

**Universidad CENTRAL “Marta Abreu” de Las Villas
Facultad: Matemática, Física y Computación
Carrera: Licenciatura en Ciencia de la Computación**

TRABAJO DE DIPLOMA



SISTEMA DE MANEJO DE CONTENIDOS

SITIO WEB DE LA FACULTAD MATEMÁTICA FÍSICA Y COMPUTACIÓN

**Autor: Yoandry Pacheco Águila
Tutor: Ing. Leonardo. Flavio Del Toro Melgarejo
Julio 2005 Año de la Alternativa Bolivariana para las Américas**

Dirección: Ing. Leonardo Flavio Del Toro Melgarejo
Dpto.: Producción de software
Facultad Matemática Física y Computación
Universidad Central Marta Abreu de la Villas

SISTEMA DE MANEJO DE CONTENIDOS

SITIO DE LA FACULTAD MATEMÁTICA FÍSICA Y COMPUTACIÓN

Autor: Yoandry Pacheco Águila
Título: Sistema de Manejo de Contenido
Carrera: Licenciatura en Ciencia de la Computación
Curso: 2004-2005
Julio 2005

Hago constar que el presente trabajo fue realizado en la Universidad Central Marta Abreu de las Villas como parte de la culminación de los estudios de la especialidad de Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

Firma del Autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del Tutor

Firma de Jefe de
Seminario

Firma del Responsable de
Información Científico-Técnica

Yo te bendigo, vida porque nunca me diste ni esperanza fallida ni trabajos injustos, ni pena inmerecida porque veo al final de mi rudo camino que yo he sido el arquitecto de mi propio destino.

Amado Nervo (Bendiciones)

- A mis hermanos Juan Carlos, Yosdelbis y Yaniester.
- A mis padres por su esmerado apoyo.
- A mis amigos.

AGRADECIMIENTOS	I
RESUMEN.....	II
ABSTRACT	III
INTRODUCCIÓN.....	4
Planteamiento del problema	4
Objetivos	5
Hipótesis	6
Contenido del trabajo	6
CAPÍTULO 1: LOS SISTEMAS DE MANEJO DE CONTENIDOS	7
1.1 CONCEPTOS Y DEFINICIONES FUNDAMENTALES.....	7
1.1.1 ¿Qué es un CMS?	7
1.1.2 Creación de contenido.	7
1.1.3 Gestión de contenido.	8
1.1.4 Publicación.	8
1.1.5 Presentación.....	8
1.2 NECESIDAD DE UN CMS.	9
1.3 CMS COMERCIALES Y DE CÓDIGO ABIERTO.....	10
1.4 HISTORIA DE LOS CMS.	11
1.5 PRESENTE Y FUTURO DE LOS CMS.	12
1.6 LOS CMS EN EL E-LEARNING.....	12
1.7 CRITERIOS DE SELECCIÓN.....	13
1.8 LENGUAJE DE PROGRAMACIÓN PHP.....	15
1.8.1 ¿Qué es PHP?	15
1.8.2 Usos de PHP.....	16
1.8.3 ¿Por qué usar PHP y no otro lenguaje de programación?	16
1.8.4 Seguridad en PHP.....	20
1.9 MySQL.....	21
1.9.1 Características (versión 4.0 en adelante)	21
1.9.2 MySQL en cifras	22
1.9.3 ¿Qué licencia utilizar?	22
1.9.4 Ventajas y desventajas.....	22
1.9.5 ¿Por qué MySQL?	23
1.10 AUTENTIFICACIÓN SEGURA EN SITIOS WEB. (ALGORITMO OPT).....	23
1.10.1 Escenario.	23
1.10.2 Encriptación de contraseñas.	24
1.10.3 Contraseñas de un solo uso (OTP).	25

1.10.4	Implementación: PHP con MD5.....	26
CAPÍTULO 2:	DISEÑO DEL SISTEMA Y LOS SITIOS WEB.....	28
2.1	DISEÑO DEL SISTEMA DE MANEJO DE CONTENIDO (CMS).	28
2.1.1	Diagrama de Navegación.....	28
2.1.2	Usuarios y Privilegios.....	35
2.1.3	Seguridad del sistema.	36
2.1.4	Arquitectura del sistema	36
2.1.5	Actores y casos de uso del sistema	37
2.1.6	Diseño de Bases de Datos.....	37
2.2	DISEÑO DEL SITIO WEB PRINCIPAL DE LA FACULTAD	39
2.3	DISEÑO DEL SITIO WEB PARA UN DEPARTAMENTO	39
2.4	DISEÑO DEL SITIO WEB PARA UNA CARRERA	40
2.5	DISEÑO DEL SITIO WEB PARA UNA DISCIPLINA	40
2.6	DISEÑO DEL SITIO WEB PARA UNA ASIGNATURA.....	41
CAPÍTULO 3:	GUÍA DEL NAVEGADOR.....	42
3.1	INICIAR SESIÓN.	42
3.2	REGISTRARSE.	42
3.3	CERRAR SESIÓN.....	42
3.4	ADMINISTRACIÓN.	43
CONCLUSIONES.....		46
RECOMENDACIONES.....		47
ANEXO 1: PHP.....		48
ANEXO 2: ARQUITECTURA DEL CMS.		50
ANEXO 3: CASOS DE USO DEL SISTEMA.		54
ANEXO 4: DISEÑO DE LA BASE DE DATOS.....		55
ANEXO 5: DESCRIPCIÓN DETALLADA DE LAS DE LAS TABLAS.....		57
ANEXO 6: GUÍA DEL NAVEGADOR.		67
GLOSARIO		74
REFERENCIAS Y BIBLIOGRAFÍAS		76

Agradecimientos

- A la revolución que me ha dado la posibilidad de cursar una carrera universitaria.
- A los profesores que sabido trasmitirme parte de sus conocimientos y compartido sus experiencias.
- A todos mis compañeros de la universidad y especialmente a los de mi año, con los cuales he compartido los buenos y malos momentos.
- A todos los que de una forma u otra han contribuido a mi formación.

Resumen

Los documentos digitales constituyen hoy en día la fuente bibliográfica y de información principal de cualquier centro de la educación superior. Por esa razón la facultad de MFC ha mantenido un sitio Web para el acceso y manejo de los mismos. Sitio Web que hoy por algunas razones no satisface todas las necesidades del centro y no cumple con algunos requisitos trazados por la universidad. Por tanto fue necesario crear un nuevo sitio Web para dar solución a estos problemas. Para lo cual se comenzó el estudio de los sistemas de manejo de contenidos, que son softwares que se utilizan principalmente para facilitar la gestión de sitios Web. También se estudia las ventajas del PHP y el MySQL como posibles candidatos en el desarrollo de un sistema de manejo de contenido y algunos aspectos que intervienen en el fortalecimiento de la seguridad del sistema.

Abstract

Nowadays the digital documents constitute the main bibliographical and information source of any center of superior education. For that reason the MFC faculty has maintained a Web site for the access and handling of such. Web site that today for some reasons does not satisfy all the necessities with the center and does not fulfill some requirements drawn up by the university. Therefore it was necessary to create a new Web site to give solution to these problems. For which the study of the Content Management Systems was begun, that constitutes software's that are used mainly to facilitate the management of Web sites. Also one studies the advantages of the PHP and the MySQL as possible candidates in the development of a Content Management Systems and some aspects that take part in the fortification of the security of the system.

Introducción

Realizar un Web puede ser un trabajo complicado y muy laborioso si no se dispone de las herramientas adecuadas. En los últimos años se ha desarrollado el concepto de sistema de manejo de contenidos (Content Management Systems o CMS¹). Se trata de herramientas que permiten crear y mantener un Web con facilidad, encargándose de los trabajos más tediosos que hasta ahora ocupaban el tiempo de los administradores de los sitios Web.

Los gestores de contenidos proporcionan un entorno que posibilita la actualización, mantenimiento y ampliación de la Web con la colaboración de múltiples usuarios. En cualquier entorno virtual ésta es una característica importante, que además puede ayudar a crear una comunidad cohesionada que participe más de forma conjunta.

En este artículo se describen los criterios más importantes a la hora de seleccionar o crear un gestor de contenidos y los requerimientos en función de los objetivos que se quieran alcanzar. Por eso, se hace un breve repaso de las herramientas de código abierto que permiten construir sistemas gestores de contenido generales.

Planteamiento del problema

Con el objetivo de mejorar la documentación la Facultad de MFC ha mantenido durante algunos años el sitio Web “www.mfc.uclv.edu.cu” en el cual publica información referente a los departamentos, carreras, cursos de postgrado, maestrías, disciplinas y gran parte de la documentación correspondiente a cada asignatura del curso de pregrado.

A pesar de que dicho sistema, facilitó en gran medida el manejo de contenido, hoy no responde a las necesidades del centro y no cumple todos los requisitos que debe tener el sitio Web de una facultad según acuerdo de la Universidad, debido a:

- 1) Está implementado en más de un lenguaje de programación (asp, php, javascript y vbscript).
- 2) Cada disciplina y asignatura en particular, a pesar de la requerida homogeneidad en el diseño y estructura, es un directorio independiente, lo cual implica la existencia de mucho código redundante.
- 3) Requiere de servicios como, Internet Information Server y SQL-Server, además de una maquina virtual de Java.

¹ En adelante se usará la sigla CMS para referirse a los Sistemas de Manejo de Contenido ya que se conocen más por sus siglas en inglés que por sus siglas en español.

- 4) Es muy complejo su instalación y posterior administración, pues requiere de muchas configuraciones adicionales.
- 5) No funciona correctamente sobre todos los navegadores de los más usados regularmente en Internet.
- 6) Es muy difícil montarlo directamente sobre otro sistema operativo diferente de Windows-k-Server.
- 7) Windows-k-Server con IIS y el SQL-Server no pertenecen a la rama de software libre. Lo que implica una inversión adicional en estos sistemas.

Objetivos

Sobre la base de lo anterior, se plantean los siguientes objetivos:

Objetivos generales:

- 1) Implementación y puesta a punto de un sistema de manejo de contenido que responda a las necesidades y requisitos que hoy se exigen para el sitio Web de una facultad. Teniendo como objetivo principal el desarrollo de módulos que garanticen a los diferentes usuarios del sistema un fácil manejo de la información, los servicios y del sistema en general; mejorando de este modo la capacidad de actualización y la utilidad práctica del Sitio Web.

Objetivos específicos:

- 1) Revisar la bibliografía existente sobre los sistemas de manejo de contenido para su implementación.
- 2) Constatar en Internet artículos donde se evalúe la existencia y eficacia de otros sistemas de manejo de contenido.
- 3) Implementación del nuevo sistema.
 - a) Perfeccionar el sistema de publicación y el acceso a materiales en formato electrónico, implementando o incorporando módulos para:
 - i) Carga y descarga de archivos al sitio.
 - ii) Edición de texto WYSIWYG².
 - iii) Artículos y noticias.

² (What You See Is What You Get): lo que ves es lo que obtienes, que aplicado a la edición significa trabajar con un documento con el aspecto real que tendrá.

iv) Buscador.

Los cuales servirán de apoyo a la docencia y la investigación.

b) Incorporar otros módulos de interés secundario tales como, sistema de encuestas, galería de imágenes, foro de discusión, chat y otros.

4) Instalación del Sitio Web interno y externo de facultad, así como el correspondiente a cada departamento, carrera, disciplina y asignatura para mantener una completa información del proceso docente educativo y de los cursos de pregrado y postgrado.

Hipótesis

Con la confección de un CMS con facilidades para el manejo de documentos digitales de una forma sencilla y rápida, sin exigir ningún tipo de conocimiento de programación por parte del publicador, se ameniza y agiliza el proceso de publicación de los materiales correspondientes a la docencia y la investigación. Lográndose de este modo que la documentación o información deseada llegue a los usuarios finales en un menor espacio de tiempo, pues se requiere menos esfuerzo por parte del publicador.

Contenido del trabajo

El presente trabajo cuenta con tres capítulos de desarrollo:

Un primer capítulo de marco teórico, donde se aborda sobre los conceptos fundamentales de los términos que interviene en el estudio de los “Sistemas de Manejo de Contenido”, tales como creación y gestión de contenido, publicación y presentación; así como la necesidad de uso, la historia, el criterio de selección, el presente y el futuro de los CMS. También trata las ventajas y desventajas del PHP y el MySQL como posibles candidatos en la implementación de un CMS. Más el estudio de algunos aspectos relacionados con la seguridad de un sitio Web y encriptación de datos confidenciales.

El segundo capítulo de este trabajo se refiere al diseño de un CMS que cubra los objetivos planteados. Así como el diseño de cada una de las clases de los subsitios que conformarán el sitio Web de la facultad. Donde un subsitio no es más que el sitio Web correspondiente a un departamento, carrera, disciplina o asignatura.

Y un tercer y último capítulo que constituye la guía del navegador, donde se muestra cuales son como se accede a cada una de las opciones que brinda el sistema.

Capítulo 1: Los Sistemas de Manejo de Contenidos

1.1 Conceptos y definiciones fundamentales.

1.1.1 ¿Qué es un CMS?

Un sistema de manejo de contenidos (Content Management Systems o CMS) es un software que se utiliza principalmente para facilitar la gestión de sitios Web, ya sea en Internet o en una intranet, y por eso también son conocidos como gestores de contenido Web (Web Content Management o WCM). Hay que tener en cuenta, sin embargo, que la aplicación de los CMS no se limita sólo a los sitios Web [3].

La funcionalidad de los sistemas de manejo de contenidos se divide en cuatro categorías: creación de contenido, gestión de contenido, publicación y presentación [1].

1.1.2 Creación de contenido.

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

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

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

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

1.1.3 Gestión de contenido.

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

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

1.1.4 Publicación.

Una página aprobada se publica automáticamente cuando llega la fecha de publicación, y cuando caduca se archiva para futuras referencias. En su publicación se aplica el patrón definido para toda la Web o para la sección concreta donde está situada, de forma que el resultado final es un Web con un aspecto consistente en todas sus páginas. Esta separación entre contenido y forma permite que se pueda modificar el aspecto visual de un Web sin afectar a los documentos ya creados y libera a los autores de preocuparse por el diseño final de sus páginas.

1.1.5 Presentación.

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

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

1.2 Necesidad de un CMS.

En el apartado anterior se han presentado bastantes motivos para ver la utilidad de un sistema que gestione un entorno Web, pero se podría pensar que no es necesario para un Web relativamente pequeño o cuando no se necesitan tantas funcionalidades. Eso sólo podría ser cierto para un Web con unas pocas páginas estáticas para el que no se prevea un crecimiento futuro ni muchas actualizaciones, lo que no es muy realista. En cualquier otro caso, la flexibilidad y escalabilidad que permiten estos sistemas, justifican su utilización en prácticamente cualquier tipo de Web [2].

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

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

- **Inclusión de nuevas funcionalidades en el Web.** Esta operación puede implicar la revisión de multitud de páginas y la generación del código que aporta las funcionalidades. Con un CMS eso puede ser tan simple como incluir un módulo realizado por terceros, sin que eso suponga muchos cambios en la Web. El sistema puede crecer y adaptarse a las necesidades futuras.
- **Mantenimiento de gran cantidad de páginas.** En una Web con muchas páginas hace falta un sistema para distribuir los trabajos de creación, edición y mantenimiento con permisos de acceso a las diferentes áreas. También se tienen que gestionar los metadatos de cada documento, las versiones, la publicación y caducidad de páginas y los enlaces rotos, entre otros aspectos.
- **Reutilización de objetos o componentes.** Un CMS permite la recuperación y reutilización de páginas, documentos, y en general de cualquier objeto publicado o almacenado.
- **Páginas interactivas.** Las páginas estáticas llegan al usuario exactamente como están almacenadas en el servidor Web. En cambio, las páginas dinámicas no existen en el servidor tal como se reciben en los navegadores, sino que se generan según las peticiones de los usuarios. De esta manera cuando por ejemplo se utiliza un buscador, el sistema genera una página con los resultados que no existían antes de la petición. Para conseguir esta interacción, los CMS conectan con una base de datos que hace de repositorio central de todos los datos de la Web.

- **Cambios del aspecto de la Web.** Si no hay una buena separación entre contenido y presentación, un cambio de diseño puede comportar la revisión de muchas páginas para su adaptación. Los CMS facilitan los cambios con la utilización, por ejemplo, del estándar CSS (Cascading Style Sheets u hojas de estilo en cascada) con lo que se consigue la independencia de presentación y contenido.
- **Consistencia de la Web.** La consistencia en un Web no quiere decir que todas las páginas sean iguales, sino que hay un orden (visual) en vez de caos. Un usuario nota enseguida cuándo una página no es igual que el resto de las de la misma Web por su aspecto, la disposición de los objetos o por los cambios en la forma de navegar. Estas diferencias provocan sensación de desorden y dan a entender que el Web no lo han diseñado profesionales. Los CMS pueden aplicar un mismo estilo en todas las páginas con el mencionado CSS, y aplicar una misma estructura mediante patrones de páginas.
- **Control de acceso.** Controlar el acceso a un Web no consiste simplemente al permitir la entrada al Web, sino que comporta gestionar los diferentes permisos a cada área del Web aplicados a grupos o individuos.

1.3 CMS comerciales y de código abierto.

Se puede hacer una primera división de los CMS según el tipo de licencia escogido. Por una parte están los CMS comercializados por empresas que consideran el código fuente un activo más que tienen que mantener en propiedad, y que no permiten que terceros tengan acceso. Por la otra tenemos los de código fuente abierto, desarrollados por individuos, grupos o empresas que permiten el acceso libre y la modificación del código fuente [5].

