71                     Username"; ?></div>
72                     <div class="form_value"><input name="nickname" id="
73                     nickname" type="text" value="<?php echo set_value('
74                     nickname'); ?>" size="22" /></div>
75
76                     <div class="clearer">&nbsp;</div>
77                 </div>
78                 <div class="form_row">
79                     <div class="form_property form_required"><?php echo "
80                     Email"; ?></div>
81                     <div class="form_value"><input name="email" id="email"
82                     type="text" value="<?php echo set_value('email'); ?>"

```

```

        size="22" /></div>
69
70     <div class="clearer">&nbsp;</div>
71 </div>
72 <div class="form_row">
73     <div class="form_property"><?php echo "Website"; ?></div>
74     <div class="form_value"><input name="website" id="website"
        type="text" value="<?php echo set_value('website');
        ?>" size="22" /></div>
75
76     <div class="clearer">&nbsp;</div>
77 </div>
78
79 <?php else: ?>
80     <div class="form_row">
81         <div class="form_property form_required"><?php echo "Nick
            "; ?></div>
82         <div class="form_value"><input name="nickname" id="
            nickname" type="text" value="<?php echo $this->session
            ->userdata('username'); ?>" size="22" disabled /></div
            >
83
84         <div class="clearer">&nbsp;</div>
85     </div>
86 <?php endif; ?>
87     <div class="form_row">
88         <div class="form_property form_required"><?php echo "
            Comentario"; ?></div>
89         <div class="form_value"><textarea rows="10" cols="46"
            name="comentario" id="comentario"><?php echo set_value
            ('comentario'); ?></textarea></div>
90
91         <div class="clearer">&nbsp;</div>
92     </div>
93
94     <div class="form_row form_row_submit">
95         <div class="form_value"><input type="submit" name="submit"
            value="<?php echo 'Submit'; ?>" class="button" /></
            div>
96
97         <div class="clearer">&nbsp;</div>
98     </div>
99 </form>
100 </fieldset>
101 <?php else: ?>
102     <h3><?php echo "Deja un comentario"; ?></h3>
103     <p><?php echo "Los comentarios est&aacute;n desactivados para este
        post"; ?></p>
104 <?php endif; ?>

```

Archivo: pages/pages/page.php

```
1 <h2><?php echo $page_data['titulo']; ?></h2>
2 <?php echo $page_data['contenido']; ?>
```

Archivo: pages/usuario/forgotten_password.php

```
1 <h2><?php echo "Recordatorio de password"; ?></h2>
2 <p><?php echo "Por favor, introduzca su nombre de usuario y su email, donde
  recibir&aacute; las instrucciones para recuperar su password."; ?></p>
3
4 <?php if($this->session->flashdata('message')): ?>
5     <div class="message">
6         <?php echo $this->session->flashdata('message');?>
7     </div>
8 <?php endif; ?>
9
10 <?php if (validation_errors()): ?>
11     <div class="error">
12         <?php echo validation_errors(); ?>
13     </div>
14 <?php endif; ?>
15
16 <table width="100%">
17 <tr>
18     <td colspan="2">
19         <fieldset id="password">
20             <legend><?php echo "Recordatorio de password"; ?></legend>
21             <table>
22                 <tr>
23                     <td>
24                         <form action="<?=site_url('usuario/
25                             password_olvidado'); ?>" method="post">
26                             <td width="150px"><?php echo "Username";
27                                 ?></td>
28                             <td><input name="username" id="username"
29                                 type="text" value="<?php echo set_value
30                                 ('username'); ?>" size="15" class="
31                                 styled" /> <font color="red">*</font></td>
32                             </tr>
33                             <tr>
34                                 <td width="150px"><?php echo "Email"; ?></td>
35                                 <td><input name="email" id="email" type="
36                                     text" value="<?php echo set_value('email
37                                     '); ?>" size="25" class="styled" /> <
38                                     font color="red">*</font></td>
39                             </tr>
40                         </table>
41                     </td>
42                 </tr>
43             </table>
44         </fieldset>
45     </td>
46 </tr>
47 </table>
```

