

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



Proyecto Fin de Carrera

**“Diseño e implementación de una aplicación Web
para la administración de recursos de
investigación del Área de Ingeniería Telemática”**



AUTOR: Juan Antonio Castillo Dayer
DIRECTOR: M^a Victoria Bueno Delgado

Mayo / 2007



Autor	Juan Antonio Castillo Dayer
E-mail del Autor	dayer@totana.com
Director(es)	M ^a Victoria Bueno Delgado
E-mail del Director	Mvictoria.bueno@upct.es
Codirector(es)	
Título del PFC	Diseño e implementación de una aplicación Web para la administración de recursos de investigación del Área de Ingeniería Telemática
Descriptores	Apache,PHP, MySQL
Resumen	
<p>El desarrollo de la aplicación que se presenta en este Proyecto Final de Carrera surge de la necesidad de almacenamiento y organización de los recursos de investigación del Área de Ingeniería Telemática. Para ello, se propone el diseño de un repositorio de recursos de investigación, centralizado y coordinado donde los investigadores del área puedan almacenar en una base de datos los artículos de investigación generados así como realizar consultas, búsquedas y descargas de los mismos.</p> <p>En un futuro próximo esta aplicación se instalará en un ordenador del área que realiza las tareas de servidor, concretamente en el servidor donde se aloja la Página Web del Área (http://ait.upct.es). Por tanto, esta aplicación estará orientada a dar mayor servicio y utilidad a la página principal de la mencionada área de investigación.</p>	
Titulación	Ingeniería Técnica de Telecomunicaciones, especialidad Telemática
Intensificación	
Departamento	Tecnologías de la Información y las Comunicaciones
Fecha de Presentación	Mayo 2007

Índice

<u>CAPÍTULO 1 INTRODUCCIÓN</u>	1
1.1. ANTECEDENTES Y OBJETIVOS.	1
1.2. ESTRUCTURA DEL CONTENIDO	2
<u>CAPÍTULO 2. ESTUDIO DE LAS TECNOLOGÍAS Y HERRAMIENTAS DE DESARROLLO</u>	3
2.1 INTRODUCCIÓN	3
2.2 BASE DE DATOS	5
2.2.1. INTRODUCCIÓN	5
2.2.2. SISTEMAS DE ADMINISTRACIÓN DE BASES DE DATOS	6
2.2.3. MYSQL	8
2.3 SERVIDOR WEB.....	12
2.3.1. INTRODUCCIÓN	12
2.3.2. CLASIFICACIÓN DE SERVIDORES WEB	13
2.3.3. SERVIDOR APACHE	14
2.4 LENGUAJE INTERPRETADO	20
2.4.1. INTRODUCCIÓN	20
2.4.2. LENGUAJES INTERPRETADOS.....	20
2.4.3. PHP.....	22
<u>CAPÍTULO 3. DESCRIPCIÓN DE LA APLICACIÓN</u>	25
3.1 INTRODUCCIÓN	25
3.2 ARQUITECTURA DE LA APLICACIÓN	25
3.2.1 SERVIDOR WEB	25
3.2.2 BASE DE DATOS.....	26
3.2.3 MÓDULO DE LENGUAJE INTERPRETADO.....	27
3.3 APLICACIÓN EN EL LADO CLIENTE	28
<u>CAPÍTULO 4. PUESTA EN MARCHA DE LA APLICACIÓN. MANUAL DEL USUARIO</u>	31
4.1 INSTALACIÓN	31
4.2 BASE DE DATOS	31
4.3 SERVIDOR WEB Y MÓDULO DE LENGUAJE INTERPRETADO.....	31
4.4 MANTENIMIENTO.....	32
<u>CAPÍTULO 5. INSERCIÓN Y BÚSQUEDA DE DOCUMENTACIÓN. MANUAL DEL USUARIO</u>	35
5.1 INTRODUCCIÓN	35
5.2 AUTENTICACIÓN	37
5.3 BÚSQUEDA DE DOCUMENTACIÓN	38

5.4	INSERCIÓN Y MANTENIMIENTO DE DOCUMENTACIÓN	41
5.4.1	INSERCIÓN.....	41
5.4.2	EDICIÓN	43
5.4.3	BORRADO	44
<u>CAPÍTULO 6. CONCLUSIONES Y LÍNEAS FUTURAS.</u>		<u>47</u>
6.1	CONCLUSIONES	47
6.2	LÍNEAS FUTURAS	48
<u>APÉNDICE A. DETALLES DEL LADO DEL SERVIDOR.....</u>		<u>51</u>
<u>APÉNDICE B. DETALLES DEL LADO DEL CLIENTE</u>		<u>55</u>
<u>ACRÓNIMOS Y DESCRIPCIÓN DE TÉRMINOS RELATIVOS A LA APLICACIÓN ..</u>		<u>59</u>
<u>REFERENCIAS</u>		<u>1</u>

Índice figuras

Figura 2.1 - Funcionamiento de una Base de Datos.....	6
Figura 2.2 - Arquitectura del Servidor Web Apache (modelo antiguo)	15
Figura 2.3 - Arquitectura actual del Servidor Web Apache	15
Figura 2.4 - Componentes fundamentales del Núcleo de Apache	16
Figura 2.5 - Procesos a seguir por la petición de una dirección.....	18
Figura 2.6 - Resumen de la Arquitectura utilizada	23
Figura 3.1 - Modelación y relación entre tablas de la Base de Datos	26
Figura 5.1 - Funcionalidades de la aplicación.....	36
Figura 5.2 - Página principal de la aplicación	37
Figura 5.3 - Formulario de autenticación de usuarios	37
Figura 5.4 - Datos del usuario autenticado correctamente	38
Figura 5.5 - Verificación de sesión de usuario cerrada correctamente	38
Figura 5.6 - Indicación de error en la conexión con directorio LDAP	38
Figura 5.9 - Información específica de una entrada (abstract, keywords e idioma).....	40
Figura 5.10 - Información específica de una entrada (volumen, lugar, aportador, etc) 40	
Figura 5.11 - Resultado de búsqueda en modo Referencias.....	41
Figura 5.13 - Formulario de inserción de nuevos documentos	42
Figura 5.14 - Acuse de envío satisfactorio	43
Figura 5.15 - Acuse de actualización satisfactoria.....	44
Figura 5.16 - Petición de confirmación de eliminación de un documento	45
Figura B.1 - Formulario de búsqueda con aviso del no funcionamiento de <i>JavaScript</i> 56	