La disponibilidad del código fuente posibilita que se hagan personalizaciones del producto, correcciones de errores y desarrollo de nuevas funciones. Este hecho es una garantía de que el producto podrá evolucionar incluso después de la desaparición del grupo o empresa creadora.

Algunas empresas también dan acceso al código, pero sólo con la adquisición de una licencia especial o después de su desaparición. Generalmente las modificaciones sólo pueden hacerlas los mismos desarrolladores, y siempre según sus prioridades.

Los CMS de código abierto son mucho más flexibles en este sentido, pero se podría considerar que la herramienta comercial será más estable y coherente al estar desarrollada por un mismo grupo. En la práctica esta ventaja no es tan grande, ya que los CMS de código abierto también están coordinados por un único grupo o por empresas, de forma similar a los comerciales.

Utilizar una herramienta de gestión de contenidos de código abierto tiene otra ventaja que hace decidirse a la mayoría de usuarios: su coste. Habitualmente todo el software de código abierto es de acceso libre, es decir, sin ningún coste en licencias. Sólo en casos aislados se hacen distinciones entre empresas y entidades sin ánimo de lucro o particulares. En comparación, los productos comerciales pueden llegar a tener un coste que sólo una gran empresa puede asumir.

En cuanto al soporte, los CMS comerciales acostumbran a dar soporte profesional, con un coste elevado en muchos casos, mientras que los de código abierto se basan más en las comunidades de usuarios que comparten información y solución a los problemas. Las formas de soporte se pueden mezclar, y así encontramos CMS de código abierto con empresas que ofrecen servicios de valor añadido y con activas comunidades de usuarios. En el caso comercial también sucede, pero el coste de las licencias hace que el gran público se decante por otras opciones y por lo tanto las comunidades de soporte son más pequeñas.

Un problema que acostumbra a tener el software de código abierto es la documentación, generalmente escasa, dirigida a usuarios técnicos, o mal redactada. Este problema se agrava en el caso de los módulos desarrollados por terceros, que no siempre incorporan las instrucciones de su funcionamiento de forma completa y entendible.

En el mercado hay CMS de calidad tanto comerciales como de código abierto. Muchos CMS de código abierto están poco elaborados (aunque en plena evolución), pero también lo encontramos entre los comerciales. En definitiva, un buen CMS de código abierto es mucho más económico que su homólogo comercial, con la ventaja de disponer de todo el código fuente y de una extensa comunidad de usuarios.

Por todos estos motivos, y como apuesta por la filosofía del software libre, en este trabajo sólo se presentan algunos CMS de código abierto.

1.4 Historia de los CMS.

A principios de los años noventa, el concepto de sistemas de gestión de contenidos era desconocido. Algunas de sus funciones se realizaban con aplicaciones independientes: editores de texto y de imágenes, bases de datos y programación a medida.

Ya el año 1994 Illustra Information Technology utilizaba una base de datos de objetos como repositorio de los contenidos de una Web, con el objetivo de poder reutilizar los objetos y ofrecía a los autores un entorno para la creación basado en patrones. La idea no cuajó entre el público y la parte de la empresa enfocada a la Web fue comprada por AOL, mientras que Informix adquirió la parte de bases de datos [1].

RedDot es una de las empresas pioneras que empezó el desarrollo de un gestor de contenidos el año 1994. No fue hasta a finales del año siguiente que presentaron su CMS basado en una base de datos.

Entre los CMS de código abierto uno de los primeros fue Typo 3, que empezó su desarrollo el año 1997, en palabras de su autor, Kasper Skarhoj, "antes de que el término gestión de contenidos fuera conocido sobradamente".

PHPNuke, la herramienta que popularizó el uso de estos sistemas para las comunidades de usuarios en Internet, se empezó a desarrollar el año 2000. La primera versión supuso tres semanas de trabajo al creador, rescribiendo el código de otra herramienta, Thatware.

1.5 Presente y futuro de los CMS.

En la actualidad, aparte de la ampliación de las funcionalidades de los CMS, uno de los campos más interesantes es la incorporación de estándares que mejoran la compatibilidad de componentes, facilitan el aprendizaje al cambiar de sistema y aportan calidad y estabilidad.

Algunos de estos estándares son CSS, que permite la creación de hojas de estilo; XML, un lenguaje de marcas que permite estructurar un documento; XHTML, que es un subconjunto del anterior orientado a la presentación de documentos vía Web; WAI, que asegura la accesibilidad del sistema; y RSS, para syndicar contenidos de tipo noticia.

También las aplicaciones que rodean los CMS acostumbran a ser estándar (de facto), como los servidores Web Apache e IIS; los lenguajes PHP, Perl y Python; y las bases de datos MySQL y PostgreSQL. La disponibilidad para los principales sistemas operativos de estas aplicaciones y módulos, permite que los CMS puedan funcionar en diversas plataformas sin muchas modificaciones [1].

1.6 Los CMS en el e-learning.

El e-learning tiene unas necesidades específicas que un CMS general no siempre cubre, o si lo hace, no da las mismas facilidades que una herramienta creada específicamente por esta función.

En general, los sistemas de gestión del aprendizaje (Learning Management Systems o LMS) facilitan la interacción entre los profesores y los estudiantes, aportan herramientas para la gestión de contenidos académicos y permiten el seguimiento y la valoración de los estudiantes. Es decir, facilitan una translación del modelo real en el mundo virtual.

Un buen ejemplo de sistema de gestión de cursos es Moodle <<http://www.moodle.org>>, uno de los más conocidos con licencia de código abierto. Sus características pueden servir para concretar algunas de las funcionalidades que se esperan de este tipo de herramientas:

- Administración de profesores y alumnos.
- Aulas virtuales que contienen toda la información de un curso y permiten la comunicación con foros o con chats.
- Creación, mantenimiento y publicación del material de un curso, con soporte de diferentes formatos, incluidos audio y vídeo.
- Talleres virtuales.
- Exámenes y tests con valoraciones.
- Trabajos con fecha de límite de entrega y aviso al profesor en caso de incumplimiento.
- Seguimiento estadístico de las acciones del estudiante.

Estos sistemas son diferentes a los CMS, tanto por el objetivo como por las características, pero actualmente empiezan a incluir capacidades de los sistemas de gestión de contenidos. Con la integración de las dos herramientas nace un nuevo concepto, los LCMS (*Learning Content Management Systems* o *sistemas de gestión de contenidos para el aprendizaje*).

1.7 Criterios de selección.

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

La siguiente lista está basada en las funciones principales de los CMS expuestas anteriormente, las indicaciones de Robertson, J. y una recopilación de los requerimientos básicos de una Web [2].

- **Código abierto.** Por los motivos mencionados anteriormente, el CMS tendría que ser de código fuente abierta (o libre).
- **Arquitectura técnica.** Tiene que ser fiable y permitir la escalabilidad del sistema para adecuarse a futuras necesidades con módulos. También tiene que haber una separación de los conceptos de contenido, presentación y estructura que permita la modificación de uno de ellos sin afectar a los otros. Es recomendable, pues, que se utilicen hojas de estilo (CSS) y patrones de páginas.

- **Grado de desarrollo.** Madurez de la aplicación y disponibilidad de módulos que le añaden funcionalidades.
- **Soporte.** La herramienta tiene que tener soporte tanto por parte de los creadores como por otros desarrolladores. De esta manera se puede asegurar de que en el futuro habrá mejoras de la herramienta y que se podrá encontrar respuesta a los posibles problemas.
- **Posición en el mercado y opiniones.** Una herramienta poco conocida puede ser muy buena, pero hay que asegurar de que tiene un cierto futuro. También son importantes las opiniones de los usuarios y de los expertos.
- **Facilidad de uso.** La herramienta tiene que ser fácil de utilizar y aprender. Los usuarios no siempre serán técnicos, por lo tanto hace falta asegurar que podrán utilizar la herramienta sin muchos esfuerzos y sacarle el máximo rendimiento.
- **Accesibilidad.** Para asegurar la accesibilidad de una Web, el CMS tendría que cumplir un estándar de accesibilidad. El más extendido es WAI (Web Accessibility Initiative) del World Wide Web Consortium.
- **Velocidad de descarga.** Teniendo en cuenta que no todos los usuarios disponen de líneas de alta velocidad, las páginas se tendrían que cargar rápidamente o dar la opción.
- **Funcionalidades.** No se espera que todas las herramientas ofrezcan todas las funcionalidades, ni que éstas sean las únicas que tendrá finalmente la Web. Entre otras:
 - Editor de texto WYSIWYG a través del navegador.
 - Herramienta de búsqueda.
 - Comunicación entre los usuarios (foros, correo electrónico, chat).
 - Noticias.
 - Artículos.
 - Ciclo de trabajo (workflow) con diferentes perfiles de usuarios y grupos de trabajo.
 - Fechas de publicación y caducidad.
 - Webs personales.
 - Carga y descarga de documentos y material multimedia.
 - Avisos de actualización de páginas o mensajes en los foros, y envío automático de avisos por correo electrónico.
 - Envío de páginas por correo electrónico.
 - Páginas en versión imprimible.

- Personalización según el usuario.
- Disponibilidad o posibilidad de traducción al catalán y al castellano.
- Soporte de múltiples formatos (HTML, Word, Excel, Acrobat, etc.).
- Soporte de múltiples navegadores (Internet Explorer, Netscape, etc.).
- Soporte de sindicación (RSS, NewsML, etc.).
- Estadísticas de uso e informes.
- Control de páginas caducadas y enlaces rotos.

1.8 Lenguaje de programación PHP.

1.8.1 ¿Qué es PHP?

PHP (acrónimo recursivo de "PHP: Hypertext Preprocessor", originado inicialmente del nombre PHP Tools, o Personal Home Page Tools) es un lenguaje de programación implantado (enraizado) en HTML. La mayoría de sus sintaxis están prestadas de los lenguajes de programación C, Java y Perl, con la inclusión de algunos rasgos únicos de PHP. La meta del lenguaje es permitir a los desarrolladores de sitios web escribir rápidamente páginas dinámicas. Es un lenguaje de programación interpretado. Aunque fue concebido en el tercer trimestre de 1994 por Rasmus Lerdorf no fue hasta el día 8 de Junio de 1995 que fue lanzada la versión 1.0. Se utiliza entre otras cosas para la programación de páginas web activas, y se destaca por su capacidad de mezclarse con el código HTML. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un compilador comercial denominado (Zend Optimizer).

El uso más extendido del lenguaje PHP, es el de formar parte de una página web con el propósito de agregarle características dinámicas ^(ver Fig. 1-1). Su interpretación y ejecución se da en el servidor en el cual se encuentra almacenada la página y el cliente solo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página web, enriquecida con código PHP, el servidor interpretará las instrucciones mezcladas en el cuerpo de la página y las sustituirá con el resultado de la ejecución antes de enviar el resultado a la computadora del cliente. Además es posible utilizarlo para generar archivos PDF, Flash o JPG, entre otros [8].

1.8.2 Usos de PHP.

Los principales usos del PHP son los siguientes:

- 1). Programación de páginas web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo la norma ODBC, lo que amplía en gran medida sus posibilidades de conexión.
- 2). Programación en consola, al estilo de Perl, en Linux, Windows y Macintosh.
- 3). Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y GTK (Gimp Tool Kit), que permite desarrollar aplicaciones de escritorio tanto para los sistemas operativos basados en Unix, como para Windows y Mac OS [8].

1.8.3 ¿Por qué usar PHP y no otro lenguaje de programación?

Ventajas de PHP:

- 1). PHP corre en (casi) cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en algo así como 25 plataformas, incluyendo diferentes versiones de Unix, Windows (95,98,NT,ME,2000,XP) y Macs. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al OS [9].
- 2). La sintaxis de PHP es similar a la del C, por esto cualquiera con experiencia en lenguajes del estilo C podrá entender rápidamente PHP. Entre los lenguajes del tipo C incluimos al Java y Javascript, de hecho mucha de la funcionalidad del PHP se la debe al C en funciones como fread() o strlen(), así que muchos programadores se sentirán como en casa.
- 3). Muchas interfaces distintas para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THTTPD. Otra alternativa es configurarlo como modulo CGI.
- 4). Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos. Siempre podrás disponer de ODBC para situaciones que lo requieran.
- 5). Una gran variedad de módulos cuando un programador PHP necesite una interfase para una librería en particular, fácilmente podrá crear una API para esta. Algunas de las que ya vienen implementadas permiten manejo de gráficos, archivos PDF, Flash, Cybercash, calendarios, XML, IMAP, POP, etc.

- 6). Rapidez. PHP generalmente es utilizado como modulo de Apache, lo que lo hace extremadamente veloz. Esta completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria [9].

Desventajas de PHP:

- 1). Modelo de programación: No se hace uso de patrones, tampoco de buenas prácticas, no se utiliza la arquitectura multicapa. Por todo lo anterior la escalabilidad es limitada y el mantenimiento, si no se parte de un diseño previo potente, es tedioso.

PHP contra ASP: facilidad, portabilidad, precio, velocidad y más.

¿Por qué utilizo PHP y no ASP?

Lo primero que pensé es que era por su facilidad. Sin embargo viendo otras opciones en Internet me doy cuenta que además de lo sencillo del lenguaje, PHP tiene muchas otras características que lo hacen sin duda, no solo mejor que ASP, sino el mejor lenguaje de Script de todos. En mi caso, una de las razones por las que lo prefiero, es por que tiene funciones nativas para muchas Bases de Datos, sin embargo muchos lo prefieren por otras razones [10].

A continuación se describen algunos factores comparativos, que al final de cuentas sustentan la supremacía de PHP sobre ASP.

- **Es fácil de aprender.**

En un inicio era renuente a aprender PHP, en cuanto a lenguajes de Script solamente conocía ASP, sin embargo cuando comencé con PHP me di cuenta que estaba frente al lenguaje de Script mas sencillo que jamás había visto, al menos para alguien que ya poseía algunos conocimientos de "Lenguaje C".

Hoy en día están de sobra los sitios en Internet adonde se puede aprender a programar en PHP, en lo personal jamás tome en curso de PHP, todo lo he aprendido a través de Internet y algunos libros. Estoy seguro que la Mayorga de programadores de PHP, lo han aprendido de la misma forma.

ASP.NET incorpora muchas características que no tenia ASP y que son buenas, sin embargo es un lenguaje mucho más difícil de aprender que cualquier otro del mercado. En un inicio pensé que ASP.NET podría dar más y por ello me interese, pues incorpora nuevos conceptos importantes, sin embargo con el tiempo es fácil darse cuenta que por su complejidad jamás tendría el éxito de PHP y al final de todo, lo mismo que se puede hacer con ASP.NET se puede hacer con PHP5.

- **Es multiplataforma.**

Sin duda una de las grandes ventajas de PHP es esta, pues se puede utilizar el mismo código en casi cualquier Sistema Operativo del mercado. Esto es de gran importancia puesto que se puede usar sobre Software gratuito (Linux+Apache), el cual además de todo, es el mas seguro del mercado, en lo que ha servidores respecta. Y si soy alguien que no sabe mucho de LINUX, no tengo de que preocuparme, pues en el servidor no necesito la "súper interfaz" de Windows, bastará algún programa gratuito que pueda descargar e instalar fácilmente para administrarlo, por lo que necesitare inicialmente muy poco conocimiento de LINUX, el cual puedo ir adquiriendo en el camino, leyendo en algunas de las millones de páginas web que tienen información sobre este.

Si lo deseo puedo desarrollar la aplicación en Windows o MAC, pero luego subir el mismo código al servidor LINUX, Solaris, etc.

Si no tengo mi propio servidor, prefiero pagar uno con Linux, el cual es mas barato que uno con Windows Server y mi código lo puedo desarrollar en mi PC con Windows o MAC. La portabilidad de PHP es sin duda algo que lo hace más poderoso que ASP. PHP corre en más de 25 Plataformas diferentes, entre ellas LINUX, Windows, Solaris y MAC. Todo esto sin necesidad de un componente adicional que deba comprar.

- **Tiene funciones nativas para la mayoría de Bases de Datos.**

Esta es una de las razones que más me gusta de PHP, pues por diferentes razones he tenido que utilizar diferentes Bases de Datos, en lo personal prefiero no utilizar ningún "mediador" (ODBC, ADO, etc.). Hasta el momento he utilizado las funciones nativas que PHP tiene para MYSQL y MSSQL. En la actualidad el 90% del tiempo utilizo PHP+ MYSQL y todo funciona de maravilla. Además de las funciones para estas Bases de Datos, también tiene funciones nativas para muchas mas, como Postgre, SYBASE, SQLite, Informix, DBase, Adabas y muchas otras.

Sin duda la velocidad de respuesta es mejor con funciones nativas que con algún mediador. Por otro lado cuando se utilizan Bases de Datos potentes como Oracle y se utiliza ODBC, OLE o ADO, se pierde gran parte del poder, pues hay funciones propias de la Base de Datos que no se pueden utilizar con un mediador genérico como estos. Por lo tanto si se quiere utilizar todo el poder de la Base de Datos, solamente se puede con funciones nativas para esta.