```

36         <td colspan="2">&nbsp;   </td>
37     </tr>
38     <tr>
39         <td colspan="2">
40             <input type="submit" name="submit" value="<?php echo '
41                 Enviar &rsquo;&rsquo;; ?>" />
42             </form>
43         </td>
44     </tr>
</table>

```

Archivo: pages/usuario/login.php

```

1 <h2><?php echo "Login"; ?></h2>
2 <div class="login">
3     <?php if($this->session->flashdata('message')): ?>
4     <div class="message">
5         <?php echo $this->session->flashdata('message'); ?>
6     </div>
7     <?php endif; ?>
8
9     <?php if($this->session->flashdata('error')): ?>
10    <div class="error">
11        <?php echo $this->session->flashdata('error'); ?>
12    </div>
13    <?php endif; ?>
14
15    <table class="login">
16        <form action="<?php echo site_url('usuario/login'); ?>"
17            method="post">
18            <tr><td><?php echo "Username:"; ?>&nbsp;  </td> <td><
19                input name="username" id="username" type="text"
20                size="15" /></td></tr>
21            <tr><td><?php echo "Password:"; ?>&nbsp;  </td> <td><
22                input name="password" id="password" type="
23                password" size="15" /></td></tr>
24            <tr><td colspan="2" class="login"><input type="
25                submit" name="login" id="login" value="<?php
26                echo 'Login'; ?>" class="login" /></td></tr>
27        </form>
28    </table><br />
29    <p align="center"><a href="<?php echo site_url('usuario/registro');
30        ?>"><?php echo "Registro"; ?></a> | <a href="<?php echo
31        site_url('usuario/password_olvidado'); ?>"><?php echo "
32        Recordatorio Password"; ?></a></p>
33 </div>

```

Archivo: pages/usuario/profile.php

```
1 <h2><?php echo "Perfil"; ?></h2>
2 <p><?php echo "Aqu&iacute; puedes editar tu perfil"; ?></p>
3
4 <?php if($this->session->flashdata('message')): ?>
5     <div class="message">
6         <?php echo $this->session->flashdata('message'); ?>
7     </div>
8 <?php endif; ?>
9
10 <?php if (validation_errors()): ?>
11     <div class="error">
12         <?php echo validation_errors(); ?>
13     </div>
14 <?php endif; ?>
15
16 <table width="100%">
17 <tr>
18     <td colspan="2">
19         <fieldset id="name" class="profile">
20             <legend class="profile"><?php echo "Nombre"; ?></legend>
21             <table>
22                 <tr>
23                     <form action="<?php echo site_url('usuario/perfil'); ?>"
24                         method="post">
25                         <td width="150px"><?php echo "Username"; ?></td>
26                         <td><input name="username" id="username" type="text" value=
27                             "<?php echo set_value('username', isset($usuario['
28                                 username']) ? $usuario['username'] : ''); ?>" size="15"
29                             class="styled" disabled /> (<?php echo "Username no
30                                 puede ser cambiado"; ?>)</td>
31                     </tr>
32                     <tr>
33                         <td width="150px"><?php echo "Display name"; ?></td>
34                         <td><input name="display_name" id="display_name" type="text"
35                             value="<?php echo set_value('display_name', isset(
36                                 $usuario['display_name']) ? $usuario['display_name'] :
37                                 ''); ?>" size="15" class="styled" /></td>
38                     </tr>
39                     <tr>
40                         <td width="150px"><?php echo "Nombre"; ?></td>
41                         <td><input name="nombre" id="nombre" type="text" value="<?
42                             php echo set_value('nombre', isset($usuario['nombre']) ?
43                             $usuario['nombre'] : ''); ?>" size="15" class="styled"
44                             /></td>
45                     </tr>
46                     <tr>
47                         <td width="150px"><?php echo "Apellidos"; ?></td>
48                         <td><input name="apellidos" id="apellidos" type="text"
49                             value="<?php echo set_value('apellidos', isset($usuario
50                                 ['apellidos']) ? $usuario['apellidos'] : ''); ?>" size="
51                             15" class="styled" /></td>
52                     </tr>
53                 </table>
54             </fieldset>
55         </td>
56     </tr>
57 </table>
```