Capítulo 1

Introducción

1.1. Antecedentes y Objetivos.

Hoy en día las herramientas telemáticas facilitan, entre otras, las tareas relacionadas con el procesado, distribución, almacenamiento, etc de datos y documentos en diversos escenarios. El entorno docente de la ETSIT es un claro ejemplo de ello. En la página web de dicha escuela (<http://www.teleco.upct.es>), tanto alumnos como profesores pueden realizar numerosas tareas de forma telemática: consulta acceso a material de asignaturas, encuestas de profesorado, consulta de fichas personales, listado de proyectos final de carrera, preparación de programas de asignaturas, etc.

El Área de Ingeniería Telemática disfruta de dicho entorno para el almacenamiento, acceso y distribución de recursos docentes, mediante la página web de la escuela, y la página del área <http://ait.upct.es>. Sin embargo, el personal del área de Ingeniería Telemática compatibiliza la labor docente con la labor de investigación, que se desarrolla en diversos campos de las Telecomunicaciones y la Telemática: Tecnologías Inalámbricas, Redes de Sensores, Comunicaciones Ópticas, Peer to Peer, Multicast, etc.

Fruto de las investigaciones del área son los numerosos artículos y publicaciones de investigación que se generan al año. Todo este material investigador es utilizado para numerosos propósitos administrativos, docentes y de investigación. Por tanto, es obvio que dicho material debe encontrarse centralizado, disponible en cualquier lugar, y en cualquier momento, facilitando, agilizando y simplificando la tediosa tarea de acceso a los recursos de investigación que hoy en día sufre, tanto el propio personal investigador, como el alumnado, así como a cualquier persona ajena a la universidad que está interesada en la investigación del Área de Ingeniería Telemática de la UPCT.

De los párrafos anteriores se deduce la necesidad de crear un entorno web centralizado donde se puedan realizar tareas de almacenamiento de artículos de investigación en una base de datos, organización de los mismos por campos de investigación, realizar consultas, búsquedas y descargas de los mismos.

Para abordar estas necesidades, se propone el diseño de un repositorio de recursos de investigación, centralizado y coordinado mediante el desarrollo de una aplicación que en un futuro próximo se pueda instalar en el servidor donde se aloja la Página Web del Área (<http://ait.upct.es>). Esta aplicación dará un mayor servicio y utilidad a la página principal de la mencionada área de investigación, pudiendo dar acceso de los datos almacenados cualquier persona. Interesada.

1.2. Estructura del contenido

El Proyecto Final de Carrera que se presenta aborda el diseño de una aplicación Web de almacenamiento, búsqueda e inserción de información y documentación referente a los artículos de investigación generados por el Área de Ingeniería Telemática. Se proponen unas fases de desarrollo del proyecto:

1. Planteamiento de las necesidades que debe cubrir la aplicación y búsqueda de soluciones.
2. Elección de la solución más viable y adecuada para las necesidades planteadas.
3. Diseño de la Base de Datos, definición de las secciones de la aplicación. Primeras pruebas en un ordenador local.
4. Desarrollo de las funcionalidades de PHP y sentencias SQL de la Base de Datos. Primeras pruebas en el servidor.
5. Pruebas finales y optimización de la aplicación, tanto a nivel de cliente como de servidor.

Estas fases se han descrito en la memoria de este proyecto bajo los siguientes capítulos:

- En el capítulo 2, se presentan las posibles soluciones software existentes para abordar la tarea encomendada, enumerando y destacando aquellas que, por sus ventajas desde el punto de vista de la arquitectura, se eligen para realizar la aplicación. Además, se indica como la necesidad de aprovechar al máximo los recursos y servicios del servidor donde se alojará la aplicación llevan al autor a realizar una aplicación desde el principio.
- En el tercer capítulo se detalla la arquitectura de la aplicación desarrollada bajo las herramientas software elegidas en el capítulo anterior. La arquitectura de la aplicación se aborda tanto a nivel de servidor como cliente.
- En el cuarto capítulo se tienen en cuenta a nivel usuario, los pasos a seguir para la instalación y configuración de la aplicación.
- En el quinto capítulo se presenta un manual de usuario. En él se trata una descripción a nivel de usuario de la aplicación, sus secciones y procedimientos para las funcionalidades, tanto como usuario normal como autenticado.

Capítulo 2.

Estudio de las tecnologías y herramientas de desarrollo

2.1 Introducción

Cuando surge la necesidad de emplear una aplicación para la administración de recursos de una determinada materia o un determinado campo, lo que se necesita en realidad es una aplicación conocida como CMS (*Content Management System*) [CMS], o Sistema de Gestión de Contenidos. Fundamentalmente se distinguen cinco categorías de CMS:

- Foros, para la discusión de diversos temas.
- Blogs, para la publicación de noticias y opiniones.
- Wikis, para la colocación de todo tipo de información o de un tema en concreto.
- eCommerce, para el empleo de comercio electrónico.
- Sitios web, que permiten una gran variedad de utilidades y funcionalidades.