- **Es código abierto.**

Hay un grupo de desarrolladores trabajando en mejoras, pero también puedo adaptar mis funciones propias, lo cual generalmente solo lo hacen usuarios avanzados, pero en caso de necesitarlo ahí esta la posibilidad. Una ventaja del Código Abierto es que no esta ligado a caprichos de alguna empresa que puede

llegar a tomar decisiones que afecten a los usuarios. Si bien PHP hoy en día utiliza el “Zend Engine”, si ellos tomaran alguna decisión arrebatada sobre PHP, algún grupo de desarrolladores podría tomar el código fuente y darle seguimiento por otras vías. Así como lo sucedido con Netscape que desapareció, pero con sus fuentes liberados se han desarrollado Mozilla y Firefox. Por ser código abierto hay muchas herramientas que se adaptan grandemente a este, pues hay desarrolladores que han creado funciones para que PHP se pueda utilizar con Oracle (Ellos incluso recomienda en varios artículos de su web a PHP y hasta le han hecho una sección en su web); con FLASH, el cual se adapta grandemente a PHP e incluso se pueden desarrollar películas SWF desde PHP, utilizando la librería gratuita MING; con documentos PDF, los cuales se pueden generar desde PHP, agregando alguna de las librerías para dicho fin; así como estas hay funciones propias para las mas grandes bases de datos del mercado: MSSQL, MYSQL, SYBASE, Informix y muchas más.

Hay personas que piensan que por ser Código Abierto, PHP es menos seguro, pero si esto fuera así, algunos grandes sitios como Yahoo o Amazon no lo estarían utilizando. La seguridad que PHP provee en si mismo es mejor que la de ASP, sobre todo si se monta sobre Apache, el cual es mucho menos vulnerable a ataques que el Windows Server de Microsoft. Aunque al final de todo, la seguridad de una aplicación en la web, dependerá en gran medida de las buenas practicas del desarrollador.

- **Velocidad.**

PHP es bastante veloz, sobre todo cuando se utiliza como modulo de Apache. Aun y cuando es un lenguaje interpretado, es bastante rápido e incluso mejora los tiempos de respuesta que tienen sus competidores que hoy en día son en su mayoría lenguajes compilados.

- **Es gratuito.**

Cualquiera podría decir que también ASP y ASP.NET son gratuitos, los cual es totalmente cierto. Sin embargo en estos lenguajes la mayoría de componentes desarrollados por terceros son de pago. Por otro lado ASP puede ser gratuito, pero no así el Sistema Operativo sobre el cual se monta, el cual es de pago. Si bien hoy en día algunas versiones de LINUX para servidor son pagadas, el precio de estas es mucho menor que el de una copia de Windows, además que las políticas de uso son mucho más abiertas. Por otro lado, para PHP hay librerías gratuitas casi que para hacer cualquier cosa.

A continuación se presenta una tabla comparativa de los factores más importantes para seleccionar una tecnología entre PHP4, PHP5 y ASP.NET, esta ha sido extraída de un artículo de la web de Oracle:

	PHP 4	PHP 5	ASP.NET
Precio del software	gratis	gratis	gratis
Precio de la plataforma	gratis	gratis	\$\$
Velocidad	rápido	rápido	Menos rápido
Eficiencia	fuerte	fuerte	débil
Seguridad	fuerte	fuerte	fuerte
Plataforma	fuerte	fuerte	débil (solo IIS)
Plataforma	variadas	variadas	win32 (solo IIS)
Código abierto	si	si	no
Excepciones	no	si	si
OOP	débil	fuerte	fuerte

Por último, en años anteriores se hablaba mucho de que una de las desventajas de PHP era su pobre y casi nula Objeto Orientación, sin embargo esto ha cambiado desde que se lanzo PHP5. Este ya es totalmente Orientado a Objetos. Por lo que esta ya no es ninguna desventaja de PHP frente a otros lenguajes de Script.

Por todo lo anterior me doy cuenta que la razón por la que utilizo PHP, se sustenta en que en este momento, es sin duda alguna, el lenguaje de Script mas poderoso del mercado. Me parece que esta es la opinión de la mayoría de desarrolladores en la actualidad, sino lo crees, basta con que veas la extensión de las páginas donde navegas (claro sin incluir las HTML!) y te darás cuenta que la gran mayoría son ".php", algo sorprendente es que el número de páginas que lo utilizan aumenta en gran medida con el tiempo ^(ver Fig. 1-2), y la brecha con sus competidores se abre cada día mas, sobre todo sobre la extensión ".asp" o ".aspx" [10].

1.8.4 Seguridad en PHP.

PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor web en forma de módulo o ejecutado como un binario CGI separado, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor web sea insegura por naturaleza. PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la

selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación, PHP le puede dar la combinación precisa de libertad y seguridad que usted necesita.

Ya que hay muchas maneras de utilizar PHP, existen varias opciones de configuración diseñadas para controlar su comportamiento. Un amplio rango de opciones le garantizan que pueda usar PHP para muchos propósitos distintos, pero también quiere decir que hay combinaciones de éstas opciones y configuraciones de servidor que pueden resultar en un entorno inseguro.

El nivel de flexibilidad en la configuración de PHP se compara quizás solo con su flexibilidad de desarrollo. PHP puede ser usado para escribir aplicaciones completas de servidor, con todo el poder de un usuario de un intérprete de comandos, o puede ser usado para inclusiones simples del lado del servidor con muy poco riesgo en un entorno minuciosamente controlado. Cómo construir ese entorno, y qué tan seguro es, básicamente depende del desarrollador PHP.

PHP divide la seguridad en varios temas (Instalación como un binario CGI, Instalación como módulo de Apache, Seguridad del sistema de archivos, Seguridad de Bases de Datos, Reporte de Errores, Uso de Register Globals, Datos Enviados por el Usuario, Magic Quotes, Ocultando PHP) los cuales explica detalladamente en todos sus manuales y en el sitio oficial de PHP <www.php.net>.

En nuestro caso uno de los primeros pasos para mejorar en la seguridad del sitio Web es, deshabilitar el reporte de errores y el registro de variables globales para evitar que un crack o cualquier otro usuario obtengan información del código PHP, tal como nombre de variables o funciones, rutas en el disco, etc. y logre modificar su valor a través de algún parámetro pasado en el url.

1.9 MySQL.

MySQL es uno de los gestores de bases de datos más populares desarrolladas bajo la filosofía de código abierto. La desarrolla y mantiene la empresa “*MySql AB*” pero puede utilizarse gratuitamente y su código fuente está disponible [11].

1.9.1 Características (versión 4.0 en adelante)

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad; aquellos elementos faltantes fueron llenados por la vía de las aplicaciones que la utilizan.

Poco a poco los elementos faltantes en MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

1.9.2 MySQL en cifras

Según las cifras del fabricante, existirían más de seis millones de copias de MySQL funcionando en la actualidad, lo que supera la base instalada de cualquier otra herramienta de bases de datos. El tráfico del sitio Web de “MySQL AB” supera en el 2004 el tráfico del sitio de IBM [11].

1.9.3 ¿Qué licencia utilizar?

La licencia GPL de MySQL obliga a distribuir cualquier producto derivado (aplicación) bajo esa misma licencia. Si un desarrollador desea incorporar MySQL en su producto pero no desea distribuirlo bajo licencia GPL, puede adquirir la licencia comercial de MySQL que le permite hacer justamente eso.

1.9.4 Ventajas y desventajas.

Ventajas de MySQL:

- 1). Mayor rendimiento.
- 2). Mayor velocidad tanto al conectar con el servidor como al servir consultas y demás funciones.
- 3). Mejores utilidades de administración (backup, recuperación de errores, etc.).
- 4). Aunque se cuelgue, no suele perder información ni corromper los datos.
- 5). Mejor integración con PHP.

- 6). No hay límites en el tamaño de los registros.
- 7). Mejor control de acceso, es decir, qué usuarios tienen acceso a qué tablas y con qué permisos.
- 8). MySQL se comporta mejor que otros motores de BD a la hora de modificar ó añadir campos a una tabla.

1.9.5 ¿Por qué MySQL?

- 1). MySQL es un servidor de bases de datos muy rápido, seguro y fácil de usar.
- 2). Fue desarrollado originalmente para manejar grandes bases de datos mucho más rápido que las soluciones existentes. Aunque se encuentra en constante desarrollo, ofrece actualmente un conjunto muy rico y útil de funciones.
- 3). Su conectividad, velocidad, y seguridad lo hacen un servidor especialmente apropiado para bases de datos y aplicaciones de Internet.
- 4). MySQL puede ser integrada tanto al ambiente de Windows como Linux permitiendo la migración y creación de aplicaciones para ambos sistemas operativos.

1.10 Autenticación segura en sitios Web. (Algoritmo OPT).

En este epígrafe se presenta varias alternativas para llevar a cabo el proceso de autenticación de forma segura y estudia el empleo de contraseñas de un solo uso como método seguro de autenticación en aplicaciones sobre redes [7].

1.10.1 Escenario.

Un problema clásico de las aplicaciones que trabajan en red es el de la autenticación. Es un escenario frecuente tener que enviar los datos que nos identifican a través de un medio que no es seguro (Figura 1-1).

Nuestro cliente se ve expuesto a dos medios susceptibles de interceptar su nombre de usuario y contraseña: la LAN e Internet.

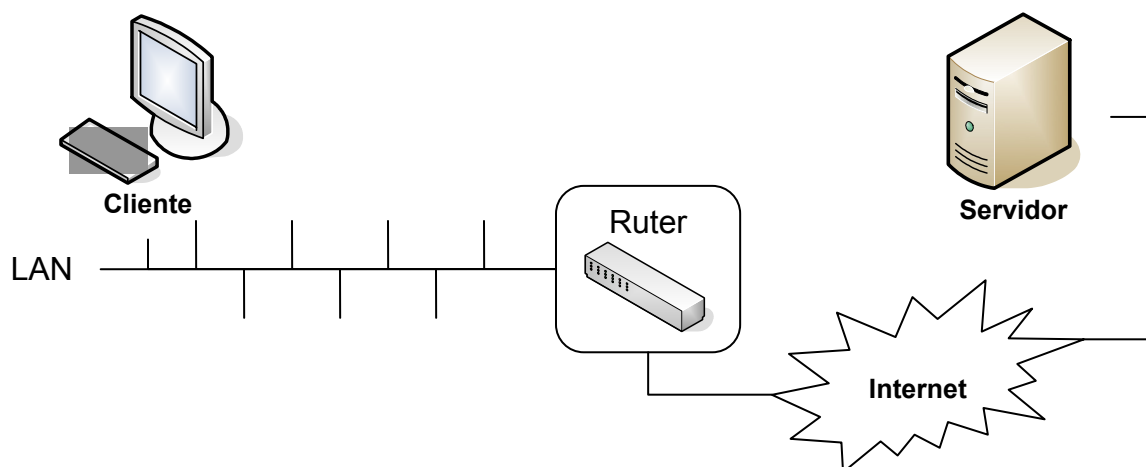


Figura 1-1: Escenario.

En estos casos hay diferentes soluciones que han demostrado ser eficaces (SSL, IpSec), pero que por un motivo u otro no siempre están disponibles para el desarrollador (no siempre es posible disponer de un servidor http con SSL).

1.10.2 Encriptación de contraseñas.

En primer lugar hay que tener en consideración la seguridad del servidor. Siempre se debe evitar el típico fichero con contraseñas en la forma: usuario contraseña. Las contraseñas siempre deben permanecer encriptadas en la máquina que realiza la autenticación.

Es recomendable el empleo de funciones hash (MD5, SHA, RIPEMD,...) para almacenar las contraseñas. Estas funciones son irreversibles (teniendo el número generado por la función hash no existe el proceso inverso para obtener los datos que generaron ese número), por lo que es permisible que pudieran ser leídas por un intruso.

Almacenar la suma de comprobación de la contraseña empleando una función hash implica que si perdemos la contraseña, nos será imposible recuperarla ya que solo disponemos de del valor generado.

Parece que la solución está clara: como el valor resultante de la fórmula hash no es reversible, se puede enviar ese número para realizar la autenticación, evitando así el envío de contraseñas visibles.

Nada más lejos de la realidad: el hecho de enviar las contraseñas encriptadas no soluciona en absoluto nuestro problema ya que, un intruso situado en el medio

inseguro, puede interceptar la suma de comprobación para luego simplemente enviarla directamente al servidor.

Como no podemos evitar que los datos enviados se intercepten, nos encontramos ante una difícil situación.

1.10.3 Contraseñas de un solo uso (OTP).

Si los datos de identificación se pueden interceptar, podríamos disponer de contraseñas que solo fueran válidas para un uso, de forma que los datos interceptados estuvieran caducados cuando el intruso quisiera emplearlos.

Esta forma de actualización parece poco viable, ya que implica un cambio de contraseña por cada acceso, y además esa contraseña debería enviarse por un medio seguro. No obstante, se puede implementar un sistema de contraseñas de un solo uso empleando el mismo medio inseguro para enviar las nuevas contraseñas.

Para emplear el sistema OTP necesitamos definir:

- **Contraseña:** es la contraseña que realmente conoce el cliente.
- **Llave temporal:** una contraseña temporal asociada a un instante en el tiempo y a los datos disponibles del cliente.
- **Sesión:** intervalo de tiempo en el cual tiene validez una llave temporal.

Para la implantación del sistema necesitamos soporte para encriptación tanto en el cliente como en el servidor. Supongamos para la explicación el uso de MD5.

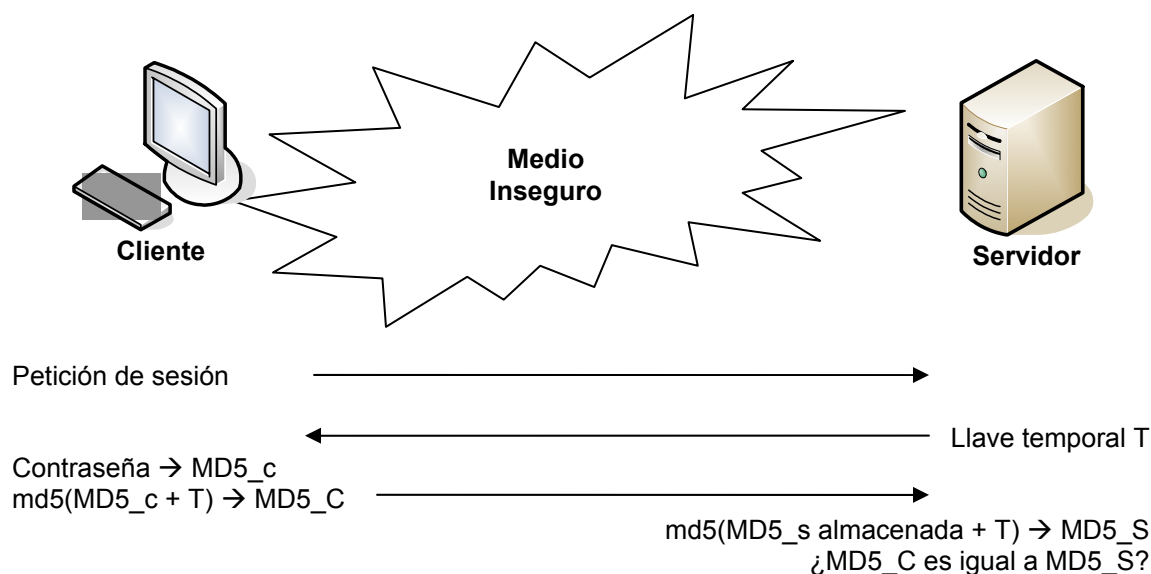


Figura 1-2: Algoritmo OTP.

En la (Figura 1-2) se observa el algoritmo de contraseñas de un uso:

1. El cliente pide una nueva sesión.
2. El servidor le devuelve una llave temporal t que solo valdrá para esa sesión.
3. El cliente encripta la contraseña proporcionada por el usuario con MD5 obteniendo MD5_c. Es necesario ya que el servidor tiene almacenadas las contraseñas en esa forma. Entonces se procesa MD5_c empleando T . Ese procesamiento puede ir desde una concatenación hasta la aplicación de los algoritmos de cifrado. De ese proceso se obtiene un MD5 que llamamos MD5_C, y es el valor que enviaremos al servidor.
4. Ahora el servidor debe realizar el proceso que ya ha hecho el cliente para ver si obtiene el mismo MD5 recibido MD5_C. Para ello procesa MD5_s (la contraseña almacenada previamente encriptada con MD5) con T y aplica el MD5 para obtener MD5_S que finalmente es comparado con MD5_C.

Analicemos los puntos débiles del algoritmo (envío de datos).

Cuando el cliente pide una nueva sesión, el servidor genera T empleando la hora del sistema o un número aleatorio, asociando la IP del cliente a ese valor. El servidor no debe guardar T , sino el valor temporal o aleatorio. Posteriormente recalculará T en el paso 4 con la IP del cliente que envía los datos del paso 3. De esta forma evitamos que una máquina pida sesión y otra distinta conteste. Por supuesto esto no es infalible, aunque si proporciona un nivel aceptable de seguridad para muchos casos.

El cliente, en el paso 3, envía un MD5 por el medio inseguro. Ese dato podría ser interceptado, pero solo tiene validez para esa sesión, por lo que resultaría completamente inútil para un intruso.

Como se puede observar, realmente se emplea una contraseña cada vez, pero de forma automática y transparente para el usuario y el administrador.

1.10.4 Implementación: PHP con MD5.

Supongamos una aplicación web sobre http que requiere de una autenticación segura vía formulario y no se dispone de SSL.

La implementación requiere de:

- Servidor: PHP (versión 4, adaptable a anteriores)
- Cliente: JavaScript (necesario para encriptación)

El sistema requiere de dos partes diferenciadas: cliente y servidor.

La parte cliente consta de un formulario en el que se introducen los datos de autenticación. El documento lleva código JavaScript con una llave temporal, y se encarga de encriptar y procesar la contraseña mediante MD5 antes de enviar los datos mediante POST al servidor.

