

Títol: *Sistema de Gestión de Eventos para Seminarios Europeos*

Volum: *1*

Alumne: *Pol Martínez Orta*

Director/Ponent: *Nacho Navarro Mas*

Departament: *Arquitectura de computadores (DAC)*

DADES DEL PROJECTE

Títol del Projecte: Sistema de Gestión de Eventos para Seminarios Europeos.

Nom de l'estudiant: Pol Martínez Orta

Titulació: Enginyeria tècnica en Informàtica de Sistemes

Crèdits: 22,5

Director/Ponent: Nacho Navarro Mas

Departament: Arquitectura de computadores (DAC)

MEMBRES DEL TRIBUNAL (nom i signatura)

President: Enrique Morancho Llena

Vocal: Núria Pelechano Gómez

Secretari: Nacho Navarro Mas

QUALIFICACIÓ

Qualificació numèrica:

Qualificació descriptiva:

Data:

Índice de contenido

1. Introducción.....	7
1.1 Presentación.....	7
1.2 Definición del proyecto.....	7
2. Objetivos.....	9
3. Análisis de requerimientos.....	10
3.1 Descripción de la aplicación.....	10
4. Especificación.....	13
4.1 Requisitos funcionales.....	13
4.1.1 Roles/Perfiles.....	13
4.1.2 Contenido.....	15
4.1.3 Gestión de Grupos.....	18
4.1.4 Vistas (Views).....	20
4.1.5 Secciones / Estructura de la web.....	20
4.1.6 Gestión de ficheros.....	23
4.1.7 Administración.....	24
4.1.8 RSS.....	27
4.1.9 Logs del sistema.....	27
4.1.10 Idioma.....	28
4.1.11 Recuperabilidad.....	28
4.2 Requisitos no funcionales.....	28
4.2.1 Seguridad.....	29
4.2.2 Fiabilidad.....	29
4.2.3 Usabilidad.....	29
4.2.4 Eficiencia.....	30

4.2.5 Mantenimiento	30
4.2.6 Portabilidad.....	30
4.2.7 Negocio	31
4.2.8 Integración con el site hipeac.net.....	31
4.3 Diagrama Casos de uso	31
4.3.1 Anonymous User.....	31
4.3.2 Authenticated User	32
4.3.3 Organizer	32
4.3.4 Administrator.....	33
4.4 Descripción casos de uso	35
5. Diseño.....	43
5.1 Que es un CMS?.....	43
5.2 Elección del CMS.....	45
5.3 El gestor de contenidos Drupal	46
5.3.1 Descripción general.....	46
5.3.2 Módulos en Drupal.....	47
5.3.4 Arquitectura.....	47
5.3.5 Características generales	49
5.3.5 Bases de datos y seguridad.....	52
5.4 Gestión de usuarios	52
6. Implementación	54
6.1 Herramientas de desarrollo y entorno de pruebas.....	54
6.1.1 Xampp.....	54
6.2 Instalación Drupal.....	54
6.2.1 Requerimientos del servidor	55
6.2.2 Configuración del servidor.....	55

6.2.3 Creación de las BBDD's.....	55
6.2.4 Proceso de Instalación de Drupal	56
6.2.5 Configuración inicial del sitio web	56
6.3 Módulos	57
6.3.1 Instalación de módulos	58
6.3.2 Configuración de módulos.....	58
6.3.3 Módulos utilizados	58
6.4 Configuración del site	62
6.4.1 Usuarios y permisos	62
6.4.2 Configuración del aspecto visual	65
6.4.3 Creación de bloques y menús.....	67
6.4.4 Tipos de contenido.....	69
6.4.5 Vistas	70
6.4.6 RSS. Compatibilidad con iUPC y UPCdroid	72
6.4.7 Logs del sistema	78
6.5 Gestión de ficheros.....	79
6.5.1 sistema de ficheros	79
6.5.2 Upload de ficheros	80
6.5.3 Gestor WebFileManager (WebFM).....	81
6.6 Recuperación.....	83
6.6.1 Mirroring.....	83
6.6.2 Backup	84
6.6.3 Esquema de red	85
6.6.4 Cron	86
7. Actualización	88
8. Planificación	91

9. Estudio económico.....	93
9.1 Hardware.....	94
9.2 Software.....	94
9.3 Recursos humanos.....	95
10. Conclusiones.....	96
11. Bibliografía.....	98

1. Introducción

1.1 Presentación

Este proyecto se basa en la implementación de un sistema (o aplicación) web mediante un CMS (Content Management System), que permitirá a los usuarios del site acceder a un repositorio de eventos pasados y futuros que han sido o serán impartidos por diferentes departamentos de un conjunto de universidades europeas.

El actual responsable de la promoción y gestión de dichos eventos es la red de excelencia europea HiPEAC (European Network of Excellence on High Performance and Embedded Architecture and Compilation). Tanto el Barcelona Supercomputing Center (BSC) como la Universitat Politècnica de Catalunya (UPC) son miembros de esta red europea.

En colaboración directa con esta red se encuentra el DAC (Departament d'Arquitectura de computadors) que se encarga de mantener un repositorio de seminarios que han sido impartidos en la UPC.

Debido a que el proceso de introducción de contenido al repositorio es dificultoso, y el diseño del site ha quedado obsoleto, ha surgido la necesidad de actualizar el sistema de gestión del repositorio de eventos con un gestor de contenidos eficiente y cómodo, que nos permita organizar todo el contenido según nuestras necesidades.

Esta web pretende dar soporte a los usuarios que deseen disponer de la información y contenido asociado de los eventos impartidos por la red europea HiPEAC.

1.2 Definición del proyecto

Para llevar a cabo la realización de este proyecto se utilizara Drupal, uno de los diferentes productos de libre distribución que nos permitirá crear un portal web donde alumnos y profesores podrán disponer de todo el material relacionado con los seminarios europeos.

Es fundamental que la introducción de contenido y la visualización del mismo sea fácil y coherente, así como la navegación por el site sea cómoda e intuitiva, con la información claramente estructurada, agrupada y clasificada.

La web dispondrá de diferentes perfiles, para que cada usuario según su rol, pueda crear, acceder y visualizar el contenido asociado a cada uno de los eventos.

En la página principal de la web se mostrara una breve información de los eventos y algunos de sus atributos, hasta un máximo de 20 eventos por página. El sistema dispone de diferentes vistas para los eventos futuros y pasados, así como una sección de noticias y RSS.

Un evento puede tener diferentes variantes, que pueden ser un seminario, una conferencia, una clase sobre una temática en concreto, etc. Cada uno de los eventos contiene diversa información como por ejemplo el *speaker*, su biografía, la ubicación física donde se impartirá el evento, la hora que empieza, ficheros adjuntos de soporte, etc. Podemos acceder al contenido completo del evento clicando en su título. Para controlar toda la gestión de esta información debemos tener en cuenta que la gestión de usuarios es muy importante en esta aplicación, ya que se pretende que se pueda controlar el acceso a los eventos i a su contenido asociado mediante el rol de usuario que accede a la aplicación.

2. Objetivos

El objetivo principal del proyecto a nivel técnico es adquirir un conocimiento general en algunas de las tecnologías Web que se están utilizando en la actualidad por los principales creadores de software.

Además hay otros objetivos que me he planteado desde el inicio del proyecto. Éstos son, de alguna forma, metas, beneficios y motivaciones que pretendo llegar a alcanzar durante el desarrollo de todo el proceso. De cumplirse todos estos objetivos, habrá resultado ser una experiencia más enriquecedora, más útil y de mejora en lo que respecta a mi faceta académica y profesional.

- El aprendizaje de una serie de nuevas tecnologías que a pesar de conocer antes del proyecto, no había trabajado en ellas con anterioridad.
- El aprendizaje de un Content Management System de código abierto: Drupal. Este es otro punto de mucho interés, nunca antes había trabajado con un CMS.
- Realizar un proyecto que consista en un diseño integral de un sistema.
- Ser capaz de superar con éxito todas las etapas que un proyecto supone.

Quiero aprovechar esta oportunidad de aprender, con la intención de desarrollar una tarea similar en un futuro a corto plazo.

En definitiva, los objetivos están planteados sobre cosas con las que no he trabajado, ni en mi carrera académica ni mi experiencia laboral. Es algo diferente y nuevo, y esto en conjunto, es lo más motivador y estimulante a la hora de realizar el proyecto.

A continuación, detallaré el proceso que he llevado a cabo para desarrollar el software, que se divide en las siguientes etapas: análisis de requerimientos, especificación, diseño e implementación.

3. Análisis de requerimientos

El análisis de requerimientos consiste en la determinación de una serie de condiciones a satisfacer por el software que se debe desarrollar. En este apartado se hará una descripción del proyecto de carácter general, con sus funcionalidades principales y los usuarios que la conformarán. En el siguiente capítulo se especificarán cada una de las funcionalidades requeridas por el cliente.

Para la obtención de los requerimientos, se ha considerado como la opción más adecuada, la realización de reuniones periódicas con el cliente (en este caso, el director del proyecto) para ir definiendo las necesidades que debemos cubrir. Muchas de las decisiones tomadas han sido fruto de dificultades que nos hemos ido encontrando durante la realización del proyecto.

3.1 Descripción de la aplicación

La aplicación a desarrollar se basa en un repositorio de eventos pasados y anuncio de eventos futuros, algunos de ellos con material asociado, como por ejemplo: documentos de apoyo, diapositivas, video de la retransmisión (en el caso de que el evento sea pasado), etc. La web dispondrá también de una sección de noticias, subscripción de RSS, así como diferentes vistas para filtrar los eventos. Es necesario crear una intranet que permita gestionar todo el contenido de la web, los usuarios, el acceso al contenido, etc.

Se requiere también que la web sea compatible con dos aplicaciones desarrolladas por dos alumnos de la FIB (Facultat d'Informàtica de Barcelona). Estas aplicaciones recogen el código XML que genera nuestro site, y lo tratan, con el fin de ofrecer, a través de dispositivos móviles, la información que se muestra en la web, organizando el contenido a su manera. Estas aplicaciones son:

- **iUPC:** aplicación desarrollada para iPhoneOS. Una de sus funcionalidades es mostrar los 50 eventos más recientes, con su Topic, Description, y los links (si existen) al contenido asociado.
- **UPCdroid:** aplicación desarrollada para Android. No se conocen las funcionalidades que ofrecerá debido a que se está empezando su desarrollo.

Para mostrar el código XML que ambas aplicaciones necesitan, deberemos seguir un proceso detallado más adelante.

Para poder controlar de una forma estructurada y coherente todas las herramientas que nos ofrece la aplicación, se han definido diferentes roles/perfiles para los usuarios de la aplicación, que nos determinaran el grado de acceso para realizar acciones sobre la aplicación.

Los roles/perfiles que se han definido para dicho propósito son:

- usuario no registrado
- usuario registrado
- organizador
- administrador.

Los *usuarios no registrados* solo podrán acceder al contenido público de la web. El contenido privado estará asociado a diferentes grupos (previamente creados por los organizadores), que determinaran el grado de privacidad del contenido adjunto. El tipo de grupo y sus restricciones determinaran si el usuario puede acceder al contenido o no.

Para poderse registrar a un grupo, el usuario deberá primero registrarse a la aplicación, y posteriormente solicitar la afiliación al grupo que le interese. El administrador del grupo será quien decida a que categoría pertenece el grupo, es decir, que restricciones tendrá el grupo, lo que determinará los permisos que tiene el usuario para acceder al contenido de dicho grupo.

El *usuario registrado* podrá solicitar la afiliación al grupo que le interese para ver los eventos y material adjunto a dicho grupo.

Los administradores de los grupos son los usuarios con el rol de *organizador*. Es el usuario que se encarga de crear el contenido de la web (eventos, noticias, grupos) y asignar los permisos de acceso al evento y a su contenido. Este acceso se puede definir como público o privado. Si es privado, el organizador deberá asignar el evento a uno de los grupos ya existentes.

Como administrador del site utilizaremos el rol de *administrador*, el cual podrá gestionar todos los usuarios, todo el contenido y ficheros asociados, podrá modificar los aspectos visuales y estructurales de la web, etc.

4. Especificación

En este capítulo se establecen de forma detallada todas las características y funcionalidades que tendrá nuestra aplicación. Se especifica el formato que se le dará al proyecto así como la distribución de espacio, estructuración de usuarios i las funcionalidades con las que se les dotará.

4.1 Requisitos funcionales

Teniendo en cuenta la descripción del apartado anterior se obtienen las necesidades que se describen a continuación.

4.1.1 Roles/Perfiles

Para poder gestionar todo el contenido del site y controlar el acceso al mismo, es necesario determinar qué usuarios podrán crear éste contenido y quiénes lo podrán ver. Se han definido 4 roles para definir diferentes grados de acceso e interacción a la aplicación.

- ***Anonymous user***

Este rol de usuario ya viene definido por Drupal, que representa el usuario que navega por nuestra aplicación sin haberse registrado o identificado. Podrá ver todo el contenido de la web que haya sido definido como “público”.

- ***Authenticated user***

Para adquirir este rol, el usuario anónimo deberá registrarse en la aplicación. Una vez registrado, el usuario podrá solicitar la afiliación a los grupos que le interese, para poder acceder a su contenido.

- ***Organizer***

Es el usuario que se encarga de crear y gestionar los eventos y las noticias que aparecen en la web. También puede crear grupos, asignar contenido a esos grupos (eventos, y material asociado), y determinar que usuarios podrán acceder a ese contenido.

- **Administrator**

Como administrador del site se ha definido este rol para poder gestionar:

- Todo el contenido del site: alta, baja o modificación de eventos, noticias y grupos.
- Los usuarios: alta, baja o modificación de perfiles y asignación de roles.
- Aspecto visual: Podrá modificar los estilos de la aplicación y controlar diferentes aspectos estructurales de la web (disposición de bloques, enlaces, primary links, etc.).

Si un usuario *Anonymous user* quiere registrarse en nuestra aplicación, deberá rellenar el formulario específico para ello, en el que se le pedirán los siguientes datos:

- **Username:** nombre de usuario para entrar en la aplicación. Este nombre de usuario aparecerá también en el campo *author* de un evento o noticia, que define el autor o creador del post (solo afectará a los organizadores y administradores).
- **E-mail address:** dirección de correo electrónico válida. Se utilizará esta dirección para la validación del registro de usuario.

Además de esos dos campos, puede ser posible que aparezcan en el mismo formulario la subscripción simultánea a determinados grupos de la aplicación. El hecho de que aparezcan o no algunos de los grupos de la aplicación dependerá del creador del grupo, el cual definirá en el momento de su creación, si desea que aparezca la subscripción a dicho grupo en el formulario de registro de la aplicación o no. Esta opción será detallada más adelante.

Una vez rellenado el formulario y enviado, el usuario recibirá un correo electrónico en la cuenta introducida en el campo *e-mail address*. En el mail recibido aparecerá un link de confirmación el cual deberá aceptar el usuario para completar su registro.

A partir de este momento el usuario ya formará parte de la aplicación y, de acuerdo con su rol, podrá ver contenido al cual antes no tenía acceso.

Cuando un usuario accede a la aplicación, le parecerá, en función de su rol, la información de su perfil. En el caso del *authenticated user*, le aparecerán los datos básicos de su registro. Para ver los grupos a los que está afiliado deberá ir a la sección “My Subscriptions” en el bloque inferior izquierdo del site. En el caso de los *organizers* y *administrators*, tendrán la opción de visualizar, además de su información personal, los grupos de los cuales son administradores, para posteriormente modificarlos, añadirles contenido o bien gestionar sus afiliados.

Cualquier usuario registrado en la aplicación tiene permisos para cambiar los datos básicos de su perfil.

4.1.2 Contenido

Casi todo el contenido de Drupal se almacena como un nodo. Un nodo se puede definir como un objeto de contenido, al que podremos acceder siempre mediante una url fija (permalink).

Drupal nos permite crear y definir nuestros propios tipos de contenido. Por defecto, la instalación básica de Drupal, solo se nos permite crear 3 tipos de contenido básico: Page, Story y Poll. En nuestro caso reutilizaremos el objeto *Page* para definirlo como noticia (new).

Mediante la instalación de un modulo específico, podremos crear nuevos tipos de contenido con los atributos que queramos definir para ellos. Los usuarios autorizados a publicar, modificar o eliminar cualquier tipo de contenido son el *administrator* y *organizer*.

A continuación se detallan los tipos de contenido definidos para nuestra aplicación:

- **Evento (Event)**

Un evento es uno de los tipos de contenido más utilizados en nuestro site. En el formulario de alta de un evento deben aparecer los siguientes campos:

- Start Date: Fecha de inicio del evento.
- End Date: Fecha de finalización del evento.
- Repeat Event options: Opciones de repetición del evento.

- *Topic*: Título representativo del evento.
- *Tags*: Son palabras clave asociadas al evento. Nos ayudaran a clasificar o filtrar el contenido de un mismo tipo de objeto, en este caso, los eventos. Un ejemplo seria asociar un mismo tag para agrupar los eventos que se impartieron un año determinado, o fueron impartidos por una entidad o universidad especifica, etc.¹
- *Abstract*: Es el cuerpo del evento. Contiene la descripción completa del mismo; se expone la temática de la que se hablará en el evento.
- *Audience Groups*: Este campo definirá la privacidad del evento (se detalla en el siguiente apartado).
- *Speaker*: Ponente del evento.
- *Speaker's Picture*: Fotografía del ponente.
- *Bio*: Biografía del ponente.
- *Affiliation*: Procedencia del ponente. (Ej: Universidad, empresa, etc.).
- *Video Streaming*: Se utilizara este campo para mostrar el link de online streaming de los eventos futuros (si se dispone de él). Si el evento es pasado, se puede utilizar el este campo para linkar algún video especifico.
- *Event Type*: Campo obligatorio con un desplegable. Los tipos de evento son: seminar, conference, tutorial, workshop, lecture y talk.
- *Sponsor*: Entidad que esponsoriza el evento.
- *Location*: Ubicación física de la impartición del evento.
- *Telephone Contact*: Teléfono de contacto para asistencia durante el evento.
- *E-Mail Contact*: Correo electrónico de contacto.
- *Contact Organizer*: Nombre del organizador o personal de contacto.

¹ Este campo es autocompletable. Todos los tags introducidos anteriormente en el alta o modificación de un evento irán apareciendo mientras coincida la palabra que se está escribiendo en el momento.

- **Noticia (New)**

Hemos utilizado el tipo de contenido *page* (predefinido por Drupal) para crear este objeto. Las noticias serán accesibles por todos los usuarios de la aplicación, ya que hemos definido el tipo de contenido como público. Los atributos del objeto son:

- *Title*: Título de la noticia
- *Body*: Descripción de la noticia.

- **Grupo (Group)**

Un grupo no es exactamente un tipo de contenido, sino que más bien podríamos definirlo como un contenedor de uno o más objetos, tales como eventos, noticias u otro tipo de contenido. En nuestra aplicación hemos definido el evento como único tipo de contenido para los grupos. Los atributos del grupo son:

- *Group Name*: Nombre del grupo.
- *Description*: Descripción del grupo.
- *Membership Request*: Proceso de afiliación al grupo (detallado en el siguiente apartado).
- *Registration form (checkbox)*: deberemos marcar la casilla si queremos que se muestre la opción de solicitud de afiliación al grupo en el formulario de alta de usuario a la aplicación.
- *List in groups directory (checkbox)*: Marcando la casilla habilitamos la visibilidad del grupo en el directorio de grupos (Group Directory).
- *Private Group (checkbox)*: Esta opción nos permite mantener el grupo totalmente privado e inaccesible por parte del usuario. El administrador del grupo deberá especificar los usuarios afiliados al grupo.

- **Tag**

Es un atributo que se le puede añadir a un objeto o tipo de contenido, que nos permitirá, mediante la activación del módulo *Taxonomy*, clasificar o catalogar el contenido mediante palabras clave.

Esta descripción se encuentra en este apartado porque tanto el *organizer* como el *administrator* podrán crear tags independientemente de los eventos, como si de un tipo de contenido se tratara.

En la mayoría de los campos de cada objeto se puede definir el formato de entrada del texto, para poder determinar si el texto introducido es: HTML filtrado, HTML puro o PHP. Ello nos dará más flexibilidad y versatilidad cuando queramos darle formato al texto.

En cada tipo de contenido es posible asociar ficheros adjuntos al nodo, tal como presentaciones, pdf's, video, etc. En el apartado 6.5.2 se detalla cómo funciona el sistema o proceso de upload y asociación de ficheros a los nodos.

Existe un campo común para los tipos evento y noticia, que se denomina *published*. Al marcar la casilla definiremos el nodo como publicado, visible para todos los usuarios que tengan acceso a él. Si por lo contrario, queremos mantener el contenido oculto, no marcaremos la casilla, en ese caso, solo los administradores de la aplicación y el creador del nodo podrán acceder a él.