En el proyecto que se presenta, se aborda el caso de implementación de una aplicación web para la administración de recursos del Área de Ingeniería Telemática. Por tanto, la mejor solución que se adapta a las necesidades exigidas son los CMS de la categoría “Sitios web”, ya que permiten una mayor especialización y personalización de la aplicación. Dentro de esta categoría se distinguen 3 tipos de soluciones, bien diferenciadas:

- Aplicación fabricada:
iFolder es un ejemplo de este tipo de aplicaciones. No se trata exactamente de un sitio web, sino de una herramienta propia desarrollada por Novell [Novell] y diseñada para compartir contenidos, soportando GNU/Linux, Windows y MAC OS. Entre sus ventajas destaca el hecho de poder emplear una máquina que haga las funciones de servidor y almacene los contenidos. De esta forma no es necesario que los terminales de usuario estén siempre conectados. Como inconvenientes principales destacar que, para hacer uso de los recursos almacenados se requiere tener la aplicación instalada en la máquina desde donde se quiere acceder. Además, se trata de una herramienta totalmente diseñada, por lo que no permite una gran especialización de las necesidades de los usuarios ni una fácil edición de su código fuente. Existen dos versiones, una de libre distribución y otra más avanzada de pago.
- Aplicación prefabricada

En este caso lo más común es el uso de *frameworks* a los que se añaden *módulos* que aumentan y mejoran la funcionalidad de la aplicación, la mayoría de *OpenSource*. De las numerosas aplicaciones de este tipo que existen hoy en día, caben destacar:

Article Management System, con licencia GPL [GPL] desarrollado por XOOPS [XOOPS], cuyas principales características son que posee módulos para trabajar con formato PDF [pdf], sistema de encuestas y de estadísticas. Además, permite buscar un recurso por el Identificador que se le asigna al almacenarlo, mostrar una cabecera de información del recurso, etc.

TYPO3, con licencia GPL ha sido desarrollado por TYPO3 [TYPO3]. Entre sus características, destaca el empleo de un editor WYSIWYG [WYSIWYG], distintas interfaces según el usuario, accesos directos a nivel interno, multitud de plantillas de organización de la aplicación, visualización de las funciones en modo de estructura de árbol, etc. En material de seguridad permite, entre otros, historial de identificación de usuarios, autenticación empleando LDAP y conexión SSL.

MAMBO (Mambo Open Source) [Mambo], con licencia GNU [GNU], presenta la gran ventaja de ser administrador de contenidos que soporta hasta 20 idiomas, multitud de plantillas, infinidad de módulos, posibilidad de encuestas y estadísticas, así como generación de información en formato RSS [RSS]. No es de los CMS más antiguos pero sí uno de los más consolidados por su gran atractivo y usabilidad.

PHP-Nuke [PHP-Nuke], con licencia GNU GPL. Su antigüedad le otorga una gran fama, también por ello posee gran soporte en todo Internet, con multitud de tutoriales, trucos y sobre todo, módulos para implementarlos en él. Actualmente está perdiendo mucho uso debido a la poca variedad de plantillas para modificar notoriamente su aspecto, ya que, tiene como función principal servir de sistema automatizado de noticias, lo cual no siempre es lo que necesita el cliente. Además, desde hace poco tiempo se ha convertido en una aplicación de pago.

- Aplicación desarrollada desde cero

Si bien es la manera de obtener la aplicación que mejor se adapta a las necesidades requeridas, también supone la solución más tediosa.

Existe la alternativa de encargar a una empresa de desarrollo la elaboración de una aplicación que cubra las necesidades planteadas. Esta elección cuenta con la principal ventaja de que se trata de una aplicación totalmente adaptada a las necesidades y con soporte técnico, pero, su principal inconveniente reside en la total dependencia de la empresa desarrolladora, además del coste por cada modificación requerida del entorno.

Para realizar este proyecto se ha optado por el desarrollo de una aplicación desde cero porque es la solución que permite una mayor especialización en las necesidades que se han de cubrir. Además la edición de una aplicación ya fabricada, o incluso prefabricada, si es que la primera de ellas lo permite, supone un gran esfuerzo para la persona encargada de dicha tarea, al enfrentarse a un código y a un diseño cuya estructuración desconoce, siendo muy fácil que un simple error suponga que la aplicación deje de funcionar en su totalidad.

2.2 Base de Datos

2.2.1. Introducción

Henry F. Korth define en su libro “*Fundamentos de Bases de Datos*” [Fundamentos de bases de datos] a una base de datos como una serie de datos organizados y relacionados entre sí, y un conjunto de programas que permite a los usuarios acceder y modificar esos datos.

En una base de datos los datos son introducidos y almacenados de una sola vez, con independencia de los tratamientos, a nivel lógico.

Antes de la existencia de las bases de datos se empleaban archivos secuenciales para almacenar datos, lo que suponía un serio problema para el acceso a una determinada posición, ya que había que recorrer el archivo por completo, mermando seriamente la velocidad. Tiempo después hicieron su aparición los archivos indexados, con lo que ya se podía acceder a una determinada posición directamente.

Con el uso y el crecimiento de las necesidades exigidas, se requería un almacenamiento que permitiera operaciones complejas, sin que se violasen las restricciones impuestas por el sistema de archivos que se estuviese empleando, y cada usuario necesitaba tener confidencialidad e integridad en sus datos almacenados. Por todo esto surgieron las bases de datos jerárquicas, en donde los datos se situaban siguiendo una jerarquía, pero también presentaban serias limitaciones. Los accesos a datos eran unidireccionales, por lo que no se tardaba el mismo tiempo en hacer un camino que otro, (por ejemplo: en nuestro caso no sería igual de sencillo averiguar los artículos que ha publicado un investigador, que de quien es un determinado artículo).

A continuación hicieron aparición las bases de datos relacionales, que daban una mayor libertad a las relaciones entre tablas y tenían las propiedades ACID [ACID] y un lenguaje común de acceso a datos conocido como SQL [SQL]. Estas propiedades impiden que las bases de datos sean igual de rápidas que el trabajo directo con archivos, aunque a nivel interno la base de datos trabaje sobre archivos y hoy día, aún esté en serio debate si el acceso a ciertos datos a través de base de datos es más rápido que el acceso directo a disco.