La parte servidor se encarga de, empleando variables de sesión, crear una sesión cuando se genera el formulario generando la llave temporal, y posteriormente verificar el MD5 enviado.

En el cliente se emplea una implementación de MD5 en JavaScript bajo licencia BSD disponible en <http://pajhome.org.uk/crypt/md5> y se puede bajar el código completo de la implementación de un ejemplo de autenticación usando el OTP (incluye md5.js) en <http://www.usebox.net/jjm/passw/otp-php.tar.gz>.

A modo de conclusión se puede decir que el empleo de contraseñas de un solo uso puede aumentar de forma considerable la seguridad de las autenticaciones en redes inseguras cuando no se dispone de recursos especiales para ese propósito (SSL, por ejemplo), de una forma sencilla y efectiva.

Posteriormente he descubierto que Yahoo! Mail parece utilizar este sistema (o similar) para autenticar a los usuarios cuando no se emplea SSL [7].

Capítulo 2: Diseño del Sistema y los sitios Web.

En este capítulo se describen los elementos correspondientes a la etapa de análisis y diseño, utilizando la notación de UML para la modelación. Entre la información que se relaciona de esta etapa, se encuentra la expansión de los casos de uso, diagrama de navegación y actores del sistema.

2.1 Diseño del Sistema de Manejo de Contenido (CMS).

Descripción detallada del problema

Una representación general de un CMS está compuesta por:

- 1) Uno o varios menús que contienen múltiples opciones que representan enlaces a otras páginas que pueden pertenecer o no al propio sistema o sitio Web.
- 2) Un conjunto de servicios insertados al sistema los cuales pueden tener todo un diagrama de navegación propio.
- 3) Módulos para registrarse, inicial sesión y cerrar sesión.
- 4) Un conjunto de módulos que conforman el sistema de administración para el manejo de los usuarios, los menús y cada uno de los diferentes servicios que brindará el sistema.

Es importante señalar que existen servicios que están presentes en casi todos los sitios web; por lo que en nuestro caso serán insertados junto con sus correspondientes módulos de administración, como servicios por defecto en el CMS. Algunos de estos servicios son: buscador, publicación de noticias, artículos y páginas de contenido, carga y descarga de recursos, bloques informativos, además de los módulos relacionados con la administración de la base de datos y las variables de entorno del sistema de manejo de contenido.

2.1.1 Diagrama de Navegación

El diagrama de navegación está formado por un conjunto de estados y sus correspondientes enlaces, a través de los cuales un usuario puede transitar moviéndose de un estado a otro.

En el CMS, cada estado representa una o varias páginas que conforman una vista en el navegador que representan un servicio determinado, y tiene enlace con los estados principales de cada servicio e incluso con el estado inicial, esto es un problema que existe en muchos sitios Web, donde durante la navegación se llega a algunos estados de los que no se puede regresar a la página principal.

A continuación se definirán los estados a través de los cuales transita un usuario partiendo del estado inicial o página principal con un diagrama general y detallando posteriormente en los diagramas de navegación correspondientes a algunos servicios y módulos de administración.

Diagrama general de navegación del CMS:

Un usuario a partir del estado inicial (página principal) puede acceder a las diferentes opciones de cada uno de los menús que brinda el sistema. Por ejemplo en un menú de servicios puede encontrar funcionalidades referentes a chat, foro de discusión, noticias entre otras.

En la Figura 2-1, que se muestra a continuación, se representan estas funcionalidades para un usuario anónimo.

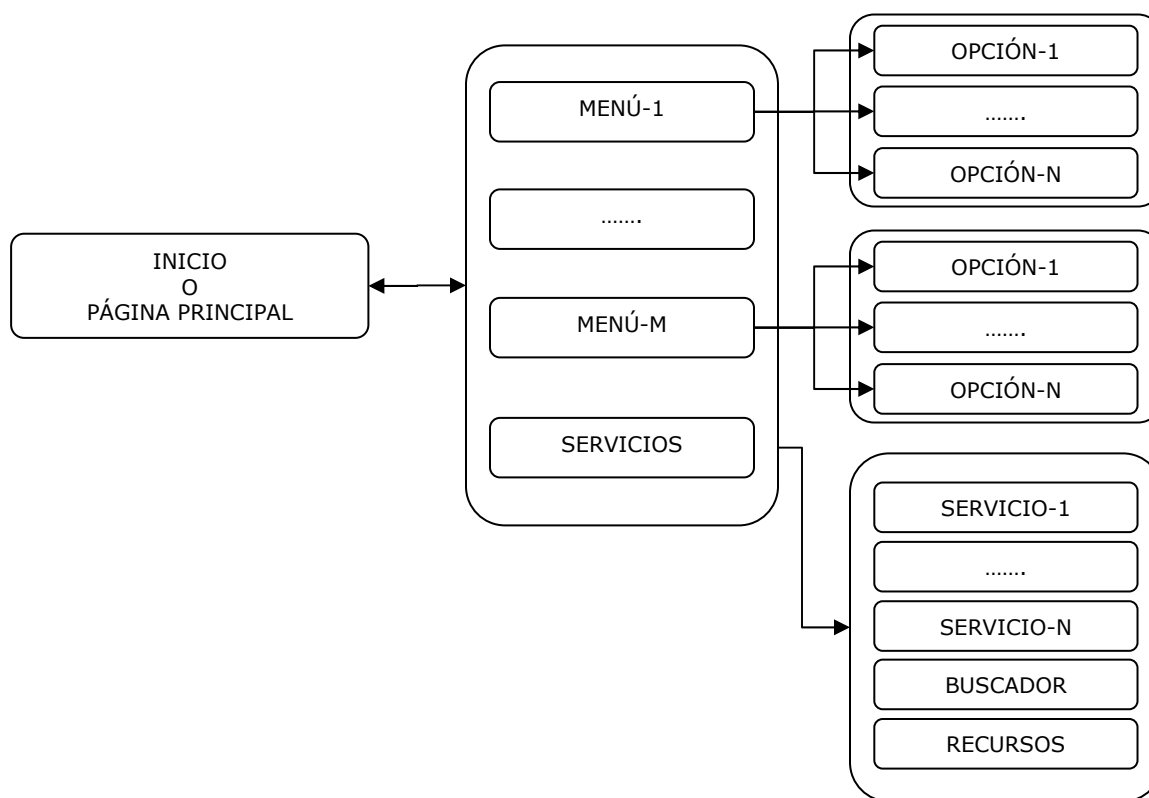


Figura 2-1: Diagrama general de navegación del CMS.

Por otro lado, podemos encontrar estados personalizados dirigidos a tareas administrativas. Para acceder a estos estados un usuario tendría que inicialmente registrarse, y una vez iniciada la sesión, dependiendo del rol que se le haya asignado, podría acceder a los módulos principales de administración, entre los

cuales podemos encontrar la administración de noticias, artículos, variables de entorno del sistema, carga y descarga de recursos, base de datos y otras que en su conjunto conforman el sistema de administración.

En la Figura 2-2, que aparece a continuación, se muestran los pasos para acceder a cada uno de estos módulos.

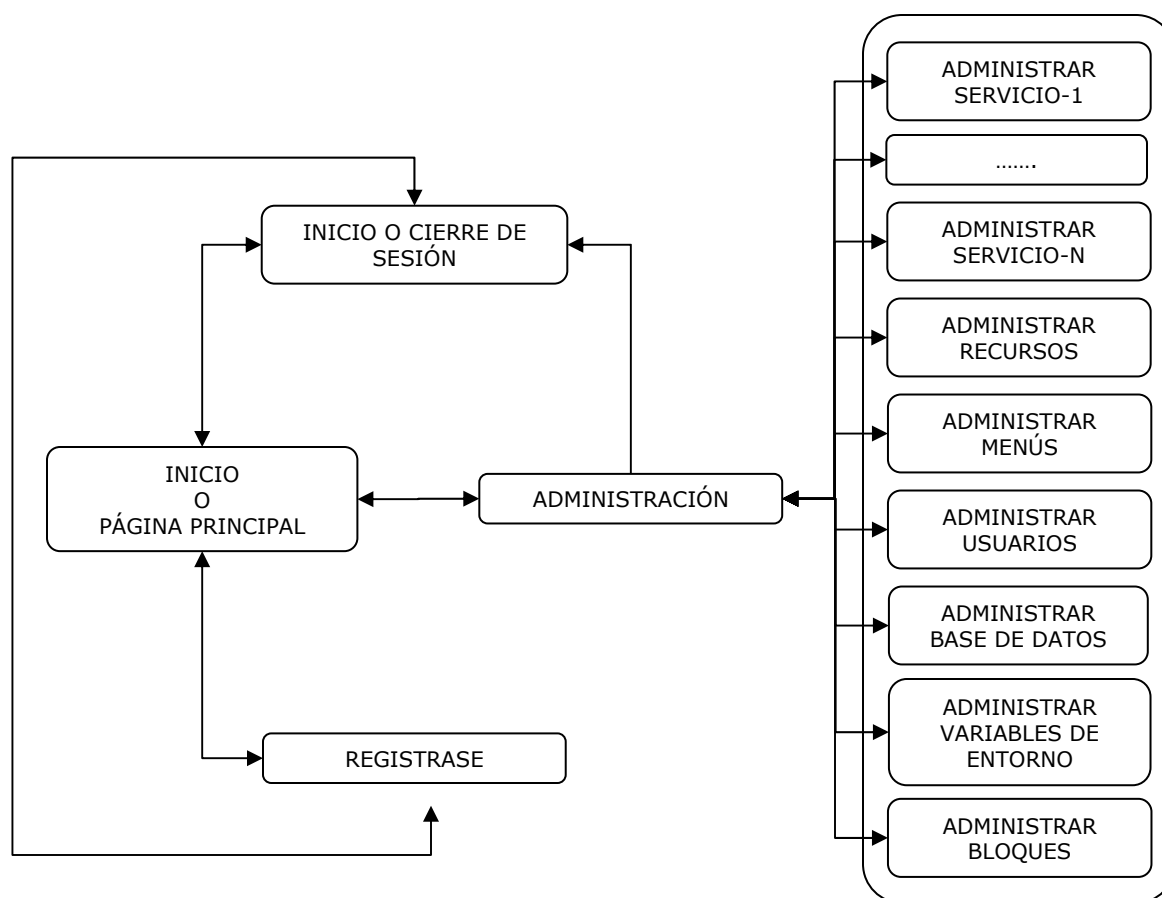


Figura 2-2: Diagrama general de navegación para los módulos de administración.

Cada uno de los estados finales que se presentan en el diagrama general de navegación, constituyen módulos o servicios independientes que a su vez brindan otro grupo adicional de opciones al usuario, por lo que cada uno contendrá su propio diagrama de navegación.

A continuación se describen algunos de estos módulos o servicios, los cuales brindan un conjunto de opciones, algunas de adición, edición o eliminación de algún registro, teniendo esta última un estado de confirmación de la acción. También notará que en algunos diagramas se agrupan varios módulos, lo cual no significa que sean el mismo módulo ni que sean iguales, sino que presentan el mismo menú

de opciones, cada opción apunta a una acción dentro del propio módulo a través del cual se accedió a ella. Este tipo de agrupación es paso muy importante para el cumplimiento de los objetivos planteados en el presente trabajo, ya que de este modo durante la implementación se pueden crear fragmentos de códigos comunes para un grupo de módulos y almacenarlos en un solo archivo, el cual será usado posteriormente por cada uno de los módulos que lo necesiten. Con esta idea se reduce en gran medida la repetición de código y el tamaño final del CMS. Además la escalabilidad del sistema sería mucho más sencilla.

Servicio de carga y descarga de recursos:

Este es un servicio de gran utilidad en un CMS ya que brinda la posibilidad de subir o descargar archivos en múltiples formatos al o del servidor respectivamente.

Un usuario que acceda a este servicio podrá navegar por el mismo según define su diagrama de navegación (Figura 2-3), obteniendo de este modo la posibilidad de explorar los diferentes directorios y bajar los recursos que desee, o en caso de tener privilegios de especiales sobre este servicio, podrá entrar al módulo de administración y acceder a las opciones de: crear o eliminar directorio, subir, renombrar o eliminar recurso y crear enlace al recurso o directorio en uno de los menús existentes.

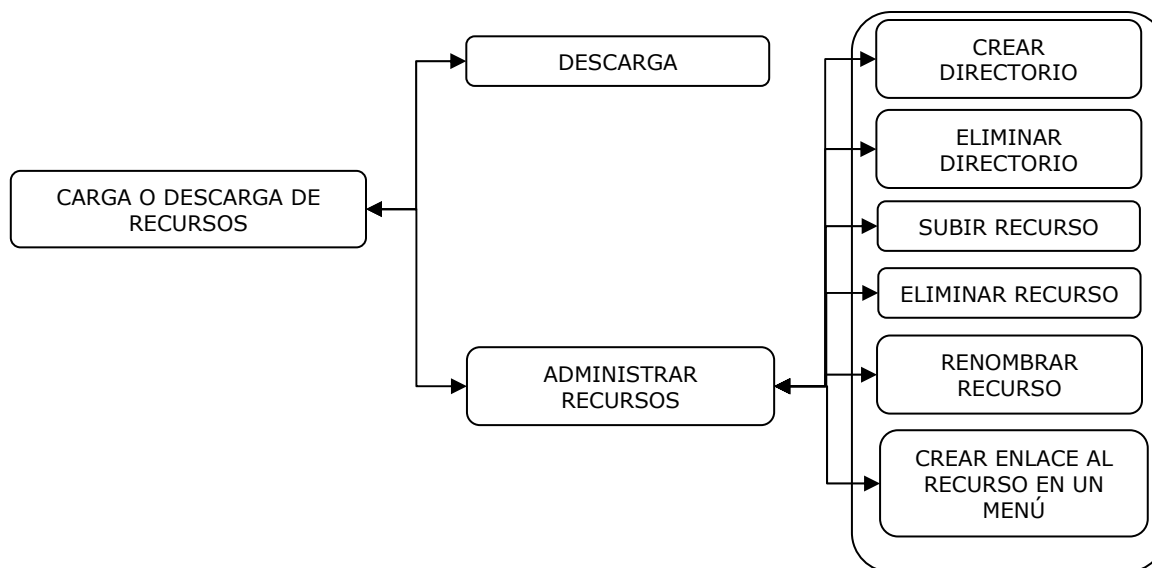


Figura 2-3: Diagrama de navegación. (Carga o descarga de recursos).

Módulos de administración de bloques, menús y opciones de menús:

Por razones de no redundar en la misma escritura, se tratarán estos módulos a la vez, ya que presentan el mismo menú de opciones aunque vale aclarar que cada una apunta a un módulo distinto en cada caso.

Antes que nada y para entender mejor este punto es necesario dar respuesta a la siguiente pregunta. ¿Qué son y donde se muestran los bloques de información o menús?

Un bloque de información no es más que una pequeña caja que se muestra el borde derecho o izquierdo o en ambos bordes del sitio Web. Estas cajas contienen un pequeño código php o html que se almacena en una base de datos, el cual es procesado para obtener un resultado que será mostrado en el borde correspondiente.

Un menú también es una caja, pero a diferencia de los bloques este puede aparecer además de en los laterales, en el tope del sitio Web, debajo de la cabecera; y en lugar de contener código lo que contiene es un grupo de opciones que también se almacenan en la base de datos y representan enlaces a otras páginas o sitios.

Conocido esto y según lo define el diagrama de navegación (Figura 2-4) para estos módulos, un usuario con privilegios de administración, que llegue a este estado, podrá: editar, adicionar o eliminar, activar o desactivar, subir o bajar un nivel en el orden de visualización, uno o varios bloques de información, menús u opciones de un menú según corresponda

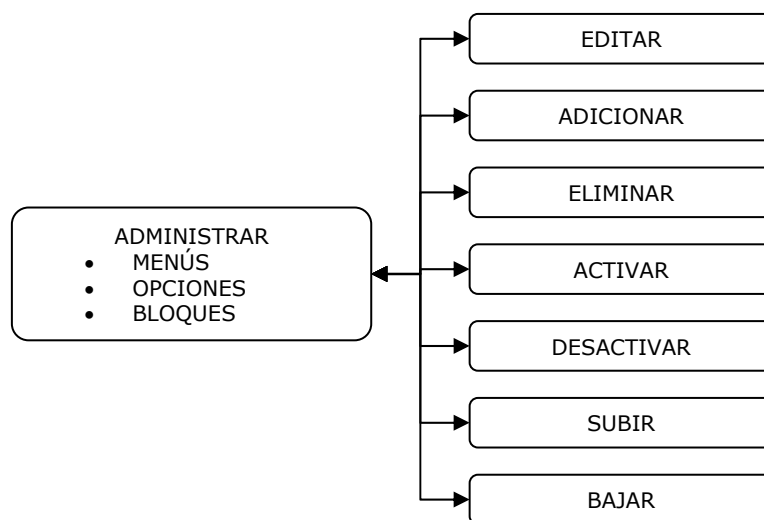


Figura 2-4: Diagrama de navegación. (Administrar menús, opciones y bloques).

En este caso los tres módulos cumplen con los términos de agrupación planteados en la introducción del tema de este epígrafe, contribuyendo de este modo a la escalabilidad del sistema.

Administración de noticias, artículos y páginas de contenido:

Las noticias, los artículos y las páginas de contenido tienen características muy parecidas, incluso las diferencias entre una noticia y un artículo son solo conceptuales, por que se pueden manejar con el mismo servicio y modulo de administración. Por otro lado una página de contenido puede ser en determinado momento una noticia, un artículo o formar parte de alguno de ellos.