4.1.3 Gestión de Grupos

Como se ha explicado en el apartado anterior, un grupo o *Audience Group* es un contenedor de objetos de tipo evento. Solo los administradores y organizadores de la aplicación tendrán permisos para crearlos.

Podríamos definir como grupo un conjunto de eventos con un elemento en común, que podría ser, una temática, una asignatura, un grupo de conferencias, un departamento de universidad, una entidad o empresa, etc. Al igual que los eventos, también se le puede adjuntar material didáctico al grupo, sin necesidad de crear un evento para dicho propósito. Un ejemplo sería información adicional del grupo, documentación común para todos los eventos del grupo, etc.

Para poder hacer la solicitud a un grupo, el usuario deberá estar registrado previamente en la aplicación. No obstante, el usuario no registrado podrá acceder a la lista de grupos (Group Directory), donde aparecen los grupos públicos listados alfabéticamente.

Cuando un usuario (administrador u organizador) da de alta un grupo, deberá definir (además del nombre, descripción, etc.) qué tipo de restricción de acceso va a asignar al grupo. En función de la opción seleccionada determinaremos el proceso de afiliación que deberá seguir el usuario para poder acceder al grupo y a su contenido adjunto.

Los cuatro tipos de restricción de afiliación son:

- *Open*: la solicitud de afiliación al grupo de un usuario será aceptada automáticamente. El usuario podrá ver todo el contenido asociado al grupo. Podríamos definir esta restricción como pública.
- *Moderated*: Cuando un usuario hace la petición de afiliación deberá esperarse a que el administrador del grupo acepte la solicitud. El administrador del grupo podrá decir si decide aceptar la solicitud o rechazarla, que en el segundo caso, prohibiría al usuario el acceso al contenido del grupo.
- *Invite only*: Los miembros deben ser introducidos por el administrador del grupo.
- *Closed*: El grupo está totalmente cerrado. Ningún usuario de la aplicación excepto el administrador del grupo podrá conocer su existencia, y interactuar con él.

El administrador del grupo será el usuario que lo haya creado, teniendo posteriormente la posibilidad de asignar nuevos usuarios como *administradores adicionales* del grupo. Si se quiere definir un usuario como nuevo *administrador adicional*, éste deberá estar afiliado al grupo previamente. Una vez cumplido el requerimiento, el administrador principal podrá (a través de la aplicación) asignar este rol al usuario en cuestión.

Si en el proceso de creación del grupo se ha marcado la casilla *Registration form*, un usuario podrá registrarse a la aplicación y pedir la afiliación al grupo simultáneamente. Se dará la posibilidad porque aparecerá una casilla para hacer la solicitud de afiliación al grupo en cuestión en el propio formulario de alta de usuario de la aplicación.

• Grupos y Eventos

Los grupos nos permiten agrupar o clasificar el contenido del site, además de controlar el acceso a él. Cuando un evento es dado de alta, éste se define por defecto como público, lo que permitirá a todos los usuarios de la aplicación ver la información del nodo. Además de público, el evento puede ser asociado a un grupo. En el caso de que el evento sea público y también esté afiliado a un grupo, aunque el nivel de restricción de afiliación de dicho grupo fuera muy alto, cualquier usuario de la aplicación podría ver el contenido del evento. Si el evento no está definido como público, la aplicación exige que se asigne el evento a un grupo. En este momento el evento será únicamente visible por los afiliados a dicho grupo.

Para terminar, apuntar que cuando accedemos al contenido de un grupo, la aplicación nos muestra todos los eventos asociados a este grupo, ordenados por fecha, empezando por los más recientes hasta los más antiguos.

4.1.4 Vistas (Views)

Drupal nos permite definir, mediante la instalación y aplicación de algunos módulos, qué contenido queremos mostrar y cómo lo queremos mostrar, pudiéndolo estructurar y organizar según nuestras necesidades.

Cada una de las vistas será personalizada para cada rol de la aplicación. Según los permisos del usuario que está solicitando la vista, se podrá interactuar con mayor o menor medida con la aplicación. Por ejemplo, según el usuario que haga la petición para ver la lista de grupos le aparecerán unas opciones u otras. En el caso del organizador le aparecerán botones de creación de nuevos grupos y modificación de existentes, y en cambio, al usuario autenticado le aparecerán los botones de afiliación a los grupos entre otras opciones.

En el apartado 6.4.5 se describe con detalle cuál es el proceso de creación de las vistas.

4.1.5 Secciones / Estructura de la web

Para satisfacer las necesidades del cliente, se ha diseñado una estructura genérica en la que se basará la vista principal de la aplicación para todos los usuarios. En el

momento que un usuario se registre, se le mostrarán los bloques específicos de las funcionalidades asignadas para su rol.

El diseño estructural de la web se ha basado en el site <http://www.hipeac.net>, siguiendo las directrices del director de este proyecto.

La estructura visual que se ha definido para la aplicación es la siguiente:

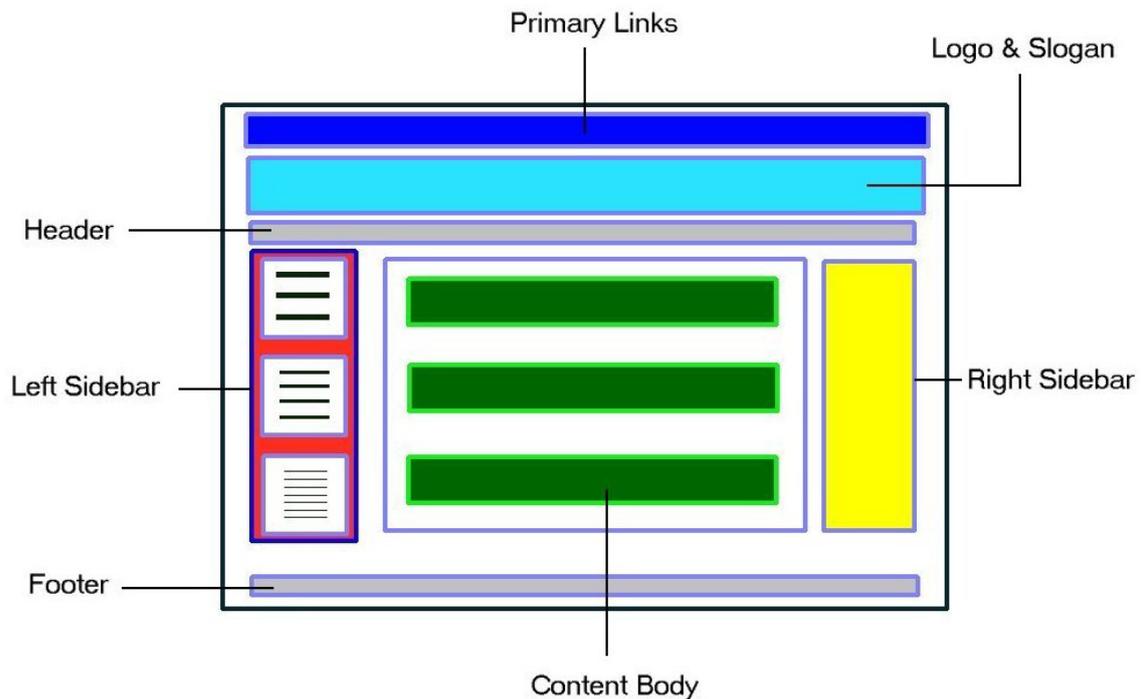


Ilustración 1: Esquema de la estructura del site

A continuación se definirán las diferentes partes que componen la estructura de la web y las funcionalidades específicas para cada uno de los roles:

- **Primary Links:** En esta parte se muestran un grupo de links considerados de los más importantes/funcionales del site.
 - *Usuario no registrado:* Los links que aparecen son:
 - ❖ HOME: Muestra la pagina principal del site.
 - ❖ News: Muestra las noticias del site.
 - ❖ LogIn: Formulario de registro a la aplicación.
 - *Demás usuarios:* Los 3 roles restantes verán una pequeña variación respecto al usuario no autenticado. Los links son:
 - ❖ HOME: Pagina principal.

- ❖ **News:** Lista de noticias.
- ❖ **My Account:** Muestra los datos de perfil del usuario, tales como los grupos afiliados del usuario, los grupos de los cuales se es administrador y historial del usuario, entre otros. También se le permite al usuario cambiar el correo electrónico definido inicialmente, y la contraseña.
- ❖ **Log Out:** Cierra la cookie de nuestra sesión.
- **Logo & Slogan:** Esta parte de la web es fija para todas las páginas. Siempre se muestra el mismo contenido, en este caso el logo del site y su slogan.
- **Header:** Sección fija para todas las páginas. Se puede definir un texto explicativo, aunque en nuestro caso lo hemos dejado en blanco.
- **Left Sidebar:** Ésta parte está formada por bloques. Los bloques son contenedores que nos permiten mostrar una determinada información que puede ser fija (por ejemplo links a otras páginas), o bien puede ser variable, es decir, información que se genera dinámicamente cuando se renderiza el contenido del bloque en cuestión.

En la parte izquierda del site se han definido 4 bloques diferentes:

- ❖ **Upcoming Seminars:** Bloque dinámico visible para todos los usuarios. Se mostraran los eventos futuros (titulo y días restantes para el seminario) y los eventos que se están impartiendo en el momento, con su titulo y el tag “now”, indicando que el evento se está impartiendo en ese preciso momento.
Si el evento es público todos los usuarios lo verán expuesto en el bloque, pero si el evento está asociado a un grupo privado, solo los usuarios registrados en la aplicación y afiliados al grupo lo verán en el bloque.
- ❖ **User:** Este bloque está dedicado únicamente a los usuarios con el rol *organizer* y *adminisitrator*. Aquí aparecen las opciones que dispone el usuario para crear contenido y administrarlo. En el caso del *administrator* dispondrá también de opciones de administración del site.

- ❖ *Site navigation*: Bloque de navegación visible para todos los usuarios. Si el usuario está registrado le aparecerá algún elemento más en el bloque. Nos permite buscar cualquier tipo de contenido desde cualquier parte del site. Está compuesto por diferentes links, que son: lista de grupos, diferentes vistas para filtrar los eventos, grupos a los que estamos afiliados y búsqueda de tags.
- ❖ *Related Links*: Este bloque es visible e idéntico para todos los usuarios, es totalmente estático. Contiene links externos a páginas “amigas” o relacionadas con nuestro site.
- **Content Body**: En esta sección se muestra el contenido que el usuario ha solicitado mediante su petición. La generación de este contenido es totalmente dinámica. Cuando el usuario accede a cualquier página de nuestra web, el navegador hace una petición a la aplicación, lo que implica que ésta haga la consulta a la base de datos para poder ofrecer el contenido solicitado. Previamente se habrá comprobado que el usuario tenga acceso a dicho contenido.
- **Right Sidebar**: Esta parte de la aplicación ha sido inhabilitada para engrandecer el espacio para el content body. En este proyecto se había propuesto inicialmente mostrar un calendario de eventos en la parte derecha, pero debido a exigencias del cliente se ha desestimado dicha opción, y se ha dejado como oculta.
- **Footer**: Sección fija visible para todos los usuarios. El contenido de esta sección será común en todas las páginas.

4.1.6 Gestión de ficheros

La gestión de ficheros es muy importante en nuestra aplicación, ya que se requiere poder controlar el acceso a todo el contenido adjunto a los eventos. Además, se pretende hacer el upload de ficheros con un tamaño superior a 500Mb, por lo que tenemos que buscar una alternativa al upload por http. Por un lado existe la limitación

del servicio PHP respecto al tamaño de los ficheros que se quieren subir², y por otra parte el upload de ficheros grandes por http sería muy lento.

Como solución rápida y sólida se ha decidido que el usuario realizará el upload de ficheros por ftp, y posteriormente añadirá los ficheros al nodo que desee. Para realizar esta acción necesitaremos instalar un modulo específico que nos ofrecerá una API³ implementada en AJAX⁴, que nos permitirá insertar en la base de datos los ficheros subidos al servidor, y posteriormente adjuntarlos al nodo que deseemos, definiendo el grado de restricción de acceso al mismo por parte del usuario. Estas acciones se pueden realizar mediante un procedimiento descrito en el apartado 6.5.3.

4.1.7 Administración

El área de administración será únicamente accesible por los usuarios con el rol *administrator* o bien, por el super-administrador de la aplicación. Los servicios que ofrece esta sección son: gestión total de los usuarios, gestión del contenido, modificación de los aspectos funcionales y visuales del site, entre otros.

A continuación se describen las funcionalidades que se han requerido para esta sección:

• Gestión del Contenido

- **Creación de contenido:** Posibilidad de crear eventos, noticias o grupos.
- **Gestión Categorías:** Nos permitirá dar de alta categorías (tags), y modificar las existentes.
- **Content Templates:** Permite crear un template personalizado para el RSS.
- **Publicación RSS:** Configuración general del RSS.
- **Contenido:** Ver, editar y eliminar contenido del site.

• Construcción del site

- **Bloques:** Configuración de los bloques de contenido que aparecen en los laterales de la web.

² La configuración del PHP del servidor tiene limitado el upload de ficheros a 2Mb.

³ Application Programming Interface.

⁴ Asynchronous JavaScript And XML

- **Menús:** Creación, edición, eliminación y reorganización de los primary links y menús.
 - **Módulos:** Habilitación o inhabilitación de los módulos de la aplicación.
 - **Temas:** Edición del aspecto visual general del site.
 - **Vistas:** Creación, edición y eliminación de vistas.
- **Grupos Orgánicos**
 - Configuración de grupos
 - Configuración del acceso a los grupos
- **Configuración del site**
 - **Tema de administración:** Modificación del tema de la aplicación de administración.
 - **Clean URL's:** Configuración de url's amigables.
 - **Preferencias Content Templates:** Configuración específica del modulo Content Templates.
 - **Fecha y hora:** Configuración de la zona horaria, hora y fecha, y formato de la misma.
 - **Error reporting:** Configuración del redireccionamiento en caso de errores 403 (access denied) y 404 (page not found).
 - **Configuración eventos:** Configuración específica para del modulo que gestiona los eventos.
 - **File System:** Configuración del sistema de ficheros. Las dos opciones que nos ofrece Drupal son *Public* o *Private*. En nuestro caso hemos marcado la segunda opción.
 - **File Uploads:** Restricción de ficheros según la extensión, tamaño y usuario.
 - **Google Analytics:** Configuración de los parámetros específicos para el tracking de páginas mediante la aplicación Google Analytics.
 - **Kit de herramientas para las imágenes:** Selección de la calidad de las imágenes. Un valor porcentual definirá este campo.
 - **Formatos de entrada:** Configuración del formato de entrada del contenido introducido por el usuario.

- **Caching:** Activación o desactivación de la cache del site.
 - **Configuración repetición de eventos:** Configuración específica del modulo *Repeat Events*.
 - **Información del site:** Configuración del nombre del site, configuración de la dirección de correo utilizada como remitente de los mails enviados por la aplicación, edición del slogan y footer de la web, y configuración de la página principal por defecto.
 - **Site Maintenance:** Configuración del estado del site. Existen dos modos posibles, "Online" (la web está funcionando), y "OffLine" que desactiva el site mostrando un único mensaje al usuario. En nuestro caso: *"HiPEAC Web Seminars is currently under maintenance. We should be back shortly. Thank you for your patience"*.
 - **Lista de Taxonomias:** Configuración específica del modulo *Taxonomy*.
- **Gestión de usuarios**
 - **Control de acceso:** Sección muy importante en la que determinaremos el acceso a las funcionalidades de la aplicación para cada uno de los roles.
 - **Reglas de acceso:** Podemos deshabilitar usernames, direcciones de correo y direcciones IP.
 - **Roles:** Creación, modificación o eliminación de roles.
 - **Usuarios:** Creación, modificación o eliminación de usuarios. Nos permitirá también asignar roles a los usuarios.
 - **Preferencias de usuario:** Permite configurar el comportamiento por defecto de los usuarios, incluyendo los requerimientos de registro, mails y foto de usuario.
- **Logs del sistema**

Drupal dispone de sus propios gestores de logs, y aunque utilicemos google analytics para recoger esta información, cabe destacar que nos puede dar datos interesantes.

 - **Recent log entries:** Log de las entradas más recientes
 - **Recent hits:** Últimas páginas visitadas.
 - **Top 'access denied' errors:** Muestra los últimos errores de acceso denegado.

- **Top 'page not found' errors:** Muestra los últimos errores de página no encontrada.
- **Top pages:** Páginas con más visitas.
- **Top visitors:** Usuarios con más páginas visitadas.
- **Access log settings:** Configuración de logs.
- **Status report:** Reporte de errores detectados en nuestro site.

4.1.8 RSS

En este proyecto se requiere como una de las funcionalidades la posibilidad de poderse suscribir a nuestra web mediante RSS. Si un usuario se suscribe a nuestra aplicación, recibirá una notificación cada vez que se cree un nuevo evento. De esta forma el usuario podrá estar informado en todo momento de los nuevos eventos publicados en el site.

- **Compatibilidad aplicaciones iUPC y UPCdroid**

También se ha especificado como requerimiento del proyecto la generación de código XML en el feed del RSS, para que las aplicaciones iUPC y UPCdroid puedan extraer la información que necesitan para mostrar los eventos del site y su contenido adjunto (pdf's, presentaciones, videos, etc.) en sus respectivos dispositivos. La estructura del código XML que debemos mostrar ha sido especificada por los proyectistas de iUPC y UPCdroid. Ellos han definido los tags y la información que debemos generar en el feed del RSS. En el apartado de implementación se describe como se ha llevado a cabo este proceso.

4.1.9 Logs del sistema

El sistema de logs de la aplicación solo podrá ser consultado por el usuario con el rol de *administrator*. Se podrá consultar toda la información referente a numero de accesos por pagina, accesos totales, estado de los servicios del servidor, etc., tal y como se ha descrito en el apartado 4.1.7.

Para complementar esta información y poder hacer un análisis más exhaustivo del usuario que accede a la aplicación, se podrán consultar (externamente a la aplicación)

los logs del sistema a través de Google Analytics⁵. Se pueden obtener informes como el seguimiento de usuarios exclusivos, el rendimiento del segmento de usuarios, el rendimiento del contenido, el análisis de navegación, los objetivos y proceso de redireccionamiento o los parámetros de diseño web. Nos permitirá saber entre otras cosas que sistemas operativos, navegadores, versión de navegador, etc., utilizan los usuarios de nuestro site.

Se debe integrar un modulo a nuestro sistema para generar el script que va incrustado en cada una de las páginas de la web.

4.1.10 Idioma

El idioma definido para la aplicación es el inglés, debido a que el site de la *Red Europea HiPEAC* está en este idioma.

4.1.11 Recuperabilidad

En una aplicación donde se almacena una gran cantidad de información, es de vital importancia disponer de un sistema de recuperación de datos que nos permita restablecer nuestro site en caso de fallida.

Los dos procedimientos que se utilizarán para disponer de este sistema serán el *mirroring* y el *backup*.

Nuestra aplicación deberá replicarse en otro servidor con el fin de poder redireccionar las peticiones en caso de caída del servidor principal, para normalizar el estado del site. Se realizará además un backup diario en un tercer servidor, separado físicamente de los dos anteriores, para garantizar una mayor seguridad.

En el apartado “6.6 *Recuperación*” se detalla el proceso de implementación de este sistema.

4.2 Requisitos no funcionales

A diferencia de los requerimientos funcionales, los requerimientos no funcionales definen las calidades que debe cumplir la aplicación al realizar su función. Por tal de evaluarlos en profundidad, se toma como base el estándar ISO/IEC 9126-1, el cual

⁵ Herramienta web gratuita ofrecida por Google.

permite evaluar la calidad del software. Se utiliza la extensión NT-ISO/IEC 9126 del estándar mencionado para poder evaluar aspectos no técnicos, como por ejemplo, la vertiente económica del software usado.

Acto seguido, se describen los requerimientos no funcionales de la aplicación agrupados según las características que define el estándar mencionado.

4.2.1 Seguridad

Capacidad del producto software para proteger la información y los datos de forma que las personas o sistemas no autorizados no las puedan leer ni modificar, al mismo tiempo que no se deniega el acceso a personas ni sistemas autorizados.

- La aplicación debe utilizar contraseñas para identificar y autenticar los usuarios.
- La aplicación debe almacenar cifradas las contraseñas de todos los usuarios de la aplicación.

4.2.2 Fiabilidad

Capacidad del producto software de mantener el nivel de prestación bajo condiciones establecidas durante un periodo de tiempo establecido (madurez, recuperabilidad).