Esquemáticamente el funcionamiento de una base de datos se puede resumir a rasgos generales de la siguiente manera:

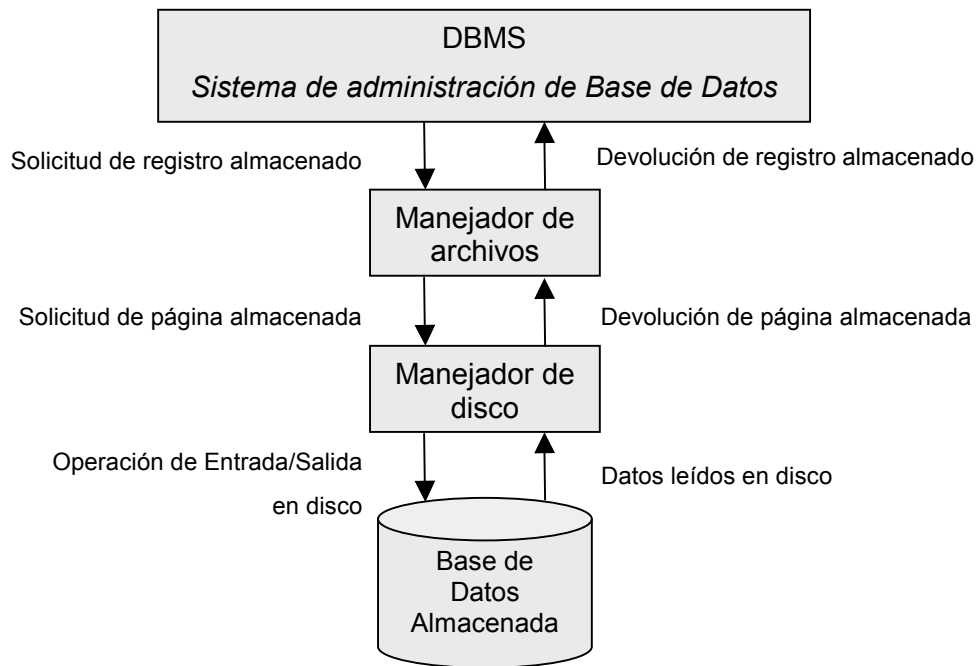


Figura 2.1 - Funcionamiento de una Base de Datos

Usar una base de datos presenta las siguientes ventajas:

- Globalizar la información: permite a los diversos usuarios considerar la información como un recurso sin dueños específicos.
- Eliminar información redundante: elimina la información duplicada que podría ocasionar inconsistencias en el sistema de la base de datos.
- Eliminar información inconsistente: una operación ha de ser tenida en cuenta a nivel de todo el sistema, como por ejemplo si se vende un componente en una empresa de distribución, este tendrá que ser eliminado del stock y a la vez añadido en la facturación.
- Compartir información: varios sistemas y/o usuarios pueden emplear una misma entidad.
- Mantener integridad en la información: no se almacena información incorrecta, sólo se almacena la información que ha sido añadida correctamente y cuyo proceso de inserción ha terminado con éxito.
- Independencia de datos y tratamiento: se puede alterar la información sin cambiar por ello la aplicación que accede a la base de datos, y viceversa, lo que supone menor coste de mantenimiento.
- Restricciones de seguridad: no permite acceder ni efectuar operaciones sobre los datos a cualquier usuario.

2.2.2. Sistemas de Administración de Bases de Datos

Existen una gran diversidad de sistemas de administración de bases de datos relacionales (RDBMS: *Relational Data Base Management System*) [base de datos relacionales] los más conocidos se explican a continuación:

- MySQL: disponible bajo S.O. MS Windows y GNU/Linux, posee una gran velocidad, es muy empleada sobre todo en aplicaciones desarrolladas en PHP o Perl y tiene un bajo coste en cuanto al espacio que necesita. Es de código abierto y la base de datos más popular del mundo. Como principales inconvenientes destacan las grandes limitaciones para transacciones [transacción], procedimientos almacenados y complejas sentencias de SQL.
- Oracle: fabricado por Oracle Corporation [Oracle], es un producto vendido a nivel mundial, aunque su gran potencia y su elevado precio lo hace sólo accesible a grandes empresas. Otorga gran robustez y numerosas características que garantizan la seguridad e integridad de los datos, que las transacciones se ejecuten de una forma correcta sin inconsistencias, ayuda a administrar y almacenar gran cantidad de información, estabilidad, escalabilidad y está disponible bajo MS Windows y GNU/Linux. Hasta hace poco tiempo ha presentado un dominio casi total en el mercado de servidores empresariales.
- Access: disponible únicamente bajo S.O. MS Windows, su uso se aconseja únicamente para volúmenes de datos pequeños, bajo número de visitas simultáneas y que la aplicación que acceda a ella no efectúe cambios. Se deben de emplear aplicaciones elaboradas en ASP para acceder a ella.
- MS SQL Server: disponible únicamente bajo S.O. MS Windows, está orientada para programadores especializados en ASP y Visual Basic, entre otros de Microsoft, y es adecuada para aplicaciones críticas y con cualquier grado de complejidad.
- PostgreSQL: disponible bajo S.O. MS Windows y GNU/Linux. Se trata del servidor de bases de datos de código abierto más potente, y es, de los de código abierto, el más adecuado para aplicaciones complejas o críticas. Entre sus principales características se encuentran: funciones (que pueden escribirse en variedad de lenguajes para que se ejecuten en el servidor, como C, C++ y JAVA), alta concurrencia (que permite que un proceso escriba en una tabla y mientras otros accedan a esa misma tabla sin necesidad de bloqueos), amplia variedad de tipos nativos (como texto ilimitado y direcciones IP y MAC), integridad transaccional y herencia de tablas. Su principal inconveniente está en que se trata de los servidores de bases de datos más lentos.

Para la elaboración de este administrador de contenidos, resulta necesario el empleo de una base de datos en la que se almacenen datos diversos de la aplicación. Se toma esta decisión debido a que se considera más ventajoso que el empleo de archivos para el almacenamiento de la información puesto que:

- el empleo de archivos produce una ocupación inútil de memoria y un aumento en los tiempos de proceso (al repetirse los mismos controles y operaciones en distintos archivos).
- empleando directamente archivos se está más a merced de las inconsistencias que estas prácticas presentan, ya que la actualización de

los datos cuando estos se encuentran en más de un archivo no se puede realizar de manera simultánea en todos ellos.