El principal aspecto que relaciona a estos módulos es que un documento de cualquiera de estos tres tipos está formado fundamentalmente por un “título” y un “contenido” o “cuerpo del documento”, para el cual los tres módulos usan un edición de texto WYSIWYG.

Basado en lo anterior se construye el diagrama de navegación para la administración de estos tipos de documentos (Figura 2-5), el cual muestra como un usuario con privilegios administrativos, partiendo de un estado que representa alguno de estos módulos de administración, puede transitar hacia otros estados representados por las opciones de: editar, adicionar, eliminar y crear un enlace a un documento uno de los menús existentes.

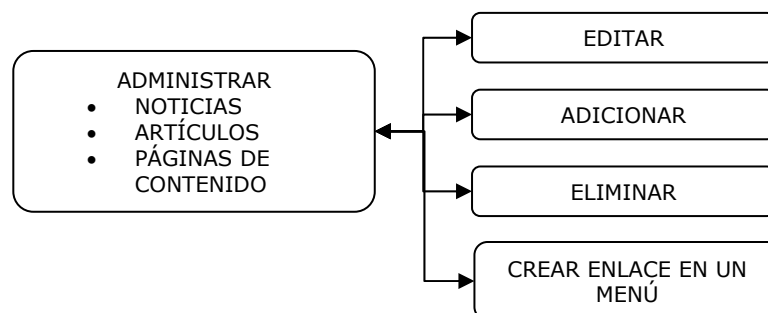


Figura 2-5: Diagrama de navegación. (Administrar noticias, artículos y páginas de contenido).

Note que en este caso también tenemos una agrupación de tres módulos que también cumplen con los términos de agrupación.

Administración de usuarios:

En este módulo se definen o manipulan los datos de los usuarios del sitio Web y se definen los privilegios del mismo. Aunque este es uno de los módulos más importantes, es uno de los más sencillos, puesto que solo incorpora tres nuevos estados identificados por las opciones de: editar, adicionar y eliminar un o varios usuarios tal y como se muestra en la Figura 2-6.

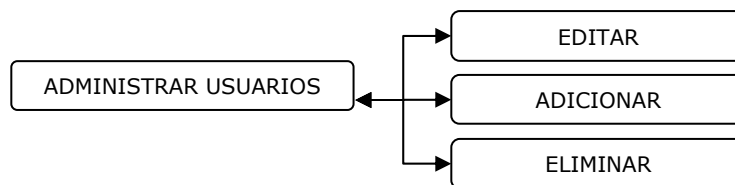


Figura 2-6 : Diagrama de navegación. (Administrar usuarios).

Administrar variables de entorno del CMS:

Es un módulo muy simple. Su única opción del es editar tal y como se muestra en el diagrama de navegación de la Figura 2-7.

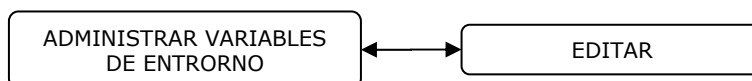


Figura 2-7: Diagrama de navegación. (Administrar variables de entorno).

Este es un módulo muy valioso puesto que a través de el se manipula la apariencia, el idioma y el título del sitio Web, además define el servidor de LDAP para la validación de los registros de usuarios algo muy importante para la seguridad, también controla el contador de visitas globales, importante para las estadísticas del sitio Web.

Administrar la base de datos:

Este es un módulo que por el momento solo incluye la opción de realizar copias de seguridad de la base de datos (Figura 2-8).

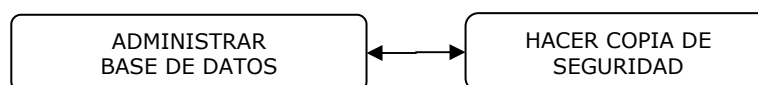


Figura 2-8: Diagrama de navegación. (Administrar base de datos).

Este módulo puede ser ampliado en un futuro para la administración avanzada del sistema, con usuarios administradores con un profundo conocimiento del diseño del sistema y la base de datos. Una posible ampliación de este módulo podría ser la instalación de una versión de PHPAdmin, que es un buen sistema para manejar bases de datos de MySQL sobre la web.

Habiendo concluido el análisis de los diagramas de navegación, es importante señalar que los servicios más importantes en función de los objetivos trazados son: carga y descarga de recurso y la Publicación de páginas de contenido. Ya que con ellos unidos a la administración de menús, usuarios y variables de entorno se logra publicar toda la documentación referente al proceso docente educativo, los cursos, las bibliografías digitales, las conferencias, las clases prácticas y cualquier otro tipo de información.

2.1.2 Usuarios y Privilegios

Al igual que en la mayoría de los sitios Web dinámicos, un CMS cuenta con un usuario anónimo y un usuario para la administración, además de un grupo de usuarios registrados con ciertos y determinados privilegios. A continuación se muestra un listado donde se describen los privilegios de los usuarios separados por categorías.

- 1) **Anónimo:** Este es el usuario inicial del sitio Web y tiene acceso de solo leer sobre la información publicada, lo que significa que no puede publicar ningún tipo de información.
- 2) **Administrador:** Usuario que tiene todos los privilegios sobre el sitio Web y sus servicios. Su función principal es el manejo de los usuarios y de las variables de entorno del sistema, aunque puede realizar cualquier otra tarea.
- 3) **Publicador o Editor:** Usuario con la potestad de crear, editar o eliminar documentos. Un usuario de este tipo puede administrar el servicio de noticias o el de artículos o ambos servicios a la vez.
- 4) **Usuario:** Los usuarios de esta categoría son similares al usuario anónimo, salvo que al estar registrados pueden ingresar comentarios en las noticias, los

artículos y las encuestas y pueden acceder a algunos servicios que por algún motivo es necesario llevar un registro de acceso de los usuarios.

Los usuarios pueden registrarse en el sitio Web a través del administrador o de forma automática mediante el módulo que brinda el CMS para registrarse. Estos usuarios pueden tener ciertos y determinados privilegios, los cuales son definidos posteriormente por el administrador sistema.

2.1.3 Seguridad del sistema.

Para lograr mayor seguridad e integridad de la información y los datos que se transmiten por la red entre el cliente y el servidor, además de las configuraciones del PHP que intervienen en la seguridad de un sitio Web ^(ver epígrafe 1.8.4), en la programación del CMS se evito el pase de parámetro confidenciales a través de método “GET” usando al mayor parte del tiempo el método “POST” con métodos de codificación y encriptación en los casos que fuese necesario.

Durante el proceso en que un usuario se registra en el sitio, en necesario que este llene un formulario con el nombre de usuario, contraseña y dominio para ser autenticado por un servidor de LDAP, estos datos serán enviados al servido y por supuesto deben ser encriptados antes del envío. Esto es un problema que presentan muchos CMS ya que envían los datos en texto plano o no usan algoritmos de encriptación muy eficientes y fáciles de burlar. En nuestro caso se usa el método OTP ^(ver epígrafes 1.10.3).

Es importante señalar que para mayor seguridad la contraseña para autenticarse contra le servidor LDAP, solo se utilizará una vez al registrase, para obtener los datos del usuario que se almacenan en dicho servidor. Posterior a esto el usuario escogerá una nueva contraseña para autenticarse en el sitio, que es la que se almacena encriptada en la base de datos.

2.1.4 Arquitectura del sistema

El CMS está dividido en tres componentes fundamentales, la plantilla, los módulos y el controlador.

Un plantilla es un directorio que contiene un grupo de archivos y entre ellos un script llamado “index.php”, que contiene un conjunto de función ^(ver Tabla 2-1) en las que se define la apariencia del sitio Web. Para lo cual fue necesario dividir una página en cinco sección principales, ^(ver Fig. 2-1) cabecera de página, borde derecho, cuerpo de la página, borde izquierdo y pie de página; lo que facilita la creación de múltiples y variados modelos de diseños.

Un módulo es una o varias páginas que forman una vista en el navegador, puede ser un servicio o simplemente páginas. Cualquier página dentro del CMS se compone de la siguiente forma³:

1. Incluir “controlador”.
2. Incluir “cabecera de página”.
3. Incluir “borde izquierdo”.
4. Código de la página.
5. Incluir “borde derecho”.
6. Incluir “pie de página”

La cabecera de página, el borde izquierdo, el borde derecho y el pie de página son scripts independientes escritos en código php, que pueden recibir mantenimiento por separado. De este modo se separa más el código del diseño, lo que hace al que el sistema sea más escalable y evita la repetición innecesaria de código.

Por último el controlador es el script principal del CMS. Es en donde se especifica el flujo de control entre todos los módulos que conforman el Sistema. Es decir, en este script es donde se define la conexión a la base de datos, de donde se extrae las variables de entorno del sitio Web (plantilla, idioma, título, servidor de ldap, palabras claves, descripción del sitio, etc); y es donde se definen las sesiones de trabajo de cada cliente conectado.

2.1.5 Actores y casos de uso del sistema

Los casos de uso de este CMS atienden a cuatro categorías de usuarios, donde puede haber uno o mas usuarios editores que publican documentos, usuarios registrados que pueden participar de los comentarios a las documentos y el resto del contenido, usuarios administradores con todos los privilegios reservados y por último el usuario huésped o anónimo de acceso muy limitado ^(ver Fig. 3-1).

2.1.6 Diseño de Bases de Datos.

El CMS cuenta con una base de datos sobre MySQL versión 5, que representa el corazón del sistema. En esta base de datos se almacena toda la información referente a los usuarios, los documentos, los menús, los bloques, las variables de entorno, las notas de actividades (logs) y todo un conjunto de información referida al diseño del sitio Web ^(ver Fig. 4-1).

³ Las opciones resaltadas son obligatorias el resto puede variar según lo desee el programador o el diseñador

Esta base de datos cuenta con 22 tablas para almacenar toda la información necesaria. En el Tabla 4-1 del Anexo 4 se muestra una descripción general de cada una de las tablas.

En el Anexo 5: se muestran los detalles de la definición de cada tabla, aspecto importante para la administración avanzada del CMS.

Por ser la tabla “variables” una tabla espacial por contener los valores de las variables que definen la apariencia, idioma, seguridad, entre otras explicare cada uno de sus posibles valores.

Esta tabla contiene 2 columnas y un número fijo de filas igual a 13, que representan las variables de entorno, a continuación se explican cada una de ellas.

- 1) **TITLE:** Título del sitio Webs. (*)
- 2) **CLASSIFICATION:** Clasificación del sitio web. (*)
- 3) **DESCRIPTION:** Descripción del sitio Web. (*)
- 4) **KEYWORDS:** Palabras claves para los buscadores. (*)
- 5) **LDAP_SERVER:**
IP del servidor de LDAP que autentificará a los usuarios en el intento de registrarse.
- 6) **DOCUMENT_PATH:**
Ruta en el disco local, absoluta o relativa a la raíz del sitio Web, donde se depositarán los recursos subidos al mismo.
- 7) **COUNTER:** Contador global de visitas.
- 8) **AUTHOR:** Autores del sitio, no el webmaster, sino los publicadores. (*)
- 9) **FOOT:** Texto para el pie de página.
- 10) **LANGUAGE:** Idioma en que está montado el sitio Web sobre el CMS.
- 11) **TEMPLATE:** Plantilla que defina la apariencia del sitio (solo el nombre).
- 12) **WEBMASTER:** Nombre del WebMaster.
- 13) **WEBMASTER_EMAIL:** Correo electrónico del WebMaster.

Todas las variables que fueron marcadas con (*) forman parte de la cabecera de página y constituyen metatags⁴.

⁴ Etiquetas HTML invisibles en una página formateada, pero utilizadas por los motores de búsqueda para fines de indexación.

2.2 Diseño del sitio Web principal de la facultad

Este sitio Web cuenta con los servicios de noticias, artículos, chat, foro, galería de imágenes, calendario, efemérides, manuales, libros digitales; y brinda información general de la facultad a través de las páginas de contenido.

Menús del sitio Web	
Título del menú	Contenido del menú (opciones)
Departamentos	Enlaces a los sitios Web de cada uno de los departamentos de la facultad.
Carreras	Enlaces a los sitios Web de cada una de las carreras de la facultad.
Enlaces	Enlaces a otros sitios Web de interés
Servicios	Servicio de noticias, artículos, Chat, foro, galería de imágenes, etc.
Utilitarios	Enlace a páginas o servicios más pequeño de marcada utilidad (tabla ASCII, calendario, calculadora, etc.)

2.3 Diseño del sitio Web para un departamento

Este sitio Web cuenta con los servicios de noticias, artículos, recursos, calendario y brinda información general del departamento a través de las páginas de contenido.

Menús del sitio Web	
Título del menú	Contenido del menú (opciones)
Contenido	Enlaces a las páginas de contenido que guardan información del departamento.
Enlaces	Enlaces a otros sitios Web de interés.
Servicios	Servicio de noticias, artículos y recursos

2.4 Diseño del sitio Web para una carrera

Esta compuesto por con los servicios de noticias, artículos, recursos, calendario y brinda información general de la carrera a través de las páginas de contenido.

Menús del sitio Web	
Título del menú	Contenido del menú (opciones)
Contenido	Enlaces a las páginas de contenido que guardan información de la carrera.
Enlaces	Enlaces a otros sitios Web de interés
Servicios	Servicio de noticias, artículos y recursos
Disciplinas	Enlaces al sitio Webs de cada una de las disciplinas.

2.5 Diseño del sitio Web para una disciplina

Cuenta con el servicios de manejo de recursos, calendario y brinda información general de la disciplina a través de las páginas de contenido.

Menús del sitio Web	
Título del menú	Contenido del menú (opciones)
Contenido	Enlaces a las páginas de contenido que guardan información de la disciplina.
Enlaces	Enlaces a otros sitios Web de interés.
Servicios	Manejo de recursos.
Asignaturas	Enlaces al sitio Webs de cada una le las asignatura.
Carreras	Enlace al resto de las carreras.

2.6 Diseño del sitio Web para una asignatura

Cuenta con el servicios de manejo de recursos, calendario y brinda información general de la asignatura a través de las páginas de contenido.

Este sitio Web contiene varios menús entre los que se encuentran:

Menús del sitio Web	
Título del menú	Contenido del menú (opciones)
Contenido	Enlaces a las páginas de contenido que guardan información de la asignatura.
Enlaces	Enlaces a otros sitios Web de interés.
Servicios	Manejo de recursos.
Conferencias	Enlaces a cada conferencia. Esta puede ser un recuso o una página de contenido.
Clases prácticas	Enlaces a cada clase practica. Esta puede ser un recuso o una página de contenido.
Archivos	Enlaces directos a los recursos divididos por categorías.

Capítulo 3: *Guía del navegador.*

Luego de que un cliente se conecte al sitio Web, este puede navegar por el mismo a través de los enlaces que conforman las opciones de cada menú ^(Ver Fig. 6-6 y 6-13), los que puedan aparecer en los bloques de información e incluso en el cuerpo de alguna página de contenido.

3.1 *Iniciar sesión.*

Un usuario puede acceder la página de inicio de sesión a través del enlace “Iniciar sesión” que aparece en el bloque de información “Datos de la sesión” ^(ver Fig. 6-1), el cual lo conduce al formulario de autenticación ^(ver Fig. 6-3) en el que debe introducir el nombre de usuario, contraseña y enviar los datos. Los cuales viajarán encriptados con el método OTP explicado en el primer capítulo. Si la autenticación es correcta el CMS se reconfigura con los nuevos datos de la sesión y los privilegios del usuario ahora activo; en caso de fallo permanece en el formulario de autenticación.

3.2 *Registrarse.*

Un usuario puede registrarse en el sitio Web a través del enlace “Registrarse” que aparece en el bloque de información “Datos de la sesión” ^(ver Fig. 6-1), el cual lo conduce al formulario de autenticación sobre el servidor de LDAP ^(ver Fig. 6-4), en el que debe introducir el nombre de usuario, contraseña, escoger un dominio y enviar los datos. Los cuales viajarán encriptados con el método OTP explicado en el primer capítulo. Si la autenticación es satisfactoria entonces pasa a un segundo formulario donde llenará los datos que el sistema no pudo obtener del usuario del dominio ^(ver Fig. 6-5) y se pondrá una nueva clave de acceso para el sitio; luego debe enviar los nuevos datos haciendo clic en el botón enviar, estos datos se encriptarán nuevamente y se enviarán al servidor el cual insertará o actualizará los datos del usuario según sea el caso.






3.3 *Cerrar Sesión.*

Luego de haber iniciado sesión, en el mismo bloque que usó para iniciar la sesión, aparecerán sus datos y un enlace para cerrar la sesión ^(ver Fig. 7-2).




3.4 Administración.


En cualquier módulo de administración aparece una pequeña menú o barra de tareas con las opciones que tiene el cliente sobre dicho módulo. Cada opción esta representada por una imagen que identifica la acción a realizar y constituye un enlace a la misma ^(ver Fig. 7-8).