- La aplicación debe estar disponible en todo momento en el dominio *hipeac.ac.upc.edu/seminars* porque es el dominio que destina el cliente por hospedar la aplicación.
- Se debe realizar una copia de seguridad de la aplicación diariamente, de forma que en caso de baja del sistema se pueda restaurar la del día anterior.

4.2.3 Usabilidad

Capacidad del producto software para ser entendido, aprendido, usado y ser atractivo para los usuarios finales, cuando se utiliza bajo condiciones determinadas.

- La interfaz gráfica de usuario debe ser desarrollada en inglés, puesto que es el idioma que se ha definido para la aplicación.
- La aplicación debe ser fácil de utilizar y lo más intuitiva posible.

- La aplicación debe disponer de herramientas que permitan administrar los contenidos, los usuarios y la apariencia de la forma más sencilla posible.
- La aplicación deberá respetar, siempre que sea posible, los estándares web establecidos por el W3C respecto a accesibilidad, internacionalización e independencia de dispositivos.

4.2.4 Eficiencia

Capacidad del producto software para proporcionar prestaciones adecuadas, relativas a la cantidad de recursos usados, bajo condiciones determinadas. El tiempo de respuesta de la aplicación debe ser aceptable, se espera una respuesta de un máximo de 2 segundos.

4.2.5 Mantenimiento

La aplicación debe trabajar siempre con versiones estables del software. Las versiones no estables quedan restringidas a aplicaciones en pruebas. La aplicación debe disponer de un procedimiento establecido de actualización.

4.2.6 Portabilidad

Capacidad de un producto software por ser transferido de una plataforma a otra (instalación, reemplazo, adaptabilidad).

- La aplicación debe visualizarse correctamente, como mínimo, en los siguientes navegadores: Internet Explorer (versiones 6 y 7), Mozilla Firefox (versiones 2 y 3) y GoogleChrome, puesto que son los más utilizados entre los usuarios.
- La aplicación debe desarrollarse sobre un servidor Apache, se debe programar en PHP5 y debe utilizar MySQL como sistema gestor de bases de datos.
- La base de datos de la aplicación debe poder administrarse con herramientas específicas para bases de datos, como por ejemplo PHPMYAdmin.
- Sólo se permitirá subir documentos a la aplicación con los siguientes formatos: jpg, jpeg, gif, png, txt, doc, xls, pdf, ppt, pps, odt, ods, odp, mp4, avi, mpeg, mpg y mp3.

4.2.7 Negocio

Costes de licencia: El software utilizado para el desarrollo y gestión de la aplicación debe ser gratuito.

4.2.8 Integración con el site hipeac.net

El aspecto visual de la web debe basarse en el site www.hipeac.net, puesto que se pretende que este proyecto sea uno “anexo” al site mencionado.



4.3 Diagrama Casos de uso

Con la finalidad de analizar los posibles comportamientos que podrán llevar a cabo los distintos usuarios que accedan a la aplicación y empezar a definir así que funcionalidades deberá incorporar la aplicación en sí, se presentan los diagramas de caso de uso pertenecientes a cada uno de los 4 roles de usuario definidos.

Más adelante se describirán de forma más extendida cada uno de los casos de uso de los distintos roles.

4.3.1 Anonymous User

Podemos apreciar que el comportamiento de este usuario dentro de la aplicación se reducirá a acceder a las diferentes estructuras de contenidos (eventos futuros, pasados y actuales, a las noticias, la lista de grupos y lista de tags).

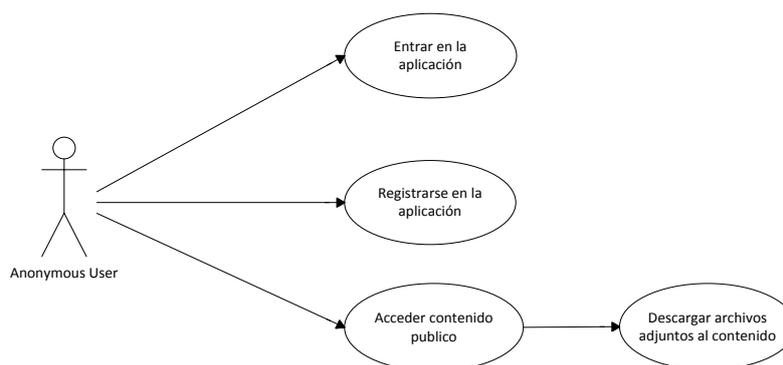


Ilustración 2: Casos de Uso Usuario Anónimo

4.3.2 Authenticated User

Este usuario tendrá algún privilegio más que el usuario no autenticado, como por ejemplo la posibilidad de afiliarse a un grupo y el acceso al contenido de los grupos a los que está afiliado y tendrá la posibilidad de modificar su perfil.

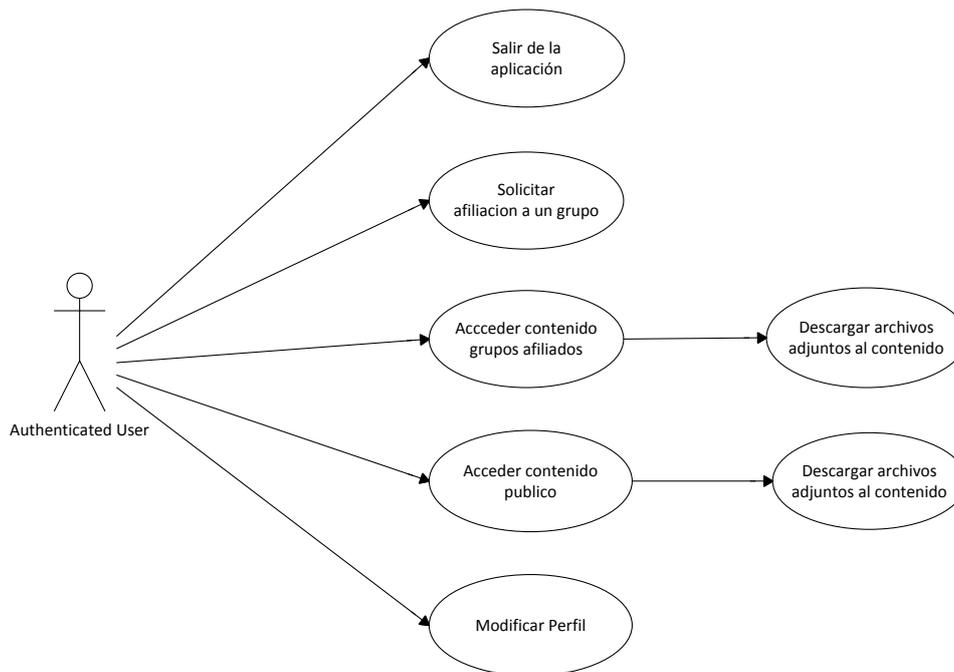


Ilustración 3: Casos de uso Usuario Autenticado

4.3.3 Organizer

El usuario con rol de organizer dispone de una mayor cantidad de competencias dentro de la aplicación, pues es el encargado de crear, modificar y eliminar cualquier tipo de contenido y gestionar los grupos que haya creado. Podrá invitar a miembros al grupo, aceptar o rechazar la solicitud de afiliación de los usuarios, entre otras funcionalidades.



Ilustración 4: Casos de uso Organizador

4.3.4 Administrator

El administrador tiene las mismas competencias respecto a la gestión de contenido y grupos que el organizer.

Además tendrá la opción de gestionar los usuarios, gestionar la configuración del site y revisar los logs del sistema.

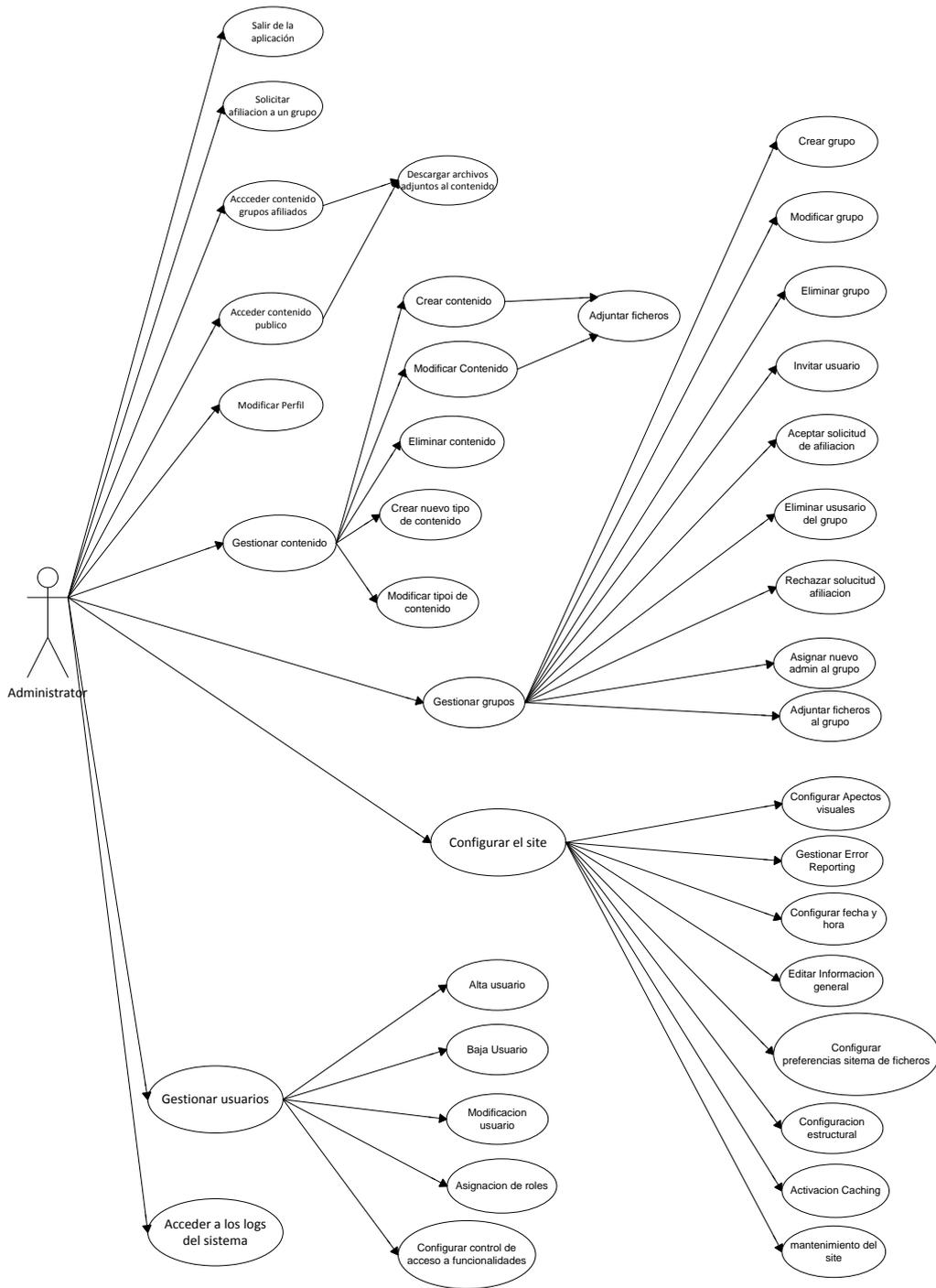


Ilustración 5: Casos de uso Administrator

4.4 Descripción casos de uso

A continuación se describen los casos de uso que se requieren en la especificación de la aplicación:

• Generales

Caso de uso:	Registrarse en la aplicación
Perfil:	Anónimo
Descripción:	Se muestra el formulario de registro a la aplicación, donde deberá introducir un nombre de usuario y un mail válido.
Caso de uso:	Entrar en la aplicación
Perfil:	Anónimo
Descripción:	Se muestra el formulario de login de la aplicación.
Caso de uso:	Acceder al contenido público
Perfil:	Todos
Descripción:	Se puede acceder al contenido público desde cualquier parte del site, tanto evento pasados, futuros y actuales, noticias, lista de grupos, lista de tags, y eventos clasificados por tags, links externos, etc.
Caso de uso:	Descargar ficheros
Perfil:	Todos
Descripción:	Se pueden descargar todos los ficheros adjuntos al contenido al cual tengamos acceso.

Caso de uso:	Solicitar afiliación a un grupo
Perfil:	Autenticado, organizador y administrador
Descripción:	Podemos solicitar la afiliación a un grupo. Si el grupo es de libre afiliación seremos miembros del grupo automáticamente. Si se necesita supervisión del administrador del grupo, deberemos esperar a que este nos acepte.

Caso de uso:	Modificar perfil
Perfil:	Autenticado, organizador y administrador
Descripción:	Cualquier usuario registrado podrá modificar su perfil (username, mail, afiliaciones a los grupos).

Caso de uso:	Acceder al contenido de grupo afiliado
Perfil:	Autenticado, organizador y administrador
Descripción:	Una vez afiliado a un grupo, el usuario podrá acceder al contenido asociado al grupo.

• *Gestión contenido*

Caso de uso:	Crear evento
Perfil:	Organizador, administrador
Descripción:	El usuario podrá dar de alta un evento mediante el formulario correspondiente.

Caso de uso:	Modificar evento
Perfil:	Organizador, administrador
Descripción:	El usuario creador del evento podrá modificar el evento en cuestión, o bien podrá ser modificado por un administrador.

Caso de uso:	Eliminar evento
Perfil:	Organizador, administrador
Descripción:	El usuario creador del evento podrá eliminar el evento en cuestión, o bien podrá ser eliminado por un administrador.

Caso de uso:	Crear noticia
Perfil:	Organizador, administrador
Descripción:	El usuario podrá dar de alta una noticia mediante el formulario apropiado.

Caso de uso:	Modificar noticia
Perfil:	Organizador, administrador
Descripción:	El usuario creador de la noticia podrá modificar el nodo en cuestión. También podrá ser editado por un administrador.

Caso de uso:	Eliminar noticia
Perfil:	Organizador, administrador
Descripción:	El usuario creador de la noticia podrá eliminarla. También podrá ser eliminada por un administrador.

Caso de uso:	Crear nuevo tipo de contenido
Perfil:	Administrador
Descripción:	El usuario podrá crear nuevos tipos de contenido, definiendo los campos y atributos del mismo mediante diferentes formularios.

Caso de uso:	Modificar tipo de contenido
Perfil:	Administrador
Descripción:	El usuario podrá modificar los tipos de contenido creados por él, pero no los creados por los demás.

Caso de uso:	Eliminar tipo de contenido
Perfil:	Administrador
Descripción:	Podrá eliminar los tipos de contenido que el mismo usuario ha creado.

Caso de uso:	Asignar ficheros al contenido
Perfil:	Organizador, administrador
Descripción:	Los usuarios podrán subir ficheros y añadirlos al contenido que deseen (evento, noticia, grupo).

• *Gestión de grupos*

Caso de uso:	Crear grupo
Perfil:	Organizador, administrador
Descripción:	Mediante un formulario, el usuario podrá dar de alta un grupo en la aplicación, donde definirá todos sus atributos relacionados.

Caso de uso:	Modificar grupo
Perfil:	Organizador, administrador
Descripción:	El usuario podrá modificar el grupo que haya creado, y el contenido asociado a él.

Caso de uso:	Eliminar grupo
Perfil:	Organizador, administrador
Descripción:	El creador del grupo podrá eliminarlo. Tendrá que decidir si desea mantener el contenido asociado al grupo y asociarlo a otro grupo, o bien eliminarlo también.

Caso de uso:	Invitar usuarios
Perfil:	Organizador, administrador
Descripción:	El creador del grupo podrá invitar directamente usuarios al grupo sin la necesidad de que soliciten la afiliación.

Caso de uso:	Aceptar solicitud de afiliación
Perfil:	Organizador, administrador
Descripción:	Cuando un usuario requiere la afiliación al grupo, el administrador de este, podrá aceptar la solicitud o rechazarla.

Caso de uso:	Rechazar solicitud de afiliación
Perfil:	Organizador, administrador
Descripción:	Si el administrador del grupo decide no admitir a un usuario, puede rechazar su solicitud de afiliación.

Caso de uso:	Eliminar usuario del grupo
Perfil:	Organizador, administrador
Descripción:	El administrador del grupo puede desafiliar a un usuario de un grupo. Entonces este no podrá acceder al contenido, hasta que haga la solicitud de nuevo y se acepte.

Caso de uso:	Asignar ficheros al grupo
Perfil:	Organizador, administrador
Descripción:	El grupo puede contener ficheros adjuntos, que pueden ser reglas del grupo, descripción del mismo, o documentación necesaria para los usuarios del grupo. El administrador del grupo podrá asociar ficheros al grupo para que sus afiliados puedan descargarlo.

• Configuración del site

Caso de uso:	Configurar aspectos visuales
Perfil:	Administrador
Descripción:	La configuración de aspectos visuales es un conjunto de posibilidades que tienen al administrador para modificar los estilos del site.
Caso de uso:	Gestionar error reporting
Perfil:	Administrador
Descripción:	El usuario podrá configurar los redireccionamientos de páginas en caso de que se le muestre una página de error al usuario.
Caso de uso:	Configurar fecha y hora
Perfil:	Administrador
Descripción:	El usuario podrá modificar los parámetros hora, fecha y zona horaria.
Caso de uso:	Configurar información general del site
Perfil:	Administrador
Descripción:	El usuario dispondrá de un formulario para editar esta información.
Caso de uso:	Configurar preferencias sistema de ficheros
Perfil:	Administrador
Descripción:	El usuario podrá definir qué tipo de gestión de ficheros se realiza en la aplicación, definición de directorios de almacenamiento de ficheros, acceso general a los mismos, etc.

Caso de uso:	Configurar estructura del site
Perfil:	Administrador
Descripción:	El usuario podrá definir la estructura de los bloques del site, así como de los primary links y vistas disponibles para los usuarios.

Caso de uso:	Control de caching
Perfil:	Administrador
Descripción:	Se podrá activar o desactivar la gestión de cache para el site.

Caso de uso:	Mantenimiento del site
Perfil:	Administrador
Descripción:	El usuario podrá desactivar la aplicación mediante un formulario que le permitirá mostrar una página principal en el que se muestre el mensaje "Site under maintenance"

- **Gestión de usuarios**

Caso de uso:	Alta usuario
Perfil:	Administrador
Descripción:	El administrador podrá dar de alta un usuario, editando sus datos de perfil.

Caso de uso:	Baja usuario
Perfil:	Administrador
Descripción:	Eliminación del usuario de la aplicación.

Caso de uso:	Modificación usuario
Perfil:	Administrador
Descripción:	El administrador podrá modificar los datos de perfil de cualquier usuario.

Caso de uso:	Asignación de roles
Perfil:	Administrador
Descripción:	EL usuario podrá asignar los diferentes roles a los usuarios de la aplicación.

Caso de uso:	Configuración control de acceso a funcionalidades del site
Perfil:	Administrador
Descripción:	Funcionalidad de la aplicación que permitirá al administrador determinar que funcionalidades se atribuyen a cada uno de los roles de la aplicación.

• *Logs*

Caso de uso:	Acceder a los logs del sistema
Perfil:	Administrador
Descripción:	Conjunto de funcionalidades que le permitirán al administrador consultar todos los logs que se generan en el sistema.

5. Diseño

Un trabajo importante de la fase de diseño, es escoger la tecnología con la que se implementará el entorno visual. Esta elección condiciona en gran medida el éxito del proyecto y es necesario tomársela con detenimiento.

Una de las opciones es utilizar un editor web. Los editores web nos facilitan la tarea de creación de páginas y formularios en HTML, también podemos encontrar algunos que interpretan PHP, ASP, JSP, Python o Ruby, lenguajes web con los que podamos crear las funcionalidades del entorno visual. Esta elección sólo se ve limitada por el conocimiento y la habilidad del programador al desarrollar el código que dará forma al proyecto.

Existen también otras herramientas de construcción de espacios virtuales. Herramientas con las que se puede construir rápidamente una página web, personalizarla y adaptarla a las propias necesidades, y todo esto con unas garantías de seguridad, son los llamados gestores de contenidos. En este capítulo se explica qué es un gestor de contenidos y porque nos puede ser útil.

5.1 Que es un CMS?

Los sistemas gestores de contenidos, del inglés *Content Management System* (CMS), son aplicaciones que facilitan una interficie para la creación, edición, administración y publicación de contenidos de forma rápida y sencilla. En el caso de los gestores de contenidos web (WCMS), son aplicaciones web diseñadas específicamente para el tratamiento del contenido web, de manera que se simplifican todas las tareas (creación, edición, administración, publicación y mantenimiento) hasta el punto que una persona sin conocimientos de HTML (*HyperText Markup Language*) ni PHP podría ser capaz de gestionar un sitio web, aunque el hecho de tener conocimientos de programación ayuda a que los CMS aún sean más potentes y versátiles.