En este proyecto se va a emplear el Sistema de Administración de Bases de Datos **MySQL**. Esta decisión se adopta por los siguientes motivos:

- Es necesaria la integridad referencial para que la información esté almacenada con una mayor seguridad frente a borrados accidentales, pero no es necesaria la alta seguridad y gran soporte para estos menesteres que Oracle y Postgre SQL, por ejemplo, brindan.
- En apartados posteriores se explican los motivos del empleo de GNU/Linux como Sistema Operativo. Como consecuencia de esto, se descarta por completo el uso de MS SQL Server y Access que necesitan MS Windows.
- Ante todo se busca una buena relación entre calidad, usabilidad y coste. En este ámbito Oracle por ser un sistema de pago y muy costoso queda descartado. Postgre SQL presenta ante todo la mayor lentitud de todos los sistemas anteriormente explicados, lo que le otorga una demora al sistema innecesaria dado que no es un sistema que vaya a almacenar cientos de miles de entradas de datos.
- <más motivos que se me puedan ocurrir>

2.2.3. MySQL.

2.2.3.1 Breve reseña histórica

Los inicios de MySQL se remontan a la necesidad de conectar el gestor [mSQL] a las propias tablas de [MySQL AB], para lo que se comenzó con unas rutinas a bajo nivel. Después de diversas pruebas se hizo más evidente que mSQL no brindaba la suficiente flexibilidad para estos menesteres, motivo por el cual se comenzaron a desarrollar nuevas funciones, que dieron lugar a una interfaz SQL totalmente compatible con mSQL. Desde sus comienzos se han ido desarrollando multitud de versiones, siendo actualmente la versión estable 5.0.27 la más reciente.

El origen del nombre “MySQL” no se conoce con certeza, parece ser que durante los últimos años sus librerías han llevado el prefijo “my”, lo que justamente coincide con el nombre de la hija de un miembro del equipo de desarrollo.

2.2.3.2 Características principales

Actualmente la versión más reciente es la versión 5, aunque la versión 4 aún se emplea mucho a pesar de que MySQL AB comunicó recientemente la finalización de actualizaciones de seguridad para dicha versión. Aunque en el servidor donde en principio se va a instalar la aplicación cuenta con MySQL 5.0.32, a continuación se detallan las características comunes a ambas versiones para una mayor generalización:

- Internas y portabilidad
 - Programado en C y C++
 - Verificado en una gran variedad de compiladores

- Multiplataforma, soportando varios Sistemas Operativos como GNU/Linux, Mac OS X, SunOS y MS Windows, entre otros.
 - Emplea GNU Automake, Autoconf y Libtool para la portabilidad.
 - Tiene disponibles APIs para C, C++, Eiffel, Java, Perl y PHP, entre otros lenguajes.
 - Soporta multi-hilo (*multi-threaded*) para múltiples microprocesadores.
 - Posee sistemas de almacenamiento transaccionales y no transaccionales.
 - Permite trabajar con tablas MyISAM, con gran rapidez con compresión de índice.
 - Sencillez para añadir otro sistema de almacenamiento, permitiendo añadir una interfaz SQL a una base de datos propia.
 - Reserva de memoria muy rápida basada en threads (*hilos*).
 - Joins muy rápidos empleando *multi-join* de una sola pasada.
 - Tablas hash empleadas como tablas temporales, al estar alojadas en memoria.
 - Trabaja con funciones SQL implementadas mediante una librería altamente optimizada, con una rapidez tan grande como sea posible, y por lo que normalmente no se reserva memoria después de toda la inicialización para consultas.
 - Código probado con Purify y Valgrind, aplicaciones que permiten encontrar memoria perdida.
 - El servidor se puede utilizar como un programa separado para el caso de un entorno de red cliente-servidor. A su vez también está disponible como biblioteca y puede ser embebido en aplicaciones autónomas, con disponibilidad o no de red.
- Tipos de datos, sentencias y funciones
 - Soporte para una gran variedad de tipos de datos: enteros con y sin signo, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM y especiales de [OpenGIS].
 - Permite registros tanto de longitud fija como variable.
 - Da soporte completo para operadores y funciones en las cláusulas de consultas SELECT y WHERE.
 - Soporta de una manera completa las cláusulas GROUP BY y ORDER BY, además de funciones de agrupación COUNT, AVG, STD, SUM, MAX, MIN y GROUP_CONCAT, así como LEFT OUTER JOIN y RIGHT OUTER JOIN cumpliendo estándares de sintaxis SQL y ODBC.
 - Permite el uso de alias en tablas y columnas.
 - Las cláusulas DELETE, INSERT, REPLACE y UPDATE devuelven el número de filas que han sido afectadas o el número de filas que serían afectadas, según se requiera.

- Las funciones pueden llamarse igual que las tablas o las columnas, no hay colisiones entre ellos, salvo que no se permiten espacios entre el nombre de la función y los paréntesis empleados para invocarla.

- Seguridad y conectividad
 - Empleo de un sistema de privilegios y contraseñas con gran flexibilidad y seguridad, permitiendo la autenticación basada en la máquina servidora, además de intercambiar contraseñas de manera cifrada.
 - Se puede conectar al sistema empleando sockets TCP/IP en cualquier plataforma.
 - Los servidores basados en MS Windows permiten conexiones con memoria compartida si se inicializan con la opción para tales efectos, así como los clientes también pueden utilizar esta prestación.
 - La interfaz para el conector *MyODBC*, disponible para todas las plataformas, proporciona a MySQL soporte para programas clientes que utilicen conexiones ODBC (*Open Database Connectivity*), como es el caso de emplear MS Access para conectarse al servidor MySQL.
 - De igual manera existen interfaces para conectores J y NET que permiten conexiones a la base de datos desde aplicaciones desarrolladas en Java y .NET.

- Escalabilidad y localización
 - MySQL es empleado en bases de datos que contienen 50 millones de registros. También hay casos con 60.000 tablas y alrededor de 5 millones de registros.
 - Permite hasta 32 índices por tabla (64 a partir de la versión 4.1.2), de los que cada uno puede tener desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 500 bytes (1000 a partir de la versión 4.1.2).
 - Los mensajes de error pueden ser mostrados en multitud de idiomas, según en el que sea configurado el servidor.
 - Soporte para gran variedad de conjuntos de caracteres.
 - Datos guardados en el conjunto de caracteres elegido, además de que todas las comparaciones, salvo si se indica en el juego de caracteres, son insensibles a mayúsculas (*case-insensitive*).
 - Las ordenaciones son llevadas a cabo en el conjunto de caracteres elegido.