```

40     </fieldset>
41 </td>
42 </tr>
43 <tr>
44     <td colspan="2">&nbsp;   </td>
45 </tr>
46 <tr>
47     <td colspan="2">
48         <fieldset id="password" class="profile">
49             <legend class="profile"><?php echo "Password"; ?></legend>
50             <table>
51                 <tr>
52                     <td width="150px"><?php echo "Password"; ?></td>
53                     <td><input name="password" id="password" type="password"
54                         size="15" class="styled" /></td>
55                 </tr>
56                 <tr>
57                     <td width="150px"><?php echo "Repite Password"; ?></td>
58                     <td><input name="password_retype" id="password_retype" type
59                         ="password" size="15" class="styled" /></td>
60                 </tr>
61             </table>
62         </fieldset>
63 </td>
64 </tr>
65 <tr>
66     <td colspan="2">
67         <fieldset id="contact_info" class="profile">
68             <legend class="profile"><?php echo "Contacto"; ?></legend>
69             <table>
70                 <tr>
71                     <td width="150px"><?php echo "Email"; ?></td>
72                     <td><input name="email" id="email" type="text" value="<?php echo
73                         set_value('email', isset($usuario['email']) ? $usuario['
74                         email'] : ''); ?>" size="25" class="styled" /></td>
75                 </tr>
76                 <tr>
77                     <td width="150px"><?php echo "Website"; ?></td>
78                     <td><input name="website" id="website" type="text" value="<?php
79                         echo set_value('website', isset($usuario['website']) ?
80                         $usuario['website'] : ''); ?>" size="25" class="styled" /></
81                     td>
82                 </tr>
83             </table>
84         </fieldset>
85 </td>
86 </tr>
87 <tr>
88     <td colspan="2">

```

```

88     <input type="submit" name="submit" value="<?php echo 'Editar &
      rsaquo;&rsaquo;'; ?>" />
89 </form>
90 </td>
91 </tr>
92 </table>

```

Archivo: pages/usuario/registration.php

```

1 <h2><?php echo "Registro"; ?></h2>
2
3 <?php if ($this->system_library->settings['permitir_registros'] == 1): ?>
4 <p><?php echo "Por favor, rellena el formulario y pulsa el bot&ocute;n
      Registrar"; ?></p>
5
6 <?php if($this->session->flashdata('message')): ?>
7 <div class="message">
8 <?php echo $this->session->flashdata('message');?>
9 </div>
10 <?php endif; ?>
11
12 <?php if (validation_errors()): ?>
13 <div class="error">
14 <?php echo validation_errors(); ?>
15 </div>
16 <?php endif; ?>
17
18 <table width="100%">
19 <tr>
20 <td colspan="2">
21 <fieldset id="name" class="profile">
22 <legend class="profile"><?php echo "Datos Personales"; ?></
      legend>
23 <table>
24 <tr>
25 <td colspan="2"><form action="<?php echo site_url('usuario/registro'); ?>
      " method="post">
26 <td width="150px"><?php echo "Username"; ?></td>
27 <td><input name="username" id="username" type="text"
      value="<?php echo set_value('username'); ?>" size="15"
      class="styled" /> <font color="red">*</font></td>
28 </tr>
29 <tr>
30 <td width="150px"><?php echo "Display name"; ?></td>
31 <td><input name="display_name" id="display_name" type="
      text" value="<?php echo set_value('display_name'); ?>"
      size="15" class="styled" /></td>
32 </tr>
33 <tr>
34 <td width="150px"><?php echo "Nombre"; ?></td>
35 <td><input name="nombre" id="nombre" type="text" value="
      <?php echo set_value('nombre'); ?>" size="15" class="
      styled" /></td>
36 </tr>

```