El contenido que generan es almacenado en las bases de datos asociadas a la aplicación, el cual, es frecuentemente guardado como XML (*eXtensible Markup Language*) para facilitar la reusabilidad y permitir opciones de presentación flexibles. La presentación se basa en un conjunto de plantillas concretas y personalizables en las

cuales se define cómo se debe mostrar cualquier contenido almacenado en la base de datos.

Por otra parte, los CMS definen diferentes roles de usuario (tal y como hemos definido en el apartado anterior) a los cuales se les otorgan diferentes partes del contenido y la ejecución de las tareas concretas. Además, los CMS acostumbran a disponer de un área de administración integrada en la aplicación a través de la cual se pueden gestionar los contenidos, la presentación, los usuarios, los permisos... es decir, casi la totalidad de la aplicación.

- **Justificación del uso e implantación de un CMS:**

El hecho de utilizar un CMS en el desarrollo y el mantenimiento de una aplicación aporta muchos beneficios si se compara con el desarrollo basado en una metodología tradicional de una aplicación (programada también en PHP y que utilice bases de datos).

A continuación se exponen los más significativos.

- Facilidad de uso. Cualquier usuario habituado al uso de Internet es capaz de navegar e interactuar con la aplicación. Además, por parte del administrador tan solo se requiere el aprendizaje de funcionamiento del propio CMS.
- Fácil de poner en marcha y configurar. La instalación del CMS se acostumbra a basar en una serie de pasos automatizados. Una vez realizada, la gestión del CMS se realiza a través menús y formularios incluidos en el apartado de administración que configuran el contenido, su organización y disposición al navegador, gestionar los usuarios, sus permisos, las características del sitio web y las herramientas que permiten llevar a cabo estas acciones.
- Aplicación más estable, más segura y con herramientas útiles incorporadas (o con posibilidad de incorporarse). El CMS que se instala ha sido probado, y se han resuelto muchos problemas para garantizar su seguridad. A parte, el hecho de que sea utilizado por muchos usuarios finales facilita el descubrimiento de nuevos problemas que pueden ser solucionados por el equipo de desarrolladores. De otra parte, los CMS incorporan muchas herramientas

(como la generación automática de menús de navegación o buscadores) y dan la oportunidad de instalar nuevas.

- Bajo coste económico. Si la aplicación es de software libre, el coste se basará en el coste de instalación y el coste de aprendizaje por parte del desarrollador. Respecto al coste de mantenimiento de un CMS, se ahorra mucho más tiempo al crear o editar el contenido así como instalar nuevas funcionalidades.
- Cumplimiento de estándares de Internet. Casi todos los CMS cumplen (o intentan cumplir) los estándares de Internet establecidos por el W3C⁶ para garantizar la accesibilidad del lugar. De esta manera la aplicación será correctamente visualizada en cualquier navegador y sistema operativo.
- Más atractivo para los usuarios finales. La adaptación a una imagen moderna, de acuerdo con otras aplicaciones (hipeac.net) hará que sea más atractiva y permitirá una mayor interacción con los usuarios finales.

En definitiva, la elección de realizar el proyecto mediante un CMS creemos que es la más acertada.

5.2 Elección del CMS

Para el desarrollo de este proyecto se ha escogido el gestor de contenidos Drupal, debido a que se ha querido integrar visual y estructuralmente esta aplicación con el site www.hipeac.net, y éste utiliza Drupal.

Respecto a la versión de Drupal que se iba a usar, se ha determinado la versión 5, porque los módulos que se necesitaban instalar para satisfacer las funcionalidades que se requerían en la aplicación solo estaban implementados para Drupal 5, lo que nos ha obligado a utilizar esta versión. Cuando estén disponibles todos los módulos utilizados en la aplicación para la versión 6 de o 7 de Drupal⁷ se podrá proceder a la actualización del site.

⁶ <http://www.w3c.es/>

⁷ En la comunidad de Drupal se puede consultar esta información (drupal.org)

5.3 El gestor de contenidos Drupal

Ahora procedemos a entrar un poco más a fondo en el sistema gestor de contenidos Drupal.

5.3.1 Descripción general

Drupal es un CMS y como tal, proporciona un entorno de desarrollo orientado a gestionar contenidos. Dentro del amplio abanico de posibilidades, Drupal ofrece una API (application programming interface), que gracias a esta capa intermedia de programación eliminamos muchos riesgos de seguridad programando y ganamos mucho tiempo de desarrollo. Estas funciones son soportadas por los desarrolladores encargados del proyecto y mejoradas por la comunidad y en ocasiones son actualizadas para eliminar errores, ya sean funcionales o de seguridad. En este sentido Drupal de nuevo es una buena herramienta, ya que permite la actualización programada, manual y al más mínimo detalle.



A grandes rasgos Drupal se caracteriza por tener un buen diseño y un código de buena calidad (documentado para ser aceptado y con una gran comunidad). Es muy extensible ya que está basado en un sistema de módulos (hay miles disponibles en la propia página del proyecto) que se pueden añadir o quitar independientemente del resto. Entre ellos destaca la posibilidad de internacionalización y todo tipo de gestores de tipo de datos.

La API de Drupal también nos ofrece toda una capa de acceso a bases de datos con funciones que prestan especial atención a la seguridad. En la presentación de datos Drupal cuenta con una potente gestión de temas gráficos, extensiones para usar AJAX (Asynchronous JavaScript And XML), Java Script y de manera transparente añadir una gran cantidad de efectos gráficos a nuestras aplicaciones Web. El aspecto visual, aunque no imprescindible funcionalmente no deja de ser un punto importante de cara al usuario final.

Básicamente Drupal se basa en:

- Sistema modular basado en eventos (hooks)
- Capa de abstracción de base de datos

- Sistema de menús
- Generación de formularios
- Sistema de control de acceso
- Sistema de temas gráficos (templates)

5.3.2 Módulos en Drupal

Los módulos de Drupal son extensiones de sus capacidades básicas que aportan funcionalidades genéricas como podría ser el uso de un calendario, creación de nuevos tipos de contenido, o específicas, si se ha implementado un módulo bajo demanda.

El sistema de módulos de Drupal es uno de sus puntos fuertes y para manejarlos se basa en los llamados Hooks. Los Hooks son una serie de eventos definidos por los diseñadores de Drupal que permiten que cualquier módulo interactúe con el sistema gestor.

La idea de los Hooks es que al haber un evento, todas las funciones mencionadas con el nombre de este evento son ejecutadas; sencillo y potente. Por ejemplo, el Hook menú, muestra los datos en el menú de Drupal. Hay más de 70 Hooks implementados al núcleo de Drupal⁸. Todo esto permite desarrollar módulos que no interfieran en la codificación del resto, y así poder añadir y eliminar módulos de forma transparente a los demás.

5.3.4 Arquitectura

Para empezar, el hecho de utilizar Drupal hace que sea este sistema el que defina la arquitectura del software. Drupal propone una arquitectura por capas que guarda mucha relación con un MVC (Modelo-Vista-Controlador) aunque muchos usuarios le atribuyen más aspectos de una arquitectura PAC (Presentación-Abstracción-Control).

No vamos a entrar en detalles de la arquitectura software que utiliza Drupal, pero podemos resumirlo en la idea de que, a grandes rasgos, utiliza una arquitectura por capas con distinción entre presentación, lógica de la aplicación y gestión de la persistencia de datos. Drupal tiene también un diseño que utiliza algunos patrones

⁸ <http://api.drupal.org/api/group/hooks/5>

muy conocidos en la ingeniería del software, como por ejemplo: Singleton, Decorator, Observer, Bridge, Chain of Responsibility y Command.

Llegados a este punto, vale la pena mencionar que Drupal, pese a utilizar PHP para su codificación, no está encasillado dentro del paradigma de orientación a objetos. PHP, a partir de la versión 5 soporta orientación a objetos, y Drupal trabaja en algunos aspectos con estos, pero su núcleo está basado en código procedural. La notación UML funciona y encaja muy bien en sistemas que se piensan en orientación a objetos.

Igual que sucede con cualquier sistema de notación, UML es capaz de representar unos sistemas de forma más concisa o eficiente que otros. Así pues, los desarrolladores tienden a ir hacia soluciones que resultan de la intersección entre las capacidades de UML y el lenguaje de implementación utilizado. Este problema es especialmente importante si dicho lenguaje de implementación no se adhiere a la doctrina de orientación a objetos más ortodoxa, y se agrava en la medida de las diferencias existentes entre el lenguaje de implementación y la capacidad expresiva de UML.

Por otra parte, también hay que tener en cuenta que Drupal en sí mismo está constituido por un sistema central, que se va extendiendo y adquiriendo funcionalidades nuevas mediante los módulos. Al tratarse de Open Source Software, Drupal tiene detrás una comunidad muy activa de usuarios que mejoran y amplían los módulos constantemente, y también producen módulos nuevos para suplir funcionalidades que la versión por defecto no cubre. En esta situación, muchas veces, cuando necesitamos añadir una nueva funcionalidad a nuestro web site, es probable que otro usuario lo haya necesitado antes y haya diseñado e implementado un módulo para cubrir esa necesidad, por lo que podemos adoptar este módulo y re-utilizarlo para nuestro sistema. De este modo agilizamos el desarrollo de nuestro sistema y no “reinventamos la rueda”.

Sin embargo, al hacer esto, traemos a nuestro sistema piezas que ya han sido diseñadas de una forma concreta, y sirven para un propósito determinado. Si esto no acaba de encajar en nuestro desarrollo, tenemos que tomar la decisión de, o bien adaptar nuestro diseño al del nuevo módulo para que este encaje de forma

consistente en nuestro sistema, o bien renunciar a este módulo que es “casi” lo que queremos, pero no lo suficiente, y hacer un desarrollo propio más a medida.

5.3.5 Características generales

A continuación se mencionan las características más relevantes de Drupal.

- **Ayuda on-line:** un robusto sistema de ayuda on-line y páginas de ayuda para módulos del “núcleo” tanto para usuarios como por los administradores.
- **Búsqueda:** todo el contenido en Drupal está totalmente indexado en tiempo real y se puede consultar en cualquier momento.
- **Código abierto:** el código fuente de Drupal esta libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario que otros sistemas de “blogs” o de gestión de contenido propietarios es posible extender o adaptar Drupal según nuestras necesidades.
- **Módulos:** la comunidad de Drupal ha contribuido con muchos módulos que proporcionan funcionalidades como “página de categorías”, autenticación mediante LDAP, mensajes privados, bookmarks, etc.
- **Personalización:** un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo a las preferencias definidas por el usuario.
- **URL’s amigables:** Drupal utiliza el mod rewrite de Apache para crear URLs que es manejable por los usuarios y los motores de búsqueda.
- **Gestión y validación de usuarios:**
 - Los usuarios se pueden registrar e iniciar sesión de forma local o utilizando un sistema de autenticación externo como Jabber, Blogger, Live Journal u otro sitio Drupal.
 - Para su uso en una intranet, Drupal se puede integrar con un servidor LDAP.
 - En Drupal, los permisos están basados en roles. Así los administradores de Drupal no tienen que establecer permisos para cada usuario y pueden asignar permisos a un rol y agrupar los usuarios por roles.

- **Gestión de contenidos:** para la gestión de contenidos Drupal utiliza el control de versiones. El sistema de control de versiones de Drupal permite seguir y auditar totalmente las sucesivas actualizaciones del contenido: qué se ha cambiado, la hora y la fecha, quien lo ha cambiado, etc.
- **Enlaces permanentes (Permalinks):** todo el contenido creado en Drupal tiene un enlace permanente asociado a él para que pueda enlazar externamente sin temor a que el enlace falle en el futuro.
- **Objetos de contenido (Nodes):** el contenido creado en Drupal es, funcionalmente, un objeto (Node). Esto permite un tratamiento uniforme de la información como una misma cola de moderación para los envíos de diferentes tipos, promocionar cualquiera de estos objetos a la página principal o permitir comentarios –o no- sobre cada objeto.
- **Plantillas (template):** el sistema de temas de Drupal separa el contenido de la presentación permitiendo controlar o cambiar fácilmente el aspecto del sitio web. Se pueden crear plantillas con HTML y/o con PHP.
- **Sindicación del contenido:** Drupal exporta el contenido en formato RDF/RSS (Resource Description Framework/Really Simple Syndication) para ser utilizado por otros sitios web. Esto permite que cualquiera con un navegador agregador de noticias visualice el contenido publicado en la web desde el escritorio.
- **Blogging:** Drupal incluye un potente agregador de noticias para leer y publicar enlaces a noticias de otros sites. Incorpora un sistema de cache en la base de datos, con temporización configurable.
- **Soporte de Blogger API:** el API de Blogger permite que un sitio Drupal sea actualizado utilizando diversas herramientas que pueden ser herramientas web o herramientas de escritorio que proporcionen un entorno de edición más manejable.
- **Plataforma:** Independencia de la base de datos. Aunque la mayor parte de las instalaciones de Drupal utilicen MySQL, existen otras opciones. Drupal incorpora una capa de abstracción de base de datos, que actualmente esta implementada para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.

- **Multiplataforma:** Drupal ha sido diseñado desde el principio para ser multiplataforma. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X. Por otra parte, al estar implementado en PHP, es totalmente portable.
- **Múltiples idiomas y localización:** Drupal está pensado para una audiencia internacional y proporciona opciones para crear un portal multilingüe. Todo el texto puede ser fácilmente traducido utilizando una interficie web, importando traducciones existentes o integrando otras herramientas de traducción GNU Gettext.
- **Administración y análisis vía web:** la administración y configuración de Drupal se puede realizar completamente con un navegador y no necesita ningún programario adicional.
- **Análisis, seguimiento y estadísticas:** Drupal puede mostrar a las páginas de administración informes sobre referals (enlaces entrantes), popularidad del contenido o de cómo los usuarios navegan por el sitio.
- **Registros e informes:** Toda la actividad y los acontecimientos del sistema son capturados en un registro de eventos que puede ser visualizado por un administrador.
- **Características de comunidad, comentarios enlazados:** Drupal proporciona un potente modelo de comentarios enlazados que hace posible seguir y participar fácilmente en la discusión sobre el comentario publicado. Los comentarios son jerárquicos, como en un grupo de noticias o un fórum.
- **Encuestas:** Drupal incluye un módulo que permite a los administradores y/o usuarios crear encuestas on-line totalmente configurables.
- **Fórums de discusión:** Drupal incorpora fórums de discusión para crear sitios comunitarios vivos y dinámicos.
- **Libre colaborativo:** esta característica es única de Drupal y permite crear un proyecto o 'libro' a ser escrito y que otros usuarios contribuyan con el contenido. El contenido se organiza en páginas cómodamente navegables.
- **Rendimiento y escalabilidad:** Drupal incorpora un mecanismo de control de congestión que permite habilitar y deshabilitar determinados módulos o

bloques dependiendo de la carga del servidor. Este mecanismo es totalmente configurable y ajustable.

- **Sistema de Caché:** el mecanismo de caché elimina consultas en la base de datos incrementando el rendimiento y reduciendo la carga del servidor.

Aunque el desarrollo de Drupal se centraliza en drupal.org, existen además de Drupal Hispano (comunidad hispanica de Drupal), diferentes comunidades o grupos de usuarios sobre Drupal en castellano y catalán.

5.3.5 Bases de datos y seguridad

La capa de abstracción de bases de datos es muy importante por todo lo que pretendemos ofrecer, ya que el uso que haremos mayoritariamente será recoger datos de formularios, guardarlos en la BBDD y mostrarlo posteriormente.

Aunque PHP ya ofrece muchas funciones de acceso a la BBDD, la API de Drupal aporta una funcionalidad extra; un alto grado de seguridad. Mientras que en PHP deben implementarse las propias funciones de seguridad para evitar inyecciones SQL o datos no saneados, gracias a la capa de abstracción de acceso a la BBDD ya no deberemos preocuparnos por esto.

5.4 Gestión de usuarios

La gestión de los usuarios en un sitio web es una tarea de gran importancia puesto que en muchas ocasiones no es suficiente con la clásica gestión de un único tipo de usuario que crea su propia cuenta y accede a todas las secciones del sitio con libertad.

Drupal incorpora un potente sistema de gestión de usuarios para que estos se registren, conecten, desconecten y mantengan sus perfiles de usuario actualizados de una forma organizada y controlada por parte de los administradores.

Los usuarios pueden crear sus cuentas y autenticarse con o sin permiso del administrador, según este crea conveniente. Un correo es enviado cuando la nueva cuenta se crea, y si necesita permiso se envía cuando la cuenta es activada por el administrador.

Una vez creados los usuarios, Drupal permite agruparlos por roles que permiten ajustar fácilmente la seguridad y administración del sitio. Un rol define un grupo de

usuarios que tienen ciertos privilegios que deben definirse en la sección permisos del usuario.

Drupal incluye de forma predefinida dos tipos de roles, usuario anónimo y usuario autenticado:

- Usuario anónimo: se emplea para los usuarios que visitan el sitio web pero no se han creado una cuenta o para los usuarios que no se han autenticado.
- Usuario autenticado: este rol se asigna automáticamente a todos aquellos usuarios que ingresan en el sitio.

Si con este tipo de administración no es suficiente se pueden crear tantos roles como sean necesarios y darle los permisos adecuados.

Cuando los roles han sido creados es necesario darles los permisos de los que van a disponer porque cuando se crea un nuevo tipo de rol en Drupal, viene por defecto sin ningún permiso.

Para ello se dispone de la sección permisos, que permite controlar qué es lo que pueden hacer los usuarios en el sitio. Cada rol de usuario tiene su conjunto de permisos, que aparecen listados en su totalidad y pueden ser incluidos o excluidos según sea necesario.

Los permisos también permiten compartir con usuarios la carga de administración en caso de que sea necesario, puesto que a cualquier rol se le puede asignar cualquier permiso.

6. Implementación

6.1 Herramientas de desarrollo y entorno de pruebas

Durante el proceso de implementación se ha tenido que instalar un entorno de pruebas para el proceso de aprendizaje del CMS, su configuración, instalación de módulos, etc. Para no realizar todas estas pruebas en el servidor de explotación, se ha utilizado un software que nos ha permitido ir construyendo nuestro site de forma local.

Se ha decidido, por tanto, el uso de un programa que incluya todo el software necesario para el funcionamiento de Drupal. Easyphp y XAMPP son dos de los programas que incluyen las funcionalidades necesarias para programar entornos web en PHP más conocidos. Finalmente nos hemos decidido por Xampp.

6.1.1 Xampp

Xampp es un programa liberado bajo licencia GNU/GPL que actúa como servidor web libre. Es fácil de usar y permite desarrollar aplicaciones web dinámicas.



Está compuesto principalmente por la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl.

Actualmente está disponible para las plataformas GNU/Linux, Windows, MacOS y Solaris y solamente requiere descargar y ejecutar un archivo y realizar unas pequeñas configuraciones en algunos de sus componentes.

6.2 Instalación Drupal

Drupal requiere para su instalación un servidor que pueda ejecutar scripts PHP, un motor de bases de datos soportado por PHP, y PHP.

El servidor recomendado es Apache pero opcionalmente puede funcionar con otros servidores como IIS, el servidor desarrollado por Microsoft para Windows.

La base de datos generalmente es MySQL o PostgreSQL, aunque los usuarios suelen usar en su mayoría la primera.

Las últimas versiones de Drupal funcionan correctamente con PHP 5.

6.2.1 Requerimientos del servidor

- Servidor web PHP: Apache 2.0.x o superior (recomendado), también funciona en Internet Information Server (IIS).
- Interprete PHP: PHP 4.3.5 o superior.
- Servidores de BBDD: MySQL 4.1 o superior (recomendado), también PostgreSQL 7.4 o superior.



Los requerimientos para todas las versiones de Drupal se pueden encontrar en:

- <http://drupal.org/requirements/>.

6.2.2 Configuración del servidor

En primer lugar deben instalarse los servicios PHP y MySQL. Mediante sencillos comandos podremos disponer de ellos en nuestro servidor.

Posteriormente deberemos instalar un gestor de BBDD, en nuestro caso instalaremos un gestor gráfico de BBDD que nos permitirá la creación y edición de las tablas necesarias para el correcto funcionamiento de Drupal de una forma sencilla y rápida. En nuestro caso hemos instalado PhpMyAdmin.

6.2.3 Creación de las BBDD's

Entramos en el administrador de base de datos para crear la base de datos que utilizará Drupal para almacenar toda la información necesaria. Para crearla, simplemente bastará con introducir un nombre para la misma y seleccionar el cotejamiento, que es la codificación para la base de datos.

De ahora en adelante, ya no necesitaremos entrar en el gestor de BBDD, ya que la instalación de Drupal se encargará de crear todas las tablas y las relaciones entre ellas.

Una vez establecida la base de datos ya se puede iniciar la instalación de Drupal.