Opciones que pueden aparecer en el menú:

- ↑ Sube elementos en el orden de visualización, los cuales pueden ser, un menú, una opción de menú o un bloque de información. Por ejemplo, el menú de conferencias esta sobre del menú de casos prácticos, si usted selecciona el menú de clases practicas y usa esta opción entonces este último pasará a estar por encima del menú de conferencias.
- ↓ Baja elementos en el orden de visualización.
-  Edita un registro. (un registro puede ser, una noticia, un artículo, un menú u opción de menú, bloque, etc.). Esta opción aparece en todos los módulos de administración. Por ejemplo si usted quiere editar una página de contenido o una opción de menú, basta acceder al módulos correspondiente en cada caso, seleccionar el registro deseado y hacer clic en esta opción en la barra de tareas, con lo cual pasará a un formulario que contiene los datos del registro seleccionado, podrá modificar lo que desee y hacer clic en continuar para actualizar los cambios o cancelar para retornar a la vista inicial del módulo.
-  Intenta eliminar registros. Te envía a una página de confirmación. Esta opción está presente en todos los módulos de administración excepto en el de administración de variables las que ya había dicho que tiene un número fijo de 13 variables.
-  Adiciona un nuevo registro. Esta opción también está presente en todos los módulos de administración excepto en el de administración de variables por lo antes expuesto.
-  Activa bloques de información u opciones de un menú. Talvez usted deseo en determinado momento que una opción del menú o un bloque de información, desapareciera temporalmente por alguna razón, y luego desea activarla nuevamente, en tal caso esta es la opción que debe utilizar.
-  Desactiva bloques de información u opciones de un menú. Suponga que usted creó un enlace a una página de contenido en determinado menú, pero hoy esta haciendo cambios en dicha página y como es lógico, no desea que ningún cliente acceda a ella hasta su completa actualización, una opción sería eliminar la opción del menú, pero para que eliminar lo que luego


tendrás que crear idénticamente. Esta opción te permite ocultar este enlace en el menú el cual podrás activar luego con la opción de activar.

-  Abre el módulo de administración de recursos. Existen dos modos de navegar por lo recurso, un modo de solo lectura y otro con privilegios administrativos. Esta opción te permita ir del modo uno al modo dos, permaneciendo en el directorio de trabajo actual.
-  Crea una nueva carpeta o directorio de recursos. Esta opción solo está disponible en el módulo de administración de recursos.
-  Subir un nivel en el directorio de recursos (ir al directorio padre).

 Subir un archivo de recurso al directorio activo. Evidentemente disponible solo en el módulo de administración de recursos. Además esta opción te permite introducir un resumen o descripción del documento a subir, resumen que se mostrará luego durante la exploración al pasar el puntero de mouse sobre el recurso.

-  Ir al directorio raíz en durante la exploración o administración de los recursos.
-  Renombrar un archivo o carpeta de recursos.
-  Comentar un archivo o carpeta de recursos. Esta opción te permite modificar la descripción o resumen del recurso seleccionado.
-  Cortar un archivo o carpeta de recursos.
-  Copiar un archivo o carpeta de recursos.
-  Pegar un recurso previamente cortado o copiado.

Las tres opciones anteriores permiten organizar los recursos ya subidos al servidor, según deseos y gustos del publicador.

-  Crear enlaces a los documentos seleccionados en algún menú. Muy útil esta opción tanto en los recursos como en la página de contenido. Por ejemplo suponga de un profesor quiere publicar una conferencia en un su asignatura.

El tiene dos opciones para crear del documento:

- 1). Puede crear una página de contenido.
- 2). Subir el documento creado previamente en cualquier formato (doc, rtf, doc, html, etc.), como un archivo recuso.

En cualquier caso, luego puede adicionar un enlace al documento en el menú de conferencias, usado la opción de crear enlace al documento en un menú

existente, la cual está disponible tanto en el módulo de administración de recurso como en el de páginas de contenido.

- ? Enlace con la página de ayuda. Cada módulo de administración tiene esta opción que no es más que un enlace al módulo de ayuda pasando como parámetro el identificador de módulo de administración del cual se desea la ayuda.

Debe quedar claro que estas opciones no aparecen en todos los módulos, sino solo en los que son necesarios y apuntan a la página que corresponda según sea el caso. Es decir, la opción de eliminar un registro no tiene el mismo enlace o url en el módulo de administración de recursos que en de administración de menús. Por otro lado la mayoría de estas opciones apuntan a páginas con un formulario de captura de datos y/o confirmación de ejecución, con dos nuevas opciones, continuar para efectuar la operación y cancelar para regresar a la vista principal del módulo o servicio. Si una acción es confirmada entonces aparece una nueva vista con los resultados de la operación (ejecución sin errores o con error) con la opción de regresar a la vista principal del módulo a través de un enlace o después de transcurrido un tiempo de 15 a 20 segundos.

Conclusiones

Habiendo terminado el estudio de los CMS, las ventajas y desventajas de PHP y MySQL y de la seguridad de un sitio Web, se pueden crear sistemas de manejo de contenido muy económicos, rápidos y capaces de adaptarse a las necesidades de cualquier centro.

En nuestro caso se creó un sencillo pero rápido y seguro CMS, sobre el cual se montó el sitio Web de la facultad, el de cada uno de los departamentos, carreras, disciplinas y asignaturas, en los que se podrá manejar toda la información, la documentación y la bibliografía digital correspondiente en cada caso. Además como todos son montados sobre el mismo CMS con bases de datos distintas, se logra un conjunto de sitios web homogéneo y se evita repetir el mismo código en sitios web que prácticamente prestan los mismos servicios.

Por otro lado al ser implementado en PHP y usar MySQL como servidor de base de datos, el CMS es instalable en cualquier sistema operativo en el que se pueda instalar estos dos software, que ya sabemos que son multiplataformas.

Recomendaciones

Incluir los temas de “Sistemas de Manejo de Contenido”, “Sistemas de Gestión del Aprendizaje” y “Sistemas de el Gestión de Eventos” en el programa de la asignatura de Sistema de Información, ya que son sistemas que integran o se pueden incorporar casi todas las técnicas que estudia dicha asignatura.

Incorporar al sistema otros servicios de interés para la facultad, como planificaciones automatizada y no estática de los diferentes horarios docentes y reservado de tiempos de maquina y de otros equipos útiles.

Anexo 1: PHP.



Fig. 1-1: Esquema del funcionamiento de las páginas PHP.

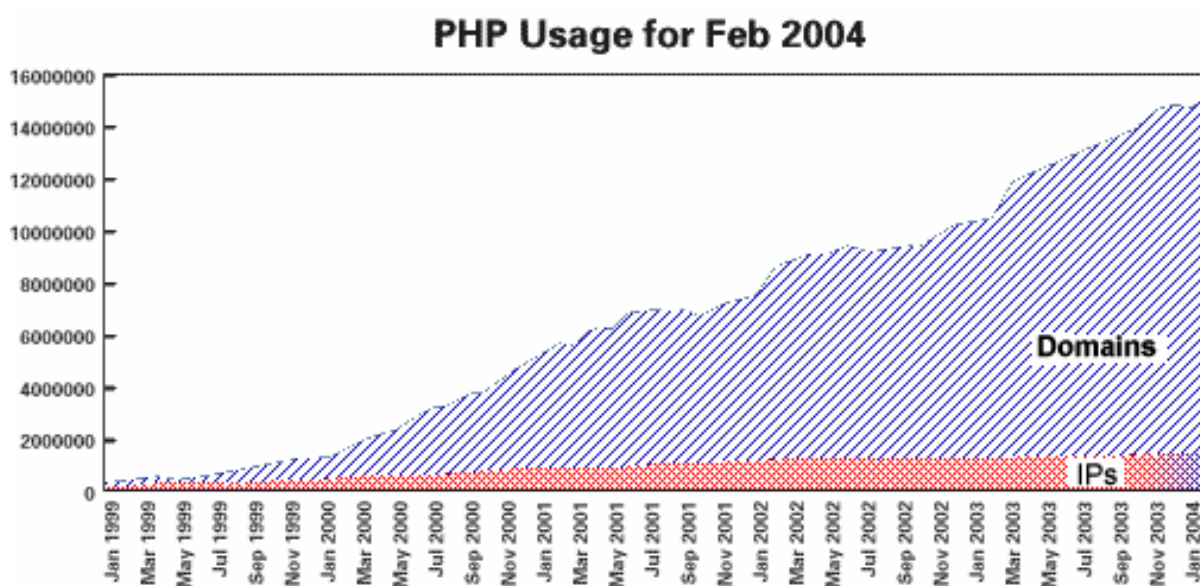


Fig. 1-2: Estadísticas de páginas PHP utilizadas entre 1999 y 2004.

Anexo 2: ARQUITECTURA DEL CMS.

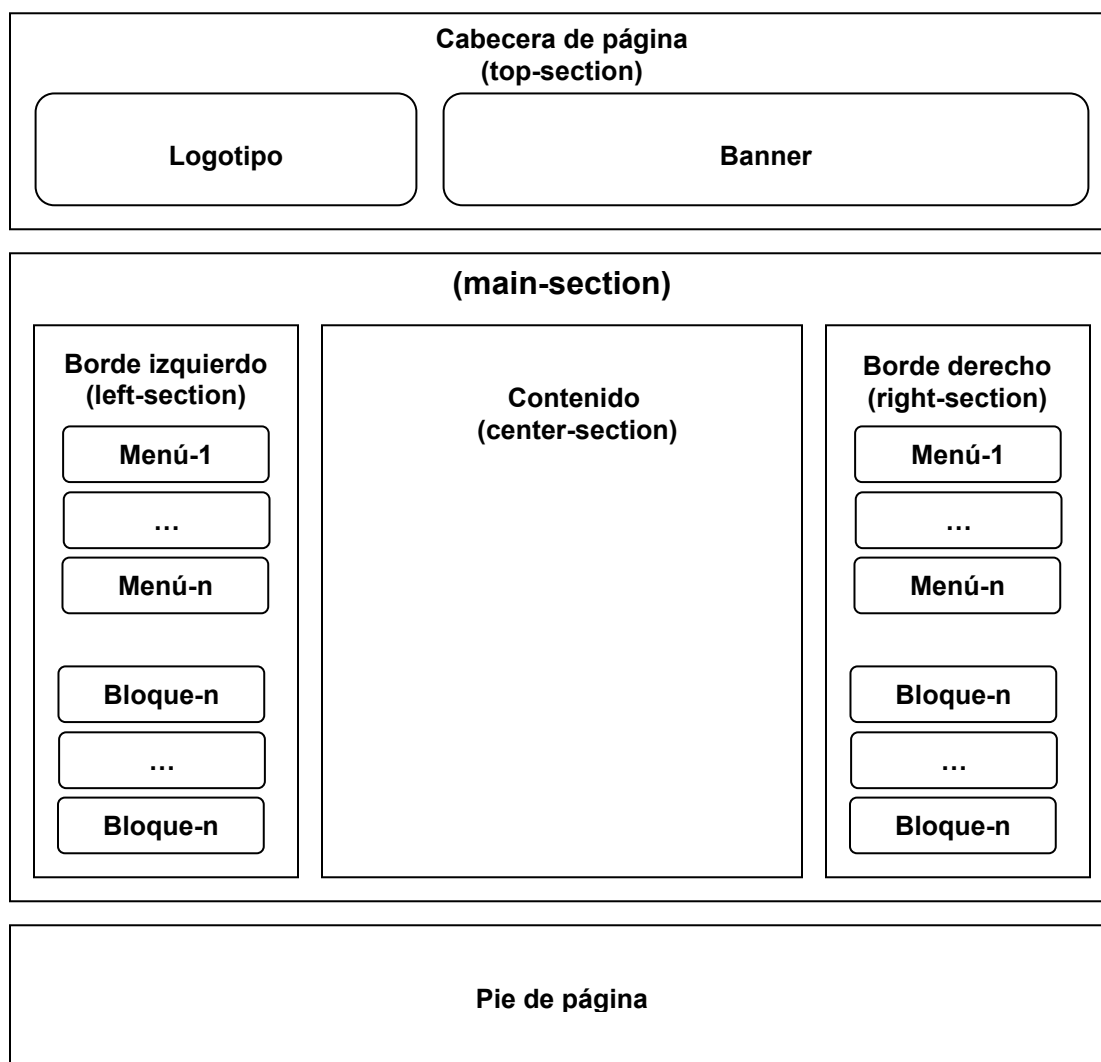


Fig. 2-1: División de una página en secciones.

Tabla 3-1: Descripción de las funciones presentes en una plantilla

Función	Parámetros	Descripción
IMG_Http_Path	\$img	Busca la imagen especificada como parámetro en el directorio "imágenes" de la plantilla y si no existe la busca en la del CMS. Si la imagen es encontrada retorna la ruta a la misma en la forma http://dns/ruta, sino retorna falso.
pcss		Define la hoja de estilo css.
BBody		Inicia cuerpo de la página (Ej: abre el tag body)
CBody		Cierra el cuerpo de la página.
BTopSection		Abre la sección superior de la página. (cabecera)

Tabla 3-1: Descripción de las funciones presentes en una plantilla

Función	Parámetros	Descripción
BLeftTopSection		Abre la sección izquierda dentro de la cabecera (sección del logotipo).
CLeftTopSection		Cierra la sección del logotipo.
BRightTopSection		Abre la sección derecha dentro de la cabecera (sección del banner).
CRightTopSection		Cierra la sección del banner.
CTopSection		Cierra la sección superior de la página.
SetLogo	\$logo	Imprime el logotipo especificado por \$logo.
BToolBar		Abre una sección de menú tipo barra de tareas.
SetItemToolBar	\$title \$url	Imprime una opción en la barra de tareas. \$title → Texto a mostrar \$url → Enlace
CToolBar		Cierra la sección de la barra de tareas
BBlock	\$title	Abre un bloque de información encabezado por un título.
BItemBlock		Abre una sección para un elemento de un bloque.
CItemBlock		Cierra la sección para el elemento del bloque.
CBlock		Cierra el bloque de información.
BMainSection		Abre la sección principal de una página.
BLeftSection		Abre la sección izquierda de una página. (borde izquierdo)
CLeftSection		Cierra la sección izquierda de una página.
BCenterSection		Abre la sección central o de contenido.
CCenterSection		Cierra la sección central
BRightSection		Abre la sección derecha de una página. (borde derecho)
CRightSection		Cierra la sección derecha de una página.
BMenu	\$title \$img = null	Abre una sección de menú lateral. \$title → Título del menú. \$img → Imagen del menú (opcional).
ItemMenu	\$title \$url \$img = null \$hint = null \$target=null	Imprime una opción en el menú lateral. \$title → Título de la opción. \$url → Enlace a otra página o sitio Web. \$img → Imagen de la opción (opcional). \$hint → Leve indicio, comentario. \$target → Define la ventana en la que se abrirá la nueva página.
CMenu		Cierra la sección del menú lateral.
CMainSection		Cierra la sección principal.
SetTitle SetRim SetHead	\$valor \$img = null	Imprimen el nombre, el título o cabecera de alguna caja o bloque de información, por ejemplo el título de una noticia, un artículo, una página de contenido, un menú, etc. \$valor → Texto a mostrar. \$img → Imagen (opcional)
SetBBody		Abre una sección para encerrar a algún tipo de documento o información.
SetBody	\$valor	Esta es una función para comodidad. La misma tiene tres pasos: 1. Llamar a SetBBody() 2. Imprime la variable \$valor 3. Llamar a SetCBody()
SetCBody		Cierra la sección abierta con SetBBody().

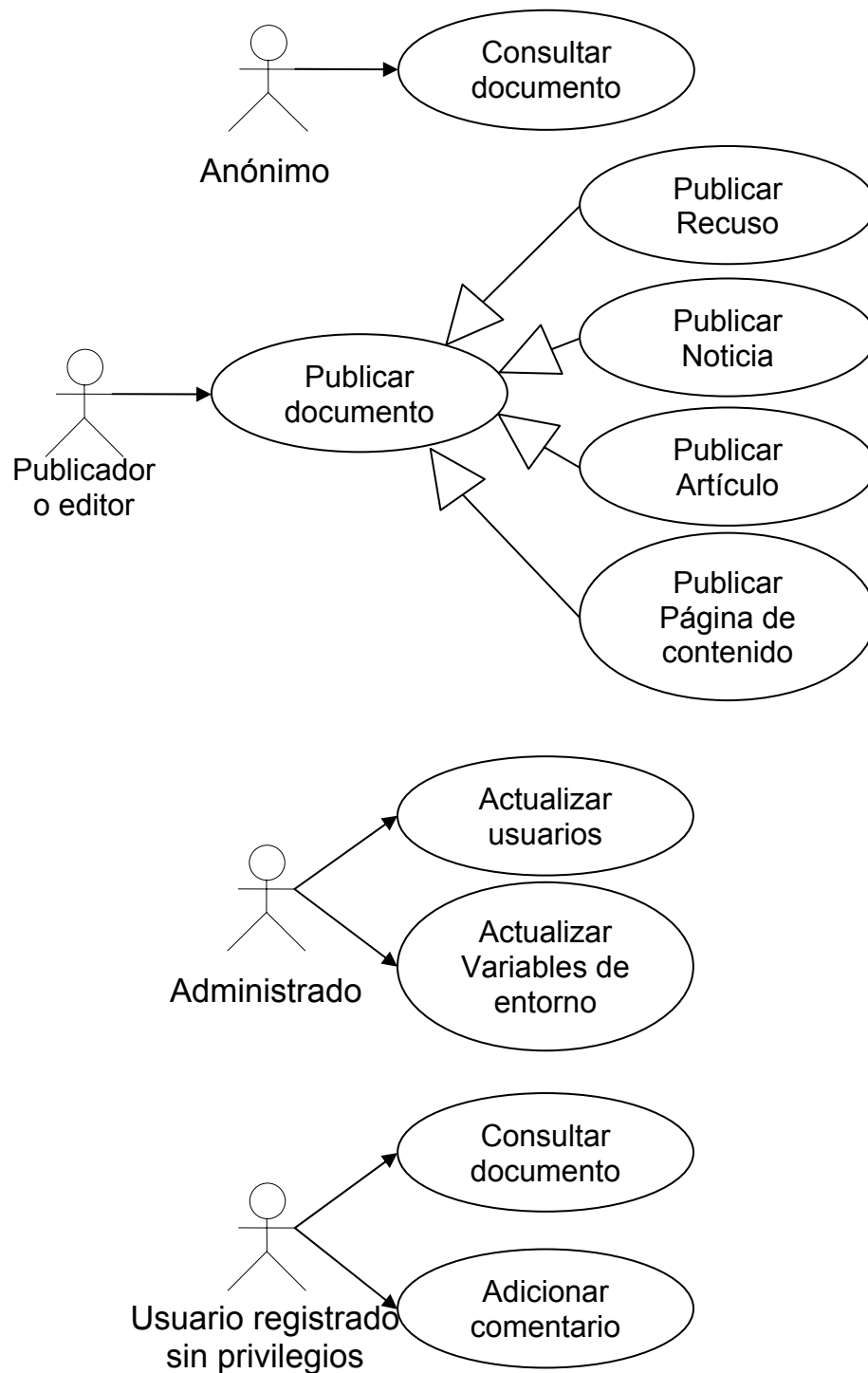
Tabla 3-1: Descripción de las funciones presentes en una plantilla