2.2.3.3 Instalación y puesta en marcha

Existen infinidad de métodos para instalar MySQL y más en un sistema con GNU/Linux, como es el caso del ordenador que va a realizar la función de servidor de esta aplicación. Para este caso, dado que este servidor cuenta con la distribución [Debian], se va a hacer uso de su eficaz y potente sistema de paquetes mediante su

herramienta apt-get, que también facilita enormemente la desinstalación del sistema de base de datos por si dejara de resultar necesario.

Partiendo y dando por hecho que se poseen unos repositorios actualizados y sin incoherencias entre paquetes ya instalados, se procede de la siguiente manera, como usuario root:

1. Instalación de los paquetes necesarios:

```
#apt-get install mysql-common mysql-client mysql-server
```

2. Iniciación del demonio (*daemon*) que controla MySQL:

```
#/etc/init.d/mysql start
```

3. Si se diera el caso de un error del estilo a “*ERROR 2002: Can't connect to local MySQL server through socket '/var/run/mysqld/mysqld.sock'*”, es necesario ejecutar las siguientes sentencias:

```
#mysqld
```

Se ejecuta el demonio de MySQL y se crea el socket en:

```
#/var/run/mysql/mysqld/mysqld.sock
```

Y de Nuevo se vuelve a iniciar MySQL con lo indicado en el paso 2.

4. A continuación es aconsejable, y muy necesario, configurar una contraseña para el usuario *root* de la base de datos, para lo que se llevan a cabo los siguientes comandos:

```
#mysql -u root
```

```
mysql> USE mysql;
```

```
mysql> DELETE FROM user WHERE User='';
```

```
mysql> SET PASSWORD FOR 'root' = PASSWORD('contraseña');
```

```
mysql> exit;
```

5. Si desde otro ordenador se desea poder acceder al *prompt* de MySQL, es necesario descomentar de */etc/mysql/my.cnf* la siguiente línea:

```
Skip-networking
```

6. Después es muy aconsejable instalar **[phpMyAdmin]**, el cual facilita enormemente las labores de administración de la/s base/s de datos del sistema (herramienta sólo operable una vez instalado PHP) con la ejecución de:

```
#apt-get install phpmyadmin
```

Si la ejecución de estos pasos ha sido satisfactoria, ya se debería de contar con un servidor de MySQL instalado, listo para crear y administrar bases de datos con sus tablas y registros. Esto puede ser llevado a cabo tanto desde un *prompt* con *mysql* o desde un navegador Web accediendo a <http://127.0.0.1/phpmyadmin> si se trabaja desde el servidor, o con http://host_servidor/phpmyadmin si se trabaja desde otro ordenador.

2.3 Servidor web

2.3.1. Introducción

Para el desarrollo y puesta en marcha de este proyecto, resulta indispensable la utilización de un servidor web, entiendo por este como el programa que implementa el protocolo [http].

Sobre la palabra servidor es necesario distinguir claramente entre la parte que identifica al programa y la parte que identifica al ordenador en sí. En este caso todo el texto se refiere por entero al programa, salvo si se indica lo contrario.

El servidor web tiene la función de permanecer a la espera de peticiones HTTP de los clientes, que suelen llevarlas acabo a través de un navegador web. Los clientes realizan peticiones HTTP al servidor, y éste les responde con el/los contenido/s que los clientes solicitan, aunque también se puede configurar el servidor para que no permita más de un cliente simultáneamente. Los contenidos devueltos por el servidor suelen ser interpretados por los navegadores web de los clientes, que los muestran por pantalla acorde a unas fuentes, colores, disposición de textos y objetos, por lo que el servidor se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de ella.

Cada vez con más frecuencia se está disponiendo de aplicaciones web, que consisten en fragmentos de código que se ejecutan tras realizar ciertas peticiones o respuestas HTTP, de las que cabe diferenciar entre:

- Aplicaciones del lado del servidor: aquellas en las que el servidor lleva acabo la/s ejecución/es de una o varias aplicaciones que producen una respuesta, normalmente un código en HTML. El servidor envía esta respuesta al cliente empleando el protocolo HTTP.
- Aplicaciones del lado del cliente: aquellas en las que el servidor proporciona al cliente el código de las aplicaciones al cliente, y éste mediante el navegador las ejecuta. Las más conocidas son las tipo Java, Javascript, Actionscript, etc. Para que la ejecución se lleve a cabo el navegador ha de tener permisos para estas ejecuciones además de soportarlas. Para el caso de Javascript, éste suele ir soportado nativamente por el navegador, pero para el resto lo normal es que el navegador tenga que emplear [plugins].

Las aplicaciones del lado del cliente presentan la ventaja de que el cliente realiza un procesado mínimo, pero el gran inconveniente de que a veces es necesario que el cliente cuente con unos plugins específicos para poder visualizar correctamente los resultados que recibe del servidor. A diferencia de las del lado del cliente, las aplicaciones del lado del servidor tienen el inconveniente de que en algunas ocasiones el servidor puede encontrarse muy saturado, por ejemplo, debido a un gran número de peticiones simultáneas pero a su vez esta tecnología presenta la gran ventaja de que apenas requiere procesado por parte del cliente, ni ninguna capacidad adicional para poder visualizar la respuesta del servidor.

2.3.2. Clasificación de servidores web

Existen infinidad de servidores web, unos propios de una plataforma en concreto y otros multiplataforma, para abarcar pequeñas cantidades de clientes o para grandes cantidades de clientes y tráfico. A continuación se detallan algunos:

- Apache: se trata de un servidor de código abierto cuyos inicios se remontan a 1995, actualmente se desarrolla dentro del proyecto *HTTP Server* de *Apache Software Foundation*, y que soporta plataformas UNIX, Windows y cualquier otra que implemente el protocolo HTTP/1.1. Es muy posiblemente el servidor web más conocido y utilizado.
- ISS: *Internet Information Services*, es una serie de servicios para los ordenadores que funcionan en la plataforma Windows. En sus inicios formaba parte del *Option Pack* para MS Windows NT, pero después fue integrado en el resto de sistemas operativos de Microsoft orientados a ofrecer servicios. El servicio de servidor web es uno más de los que ofrece, a parte de FTP, SMTP y NNTP.
- Cherokee: de código abierto, comenzó a ideado a finales de 2001 por un ingeniero español que trabaja para Sun Microsystems, aunque en el proyecto colaboran desarrolladores de todo el mundo. Está disponible para plataformas UNIX y Windows. Su mejor publicidad es su gran velocidad, facilidad de uso y modularidad. Por esto último es por lo que resulta tan eficiente y consume pocos recursos, ya que no presenta ninguna funcionalidad instalada. Es una herramienta especializada a la que se le añaden funcionalidades a través de *plugins* y sólo cuenta con funcionalidades que el servidor va a necesitar para llevar a cabo su servicio.
- Lighttpd: de software libre y disponible para plataformas UNIX, está diseñado para ser flexible, rápido, seguro y fiel a los estándares, especialmente pensado para ordenadores con escasos recursos.
- Thttpd: distribuido como software libre y disponible para plataformas UNIX, tiene como mejores prestaciones su simplicidad, poco tamaño, portabilidad, rapidez y seguridad, empleando los requerimientos mínimos de un servidor HTTP. En su página web se presume de que puede permitirse más de 1000 clientes conectados.

Para el desarrollo y funcionamiento de esta aplicación se va a emplear el servidor **Apache**. Decisión adoptaba por los siguientes motivos:

- Es necesario que el servicio funcione sobre plataforma GNU/Linux, dado que es el Sistema Operativo que tiene instalado el ordenador en el que se va a implantar el sistema. Por esta razón, a pesar de sus servicios complementarios, ISS queda descartado para esta aplicación.
- El sistema requiere estabilidad tanto a nivel de Sistema Operativo como también de la aplicación que trabaja de servidor web, motivo por el que de los servidores web restantes se considera que Apache es el más idóneo por su gran variedad de versiones, antigüedad y amplia comunidad de desarrollo.
- Resulta aconsejable que la aplicación tenga un buen soporte y asistencia técnica, y de todas las mencionadas Apache es la más extendida y utilizada.

- Todo su código está escrito en C, por lo que a pesar de no estar compuesto de las funcionalidades únicamente necesarias, el detrimento que ellas pueden ocasionar en su velocidad no es muy apreciable para el uso que en este proyecto va a tener.
- En algún momento puede ser necesario implementar un sistema de conexiones seguras a través del servidor web, funcionalidad no disponible por parte de Thttpd.

2.3.3. Servidor Apache

2.3.3.1 Breve reseña histórica

El origen de Apache se remonta a 1995, año en el que comenzó el desarrollo del proyecto del grupo Apache, también conocido como ASF (*Apache Software Foundation*). Este proyecto está basado en el servidor Apache HTTPd de la aplicación original de [NCSA].

El desarrollo de la aplicación original sufrió una pausa tras la marcha de algunos de sus desarrolladores, motivo por el que varios administradores de servidores webs que lo empleaban siguieron creando sus [parches] por labores de mantenimiento. Estos desarrolladores de parches se unieron un tiempo después y formaron el grupo actual, en el que cada desarrollador trabaja en sus propios proyectos de código abierto.

En sus inicios esta aplicación era sólo para Sistemas Operativos UNIX, y con las distintas evoluciones pasó a estar también disponible para plataformas Windows.

2.3.3.2 Características principales

Actualmente existen multitud de versiones 1.x y 2.x de Apache, pero el ordenador que efectuará las labores de servidor de este proyecto tiene instalado Apache 1.3.34 y esas decisiones no competen a este proyecto. A continuación se citarán las características disponibles en ambas versiones.

- Disponible para una gran variedad de Sistemas Operativos, como ya se citó en su presentación en otro apartado anterior.
- Al ser una tecnología gratuita de código fuente abierto, su software es transparente y quien lo administra puede saber concretamente lo que está instalando.
- Tiene diseño modular, lo que permite de una manera muy sencilla ampliar sus capacidades, y al estar escrito en C cualquier conocedor de este lenguaje y que quiera implementar más funcionalidades puede crearse sus propios módulos.
- Permite trabajar con CGI, Java, Perl y PHP, entre otros lenguajes de script, incluso en modo CGI o como módulos.
- Hace posible la personalización de los diversos errores que se pueden dar, como por ejemplo al intentar acceder a una página inexistente.
- Ofrece una alta configurabilidad en la creación y gestión de logs, por si se requiere tener un mayor control sobre el funcionamiento del servidor web.

2.3.3.3 Arquitectura y funcionamiento

El servidor web Apache se basa en un concepto muy simple de división del sistema en módulos, representando estos la división del trabajo dentro del sistema, valga la redundancia. Este servidor tiene un Núcleo (*core*), que lleva a cabo todas las funcionalidades importantes del sistema, y los módulos, externos a este núcleo, añaden funcionalidades al servidor web, configurándose cada uno de ellos por separado. Es por ello que se obtiene globalmente un sistema muy flexible, que permite ser configurado y optimizado acorde a unas necesidades mediante la implementación de módulos externos. Esta arquitectura tradicional esquemáticamente ha sido siempre la siguiente:

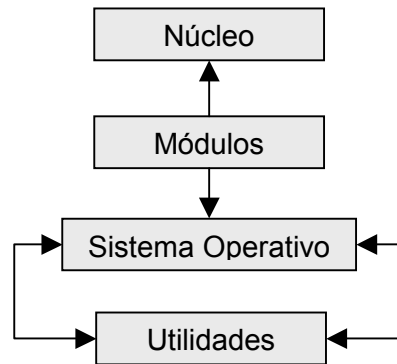


Figura 2.2 - Arquitectura del Servidor Web Apache (modelo antiguo)

La manera tradicional de ver el sistema ha consistido en pensar en él como en 4 capas, de las que cada una intercambiaba información con otra a través de una interfaz. Las cuatro capas eran Núcleo, Módulos, Sistema Operativo (SO) y Utilidades (Util). La capa de los módulos era la más independiente y consistía en un subsistema que proveía de funcionalidades adicionales, como el subsistema [MIME].