6.2.4 Proceso de Instalación de Drupal

En las primeras versiones que aparecieron de Drupal su instalación no resultaba una tarea sencilla puesto que había que instalarlo todo manualmente y crear las tablas y sus relaciones en la base de datos.

A partir de las versiones 5.x y 6.x el proceso de instalación se ha mejorado y la instalación resulta más sencilla, considerando también que ya no es necesario crear las tablas, como se ha dicho anteriormente.

Después de descargar la versión que nos interesa, se descomprimen los ficheros en la carpeta definida para dicha instalación⁹. La instalación se realiza desde el navegador web. Para ello debemos acceder a la dirección definida para nuestro site y aparecerá la pantalla de instalación de Drupal.

La instalación se realiza por defecto en inglés, pero posteriormente se puede traducir Drupal a otros idiomas, aunque en nuestro caso, no nos interesa.

Drupal nos pedirá el nombre de la base de datos que hemos creado para la instalación, y el usuario y contraseña del propietario de dicha base de datos. Una vez rellenados los campos podemos avanzar en la instalación pulsando en “Save Configuration”. En este momento nos aparece un mensaje de instalación completa y se nos muestra un link a la *home* de nuestro site.

6.2.5 Configuración inicial del sitio web

Una vez instalado Drupal ya se puede iniciar el desarrollo, pero primero deben realizarse algunas configuraciones básicas antes de empezar.

Debemos crear el primer usuario de la aplicación, que será el superusuario que dispondrá de control total de la aplicación. Este usuario será el encargado de realizar las configuraciones iniciales de nuestra web, y posteriormente podrá asignar otros usuarios como administradores de ciertos aspectos funcionales de la web.

La aplicación nos pide que creamos el primer usuario, que será el superadministrador del site. Creamos el nuevo usuario introduciendo un username y un mail válido, en el

⁹ En nuestro caso es <http://hipeac.ac.upc.edu/seminars>

que recibiremos el link de confirmación de la cuenta. Clicando en el link entraremos en la aplicación como superusuario y podremos acceder al menú de administración y a nuestro perfil para modificar la contraseña o el username.

En la información del sitio puede cambiarse el nombre del sitio web, así como el slogan, que será mostrado a menudo junto al título del sitio. Podremos cambiar también el mensaje de pie de página, texto que se muestra en la zona inferior de cada página y que es útil para añadir información, condiciones de uso y distribución.

Drupal trae en su núcleo una serie de módulos para realizar las tareas más importantes, aunque se pueden instalar muchos más, los cuales otorgan nuevas funcionalidades, tal y como se ha explicado en apartados anteriores.

Si se desea el idioma del sitio web es necesario el paquete de traducción, en este caso no nos hará falta porque lo dejaremos en inglés.

El site nos advierte de que debemos dirigirnos al reporte de estado (*status report*) y ejecutar el cron del servidor manualmente, modulo que sirve para la ejecución regular de acciones. Posteriormente deberemos definir el directorio donde se almacenaran los ficheros subidos al portal.

Realizadas las configuraciones básicas, el sitio web queda un poco más personalizado y puede iniciarse el desarrollo del sistema.

A continuación, se define por una parte el proceso de construcción del site mediante la instalación y configuración de los módulos utilizados, y por otra, la configuración general del site.

6.3 Módulos

En este apartado se especifica el procedimiento de instalación y configuración de módulos, así como los utilizados en esta aplicación.

Además de los diferentes módulos incorporados en la instalación básica de Drupal, la comunidad de usuarios de Drupal, ofrece abiertamente una lista muy extensa de módulos que nos permitirán añadir las funcionalidades requeridas en nuestra aplicación.

6.3.1 Instalación de módulos

Solo el superadministrador del site tiene permisos para añadir módulos a la aplicación. Se ha restringido el acceso a todos los demás usuarios para garantizar la estabilidad del site.

El proceso de instalación de cada uno de los módulos es el siguiente:

- Nos descargamos el modulo de la comunidad, que está en formato Zip.
- Lo descomprimos y obtenemos un directorio que contiene la implementación del mismo.
- Copiamos el directorio al destino: Drupal/modules.
- Entramos a la aplicación como superadministrador y activamos el modulo en cuestión en la sección “*Site Building* → *Modules*”.

6.3.2 Configuración de módulos

Así como la instalación y activación/desactivación de módulos solo puede ser controlada por el superadministrador del site, la configuración de los mismos puede ser controlada también por el usuario con el rol *administrator*. El superadministrador decidirá qué módulos podrán ser configurados por este rol de usuario.

Tanto el superadministrador como los usuarios que tengan acceso a la configuración del modulo en cuestión, tendrán determinadas opciones de configuración del mismo en el menú de administración.

6.3.3 Módulos utilizados

A continuación se detallan los módulos que se han utilizado para la implementación del site. La instalación básica de Drupal nos ofrece los módulos del núcleo necesarios para el correcto funcionamiento de la aplicación. Posteriormente se le han añadido más módulos para darle las funcionalidades que se han descrito en la especificación.

En el apartado anterior hemos introducido ya el concepto de los módulos, y a continuación veremos cuales se han utilizado para ofrecer las funcionalidades necesarias para satisfacer los requerimientos de nuestro sistema.

• Módulos del core

Por módulos del core, nos referimos a aquellos que ya vienen integrados con la versión básica de Drupal. Dentro del core, hay dos tipos de módulos: los requeridos y los opcionales.

Los requeridos son los siguientes:

- **Block:** Controla las cajas (o bloques) que se muestran alrededor del contenido principal.
- **Filter:** Maneja el filtrado de contenido para mostrarlo.
- **Node:** Permite que se pueda publicar contenido al sitio web y que se muestre en páginas.
- **System:** Gestión de la configuración general del sitio por administradores.
- **User:** Maneja el sistema de registro y acceso de usuarios.
- **WatchDog:** Genera y almacena los logs del sistema

Estos módulos son una parte vital del sistema, y siempre deben que estar habilitados. Sobre ellos se carga la responsabilidad de que funcionen las partes básicas del sistema.

Los módulos opcionales del core son módulos que también van incluidos por defecto en la instalación básica de Drupal, pero la mayor parte de ellos vienen desactivados inicialmente. Para el sistema que hemos desarrollado, se han activado los que se listan a continuación:

- **Agreggator:** Permite agregar contenido RSS a nuestro site. Se ha tenido que activar para poder posteriormente definir el contenido RSS que queríamos mostrar.
- **Color:** Permite cambiar el color de determinados temas del site.
- **Help:** Administra el despliegue de ayuda en línea. El uso que se hace de este módulo es principalmente interno, la mayor parte de ayuda en línea viene dada en forma de mensajes que guían, sobre todo a los administradores, a la hora de interactuar con el sitio web.
- **Menú:** Permite a los administradores personalizar los menú de navegación del site. Se trata de un módulo muy importante que gestiona el sistema de menús

de Drupal, la navegabilidad, los permisos y las conexiones de las diferentes partes de nuestro site.

- **Statistics:** Anota estadísticas de acceso a su sitio. Recopila estadísticas sobre la web que pueden ser consultadas posteriormente en los informes correspondientes.
- **Taxonomy:** Permite categorizar el contenido utilizando tanto etiquetas como términos definidos por el administrador o por el organizador. Es una herramienta flexible para clasificar contenido, con muchas funcionalidades avanzadas. Para empezar, debe crearse un vocabulario que contendrá un conjunto de términos y etiquetas. Se puede crear un vocabulario de etiquetaje libre para todo (como en nuestro caso), o bien vocabularios separados para definir las diferentes propiedades del contenido.

Con esta clasificación del contenido desaparece la idea estática de sección, y se añade flexibilidad muy grande a la hora de clasificar.

- **Upload:** Permite a los usuarios subir archivos y anexarlos a contenido. Con este módulo permitimos que los autores puedan adjuntar ficheros a sus eventos. Para hacer el upload de ficheros muy grandes utilizaremos otro módulo descrito más adelante.

• Módulos contribuidos

Los módulos contribuidos son aquellos que alguien ha desarrollado por su cuenta y ha decidido compartir con toda la comunidad, con el propósito de mejorar algún aspecto de Drupal u ofrecer alguna funcionalidad que no se puede obtener solamente con los módulos del core. Todos ellos se pueden descargar libremente del site oficial de Drupal. En ningún caso hemos necesitado modificar el código de los módulos contribuidos que hemos utilizado para este proyecto, solamente ha sido necesario instalarlos y configurarlos, por lo que tampoco nos extenderemos aquí en sus detalles, puesto que existen muchas fuentes de documentación para la mayoría de ellos.

- **Content Construction Kit (CCK):** es un módulo muy popular que consiste en un conjunto de módulos que permiten crear tipos de contenido adicionales respecto a los que ya existen por defecto en Drupal, asignándoles los atributos

que deseemos, tales como, campos de texto, radio-buttons, campos de imagen, etc.

Los módulos que hemos utilizado pertenecientes a CCK son:

- Content: Permite a los administradores definir nuevos tipos de contenido.
- Content templates: Nos permitirá definir la vista que se generará en el RSS feed y también podremos configurar el código XML que introduciremos en el RSS para que las aplicaciones iUPC i UPCdroid puedan obtener la información que necesitan.
- ImageField: Este modulo no venia por defecto en el CCK, se ha instalado posteriormente. Nos permite asignar un campo de upload de imagen para un tipo de contenido.
- Option Widgets: Permite asignar campos de radio-button, campos de texto (text-fields), y campos numéricos (numeric fields).
- Text: Nos permite crear campos de texto simple.
- Date: Nos permite crear campos de tipo Fecha, para luego ordenar el contenido en un calendario.
- **Event:** Conjunto de módulos que nos permite la gestión del tipo de contenido *evento*. Nos permitirá hacer un seguimiento de los eventos y mostrarlo o clasificarlos en un calendario. Los submodulos que se han activado son:
 - Basic event: Nos permite la creación y gestión de eventos.
 - Event: Podremos utilizar la API calendar para mostrar los eventos.
 - Event repeat: Añade la funcionalidad de hacer una secuencia de repetición de evento.
 - Event views: Permite la creación de tipo *evento* y su posterior visualización.
- **Organic Groups (OG):** Conjunto de módulos que nos permite la creación y gestión de módulos. Los submodulos utilizados son:
 - Organic Groups: Habilita a los usuarios crear y gestionar sus propios grupos.
 - Organic groups access control: Habilita el control de acceso para posts y grupos privados.

- **Google Analytics:** Añade el código que se debe incrustar en todas las páginas para hacer el tracking.
- **Taxonomy List:** Permite listar el contenido de las categorías (tags) que se han creado en el site.
- **Views:** Views es un módulo muy popular que ofrece una interfaz muy potente mediante la cual los usuarios pueden construir vistas sobre muchos aspectos de la web. Típicamente se usa para ofrecer listados de algún tema en concreto. Views permite concretar muchos aspectos de estos listados, como por ejemplo qué mostrar exactamente para cada ítem del listado que genere, condiciones que deben cumplir los elementos que se listen (mediante filtros y parámetros) y de qué forma se visualizarán los resultados de la vista generada. Otro aspecto interesante, es que también nos gestiona la generación de feeds RSS para nuestras vistas. De esta forma podemos ofrecer varios canales de este estilo por toda nuestra web. En uno de los próximos apartados se especifica las vistas definidas para la aplicación.

Los submódulos que hemos activado son:

- Views: Permite crear vistas customizadas de listas de contenido.
- Views RSS: Permite crear vistas con contenido RSS.
- Views UI: Nos permite editar las vistas creadas.
- **Web File Manager:** Módulo que nos permite añadir ficheros (previamente subidos al servidor por ftp) a la base de datos y adjuntarlos a los nodos, o a los grupos, adoptando las doctrinas de restricción de acceso del nodo o grupo en cuestión.

6.4 Configuración del site

En este apartado especificaremos las opciones de configuración que nos ofrecen Drupal y nuestra aplicación.

6.4.1 Usuarios y permisos

La definición de los diferentes roles permite que cada grupo de usuarios tenga unas funciones específicas en la aplicación. El mismo Drupal permite definir como

interactuará cada rol con el portal, pudiendo especificar a qué partes o contenido del site podrá acceder el usuario.

Drupal nos permite controlar qué es lo que pueden hacer los usuarios en el site. Cada rol de usuario tiene su conjunto de permisos, que aparecen listados en su totalidad y pueden ser incluidos o excluidos según sea necesario.

Los permisos también permiten compartir con usuarios la carga de administración en caso de que sea necesario, puesto que a cualquier rol se le puede asignar cualquier permiso.

En el siguiente esquema se muestra gráficamente los niveles de acceso de los roles de la aplicación:

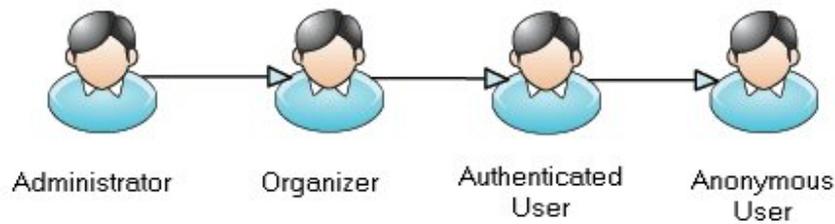


Ilustración 6: Niveles de acceso

El administrador será el rol de usuario con los permisos más elevados en el site. Seguidamente le proceden el organizador, que puede añadir contenido al site, y le siguen el usuario autenticado y no autenticado.

A continuación se muestra, a modo de ejemplo, la tabla de permisos en la que se define que funcionalidades se le atribuyen a cada uno de los perfiles de la aplicación. Mediante campos de tipo checkbox, podremos marcar, para cada uno de los cuatro roles, las funcionalidades que le atribuimos a cada uno.

Permission	administrator	anonymous user	authenticated user	organizer
aggregator module				
access news feeds	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
administer news feeds	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
basic event module				
create events	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
edit own events	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
block module				
administer blocks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
use PHP for block visibility	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
comment module				
access comments	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
administer comments	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
post comments	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
post comments without approval	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
contemplate module				
administer templates	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
content module				
Use PHP input for field settings (dangerous - grant with care)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
enc_remote module				
administer RSS Remote Enclosures	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
set RSS Remote Enclosure	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
eventrepeat module				
create repeat events	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
filter module				
administer filters	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
googleanalytics module				
administer google analytics	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
opt-in or out of tracking	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
use PHP for tracking visibility	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
imagefield module				
view imagefield uploads	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
menu module				
administer menu	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
node module				
access content	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
administer content types	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
administer nodes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
create audiencegroup content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
create infopage content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
create page content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
edit audiencegroup content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
edit infopage content	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
edit own audiencegroup content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
edit own infopage content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
edit own page content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
edit page content	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
revert revisions	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
view revisions	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
og module				
administer organic groups	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
statistics module				
access statistics	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
view post access counter	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
system module				
access administration pages	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
administer site configuration	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
select different theme	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
taxonomy module				
administer taxonomy	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
taxonomy_list module				
administer taxonomy_list	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
upload module				
upload files	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
view uploaded files	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
user module				
access user profiles	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
administer access control	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
administer users	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
change own username	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
views module				
access all views	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
views_podcast module				
views_ui module				
administer views	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
webfm module				
access webfm	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
administer webfm	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
view webfm attachments	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
webfm upload	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Ilustración 7: Tabla de permisos de usuario

6.4.2 Configuración del aspecto visual

Drupal es un sistema de gestión de contenidos altamente configurable, tanto en las funcionalidades que proporciona, como en el aspecto visual, configurable mediante los temas que incluye y los que pueden ser instalados, y posteriormente modificados.

Los temas en Drupal definen el aspecto visual del sitio web, incluyendo entre otras cosas, el tipo de letra, los colores, los menús y la posición de los bloques y el contenido mostrado en todas las páginas. Reciben el contenido que se pretende mostrar en una página y le dan formato.

Los temas que no incluye Drupal en la instalación pueden encontrarse y descargarse de muchos sitios en internet, especialmente la página oficial de Drupal, www.drupal.org, donde pueden encontrarse prácticamente todos los temas disponibles.

La configuración de los temas puede encontrarse en el apartado temas, dentro de la sección de administración, donde se listan todos los temas instalados y se ofrecen opciones para configurarlos.

Las opciones de configuración que se muestran ofrecen diferentes ajustes, entre los que se incluyen ajustes globales, que se aplican a todos los temas, y ajustes particulares para cada tema en concreto.

Las opciones de configuración permiten configurar muchos parámetros, como indicar que se muestre el logotipo que viene con el tema o uno propio, configurar las URL's que aparecerán en los enlaces e indicar que se muestren o no, diversos parámetros como fecha de publicación del contenido, avatar de los usuarios, nombre del sitio o enlaces primarios y secundarios. Drupal usa por defecto el tema Minnelli e incluye otros cuatro temas en su núcleo, pero ninguno de los cinco se adapta a las características de diseño para el sitio web que se pretende conseguir en este proyecto.

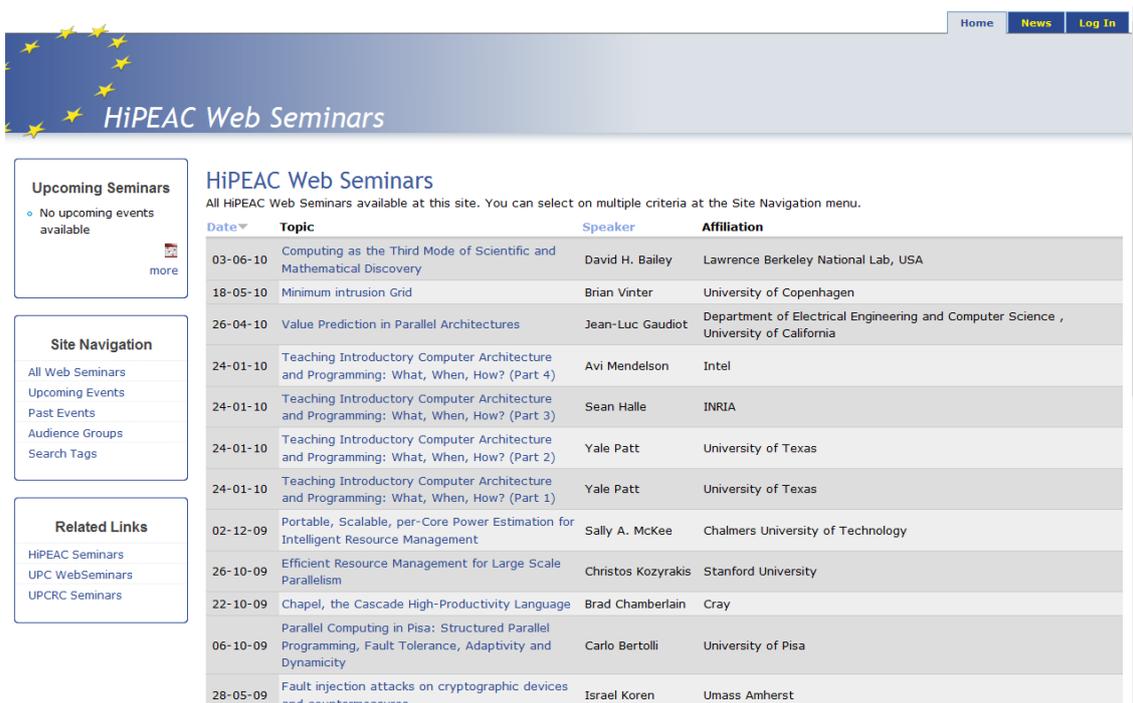
Por este motivo, deberá adaptarse el estilo que quiere otorgarse al sitio web que se está desarrollando, un estilo simple, dinámico, con un aspecto vivo y que sea intuitivo y fácil de usar para los usuarios.

Los administradores del site www.hipeac.net nos han cedido el tema que ellos utilizan actualmente en su portal. Lo hemos descomprimido y, como si de un tema nuevo se tratara, lo copiamos al directorio /Drupal/themes. Una vez hecho esto, el nuevo tema aparecerá automáticamente para ser seleccionado y configurado en el apartado temas de la sección administración.

El nuevo tema le proporcionará a nuestra web los rasgos básicos que va a tener en cuanto a aspecto visual, pero posteriormente se irá mejorando y modificando con el uso de menús, secciones, bloques y vistas, que podremos ir añadiendo gracias a los módulos que se han explicado en el apartado anterior.

Debe destacarse que, una vez instalado el tema en cuestión, hemos realizado una serie de cambios en los ficheros de estilo CSS¹⁰, que nos han permitido ajustar algunos parámetros de la configuración de la estructura de bloques y vistas.

El aspecto visual final de la aplicación ha sido:



Upcoming Seminars

- No upcoming events available

Site Navigation

- All Web Seminars
- Upcoming Events
- Past Events
- Audience Groups
- Search Tags

Related Links

- HiPEAC Seminars
- UPC WebSeminars
- UPCRC Seminars

HiPEAC Web Seminars

All HiPEAC Web Seminars available at this site. You can select on multiple criteria at the Site Navigation menu.