Función	Parámetros	Descripción
SetFoot	\$valor	Imprime cierta información especificada en la variable \$valor. Se utiliza para imprimir al pie de una caja, tabla o bloque de información.
BFecha		Abre una sección para el bloque de fecha.
SetFecha	\$date	Imprime la fecha especificada en la variable \$date.
CFecha		Cierra la sección del bloque de fecha.
Error	\$error_msg	Imprimen el mensaje de error especificado por la variable \$error_msg.
modulo_head	\$title \$pg_help \$img = null	Imprime el título o cabecera de una página. Es similar a SetTitle o SetHead, con la diferencia de que incluye un índice para crear un enlace a la página de ayuda.
OpenTable	\$style = null \$class='bdtbale'	Abre una tabla. Cada parámetro especifica el atributo de igual nombre en el tag abierto.
CloseTable		Cierra una tabla.
OpenTR	\$style = null \$class='bdtr'	Abre fila. Cada parámetro especifica el atributo de igual nombre en el tag abierto.
CloseTR		Cierra una fila.
OpenTH	\$width=null \$nowrap=null \$colspan=null \$rowspan=null \$style=null \$class = 'bdth'	Abre una celda cabecera. Cada parámetro especifica el atributo de igual nombre en el tag abierto.
SetTH	\$inneHTML \$width=null \$nowrap=null \$colspan=null \$rolspan=null \$style=null \$class = 'bdth'	Imprime una celda de encabezado. Llama a la función OpenTH, imprime el valor de la variable \$inneHTML y llama a la función CloseTH.
CloseTH		Cierra la celda de cabecera.
OpenTD	\$width=null \$nowrap=null \$colspan=null \$rowspan=null \$style=null \$class='bdtd'	Abre una celda. Cada parámetro especifica el atributo de igual nombre en el tag abierto.
SetTD	\$inneHTML \$width=null \$nowrap=null \$colspan=null \$rolspan=null \$style=null \$class='bdtd'	Imprime una celda. Llama a la función OpenTD, imprime el valor de la variable \$inneHTML y llama a la función CloseTD.
CloseTD		Cierra la celda.
toolbar_button	\$onclick \$img \$title	Crea un botón con una imagen. \$onclick → Acción a realizar. \$img → Imagen a Mostrar. \$title → Leve indicio, comentario

Tabla 3-1: Descripción de las funciones presentes en una plantilla

Función	Parámetros	Descripción
submit_bar	\$url_cancel	Imprime un botón de envío de formulario y uno para cancelar y regresar a la página especificada en la variable \$url_cancel
go_back	\$lapse \$url \$title	Imprime un enlace y una redirección automática en un periodo de tiempo, a dirección. \$lapse → Tiempo de espera. \$url → Enlace destino. \$title → Texto a mostrar. Se usa principalmente para retornar a un página luego de determinada acción.
db_pages	\$row_count \$row_limit \$url \$actual_pg	Imprime un conjunto de números que representan enlaces a las páginas en que fue dividido de alguna consulta. Por ejemplo al acceder al módulo de administración de las “páginas de contenido” se muestra un listado con de k páginas, donde k es le número máximo de registros a mostrar. Luego aparece el listado con las páginas que contienen el resto de los registros. \$row_count → Total de registros. \$row_limit → Número máximo a mostrar. \$url → Enlace a la página de trabajo. \$actual_pg → Número de la página actual.
SetTheNews SetArticle SetInquiry SetUser	\$row \$full = null	Imprime una noticia, un Artículo, una encuesta o los datos de un usuario respectivamente. \$row → Fila de resultado de una consulta a la tabla correspondiente según la información solicitada. \$full → (parámetro opcional). Si su valor es “true” entonces se imprime todo el documento y su valor e “null” se imprime solo los datos preliminares.
SetComment	\$row	Imprime un comentario. \$row → Fila de resultado de una consulta a la tabla de comentarios correspondiente según la información solicitada.
SetVote	\$row	Imprime información referente a los votos de una encuesta. \$row → Fila de resultado de una consulta a la tabla de votos.

Tabla 2-1: Descripción de las funciones presentes en una plantilla.

Anexo 3: CASOS DE USO DEL SISTEMA.**Fig. 3-1: Diagrama de casos de uso del sistema.**

Anexo 4: DISEÑO DE LA BASE DE DATOS.

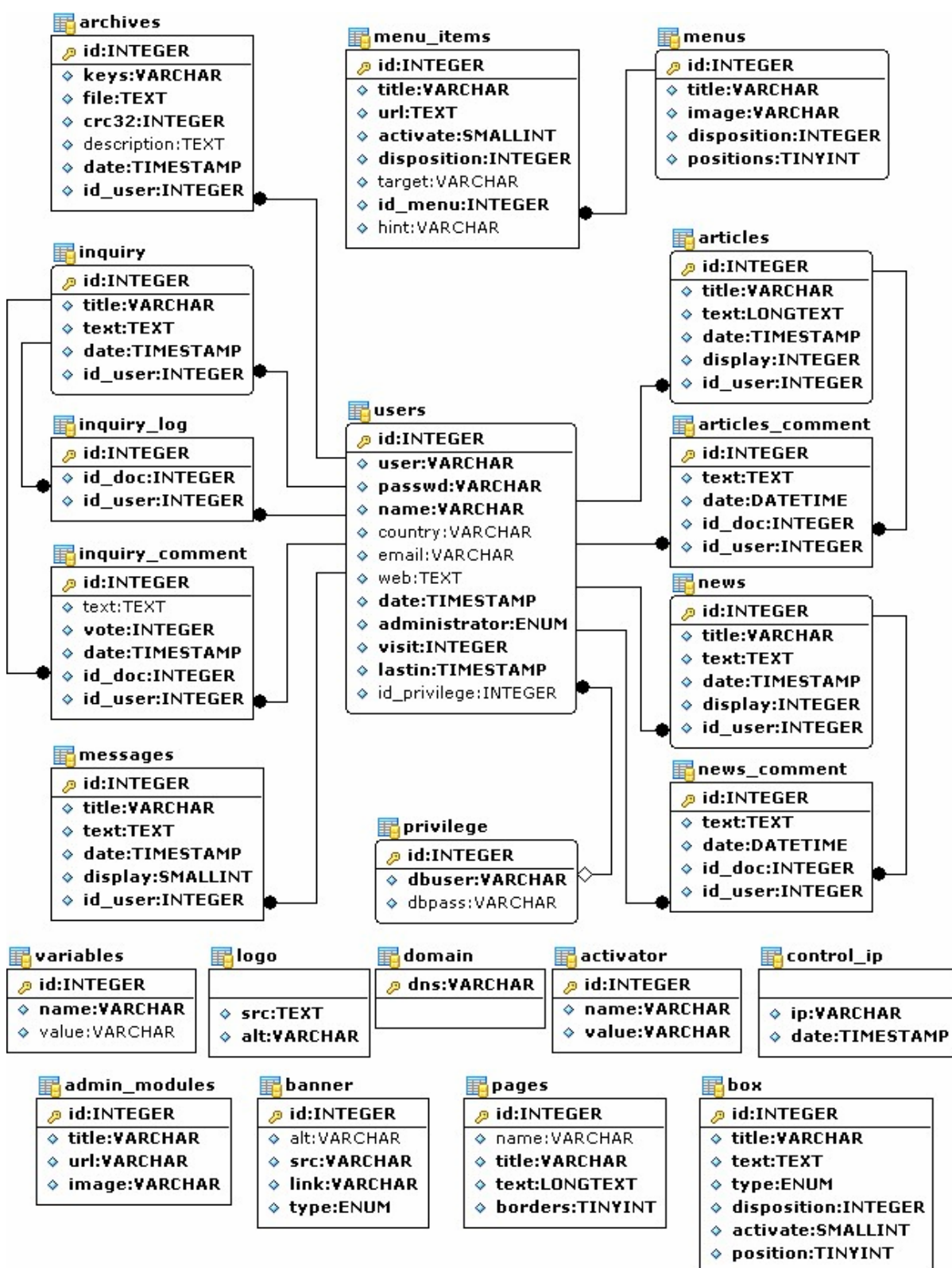


Fig. 4-1: Diseño de la base de datos.

Tabla 4-1: Descripción de las tablas de la base de datos

Tabla	Tipo	Col.	Descripción
admin_modules	MyISAM	4	Contiene información y la referencia a los módulos de administración activos en el sitio Webs.
archives	InnoDB	7	Contiene información sobre los recursos que han sido subidos al servidor.
articles	InnoDB	6	Contiene los artículos publicados.
articles_comment	InnoDB	5	Contiene los comentarios hechos a cada artículo.
banner	MyISAM	5	Contiene el listado de banners que se mostraran de forma aleatoria en la cabecera de página.
box	MyISAM	7	Contiene los bloques de información que se mostraran en los laterales las páginas.
control_ip	MyISAM	2	Contiene el listado de IP y fecha de acceso de los clientes al sitio Web.
domain	MyISAM	1	Contiene el listado de dominios a través de los cuales un usuario podrá registrarse.
inquiry	InnoDB	5	Contiene las encuestas.
inquiry_comment	InnoDB	6	Contiene los comentarios de cada encuesta.
inquiry_log	InnoDB	3	Contiene el control de votos de un usuario en cada encuesta.
logo	MyISAM	2	Contiene el logotipo del sitio Webs.
menu_items	InnoDB	8	Contiene las opciones de los menús.
menus	InnoDB	5	Contiene información sobre cada menú.
messages	InnoDB	6	Contiene los mensajes.
news	InnoDB	6	Contiene las noticias.
news_comment	InnoDB	5	Contiene los comentarios a noticias.
pages	MyISAM	5	Contiene las páginas de contenido.
privilege	InnoDB	3	Contiene un listado de usuarios y contraseñas del MySQL, que definen los permisos a las diferentes tablas.
users	InnoDB	12	Contiene el listado de usuario del sitio Webs.
variables	InnoDB	3	Contiene los valores de las variables de entrono del CMS.

Tabla 4-1: Descripción del las tablas de la base de datos.

Anexo 5: DESCRIPCIÓN DETALLADA DE LAS DE LAS TABLAS.

Tabla (activator):

Fields

PK	Name	Data type	Size	Auto Increment	Not null
✓	id	INTEGER	11	✓	✓
	name	VARCHAR	255		✓
	value	VARCHAR	255		✓

Indices

Name	Fields	Unique	Collation	Full Text
PRIMARY	id	✓	Ascending	

Tabla (articles):

Fields

P K	Name	Data type	Size	Default	Auto Increment	Not null
✓	id	INTEGER	11		✓	✓
	title	VARCHAR	255			✓
	text	LONGTEXT	16777215			✓
	date	TIMESTAMP	19	CURRENT TIMESTAMP		✓
	display	INTEGER	11	0		✓
	id_user	INTEGER	11	0		✓

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending
id_user	id_user		Ascending

Foreign Keys

Fields	Foreign Table	Foreign Fields	Update Action	Delete Action
`id_user`	users	`id`	Restrict	Restrict

Tabla (admin_modules):

Fields

PK	Name	Data type	Size	Auto Increment	Not null
✓	id	INTEGER	11	✓	✓
	title	VARCHAR	255		✓
	url	VARCHAR	255		✓
	image	VARCHAR	255		✓

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending

Tabla (archives):

Fields

PK	Name	Data type	Size	Default	Auto Increment	Not null	Unique
✓	id	INTEGER	16		✓	✓	
	keys	VARCHAR	255			✓	
	file	TEXT	65535			✓	
	crc32	INTEGER	32	0		✓	✓
	description	TEXT	65535				
	date	TIMESTAMP	19	CURRENT TIMESTAMP		✓	
	id_user	INTEGER	11			✓	

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending
crc32	crc32	✓	Ascending
id_user	id_user		Ascending

Foreign Keys

Fields	Foreign Table	Foreign Fields	Update Action	Delete Action
`id_user`	users	`id`	Restrict	Cascade

Tabla (articles_comment):

Fields

PK	Name	Data type	Size	Auto Increment	Not null
✓	Id	INTEGER	11	✓	✓
	Text	TEXT	65535		✓
	Date	DATETIME	19		✓
	id_doc	INTEGER	11		✓
	id_user	INTEGER	11		✓

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending
id_noticia	id_doc		Ascending
id_usuario	id_user		Ascending

Foreign Keys

Fields	Foreign Table	Foreign Fields	Update Action	Delete Action
`id_doc`	articles	`id`	Restrict	Cascade
`id_user`	users	`id`	Restrict	Restrict

Tabla (banner):

Fields

PK	Name	Data type	Size	Values	Default	Auto Increment	Not null
✓	Id	INTEGER	11			✓	✓
	alt	VARCHAR	255				
	src	VARCHAR	255				✓
	link	VARCHAR	255				✓
	type	ENUM	5	('image','swf')	image		✓

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending

Tabla (box):**Fields**

PK	Name	Data type	Size	Values	Default	Auto Increment	Not null
✓	id	INTEGER	11			✓	✓
	title	VARCHAR	255				✓
	text	TEXT	65535				✓
	type	ENUM	4	('text','php')	text		✓
	dispositio n	INTEGER	11		0		✓
	activate	SMALLINT	1		0		✓
	position	TINYINT	1		0		✓

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending

Tabla (control_ip):**Fields**

P K	Na me	Data type	Size	Default	Not null
	ip	VARCHAR	15		✓
	date	TIMESTAMP	19	CURRENT TIMESTAMP	✓

Indices

Name	Fields	Collation
ip	ip	Ascending

Tabla (domain):**Fields**

P K	Na me	Data type	Size	Not null
✓	dns	VARCHA R	100	✓

Indices

Name	Fields	Unique	Collation
PRIMARY	dns	✓	Ascending

Tabla (inquiry):**Fields**

PK	Name	Data type	Size	Default	Auto Increment	Not null
✓	id	INTEGER	11		✓	✓
	title	VARCHAR	255			✓
	text	TEXT	65535			✓
	date	TIMESTAMP	19	CURRENT TIMESTAMP		✓
	id_user	INTEGER	11	0		✓

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending
id_user	id_user		Ascending

Foreign Keys

Fields	Foreign Table	Foreign Fields	Update Action	Delete Action
`id_user`	users	`id`	Restrict	Restrict

Tabla (inquiry_comment):**Fields**

PK	Name	Data type	Size	Default	Auto Increment	Not null
✓	id	INTEGER	11		✓	✓
	text	TEXT	65535			
	vote	INTEGER	11	0		✓
	date	TIMESTAMP	19	CURRENT TIMESTAMP		✓
	id_doc	INTEGER	11	0		✓
	id_user	INTEGER	11	0		✓

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending
id_tema	id_doc		Ascending
id_usuario	id_user		Ascending

Foreign Keys

Fields	Foreign Table	Foreign Fields	Update Action	Delete Action
`id_user`	users	`id`	Restrict	Restrict
`id_doc`	inquiry	`id`	Restrict	Cascade

Tabla (inquiry_log):

Fields

PK	Name	Data type	Size	Auto Increment	Not null
✓	id	INTEGER	11	✓	✓
	id_doc	INTEGER	11		✓
	id_user	INTEGER	11		✓

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending
id_vote	id_doc, id_user	✓	Ascending
id_doc	id_doc		Ascending
id_user	id_user		Ascending

Foreign Keys

Fields	Foreign Table	Foreign Fields	Update Action	Delete Action
`id_doc`	inquiry	`id`	Restrict	Cascade
`id_user`	users	`id`	Restrict	Cascade

Tabla (logo):

Fields

P K	Name	Data type	Size	Not null
	src	TEXT	65535	✓
	alt	VARCHAR	100	✓

Tabla (menus):

Fields

PK	Name	Data type	Size	Auto Increment	Not null
✓	id	INTEGER	11	✓	✓
	title	VARCHAR	20		✓
	image	VARCHAR	255		✓
	disposition	INTEGER	11		✓
	positions	TINYINT	4		✓

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending

Tabla (menu_items):

Fields

PK	Name	Data type	Size	Auto Increment	Not null
✓	id	INTEGER	11	✓	✓
	title	VARCHAR	255		✓
	url	TEXT	65535		✓
	activate	SMALLINT	1		✓
	disposition	INTEGER	11		✓
	target	VARCHAR	10		
	id_menu	INTEGER	11		✓
	hint	VARCHAR	255		

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending
id_menu	id_menu		Ascending