```

37         <tr>
38             <td width="150px"><?php echo "Apellidos"; ?></td>
39             <td><input name="apellidos" id="apellidos" type="text"
              value="<?php echo set_value('apellidos'); ?>" size="15
              " class="styled" /></td>
40         </tr>
41     </table>
42     </fieldset>
43 </td>
44 </tr>
45     <tr>
46         <td colspan="2">&nbsp;</td>
47 </tr>
48     <tr>
49         <td colspan="2">
50             <fieldset id="password" class="profile">
51                 <legend class="profile"><?php echo "Password"; ?></legend>
52                 <table>
53                     <tr>
54                         <td width="150px"><?php echo "Password"; ?></td>
55                         <td><input name="password" id="password" type="password"
                          size="15" class="styled" /> <font color="red">*</font>
                          </td>
56                     </tr>
57                     <tr>
58                         <td width="150px"><?php echo "Repite Password"; ?></td>
59                         <td><input name="password_retype" id="password_retype"
                          type="password" size="15" class="styled" /> <font
                          color="red">*</font></td>
60                     </tr>
61                 </table>
62             </fieldset>
63 </td>
64 </tr>
65     <tr>
66         <td colspan="2">&nbsp;</td>
67 </tr>
68     <tr>
69         <td colspan="2">
70             <fieldset id="contact_info" class="profile">
71                 <legend class="profile"><?php echo "Contacto"; ?></legend>
72                 <table>
73                     <tr>
74                         <td width="150px"><?php echo "Email"; ?></td>
75                         <td><input name="email" id="email" type="text" value="<?
                          php echo set_value('email'); ?>" size="25" class="
                          styled" /> <font color="red">*</font></td>
76                     </tr>
77                     <tr>
78                         <td width="150px"><?php echo "Website"; ?></td>
79                         <td><input name="website" id="website" type="text" value
                          ="<?php echo set_value('website'); ?>" size="25"
                          class="styled" /></td>
80                     </tr>
81                 </table>
82             </fieldset>

```


- **Imagen:** Aquí pondremos el nombre de la imagen que aparecerá en el menú de templates. Por ejemplo, si nuestra imagen se llama **imagen.jpg**, en este campo pondremos ese mismo nombre (incluida la extensión) y luego almacenaremos dicha imagen en la carpeta *application/views/admin/static/images/templates/*. La imagen tendrá un tamaño de 240x234px.

Bibliografía

- [1] Open conference system. <http://pkp.sfu.ca/?q=ocs>, Consultado: Agosto 2009.
- [2] Arcsin. <http://templates.arcsin.se/>, Consultado: Agosto 2009.
- [3] iconpot. <http://www.iconpot.com/>, Consultado: Agosto 2009.
- [4] Julie C. Meloni. *PHP, MySQL y Apache*. ISBN: 978-84-415-2541-2. Anaya Multimedia, 2009.
- [5] Luke Welling y Laura Thomson. *Desarrollo Web con PHP y MySQL*. ISBN: 978-84-415-1569-7. Anaya Multimedia, 1ª edition.
- [6] Codeigniter user guide. http://codeigniter.com/user_guide/, Consultado: Agosto 2009.
- [7] Bernardo Cascales Salinas. *El libro de Latex*. ISBN: 978-84-205-3779-5. Prentice Hall, 2003.
- [8] Wikibooks. The Book of LaTeX. <http://en.wikibooks.org/wiki/LaTeX>.
- [9] Editor bluefish. <http://es.wikipedia.org/wiki/Bluefish>, Consultado: Septiembre 2009.
- [10] Xampp. <http://es.wikipedia.org/wiki/XAMPP>, Consultado: Septiembre 2009.
- [11] Joaquín Cervera Luis Fernández Mario Piattini, José Antonio Calvo-Manzano. *Análisis y Diseño de Aplicaciones Informáticas de Gestión. Una perspectiva de Ingeniería del Software*. ISBN: 978-84-789-7587-7. RA-MA EDITORIAL, 2003.
- [12] Antonio Falcón Aragón. forja: Openconference: Información del proyecto. <https://forja.rediris.es/projects/openconference/>.

GNU Free Documentation License

Version 1.3, 3 November 2008
Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with . . . Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.