Date	Topic	Speaker	Affiliation
03-06-10	Computing as the Third Mode of Scientific and Mathematical Discovery	David H. Bailey	Lawrence Berkeley National Lab, USA
18-05-10	Minimum intrusion Grid	Brian Vinter	University of Copenhagen
26-04-10	Value Prediction in Parallel Architectures	Jean-Luc Gaudiot	Department of Electrical Engineering and Computer Science , University of California
24-01-10	Teaching Introductory Computer Architecture and Programming: What, When, How? (Part 4)	Avi Mendelson	Intel
24-01-10	Teaching Introductory Computer Architecture and Programming: What, When, How? (Part 3)	Sean Halle	INRIA
24-01-10	Teaching Introductory Computer Architecture and Programming: What, When, How? (Part 2)	Yale Patt	University of Texas
24-01-10	Teaching Introductory Computer Architecture and Programming: What, When, How? (Part 1)	Yale Patt	University of Texas
02-12-09	Portable, Scalable, per-Core Power Estimation for Intelligent Resource Management	Sally A. McKee	Chalmers University of Technology
26-10-09	Efficient Resource Management for Large Scale Parallelism	Christos Kozyrakis	Stanford University
22-10-09	Chapel, the Cascade High-Productivity Language	Brad Chamberlain	Cray
06-10-09	Parallel Computing in Pisa: Structured Parallel Programming, Fault Tolerance, Adaptivity and Dynamicity	Carlo Bertolli	University of Pisa
28-05-09	Fault injection attacks on cryptographic devices and countermeasures	Israel Koren	Umass Amherst

¹⁰ CSS: Cascading Style Sheet

6.4.3 Creación de bloques y menús

Como todo sitio web, es necesaria la creación de menús y bloques para tener las secciones bien organizadas y disponibles, y ofrecer todas las herramientas necesarias a los usuarios para poder acceder a ellas, crear nuevos contenidos y disponer de información actualizada sobre el sitio.

Se va a crear una sección con links en la cabecera para mostrar los links básicos o primarios (primary links), y 3 o 4 bloques (dependiendo del rol del usuario) en la parte izquierda, que contendrán diferente información. Uno de ellos será un listado de links en el que añadiremos enlaces a páginas relacionadas con nuestro site, otro bloque será el de navegación, en el que encontraremos redireccionamientos a diferentes secciones de nuestra web. Los dos bloques restantes son: uno para la creación de contenido y administración y el otro para la gestión de grupos. En función del rol del usuario aparecerán unos bloques u otros.

- **Bloques laterales**

En la mayoría de sitios web existen enlaces a otras páginas que están relacionadas con el propio sitio o complementan la información que este contiene. En este caso, para el propósito de este sitio web pueden ser necesarios enlaces a otras páginas, o bien links a diferentes partes del mismo site.

Gracias al módulo menú pueden crearse bloques que contiene links de una forma sencilla, porque al fin y al cabo, los bloques de links no son más que menús donde las secciones son enlaces a otras páginas.

Para crear los bloques de links es necesario ir a la sección menú, que se encuentra en construcción del sitio, dentro del menú de administración. Una vez dentro debe seleccionarse la opción añadir menú, darle nombre y crearlo. Una vez creado, habrá que editarlo para añadir el contenido y definir la ubicación del bloque, en la parte derecha o izquierda. Una vez guardados los cambios, las secciones de links serán creadas. Para finalizar deberemos definir a que usuarios se les debe mostrar el bloque en cuestión. Para crear los demás bloques seguiremos el mismo procedimiento.

Drupal nos permite generar diferentes vistas para organizar el contenido, y para acceder a ellas lo haremos mediante un link para cada una. En nuestro caso, para crear el bloque “*Site Navigation*”, que nos enlaza a las diferentes partes de la web, crearemos un bloque con diferentes links (igual que en el bloque anterior), pero en este caso los enlaces serán links a las vistas que habremos generado anteriormente.

Además de personalizar nuestros bloques también podemos utilizar los que vienen definidos por defecto en la instalación de Drupal, o bien los que se añaden al instalar nuevos módulos. Estos bloques generan información dinámica, información que va cambiando constantemente. Algunos de estos bloques son:

- *Calendar to browse events*: genera un calendario en el que se muestra el mes actual con un icono distintivo en los días en los que se imparte un seminario.
- *Group members*: Se le muestran al administrador de un grupo todos sus usuarios.
- *Group notifications*: Información del grupo (usuarios totales, usuarios online, gestión de los usuarios, etc).
- *New groups*: Nuevos grupos creados recientemente.
- *Popular content*: Contenido más visitado.
- *Primary links*: Links de acceso directo.
- *Taxonomy List for Tags*: Lista de tags y categorías.
- *User login*: Formulario de login para entrar a la aplicación.
- *Who's new*: Nuevos usuarios de la aplicación.
- *Upcoming events*: Nos muestra en forma de lista los 5 eventos futuros más próximos a hoy.

Drupal nos ofrece más bloques que los anteriores citados, simplemente se quiere dar una idea de sus posibilidades.

• *Primary Links*

Es uno de los bloques que viene por defecto en la instalación, formado por los “links destacados”. Sin tener en cuenta el tema instalado en la aplicación, estos links siempre destacarán con respecto a los demás links de la aplicación. Aparecerán en todas las paginas generadas, pudiéndose definir cada uno de ellos para los cuatro tipos de

perfiles definidos en la aplicación. En nuestro caso hemos definido los siguientes elementos: *home*, *login/logout* y *MyAccount*.

6.4.4 Tipos de contenido

La finalidad principal de un sistema de gestión de contenido es la creación de contenido y el tratamiento del mismo. Drupal proporciona las herramientas indispensables para crear contenido y organizarlo de forma adecuada.

Todo el contenido que se crea a través de Drupal es tratado internamente como un tipo de entidad genérica conocida como nodo. Un nodo podría considerarse como cualquier tipo de contenido que se le puede asociar de forma individual un conjunto de atributos independientemente que sea una noticia, un tema del foro o un artículo.

Los tipos de contenido que integra inicialmente Drupal son páginas e historias. Una página es un método sencillo de crear y mostrar información que no suele cambiar y es similar en forma y contenido a un artículo. Una historia, que tiene una estructura similar a una página, es ideal para crear y mostrar contenido que informa a los visitantes del sitio, como una entrada informal de un blog, un anuncio del sitio o una nota de prensa.

Sin embargo, en Drupal se pueden crear otros tipos de contenido gracias al módulo CCK (descrito en el apartado anterior). Permite la construcción de tipos de contenido personalizados de manera fácil a través de un panel de control.

CCK permite añadir nuevos campos al contenido básico, tales como imagen, fecha, link, email, votaciones o cualquier tipo de texto o campo numérico, etc, para personalizar el tipo de contenido que se está creando.

Para crear un tipo de contenido, en la sección tipos de contenido que se encuentra dentro de gestión de contenido se selecciona la opción añadir tipo de contenido. Se le da un nombre, se define cual es la función del tipo de contenido y se crea. A partir de este momento se le pueden añadir los campos que sean necesarios mediante el módulo CCK.

Una vez se han creado los tipos de contenido y ya existe contenido publicado, la forma más sencilla de organizarlo es mediante taxonomías. Drupal incluye en su núcleo el módulo taxonomía (definido en el apartado anterior), que permite categorizar el contenido usando etiquetas y términos definidos por un administrador.

Las categorías se implementan mediante vocabularios, términos, y las relaciones que se establecen entre ellos, que pueden ser simples o jerárquicas, incluyendo jerarquías múltiples.

Los vocabularios se dan de alta en Drupal en la sección taxonomía que se encuentra en gestión de contenido. Una vez creado el vocabulario se le pueden añadir términos o etiquetas para definir diversas propiedades del contenido o tenerlo más esquematizado. En nuestra aplicación hemos definido una única categoría llamada *tags*, que nos bastará para clasificar el contenido en etiquetas.

Crear un nuevo vocabulario es relativamente sencillo en Drupal, basta con ir a la sección taxonomía y seleccionar la opción añadir vocabulario. Se debe rellenar el campo nombre, las opciones y elegir los tipos de contenido que formarán la taxonomía.

Una vez guardados los cambios ya se pueden añadir términos y agregar contenido a través de la interfaz de taxonomía. En el propio formulario de alta de evento también se pueden crear diferentes etiquetas, sin tener que hacerlo desde el menú de administración (al cual solo tienen acceso los administradores).

6.4.5 Vistas

Si no se diseña de otro modo, todo el contenido publicado aparece en una lista simplemente organizado por orden de publicación, pero cuando el número de posts empieza a crecer se necesita otro tipo de organización para que todos los contenidos sean fácilmente accesibles. Por ese motivo se han creado diferentes vistas para ordenar el contenido publicado.

Para crear una vista se deben seguir el siguiente procedimiento:

- Añadir la información básica (nombre y descripción).

- Definir que roles podrán acceder a la vista.
- La URL o link de la vista, para poder acceder a ella.
- Definir el “Tipo de vista”: El formato de la vista, que puede ser una tabla, un globo con un título y descripción, vista para RSS, etc.
- Definir los “Campos”: Atributos que se deben mostrar en la vista.
- Definir los “Argumentos”: Permiten flexibilidad a la hora de generar vistas con parámetros concretos. Utilizaríamos este campo si quisiéramos, por ejemplo, que se ordenara el contenido por años y luego por meses.
- Filtros: Definimos que condiciones debe cumplir la selección de contenido, por ejemplo, tipo de contenido que mostramos, si está publicado o no, etc.
- Ordenación: Podemos ordenar el contenido por algún campo en concreto, ascendente o descendente.

Hemos definido las siguientes vistas para poder estructurar el contenido de nuestro site:

- **Events_list**: se muestran una lista de eventos en forma de tabla con la información básica de cada evento:
 - *Date*: Fecha de inicio del evento.
 - *Topic*: Título del evento.
 - *Speaker*: Ponente del evento.
 - *Affiliation*: Procedencia o afiliación del ponente.

En esta vista aparecerán todos los eventos públicos que hayan sido publicados en el site, tanto futuros como pasados, hasta un máximo de 20 eventos por página. Se utilizará esta vista como página principal de la web.

- **Events_repository**: vista que muestra todos los eventos pasados respecto a la fecha actual.
- **Furure_events**: se muestran los eventos futuros a la fecha de hoy y también los que se están impartiendo en este momento.
- **News**: Vista que nos muestra las noticias publicadas en la web.

- **Rss_list:** vista definida para las aplicaciones iUPC i UPCdroid. A través de esta vista, las dos aplicaciones obtendrán la información que necesitan para mostrar los eventos del site.

Además de las vistas anteriormente citadas, existen otras vistas que ya vienen previamente definidas en la instalación básica de Drupal. A continuación se describen las que se han utilizado en este proyecto:

- **Lista de grupos (og):** es el directorio de grupos de la aplicación. Todos los grupos creados como públicos aparecerán en esta vista.
- **Grupos afiliados del usuario (og_my):** Esta vista muestra los grupos afiliados al usuario que la solicita.
- **Perfil:** Vista que nos muestra y nos permite modificar nuestros datos de usuario.
- **Login: Formulario** de entrada a la aplicación.
- **Registration:** Formulario de registro a la aplicación.
- **Node event:** Nos muestra el contenido íntegro del nodo tipo evento.
- **Node group:** Nos muestra el contenido del grupo (descripción y ficheros adjuntos).
- **Node New:** Nos muestra el contenido completo de una noticia.
- **Tags:** Vista que muestra la lista de todos los tags introducidos en la aplicación.
- **Single Tag:** En esta vista podemos ver todo el contenido asociado a un tag en concreto.

6.4.6 RSS. Compatibilidad con iUPC y UPCdroid

Tal y como se ha especificado en los requerimientos del sistema, nuestra aplicación debe ofrecer suscripción para lectores RSS. Además se utilizará el feed del RSS como entrada de datos para la ejecución de las aplicaciones iUPC y UPCdroid. Por este motivo se incluyen estos dos conceptos en el mismo apartado. Estas aplicaciones accederán a una vista dedicada para ellas, en las que el feed de RSS que se mostrará contendrá el código XML que necesitan.

• *Que es el RSS?*

RSS es una familia de formatos de fuentes web codificados en XML. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos. El formato permite distribuir contenidos sin necesidad de un navegador, utilizando un software diseñado para leer estos contenidos RSS (agregador). A pesar de eso, es posible utilizar el mismo navegador para ver los contenidos RSS. Las últimas versiones de los principales navegadores permiten leer los RSS sin necesidad de software adicional. RSS es parte de la familia de los formatos XML desarrollado específicamente para todo tipo de sitios que se actualicen con frecuencia y por medio del cual se puede compartir la información y usarla en otros sitios web o programas. A esto se le conoce como redifusión web o sindicación web (una traducción incorrecta, pero de uso muy común).

• *Implementación RSS*

Drupal nos permite activar la generación de RSS para nuestra aplicación, pero no nos permite editar el código generado, por lo que solo se muestra la información básica de los nodos (titulo y body).

Para poder editar el contenido del feed del RSS hemos instalado el modulo “*Content Template*”, que nos permitirá la edición y personalización del código XML generado en el RSS. Se puede editar la estructura del RSS (template) para cada tipo de contenido. En este proyecto se requiere solamente que se ofrezca el RSS para los eventos del site, no para las noticias ni grupos, tanto para nuestra aplicación como para las aplicaciones iUPC i UPCdroid. Cabe destacar que el modulo citado permite también configurar el aspecto de visualización de los diferentes tipos de contenidos, lo que es decir, que nos permite editar más a fondo el aspecto de visualización de los nodos. De todas formas, en este proyecto solo se utilizará la funcionalidad para editar el código del RSS.

Una vez instalado y activado el modulo, nos dirigiremos a la sección *Content Template* del menú de administración para configurar el template del RSS para los eventos.

Al clicar en *Create Template* en el tipo *Evento*, nos aparecerán tres desplegables con espacios de texto para configurar diferentes tipos de visualización como se puede apreciar en la *Ilustración 8*, que en nuestro caso solo nos interesa el del RSS.

En este espacio de texto podremos programar en HTML el código que queremos que se muestre en el feed del RSS. En cambio, para generar el código con los tags específicos requeridos por las aplicaciones iUPC y UPCdroid e insertar en ellos la información necesaria utilizaremos el lenguaje PHP. Tanto el código HTML como el PHP se introducirán en el mismo campo de texto, y Drupal se encargará de detectarlos y diferenciarlos.

Template for Event

Enable the textareas by enabling the checkbox above each. Expand the variables section to display. Then click on content attributes to insert the appropriate PHP print statements at your cursor position in the textarea.

It is also possible to create disk-based templates. Find more information about that [here](#)

Please note that by creating a template for this content type, you are taking full control of its output and you will need to manually add all of the fields that you would like to see in the output. Click *reset* to remove template control for this content type.

[more help...]

→ Teaser

→ Body

▼ RSS

Affect RSS output
 Note that if you do not enable this, Drupal will use either the teaser or body as specified in your RSS publishing settings.

RSS Template:

```
<div><?php echo $node->field_speaker[0]['view'] ;</div>
<div><h4>Location:</h4></div>
<div><?php echo $node->field_room[0]['view'] ?></div>
<?php
$live1 = $node->field_live_stream_link[0]['view'];
$live2 = strpos("$live1", "http://");
if ($live2 == 12) {
echo "<div><h4>video Streaming:</h4></div>";
echo "<div>$live1</div>";
} else {
$z=$node->webfm_files[0]->p;
if ($z!="") echo "<div><h4>video Streaming:</h4></div>";
for ($i = 0; $i < 10; $i++) {
$a=$node->webfm_files[$i]->p;
$b=$node->webfm_files[$i]->n;
if ($b!="")$files[$i]="<div>http://hipeac.ac.upc.edu/seminars/$a/$b</div>";

```

Leave this field blank to leave RSS unaffected. To add XML elements create an array \$xml_elements.

→ RSS Variables

Reset Submit

Ilustración 8: Edición Event Template

• Código HTML

Utilizaremos los tags básicos del HTML para definir la estructura del nodo tipo Evento dentro del RSS. Los atributos de los eventos que queremos mostrar son: Título, Abstract, Start Date, Speaker, Location y enlaces a ficheros. Para obtener ésta información utilizaremos el desplegable “RSS Variables” que nos ofrece las llamadas PHP a los atributos del objeto evento.

Un ejemplo del código HTML para mostrar algunos de los atributos citados sería:

```

<div><h4>Title:</h4></div>
<div><?php echo $node->title ?></div>
<div><h4>Abstract:</h4></div>
<div><?php echo $node->content['body']['#value'] ?></div>
<div><h4>Start Date:</h4></div>
<div><?php echo $node->start_format ?></div>
<div><h4>Speaker:</h4></div>
<div><?php echo $node->field_speaker[0]['view'] ?></div>
<div><h4>Location:</h4></div>
<div><?php echo $node->field_room[0]['view'] ?></div>

```

Como se puede observar puede mezclarse el código HTML con el PHP de las llamadas a los atributos del evento, sin problemas de compatibilidad, ya que Drupal convierte a un String el contenido de la consulta, por lo tanto queda incrustado en el HTML.

A continuación podemos ver el aspecto visual del feed RSS generado por la aplicación. Se mostraran los últimos eventos publicados en el site, ordenados por fecha descendiente, del mas actual al más antiguo.

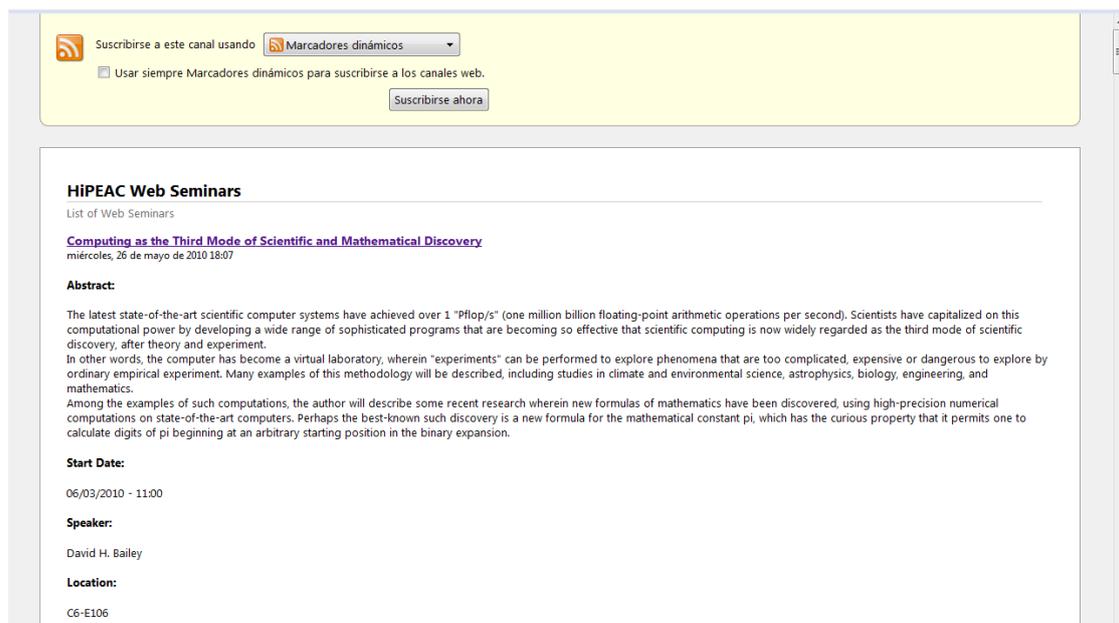


Ilustración 9: Aspecto visual del RSSfeed

• Código PHP

El código PHP lo utilizaremos para generar los tags XML que se mostraran en el código fuente del feed del RSS. Para poder generar estos tags se debe declarar, tal y como describe el modulo que estamos utilizando, un array “*xml_elements*”, que interpretará

los diferentes objetos del array como tags que se deben generar en el código fuente del feed.