Foreign Keys

Fields	Foreign Table	Foreign Fields	Update Action	Delete Action
`id_menu`	menus	`id`	Restrict	Restrict

Tabla (messages):

Fields

PK	Name	Data type	Size	Default	Auto Increment	Not null
✓	id	INTEGER	11		✓	✓
	title	VARCHAR	255			✓
	text	TEXT	65535			✓
	date	TIMESTAMP	19	CURRENT TIMESTAMP		✓
	display	SMALLINT	3	0		✓
	id_user	INTEGER	11			✓

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending
id_user	id_user		Ascending

Foreign Keys

Fields	Foreign Table	Foreign Fields	Update Action	Delete Action
`id_user`	users	`id`	Restrict	Restrict

Tabla (news):**Fields**

PK	Name	Data type	Size	Default	Auto Increment	Not null
✓	id	INTEGER	11		✓	✓
	title	VARCHAR	255			✓
	text	TEXT	65535			✓
	date	TIMESTAMP	19	CURRENT TIMESTAMP		✓
	display	INTEGER	11	0		✓
	id_user	INTEGER	11	0		✓

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending
id_usuario	id_user		Ascending

Foreign Keys

Fields	Foreign Table	Foreign Fields	Update Action	Delete Action
`id_user`	users	`id`	Restrict	Restrict

Tabla (news_comment):**Fields**

PK	Name	Data type	Size	Auto Increment	Not null
✓	id	INTEGER	11	✓	✓
	text	TEXT	65535		✓
	date	DATETIME	19		✓
	id_doc	INTEGER	11		✓
	id_user	INTEGER	11		✓

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending
id_doc	id_doc		Ascending
id_user	id_user		Ascending

Foreign Keys

Fields	Foreign Table	Foreign Fields	Update Action	Delete Action
`id_doc`	news	`id`	Restrict	Cascade
`id_user`	users	`id`	Restrict	Restrict

Tabla (pages):**Fields**

PK	Name	Data type	Size	Auto Increment	Not null
✓	id	INTEGER	11	✓	✓
	name	VARCHAR	20		
	title	VARCHAR	255		✓
	text	LONGTEXT	16777215		✓
	borders	TINYINT	4		✓

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending

Tabla (variables):**Fields**

PK	Name	Data type	Size	Auto Increment	Not null
✓	id	INTEGER	11	✓	✓
	name	VARCHAR	20		✓
	value	VARCHAR	255		

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending
nombre	name	✓	Ascending

Tabla (privilege):**Fields**

PK	Name	Data type	Size	Auto Increment	Not null
✓	id	INTEGER	11	✓	✓
	dbuser	VARCHAR	16		✓
	dbpass	VARCHAR	100		

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending
dbuser	dbuser	✓	Ascending

Tabla (users):**Fields**

PK	Name	Data type	Size	Values	Default	Auto Increment	Not null
✓	id	INTEGER	11			✓	✓
	user	VARCHAR	20				✓
	passwd	VARCHAR	50				✓
	name	VARCHAR	60				✓
	country	VARCHAR	50				
	email	VARCHAR	100				
	web	TEXT	65535				
	date	TIMESTAMP	19		CURRENT TIMESTAMP		✓
	administrator	ENUM	1	('0','1')	0		✓
	visit	INTEGER	11		0		✓
	lastin	TIMESTAMP	19		CURRENT TIMESTAMP		✓
	id_privilege	INTEGER	11				

Indices

Name	Fields	Unique	Collation
PRIMARY	id	✓	Ascending
usuario	user	✓	Ascending
nombre	user	✓	Ascending
user	user	✓	Ascending
id_privilege	id_privilege		Ascending

Foreign Keys

Fields	Foreign Table	Foreign Fields	Update Action	Delete Action
`id_privilege`	privilege	`id`	Restrict	Restrict

Anexo 6: GUÍA DEL NAVEGADOR.

Datos de la sesión
Registrados: 1 Activos: 19
Usuario: ANONYMOUS Iniciar sesión Registrarse

Fig. 6-1: Bloque “Datos le sesión”.

Cerrar sesión
Datos de la sesión
Yoandry Pacheco Águila Usuario: yoandryp Correo: a@b País: cuba

Fig. 6-2: Bloque “Cerrar sesión”.

AUTENTIFICACION
Usted intenta acceder a una zona restringida, disponible solo para usuarios registrados autorizados.
Usuario: <input type="text"/>
Contraseña: <input type="password"/>
Ingresar
Registrar nuevo usuario

Fig. 6-3: Formulario de autenticación.

REGISTRAR NUEVO USUARIO	
AUTENTIFICACION	
Usuario:	<input type="text"/>
Contraseña:	<input type="password"/>
Dominio:	<input type="text" value="uclv.edu.cu"/>
<input type="button" value="Ingresar"/>	

Fig. 6-4: Formulario para registrar o actualizar un usuario.

REGISTRAR NUEVO USUARIO	
Datos del usuario	
Campo	Valor
Usuario	andypa
Nombre	Yoandry Pacheco Águila
Correo	<input type="text"/>
País	<input type="text"/>
Dirección web	<input type="text"/>
Contraseña para autenticarse en este sitio.	
Contraseña	<input type="password"/>
<input type="button" value="Continuar"/> <input type="button" value="Cancelar"/>	

Fig. 6-5: Formulario para ingresar o modificar datos de un usuario.

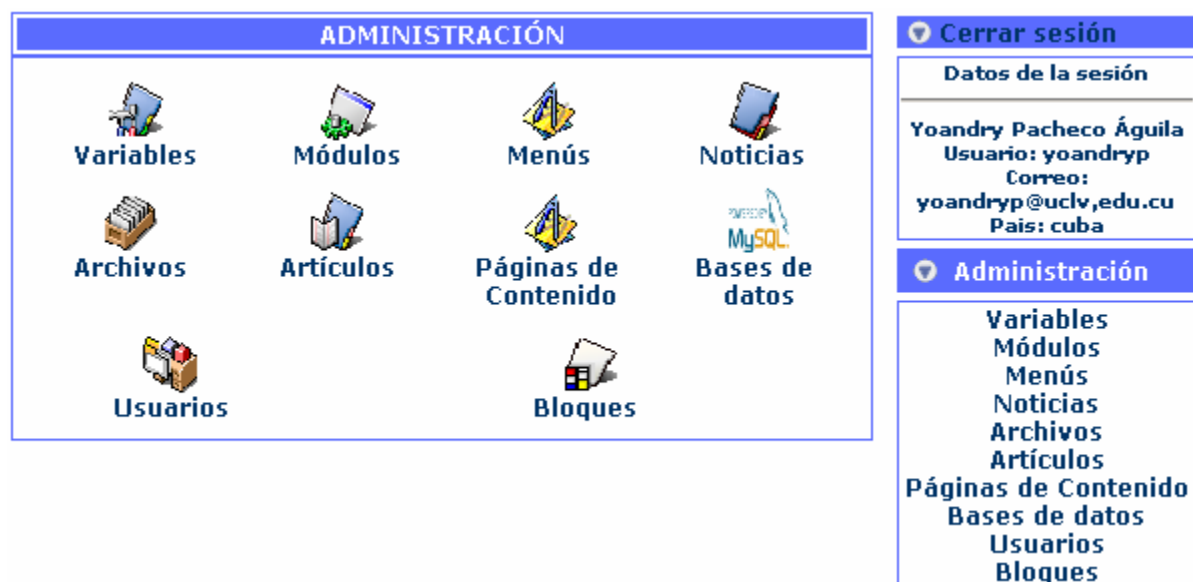


Fig. 6-6: Vista de la página principal del módulo de administración.



Fig. 6-7: Corte de la página inicial para de administración de variables.

Modificación de registros

TITLE

Facultad de Matemática Física y Computación

Continuar Cancelar

Fig. 6-8: Corte de la edición de variables.

	Nombre	Byte	Última Modificación
<input type="checkbox"/>	Book	[0]	15/06/2005 09:40:57 AM
<input type="checkbox"/>	Manuales-Tutoriales	[0]	25/05/2005 11:12:34 AM

Fig. 6-9: Administración de recursos.

SUBIR UN ARCHIVO

Campo	Valor
Archivo	<input type="text"/> Examinar...
Palabras claves	<input type="text"/>
Resumen del contenido	<input type="text"/>

Continuar Cancelar

Fig. 6-10: Formulario para subir un archivo.

NUEVA CARPETA	
/ ?	
Campo	Valor
Carpeta	Nueva carpeta
<input type="button" value="Continuar"/> <input type="button" value="Cancelar"/>	

Fig. 6-11: Formulario para crear una nueva carpeta.

Adición de un nuevo registro ?	
Campo	Valor
Título	Book
Detalle	
Enlace	/admin_archivos.php?vpath=Book
Ventana	_blank
MENÚ	SERVICIOS
<input type="button" value="Continuar"/> <input type="button" value="Cancelar"/>	

Fig. 6-12: Formulario para adicional enlace a un documento en uno de los menús.

↑ ↓ ?

	Imagen	Título	Izquierdo	Tope	Derecho
<input type="checkbox"/>		DEPARTAMENTOS	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		BARRA DE TAREAS	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		CARRERAS	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		ENLACES	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		UTILITARIOS	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>		SERVICIOS	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Pais: cuba

○ MENÚES

? DEPARTAMENTOS

BARRA DE TAREAS

? CARRERAS

ENLACES

? UTILITARIOS

SERVICIOS

○ Administración

Fig. 6-13: Administración de menús.

↑ ↓ ?

	Imagen	Título	Izquierdo	Tope	Derecho
<input type="checkbox"/>		DEPARTAMENTOS	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		BARRA DE TAREAS	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		CARRERAS	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		ENLACES	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>		UTILITARIOS	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>		SERVICIOS	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Pais: cuba

○ MENÚES

? DEPARTAMENTOS

BARRA DE TAREAS

? CARRERAS

ENLACES

? UTILITARIOS

SERVICIOS

○ Administración

Fig. 6-14: Administración de opciones de un menú.

Modificación de registros ?

Título	CHAT	
Nombre	pgchat	
Tope	<input checked="" type="checkbox"/>	
Izquierdo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Style] [Font] [Size]

B *I* U |

developing flash chat 1.6.5

Antes de entrar, lee las siguientes indicaciones.

Si hay alguien en la sala con el nombre que has escogido, te será añadida una extensión numérica; por ejemplo:

Si entras con el nombre de 'Carlos' y hay ya alguien llamado 'Carlos' en la sala, tu nombre será 'Carlos1'.

☐ View Source

Continuar Cancelar

Fig. 6-15: Edición de una página de contenido.

Glosario

Apache - servidor de páginas Web de código abierto para diferentes plataformas (UNIX, Windows, etc.)

Blog (Web log) - Diario en formato Web. Puede ser un diario personal o un conjunto de noticias, ordenado por fecha.

CMF (Content Management Framework) - Entorno a programación de aplicaciones enfocado al desarrollo de CMS.

CMS (Content Management System) - Sistema que facilita la gestión de contenidos en todos sus aspectos: creación, mantenimiento, publicación y presentación. También se conoce como Web Content Management (WCM) sistema de gestión de contenido de webs.

E-Learning -Sistemas de gestión del aprendizaje (Learning Management Systems).

GPL (General Public License) - Licencia que permite el uso y modificación del código para desarrollar software libre, pero no propietario.

CSS (Cascading Style Sheets) - Las hojas de estilo en cascada contienen un conjunto de etiquetas que definen el formato que se aplicará al contenido de las páginas de una Web. Se llaman "en cascada" porque una hoja puede heredar los formatos definidos en otra hoja de forma que no hace falta que vuelva a definirlos. Estas hojas permiten la separación entre el contenido y la presentación en una Web.

HTML (HyperText Markup Language) - Lenguaje basado en marcas que indican las características del texto, utilizado para definir documentos de hipertexto en webs.

HTTP (HyperText Transfer Protocol) - Protocolo cliente-servidor utilizado para el intercambio de páginas Web (HTML)

LAMP (Linux, Apache, MySQL y PHP, Perl o Python) - Arquitectura formada por el sistema operativo Linux, el servidor Web Apache, la base de datos MySQL y uno o más de los lenguajes de programación PHP, Perl o Python.

LCMS (Learning Content Management System) - Software para la gestión automatizada de cursos en línea, que incluye gestión de usuarios, de resultados y de recursos. Es un sistema de gestión de cursos con las capacidades de un CMS y por lo tanto de gestionar también los contenidos de los recursos.

LGPL (Lesser General Public License) - Licencia que permite el uso y modificación de librerías de código para desarrollar software libre o propietario. Antes conocida como Library GPL.

LMS (Learning Management System) - se diferencia de los LCMS en que no hay gestión de los contenidos, sino simplemente administración del curso, pero acostumbra a utilizarse como sinónimo. También conocido como Course Management System (CMS) o Virtual Learning Environment (VLE)

Metadatos - datos sobre los datos. Información que describe el contenido de los datos. Por ejemplo de un documento serían metadatos, entre de otros, su título, el nombre del autor, la fecha de creación y modificación, y un conjunto de palabras clave que identifiquen su contenido.

Metatags - Son etiquetas de código HTML que contienen palabras clave por las que tu página Web será encontrada. Los metatags son muy importantes a la hora de posicionar un sitio Web, ya que los buscadores ordenan las páginas de acuerdo a los metatags que van encontrando. Se colocan en el elemento <HEAD> del código HTML, y los más utilizados son el del título, descripción y keywords.

MySQL - base de datos relacional multiplataforma de código abierto, muy popular en aplicaciones Web.

Open source - Código abierto o código libre. Software que distribuye de forma libre su código fuente, de forma que los desarrolladores pueden hacer variaciones, mejoras o reutilizarlo en otras aplicaciones. También conocido como free software.

Perl - lenguaje de programación de alto nivel que hereda de diversos lenguajes, muy utilizado para el desarrollo de webs dinámicas.

PHP (*PHP Hypertext Preprocessor*) - Lenguaje de programación para el desarrollo de webs dinámicas, con sintaxis parecida a la C. Originalmente se conocía como Personal Home Page tools, herramientas para páginas personales (en Internet).

Python - lenguaje interpretado de alto nivel orientado a objetos.

URL (*Uniform Resource Locator*) - Dirección de un recurso en la Web. Tiene el formato protocolo://máquina.domini:port/ruta/recurs.

WAI (*Web Accessibility Initiative*) - Iniciativa del Consorcio de la World Wide Web para asegurar que las webs están diseñadas pensando en el acceso de personas con discapacidades.

Web - sistema para presentar información en Internet basado en hipertexto. Cuando se utiliza en masculino (el Web, un Web) se refiere a un sitio Web entero, en cambio si se utiliza en femenino (la Web, una Web) se refiere a una página Web concreta dentro del sitio Web.

WebDAV (*Web-based Distributed Authoring and Versioning*) - es una extensión del protocolo HTTP que permite a los usuarios editar y administrar ficheros de forma colaborativa en servidores Web remotos.

Web log - Ver 'Blog'.

WYSIWYG (*What You See Is What You Get*) - Traducido: lo que ves es lo que obtienes, que aplicado a la edición significa trabajar con un documento con el aspecto real que tendrá. Editar una página de HTML en un editor que no sea WYSIWYG, implica trabajar con los códigos que indican el formato que tendrá el texto, sin ver el resultado final.

Referencias y bibliografías

- [1] Robertson, J., Looking towards the future of CM. Step Two, 14 enero 2003 <http://www.steptwo.com.au/papers/cmb_future/index.html> [Consulta: mayo-junio 2005]
- [2] Robertson, J., How to evaluate a content management system. Step Two, 23 enero 2002 <http://www.steptwo.com.au/papers/kmc_evaluate/index.html> [Consulta: mayo-junio 2005]
- [3] ¿Robertson, J., So, what is a content management system? Step Two, 3 junio 2003 <http://www.steptwo.com.au/papers/kmc_what/index.html> [Consulta: mayo-junio 2005]
- [4] Rhyno, A. The Ten Commandments of Content Management, 18 feb 2003 <<http://usrlib.info/story/2003/2/17/82354/8716>> [Consulta: mayo 2005]
- [5] Cuerda X., Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto, dic 2004, <<http://www.uoc.edu/mosaic/articulos/cms1204.html>>
- [6] Papers " Case Studies Step Two Designs <<http://www.steptwo.com.au/papers/index.php>>
- [7] Martínez J. Contraseñas de un solo uso, 23 Nov 2002 <<http://www.usebox.net/jjm/passw/>>[Consulta: mayo 2005]
- [8] PHP, <<http://es.wikipedia.org/wiki/PHP>>[Consulta: junio 2005]
- [9] PHP, la nueva solución para dinamizar el Web, <<http://www.cgsite.cl/php1.htm>> [Consulta: junio 2005]
- [10] Rodríguez L, PHP vs ASP: facilidad, portabilidad, precio, velocidad y más, <http://www.programacion.com/blogs/60_php_land/archive/292_php_vs_asp_facilidad_portabilidad_precio_velocidad_y_mas.html>, [Consulta: junio 2005]
- [11] MySQL, <<http://es.wikipedia.org/wiki/MySQL>> [Consulta: junio 2005]
- [12] cmsInfo , cmsInfo.org, <<http://www.cmsinfo.org>> [Consulta: mayo-junio 2005]
- [13] opensourceCMS , opensourceCMS, <<http://www.opensourcecms.com>> [Consulta: mayo-junio 2005]
- [14] Open source content management , The international association for Open Source Content Management, <<http://www.oscom.org>> [Consulta: mayo-junio 2005]
- [15] portalZine , portalZine, <<http://www.portalzine.de>> [Consulta: mayo-junio 2005]