El array se debe declarar de la siguiente forma:

```
$xml_elements = array(
  array(
    'key' => 'start',
    'value' => $node->start_format,
  ),
  array(
    'key' => 'speaker',
    'value' => $node->field_speaker[0]['view'],
  )
)
```

Con esta declaración conseguiríamos que se generen los tags start y speaker, con su correspondiente información en su interior:

```
<start>23/05/2010 13:00</start>
<speaker>Pol Martinez</speaker>
```

La estructura XML del feed la veremos solo cuando miremos el código fuente del propio feed. Los tags definidos para el código XML son:

- Título del evento: `<title>TituloEvento</title>`
- Link al evento: `<link>http://LinkEvento</link>`
- Fecha de inicio del evento: `<start>Dia- Hora</start>`
- Nombre del ponente: `<speaker>Peter Griffin</speaker>`
- Descripción del evento: `<body>Descripcion...</body>`
- Ubicación física del evento: `<location>C6-E101</location>`
- Attachments: `<attachfile>http://hipeac.ac.upc.../PowerPoint.pps</attachfile>`
- Link de streaming: `<streamlink>http://hipeac.ac.upc.../video.mp4</streamlink>`

Cada uno de los eventos estará determinado por los tags `<item>` y `</item>`. Una versión básica de la estructura general del código XML sería la siguiente:

```

<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0" xml:base="http://hipec.ac.upc.edu/seminars" xmlns:dc="http://purl.org/dc/elements/1.1/">
  <channel>
    <!-- Informacion General -->
    <title>HiPEAC Web Seminars</title>
    <link>http://hipec.ac.upc.edu/seminars/?q=eventslist</link>
    <description>List of Web Seminars</description>
    <language>en</language>
    <!-- Lista de items -->
    <item>
      <title>Computing as the Third Mode of Scientific and Mathematical Discovery</title>
      <link>http://hipec.ac.upc.edu/seminars/?q=node/93</link>
      <group domain="http://hipec.ac.upc.edu/seminars/?q=node/30" xmlns="http://drupal.org/project/og">DAC-UPC</group>
      <category domain="http://hipec.ac.upc.edu/seminars/?q=taxonomy/term/11">2010</category>
      <start>06/03/2010 - 11:00</start>
      <speaker>David H. Bailey</speaker>
      <body>The latest state-of-the-art scientific computer systems have achieved over 1...</body>
      <location>C6-E106</location>
      <streamlink>(No live stream link available)</streamlink>
      <attachfile>http://hipec.ac.upc.edu/seminars/files/uploads/seminaricap/HiPEAC-SallyMcKee-20091202.m4v</attachfile>
      <pubDate>Wed, 26 May 2010 18:07:40 +0200</pubDate>
    </item>
  </channel>
  <item>
    .
    .
    .
  </item>
</rss>

```

Como se puede apreciar, además de los tags definidos por nosotros, también aparecen algunos que inserta Drupal por defecto en cada item, estos son:

- Grupos afiliados al evento: `<group>DAC-UPC</group>`
- Tags identificativos del evento: `<category>2010</category>`
- Fecha de publicación del post: `<pubDate>Wed, 26 May 2010</pubDate>`

Una vez generado el código XML, dependerá de los desarrolladores de iUPC y UPCdroid el hecho de que se visualicen los seminarios en sus respectivas aplicaciones. La desarrolladora del proyecto iUPC nos ha cedido unos screenshots de su aplicación para que podamos ver como se muestran los eventos, a modo de comentario.

Se puede apreciar en la primera imagen (que se muestra a continuación) un menú principal de las funcionalidades de la aplicación. La que ofrece la visualización de los eventos y su contenido asociado es la función *Seminars*. Entrando en esta sección obtendremos el listado de eventos que se ha obtenido a partir del código XML, como se muestra en la segunda imagen. Por último, podemos ver el contenido del evento con los links de streaming y links, si estos existen.



Mediante la comunicación constante que se ha mantenido con la desarrolladora de iUPC hemos conseguido poder “integrar” las dos aplicaciones. No ha sido tarea fácil debido a la poca disponibilidad para organizar reuniones, pero finalmente hemos cumplido nuestro objetivo y por supuesto los requerimientos especificados en ambas aplicaciones.

6.4.7 Logs del sistema

Como se ha descrito anteriormente, Drupal incorpora un sistema de logs que genera el sistema, que concretamente en nuestra aplicación, solo pueden consultar los usuarios con el rol de *administrator*.

Para activar el sistema de logs debemos activar el módulo *Statiscs*, definir qué información queremos mostrar y qué roles podrán acceder a ella. Éste módulo nos generará los logs de carácter más general, como el numero de accesos totales, páginas más visitadas, accesos de los usuarios, y algunas más que se han definido en el apartado 4.1.7 *Administración*.

Para poder obtener una mayor profundidad en las estadísticas del site y logs más específicos se ha decidido crear una cuenta de Google Analytics¹¹ para el

¹¹ <http://www.google.com/intl/es/analytics/>

superadministrador del site, el cual podrá consultar toda la información que genera ésta información.

Se ha instalado el modulo *Google Analytics*, que nos incrustará el código necesario, para que Google Analytics pueda registrar toda la actividad del site, en todas las páginas de la aplicación.

Si se pretende que otros usuarios (por ejemplo los que tienen el rol de *adminisitrator*) puedan acceder a este sistema de logs, el superadministrador de la aplicación deberá entrar en la aplicación de Google Analytics y darles de alta des de allí, agregándolos como otros usuarios de la cuenta. Para más información del proceso de agregación de usuarios a la cuenta de Google Analytics consultar el siguiente enlace:

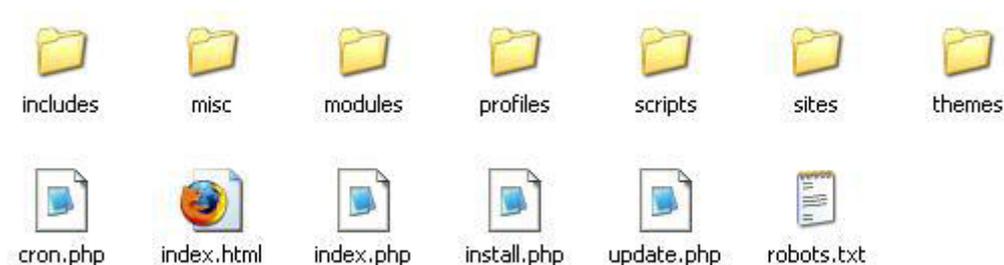
- <http://www.google.com/support/googleanalytics/bin/answer.py?hl=es&answer=55500>

6.5 Gestión de ficheros

En este apartado se definirá la estructura de directorios de la instalación de Drupal, así como la las directrices para gestionar los ficheros.

6.5.1 sistema de ficheros

En la figura siguiente podemos ver el directorio principal de la instalación básica de Drupal:



A continuación se describe de una forma general el contenido de los directorios:

- La carpeta *includes* contiene librerías de funciones comunes usadas por Drupal.
- En la carpeta *misc* hay ficheros JavaScript e iconos usados en el site, resultantes de una instalación por defecto.

- En la carpeta *modules* se encuentran los módulos que forman el núcleo de Drupal. Todos los módulos que añadamos posteriormente deberemos introducirlos en este directorio.
- La carpeta *profiles* contiene diferentes perfiles de instalación de Drupal.
- En la carpeta *scripts* hay Scripts en Shell y Perl que son utilizados para revisar la sintaxis, limpiar el código e interactuar con el cron.
- En la carpeta *sites* se guardan las modificaciones hechas sobre la web por defecto.
- La carpeta *themes* contiene ficheros de los temas Drupal predeterminados. Si instalamos nuevos temas deberemos copiarlos dentro de este directorio.
- El fichero *cron.php* realiza procesos de mantenimiento y actualización del framework y de los módulos añadidos, hace falta programar la ejecución periódica de este fichero según las necesidades de la web.
- El fichero *index.php* es la página principal del site.
- El fichero *install.php* se puede ejecutar manualmente en la instalación inicial del framework. Muestra un asistente de configuración de la web y de la base de datos.
- El fichero *update.php* se encarga de realizar la actualización de la base de datos, cuando actualizamos la versión de Drupal.
- El fichero *robots.txt* controla el indexado de contenidos por parte de robots rastreadores evitando el acceso a determinadas partes de la web.

Quizás en versiones superiores de Drupal existen pequeñas variaciones en la estructura de directorios.

6.5.2 Upload de ficheros

Drupal dispone de un módulo específico para gestionar el upload de ficheros, que nos permite subir ficheros mediante http al directorio especificado en la configuración del mismo. Los ficheros subidos tendrán una limitación de tamaño, ya que la configuración del servicio PHP del servidor prohíbe realizar un upload de un fichero superior a 2Mb.

Una de las necesidades de los usuarios de la aplicación que debemos satisfacer es permitir al *organizer* o al *adminisitrator* añadir contenido adjunto (ficheros) a los

eventos con un tamaño superior a 2Mb. Este problema lo podemos resolver fácilmente dando acceso por ftp al usuario, para que suba los ficheros al servidor directamente de una forma fácil, y por supuesto, más rápida.

Si queremos dar algún tipo de restricción de acceso al fichero subido, deberemos primero añadirlo o registrarlo a la base de datos, y posteriormente asociarlo al nodo que queramos (evento o grupo), para que los ficheros adquieran la restricción de acceso que está definida para el nodo. De esa forma solo los usuarios con acceso al grupo o evento podrán acceder a los ficheros.

Para realizar las acciones descritas utilizaremos el modulo WebFM, que nos permitirá realizar todo el procedimiento de una forma sencilla y rápida. A continuación se describe el funcionamiento y uso del modulo.

6.5.3 Gestor WebFileManager (WebFM)

El módulo WebFM presenta un cambio de paradigma en la gestión de archivos para Drupal. Este administrador de archivos se basa en una estructura jerárquica de directorios a diferencia del tradicional sistema de estructuración de archivos utilizando por defecto en Drupal.

WebFM utiliza AJAX para permitir a los usuarios organizar archivos en el servidor de la misma forma que lo hacen en sus sistemas personales. Esta habilidad para organizar jerárquicamente los archivos mejora la capacidad de gestión de grandes colecciones de ficheros.

• Configuración

Debemos instalar este modulo de la misma forma que lo hemos hecho con los demás. Una vez activado deberemos realizar las configuraciones pertinentes, que consistirán en especificar qué funcionalidades del modulo se le atribuye a cada uno de los roles que interactuaran con la aplicación.

También deberá definirse el directorio raíz de todos los ficheros subidos, ya que posteriormente podremos crear nuestros propios directorios.

• Funcionamiento

En el formulario de creación del contenido nos aparece un campo llamado *WebFM Attachments* con una interficie que nos mostrará el contenido del directorio root que hemos definido en la configuración del modulo.

La interficie nos permite crear subdirectorios respecto al directorio root, para poder clasificar el contenido bajo nuestro criterio. Estos directorios también pueden ser creados directamente en el servidor.

En primer lugar debemos subir al servidor (por debajo del directorio *root* definido en el modulo) por ftp los ficheros que queremos asociar a los nodos que vamos a crear. Una vez subidos los ficheros procederemos a la creación del contenido.

Como podemos ver en la siguiente imagen, nos aparecerá la estructura de directorios en la parte izquierda, y la lista de ficheros subidos al servidor, en la parte derecha.

The screenshot shows the 'WebFM Attachments' interface. At the top, there is a section for 'Attached Files' with a table containing one entry: 'Art_Science_SloMo_GPU_720p.mp4' (19 MB, modified 22/04/10 15:49, owner seminaricap). Below this is a 'File Browser' section. On the left, a tree view shows the directory structure: 'uploads' (root), 'uploads', 'pol', and 'seminaricap'. The main area shows the contents of the 'seminaricap' directory, listing several files with their names, modification dates, sizes, and owners.

Name	Modified	Size	Owner
HiPEAC-2005-AndreasMoshovos-RegionScout_Exploiting_Coarse-Grain_Sharing_to_Improve_Power_and_Bandwidth_in_Snoop_Coherence.pdf	28/04/10 11:57	270 KB	hpcadmin
HiPEAC-AlexSettle-2004-Architectural_Support_for_Enhanced_SMT_Job_Scheduling.pdf	28/04/10 13:31	564 KB	hpcadmin
HiPEAC-AlexSettle-2004-Architectural_Support_for_Enhanced_SMT_Job_Scheduling.ppt	28/04/10 13:31	2 MB	hpcadmin
HiPEAC-AlexSettle-20041025.avi	27/04/10 20:53	36 MB	hpcadmin
HiPEAC-AlexSettle-20041025.m4v	29/04/10 16:14	50 MB	hpcadmin

Deberemos especificar qué ficheros queremos adjuntar al nodo que estamos creando. Primero añadiremos una entrada correspondiente al nombre y propietario del fichero en la BBDD. Lo haremos clicando con el botón derecho sobre el fichero, y nos aparecerá un menú contextual con diversas opciones, entre ellas la de *Add to Database*. Una vez añadido a la base de datos, volveremos a clicar con el botón

derecho y ya nos aparecerá la opción *Attach to Node*, con lo que conseguiremos que adopte los permisos de restricción de acceso del nodo.

Si el proceso se ha realizado con éxito, nos aparecerá el fichero en cuestión como “Attached Files” en el bloque superior.

Este proceso se puede realizar tantas veces como se desee, ya que no existe limitación de ficheros adjuntos para un nodo en concreto.

6.6 Recuperación

Para mantener la integridad de nuestra aplicación y de toda la información que se almacena en el servidor utilizaremos dos procedimientos muy comunes que nos permitirán restablecer nuestro site en caso de fallida del sistema, por ejemplo, por rotura o mal funcionamiento de un disco duro. A continuación, se definen los dos procesos.

6.6.1 Mirroring

Consiste en replicar el contenido de un servidor a otro. Diariamente se actualiza el servidor secundario (mirror) respecto al servidor de explotación, para tener una réplica lo más actualizada posible de todo el contenido del site.

En el caso de que el servidor principal tuviera una avería, solo deberíamos redireccionar las peticiones del servidor principal al secundario, disponiendo nuevamente de nuestro site. La información que se perdería en este caso sería la introducida entre la hora de sincronización de los dos servidores y la hora de la fallida del sistema.

El proceso de sincronización se hace mediante el servicio *rsync*¹², que nos permite definir, entre muchas otras opciones, el servidor que actuará como *mirror*, la frecuencia de sincronización, el contenido a sincronizar, y el momento en que debe iniciarse la sincronización.

¹² Definición Wiki: <http://es.wikipedia.org/wiki/Rsync>

Se ha creado un *shellscript*¹³ llamado “backup-seminars.sh” que realizará las llamadas a los servicios necesarios para realizar este proceso. El proceso a seguir es el siguiente:

- Sincronizar el contenido de los ficheros y directorios que forman la aplicación, por lo tanto, el directorio principal de la instalación de Drupal.
- Sincronizar la base de datos. Este procedimiento lo haremos mediante el servicio *mysqldump*¹⁴ que nos permite exportar la base de datos de la aplicación.

El shellscript sincronizará los directorios de la instalación de Drupal y todo el contenido de ambos servidores. Posteriormente se realizará un export de la base de datos del servidor principal y se importará en la base de datos del servidor secundario. De esta forma tendremos los servidores replicados cuando finalice de la ejecución del script. Es importante que la base de datos del servidor mirror tenga el mismo nombre que la que se encuentra en el servidor de explotación, sino nuestra aplicación replicada no funcionaria.

6.6.2 Backup

Para asegurar aun más la integridad de nuestra aplicación utilizaremos otro servidor dedicado al almacenamiento de un repositorio de backup’s de nuestra aplicación, que se irán generando diariamente. Este servidor estará situado en un espacio físico diferente al que se encuentran los servidores de explotación y mirroring. De esta forma, en caso de que se produjera un incendio que afectara a los otros dos servidores, podríamos recuperar toda la información.

En el mismo shellcript que se ha utilizado para realizar la sincronización entre el servidor de explotación y el secundario, se ha definido el proceso de creación del backup, que es el siguiente:

- Exportar la base de datos de la aplicación.
- Comprimir el directorio principal de la instalación de Drupal, excepto el directorio `/drupal/files`.

¹³ Definición Wiki: http://en.wikipedia.org/wiki/Shell_script

¹⁴ Definición Wiki: http://wiki.welmers.net/en/MySQL_dump

- Crear un tar que contendrá los ficheros resultantes de los dos pasos anteriores, con el nombre backup-seminars\$*date*, donde *\$date* será la fecha de creación.
- Mover el tar generado al servidor de backup's.
- Sincronizar el directorio /drupal/files del servidor de explotación con el de backups. En este caso simplemente se sincronizará el contenido del directorio, debido a que el volumen de ficheros es muy grande y son muy pesados, y sería incoherente copiar ficheros como videos, pdf's, presentaciones, etc, que no son nunca modificados.

Si tuviéramos que recuperar nuestro sistema mediante el backup, el procedimiento a seguir sería el siguiente:

- Preparar el servidor con los servicios necesarios para que nuestra aplicación pueda funcionar, especificados en el apartado *"6.2.1 Requerimientos del servidor"*.
- Crear una base de datos con el mismo nombre que tenía la base de datos del servidor de explotación.
- Importar en la base de datos creada el fichero de exportación generado en el backup.
- Copiar el contenido del directorio de instalación de Drupal (que está comprimido en el backup) al servidor destino.
- Copiar el contenido de los ficheros que se encuentran sincronizados en el servidor de backup al nuevo servidor, concretamente en /drupal/files.

Una vez realizado este procedimiento, ya dispondremos de nuestro site nuevamente en funcionamiento, sin problema alguno.

6.6.3 Esquema de red

A continuación se muestra el esquema de red resultante del sistema de recuperación aplicado a nuestro site.

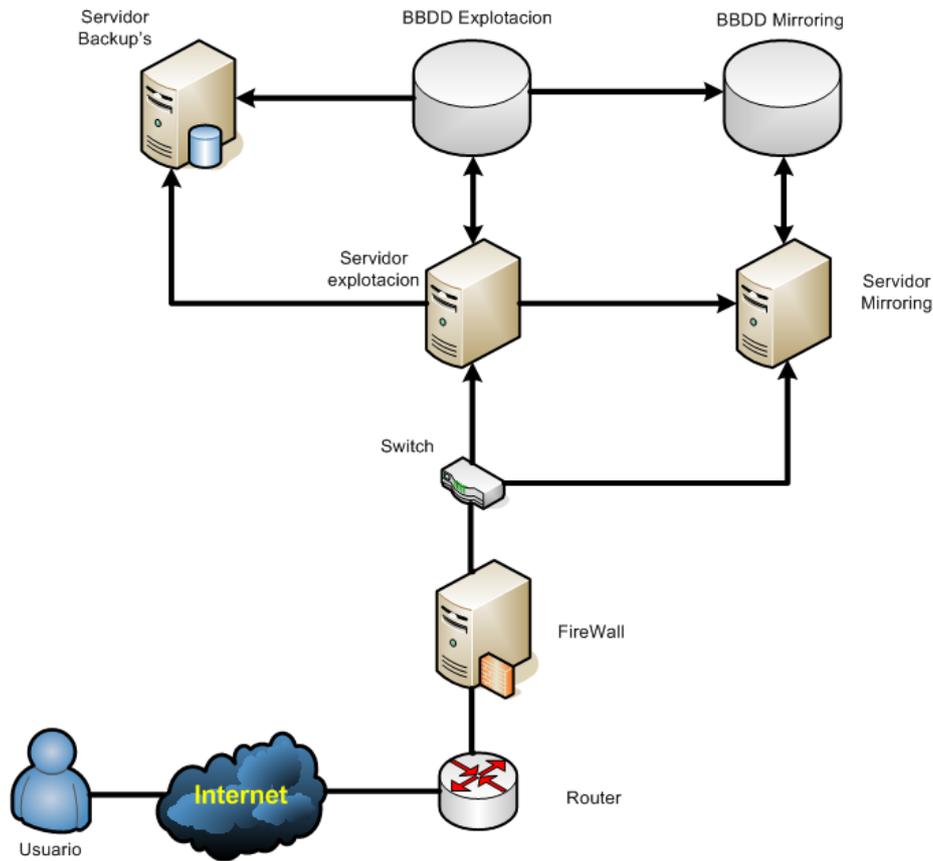


Ilustración 10: Esquema de red del servicio SGESE

Cuando un usuario accede a nuestra aplicación, será dirigido al servidor de explotación. Si este servidor hubiera caído o hubiera sufrido algún daño, las peticiones serían redirigidas al servidor secundario (mirror), de una forma totalmente transparente al usuario.

Como se puede observar, se guarda una réplica del servidor primario y de la BBDD en el servidor de backup's.

6.6.4 Cron

Para la ejecución periódica o diaria del shellscrip mencionado en los apartados anteriores utilizaremos el servicio *cron* del servidor. Cron es un administrador regular de procesos en segundo plano (daemon) que ejecuta procesos o guiones a intervalos regulares (por ejemplo, cada minuto, día, semana o mes). Los procesos que deben

ejecutarse y la hora en la que deben hacerlo se especifican en el fichero *crontab*¹⁵. En este fichero hemos especificado que se ejecute el script *backup-seminars.sh* diariamente a las doce de la noche.

Como alternativa para no editar el fichero *crontab*, podemos copiar nuestro script al directorio */etc/cron.daily*, lo que hará que se ejecute también todos los días.

¹⁵ Para editar el fichero *crontab* en el terminal: `$> crontab -e`

7. Actualización

Periódicamente salen actualizaciones de Drupal que mejoran la seguridad y corrigen errores de versiones anteriores. Por ese motivo es importante que el site se vaya actualizando cada cierto tiempo.

La actualización de Drupal debe realizarla el superadministrador del site, ya que es un proceso delicado. Será necesario realizar copias de seguridad y modificaciones en la base de datos, así que utilizaremos el gestor de BBDD phpMyAmdin, tal y como se ha descrito en apartados anteriores, para realizar este procedimiento.

En primer lugar debemos comprobar en la comunidad de Drupal que los módulos contribuidos (los que no vienen por defecto en la instalación básica de Drupal) estén actualizados y disponibles para la nueva versión a la que queremos actualizar. Si alguno de ellos aun no está disponible no se podrá actualizar el site.

El primer paso que debemos dar es realizar una copia de seguridad de la información almacenada en el servidor. Deberemos copiar todo el contenido existente en el directorio de instalación de Drupal en adelante y, posteriormente, realizar una exportación de todos los datos almacenados en la base de datos que utiliza nuestra aplicación. De esta forma nos aseguramos la restauración del site, en el caso de que el proceso de actualización fallara en algún momento.

El segundo paso es cambiar el estado del site a modo “fuera de línea” o “modo mantenimiento”. Para ello, entraremos en la aplicación como superadministrador y en la sección mantenimiento del sitio, dentro de configuración del sitio, activaremos la opción *off-line*, como se observa en la siguiente imagen.

Site maintenance

Site status:

- Online
 Off-line

When set to "Online", all visitors will be able to browse your site normally. When set to "Off-line", only users with the "administer site configuration" permission will be able to access your site to perform maintenance; all other visitors will see the site off-line message configured below. Authorized users can log in during "Off-line" mode directly via the user login page.

Site off-line message:

HiPEAC Web Seminars is currently under maintenance. We should be back shortly. Thank you for your patience.

Message to show visitors when the site is in off-line mode.

Ilustración 11: Mantenimiento del Site

En el mismo formulario podremos configurar el mensaje que verán los usuarios cuando accedan a la web. Cuando guardemos la configuración el site pasará a estar fuera de línea.

Se recomienda modificar el tema de visualización del site a uno por defecto (Garland or Bluemarine) para reducir al máximo el riesgo de no compatibilidad de la nueva versión. También deberemos desactivar los módulos contribuidos durante la instalación, debido a que substituiremos los existentes por los actualizados.

Una vez que el site está fuera de línea, configuraremos los permisos de escritura del archivo settings.php que se encuentra dentro del directorio Drupal/sites/default. Se le otorgaran solo permisos de lectura tanto a la carpeta default como al archivo settings.php. Posteriormente deberemos editar el archivo settings.php, donde se encuentran los parámetros generales de configuración del site, y cambiar la línea 106 aproximadamente:

Cambiaremos la variable

- `$update_free_access = FALSE;`

por:

- `$update_free_access = TRUE;`

Deben realizarse estos pasos para poder acceder y ejecutar el script update.php para actualizar el site.

El siguiente paso es reemplazar la instalación de Drupal. Nos descargaremos la versión que queramos instalar y los módulos contribuidos actualizados. Copiaremos todos los ficheros al servidor reemplazando la versión anterior por la nueva.

Posteriormente ejecutaremos el script `update.php` con el navegador web, mediante la siguiente URL:

- <http://hipeac.ac.upc.edu/seminars/update.php>

Se debe tener en cuenta que si simplemente se reemplazan los archivos de una versión a otra y no se realiza todo el procedimiento descrito anteriormente, cuando se intente acceder de nuevo al sitio aparecerá el error:

- "Fatal error: Call to undefined function `user_uid_optional_to_arg()` in `includes/menu.inc` on line 594"

Entonces no se podrá cambiar el sitio web del modo fuera de línea hasta que no se consiga ejecutar el script `update.php` correctamente.

Si el script `update.php` se ha ejecutado con éxito se muestra una página de actualización de la base de datos y ya podrá iniciarse el proceso de actualización del código fuente.

La versión de Drupal que se está actualizando se detecta automáticamente y después de pulsar el botón `update`, el proceso de instalación se inicia. Si se ha actualizado el site sin muestra de errores significará que se habrá realizado con éxito y ya será posible acceder a la página de administración de nuevo.

Una vez terminado el proceso, debemos restablecer el valor de la variable `$update_free_access` del archivo `settings.php`, y restaurar los permisos de escritura del directorio `Drupal/sites/default` y del archivo `settings.php`, que se encuentra dentro de éste.

Ahora debe activarse el sitio y quitar el modo fuera de línea mediante la aplicación de administración del superusuario.

A partir de este momento el sitio web se encontrará nuevamente activo y actualizado.

8. Planificación

En este apartado se presentan las planificaciones inicial y final de forma conjunta, aunque se han realizado por separado, una al inicio del proyecto i la otra al final, para poder contrastar las diferencias entre ambas.

Pese a hacer la planificación inicial, es difícil seguir-la al pie de la letra y siempre pueden surgir contratiempos que alarguen determinadas fases más de lo que se esperaba. Por eso no se ha podido cumplir uno de los objetivos secundarios que se había planteado al inicio del proyecto: la elección del *WebConferencing*.

Esta funcionalidad no se ha llegado a implementar, debido a que el desfase en la planificación nos ha obligado a desestimarla.

Se pretendía hacer un estudio de los diferentes servicios que ofrecían las empresas de videostreaming y online conferencing. Se quería utilizar este servicio para poder retransmitir los eventos en directo, mediante un canal de retransmisión que se definiría en el campo *VideoStreaming* del evento.

La elección de este servicio podría realizarse por los administradores del site, o bien, por algún proyectista que decida realizar ampliaciones en las funcionalidades de la aplicación en un futuro.

A continuación se muestran los diagramas de Gaant de la planificación inicial y la final:

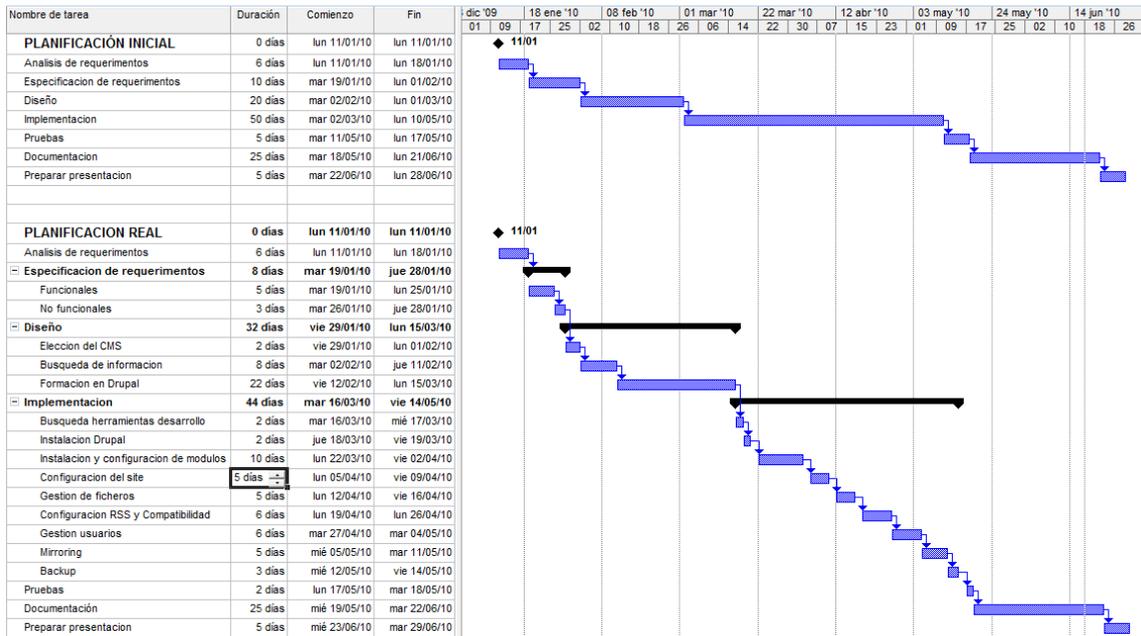


Ilustración 12: Diagramas Planificación

Como se puede apreciar en los diagramas, se ha destinado más tiempo al aprendizaje del sistema de funcionamiento de Drupal, por lo que se ha asignado menos tiempo del que se esperaba para realizar tareas de implementación. Este desfase ha supuesto reducir alguna funcionalidad y reducir los días de pruebas.

La duración del proceso de creación del proyecto ha sido de unos 120 días (sin tener en cuenta los fines de semana), asignando una media entre 4 y 5 horas diarias de dedicación.

9. Estudio económico

Antes de empezar a realizar el estudio económico del proyecto debemos definir y estructurar el procedimiento de realización de este proyecto. Para ello se han definido diferentes perfiles profesionales que han desarrollado las diferentes tareas especificadas. Posteriormente se hará el balance del trabajo realizado, desde una vertiente económica.

A continuación definimos los diferentes perfiles profesionales que intervienen en el proyecto. Éstos son:

- *Project Manager*: Se encarga de la supervisión de todas las tareas, participa en las reuniones con el cliente i da el visto bueno a los documentos que se le entregan al cliente. A grandes rasgos realiza la gestión del proyecto.
- *Analista*: Se reúne con el cliente para conocer sus necesidades, elabora la documentación sobre requerimientos, especificación y diseño del proyecto.
- *Programador*: Se encarga de transformar las especificaciones del proyecto al producto final que se le entregará al cliente.

Las tareas a desarrollar son las descritas a continuación. Todas incluyen una fase de documentación más o menos importante.

- *Análisis de requerimientos*: Comprende las entrevistas con el director del proyecto y la elaboración del documento que recoge las características y funcionalidades que tendrá la aplicación, y que una vez aceptado por el director servirá de guion de trabajo.
- *Especificación*: Consiste en establecer de forma detallada todas las características y funcionalidades que tendrá nuestra aplicación.
- *Diseño*: Consiste en definir la estructura que tendrá el entorno de la aplicación, realizar la especificación de todas las partes y decidir la tecnología que se usará en el desarrollo.
- *Implementación*: Consiste en crear el entorno virtual siguiendo las especificaciones utilizando la tecnología escogida en la fase de diseño.

- *Gestión del proyecto*: Seguimiento de la evolución del proyecto en las diferentes fases, reuniones con el cliente y revisión del documento con las especificaciones básicas y finales. Realización de tareas de gestión (cálculo del coste del proyecto, diagrama de Gantt, elaboración de la memoria, etc).
- *Pruebas*: Consiste en realizar pruebas de funcionamiento y rendimiento de la aplicación una vez terminada la implementación.

Para analizar el coste económico del proyecto debemos tener en cuenta tanto los recursos humanos utilizados como las licencias de software y hardware necesarios para la implementación y explotación de la aplicación.

9.1 Hardware

Respecto a los recursos físicos, se ha necesitado un ordenador personal para la implementación de la aplicación, un servidor en explotación para poder hospedarla, uno dedicado al mirroring, y otro a los backup's.

Podríamos definir como el coste hardware el desgaste del ordenador, aunque no se ha tenido en cuenta ya que es significativamente pequeño respecto al coste total del proyecto.

Como servidor de explotación, mirroring y backup's se han utilizado unos del DAC (Departament d'Arquitectura de Computadors) dedicados para ello, que ya estaban en funcionamiento, hospedando en la actualidad diferentes aplicaciones. Tampoco se tiene en cuenta el coste del mantenimiento el servidor de explotación, por lo que podríamos definir el coste hardware igual a cero.

9.2 Software

Para el desarrollo del proyecto se ha tenido en cuenta utilizar sistemas operativos y aplicaciones/servicios de libre distribución para abaratar al máximo el coste total del proyecto. En el caso de los sistemas operativos hemos utilizado Linux/Solaris, y en el caso de los servicios y aplicaciones para desarrollar e implementar la aplicación hemos utilizado Apache, MySQL, phpMyAdmin, Xampp y Drupal.

Todos los elementos citados tienen coste cero, por lo que reducimos en gran parte, el coste total del proyecto.

9.3 Recursos humanos

Se ha elaborado una tabla con las horas que se le ha atribuido a cada una de las tareas anteriormente descritas, con el rol/perfil profesional que las desarrolla:

Tarea	Rol	Horas	Precio/hora (€)	Coste (€)
Análisis	Analista	30	35	1050
Especificación	Analista	40	35	1400
Diseño	Analista	160	35	5600
Implementación	Programador	250	20	5000
Gestión	Project Manager	40	60	2400
Pruebas	Programador	10	20	200
Total		530		15650

Ilustración 13: Tabla de costes

Como podemos observar en la tabla, el coste total de los recursos humanos utilizados en el proyecto será de 15650€, teniendo en cuenta que se han realizando 530 horas de trabajo, repartidas en 120 días.

Como conclusión podemos afirmar que el coste total del proyecto correspondería al de los recursos humanos, ya que tanto el hardware (que se ha determinado finalmente igual a cero) como el software utilizado no han tenido coste alguno.

10. Conclusiones

Bajo mi punto de vista, Drupal aporta muchas ventajas gracias a que simplifica y automatiza la tarea de los usuarios, tanto de los desarrolladores/administradores como de los usuarios finales.

Respecto al desarrollo, se puede conseguir poner en marcha una aplicación web cómoda y rápidamente, la cual es fácil de extender a partir de módulos y dispone de un elevado grado de escalabilidad, es decir, capaz de cambiar la apariencia, la configuración y los usuarios de la aplicación.

Respecto a la administración, dispone de una sección que permite gestionar fácilmente y de forma centralizada el contenido, apariencia, configuración del site y gestión de usuarios.

En cuanto a los usuarios finales, pueden publicar y editar rápidamente los contenidos sin necesidad de tener conocimientos de programación (como por ejemplo HTML, PHP, etc.) y, gracias a su simplicidad, pueden asumir algunas tareas de administración (por ejemplo, las relacionadas con los contenidos).

De igual manera, los gestores de contenidos (no sólo Drupal, sino de forma general) también presentan ciertos inconvenientes.

Como ya se ha comentado, requiere un proceso de aprendizaje y documentación que puede ser relativamente lento. Hay que tener en cuenta, que este tiempo de aprendizaje puede ser menos que el tiempo que costaría poner en marcha una aplicación programada a medida. Así mismo, es necesario entender el aprendizaje como una inversión, ya que no será necesario (al menos en el mismo grado) de cara al desarrollo de nuevas aplicaciones en la misma plataforma.

Generalmente, también, los CMS pueden presentar ciertas vulnerabilidades (más o menos conocidas), que pueden ser objetivo de ataques para conseguir información o entorpecer el funcionamiento habitual de la aplicación. En principio, cuantos más usuarios trabajen en una aplicación más fácil es descubrirlas, y recae sobre la comunidad de desarrolladores garantizar la resolución de estos problemas.

Por otra parte, creo que se consigue una menor personalización de la aplicación, de manera que una aplicación programada a medida se puede optimizar un poco más, aunque supone mucho más esfuerzo y trabajo.

En este sentido, también la base de datos de un gestor de contenidos separa los datos en diversas tablas de acuerdo con su estructura y funcionamiento internos, cosa que condiciona una menor adecuación del modelo relacional al modelo lógico y dificulta la portabilidad a otras CMS o SGBD.

Además de los puntos comentados anteriormente, existe una dependencia de la tecnología en cuanto a extensibilidad, ya que depende, en gran medida, de la comunidad de desarrolladores, la cual ha de intentar cubrir las necesidades de los usuarios, mejorar las funcionalidades y crear nuevas. De todas formas, siempre podríamos desarrollar un módulo hecho a medida que nos ofreciera las funcionalidades que requerimos.

11. Bibliografía

En este apartado se citaran todas las fuentes bibliográficas que se han utilizado como referencia para llevar a cabo la realización de este proyecto. Los enlaces que se definen a continuación han estado activos durante el periodo Enero-Junio del 2010.

- [1] **Wikipedia**, <http://en.wikipedia.org>
- [2] **Comunidad Drupal**, <http://drupal.org/>
- [3] **Druapl Hispano**, <http://drupal.org.es/>
- [4] **Drupal USA**, <http://drupal.com/>
- [5] **DrupalMania**, <http://drupalmania.com/>
- [6] **DrupalCamp**, <http://www.drupalcamp.es/>
- [7] **Drupal en català**, <http://drupal.cat/>
- [8] **Grupo de Usuarios de Drupal 5 de habla hispana**, <http://groups.drupal.org/spain>
- [9] **Bitnami**, <http://bitnami.org/stack/drupal>
- [10] **OpensourceCMS**,
<http://php.opensourcecms.com/scripts/details.php?scriptid=191>
- [11] **Cocina con fundamento para Drupal**, <http://www.cocinandocondrupal.net/>
- [12] **Maestros del Web**, <http://www.maestrosdelweb.com/editorial/dries-buytaert-creador-de-drupal/>
- [13] **Ecosistema Drupal**, <http://ecosistema-drupal.org/>
- [14] **Desarrollo de Drupal Neurotics**, <http://www.neurotic.es/>
- [15] **Identi.ca by StatusNet**, <http://identi.ca/group/drupal>
- [16] **Drupal Mexico**, <http://drupalmexico.com/>
- [17] **Acquia**, <http://acquia.com/>

- [18] **Genbeta**, <http://www.genbeta.com/web/>
- [19] **Cambrico**, <http://cambrico.net/>
- [20] **Observatorio Tecnológico**,
<http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=547>
- [21] **Reyero**, <http://reyero.net/>
- [22] **Alzado**, http://www.alzado.org/articulo.php?id_art=849
- [23] **Error500 (Tecnología + Internet + Conocimiento)**,
<http://www.error500.net/software/open-atrium-intranets-basadas-en-drupal>
- [24] **CuencoDigital, utilizando Drupal para desarrolladores web**,
<http://cuencodigital.com/>
- [25] **Carmel Hassan**, <http://carmelhassan.es/es/desarrollando-sitios-webs-eficaces-con-drupal>
- [26] **Ayuda WordPress**, <http://ayudawordpress.com/drupal-vs-wordpress/>
- [27] **MediaLab Prado**, http://medialab-prado.es/article/drupal_jugar_con_vistas
- [28] **The INQUIRER - buscándole las cosquillas a los que mueven los hilos**,
<http://www.theinquirer.es/2009/10/26/el-portal-web-de-la-casa-blanca-migra-a-drupal.html>
- [29] **EducaSitio para principiantes**, <http://www.educasitio.com/curso-basico-drupal>
- [30] **DrupalSoul-Soluciones integrales en Drupal**, <http://www.drupalsoul.com/>
- [31] **Blog de Carlos Rincon**, <http://carlos.rinconsanchez.com/drupal-vs-joomla-una-comparativa-sincera-de-un-consultor-de-ibm>
- [32] **VSeo**, <http://www.vseo.es/buscadores/seo-drupal/>
- [33] **Barrapunto**, <http://softlibre.barrapunto.com/article.pl?sid=09/10/26/0853250>

- [34] **MIT OpenCourseWare**, <http://ocw.mit.edu/OcwWeb/HowTo/>
- [35] **Drupal hooks**, <http://api.drupal.org/api/group/hooks/>
- [36] **Linux Site**, <http://www.linux.com/articles/61713>
- [37] **Drupal hosting directory**, <http://drupalhosting.net/>
- [38] **Xampp WebSite**, <http://www.apachefriends.org/es/xampp.html>
- [39] **Comparativas de seguridad**: <http://www.securitybydefault.com>
- [41] **PHP**, <http://www.php.net/>
- [42] **WebEstilo - Usabilidad, programación, y mucho mas**,
<http://www.webestilo.com/php/>
- [43] **PHP a fondo**, <http://www.desarrolloweb.com/php/>
- [44] **PhpWebQuest**, <http://www.phpwebquest.org/>
- [45] **Geneura**, <http://geneura.ugr.es/~maribel/php/>
- [46] **Programacion en castellano**, <http://www.programacion.com/php/>
- [47] **Programacion PHP**, <http://www.programacionphp.net/>
- [48] **W3Schools**, <http://www.w3schools.com>
- [49] **PhpMyAdmin**, <http://www.phpmyadmin.net/>
- [50] **MySQL Official Site**, <http://www.mysql.com/>
- [51] **MySQL development**, <http://dev.mysql.com/>
- [52] **CristalLab**, <http://www.cristalab.com/tips/mysql-5.1-descarga-mysql-instala-y-crea-una-base-de-datos-c76897l/>
- [53] **The Apache Software Foundation**, <http://www.apache.org/>
- [54] **Wamp Server - Apache, PHP, MySQL on Windows**,
<http://www.wampserver.com/en/>

[55] **The Complete Apache Resource**, <http://www.apache.com/>

[56] **AppServ Open Project**, <http://www.appservnetwork.com/>