Actualmente, la arquitectura del sistema se contempla desde el punto de vista de otro modelo:

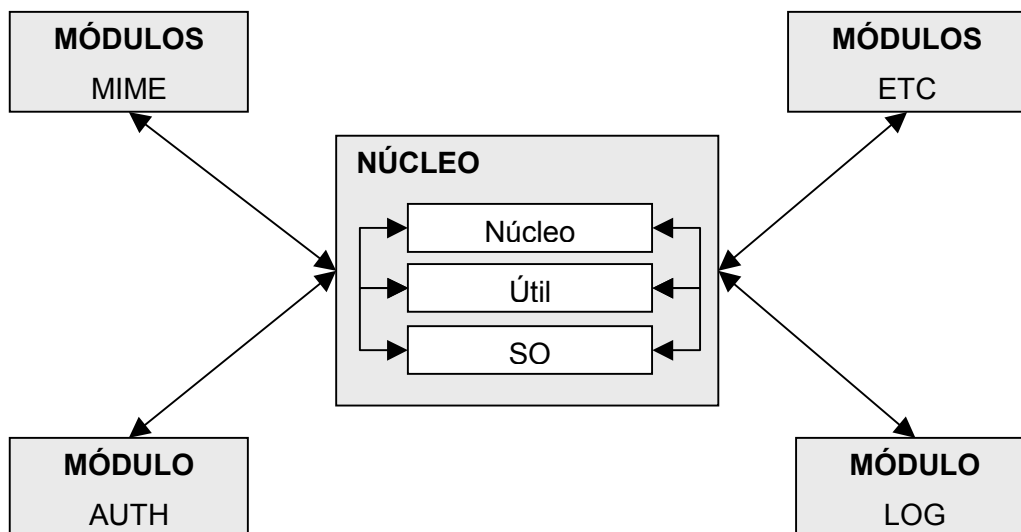


Figura 2.3 - Arquitectura actual del Servidor Web Apache

En este modelo la perspectiva es diferente. El Núcleo del servidor web puede decirse que está compuesto del Subsistema Núcleo, Subsistema de Sistema Operativo y Subsistema de Utilidades. Los tres subsistemas se encuentran formando tres capas en el Núcleo, y los módulos se comunican a través de interfaces proporcionadas por cada una de esas capas, por lo que esencialmente los módulos suplementan las funcionalidades del Núcleo.

- Núcleo

Compuesto fundamentalmente por cuatro componentes, representado esquemáticamente a continuación:

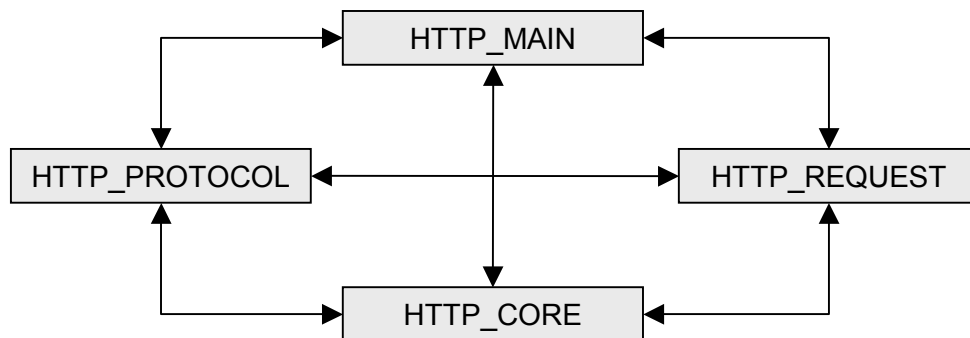


Figura 2.4 - Componentes fundamentales del Núcleo de Apache

- Módulos

El servidor Apache dispone de diversos módulos que implementan las funcionalidades de los servicios solicitados por el cliente. Estos módulos interactúan con el Núcleo y son independientes unos de otros, pero todos tienen la misma interfaz hacia él. El servidor Apache contiene varios módulos estándares, de los cuales dado su gran número sólo se describirán algunos, agrupados según su fase de uso o utilidad, aunque hay diversas clasificaciones.

- Traducción de direcciones
 - *mod_userdir*: traduce el directorio de usuario en las rutas reales de *mod_rewrite*.
 - *mod_rewrite*: proporciona un motor de reescritura, basado en reglas que reescriben las peticiones de direcciones de inmediato.
- Fases de autenticación/autorización, comprueban que el usuario tiene permiso para acceder al servidor, y se verifica mediante nombre de *host* o dirección IP.
 - *mod_auth*, *mod_auth_anon*, *mod_auth_db*, *mod_auth_dbm*: empleados para la autenticación.
 - *mod_access*: proporciona control de acceso basándose en el nombre del *host* del cliente, dirección IP u otras características de la petición del cliente.

- Determinación de tipo MIME de los objetos solicitados
 - *mod_mime*: determina el tipo de archivo a través de la extensión del archivo.
 - *mod_mime_magic*: determina el tipo MIME de un archivo analizando los bytes iniciales del contenido.

- Fase de fijación
 - *mod_alias*: traduce diferentes partes del sistema de archivos del servidor en el árbol de documentos del servidor y redirecciona direcciones.
 - *mod_env*: modifica el entorno que se envía a los scripts CGI y las páginas SSI (*Server Side Includes*).
 - *mod_speling*: puede corregir automáticamente algunos errores tipográficos de la dirección e ignora mayúsculas y minúsculas

- Envío de datos al cliente
 - *mod_actions*: se utiliza para ejecutar scripts CGI basándose en el tipo de medio o el método de petición.
 - *mod_asis*: empleado para el envío de archivos que tienen sus propias cabeceras HTTP.
 - *mod_autoindex*: lista y muestra los contenidos de un directorio de manera automática.
 - *mod_cgi*: permite la ejecución de scripts CGI.
 - *mod_include*: posibilita el incluir documentos HTML generados por el propio servidor, también conocido como SSI.
 - *mod_dir*: utilizado para redirecciones y servir los archivos de listado de directorios.
 - *mod_ldap*: necesario para el procesado de imágenes en el lado del servidor.

- Fase de registro de peticiones
 - *mod_log_config*: registro de las peticiones efectuadas al servidor.
 - *mod_logio*: registro del número de bytes recibidos y enviados en cara respuesta de petición de un cliente.

- Petición de una dirección:

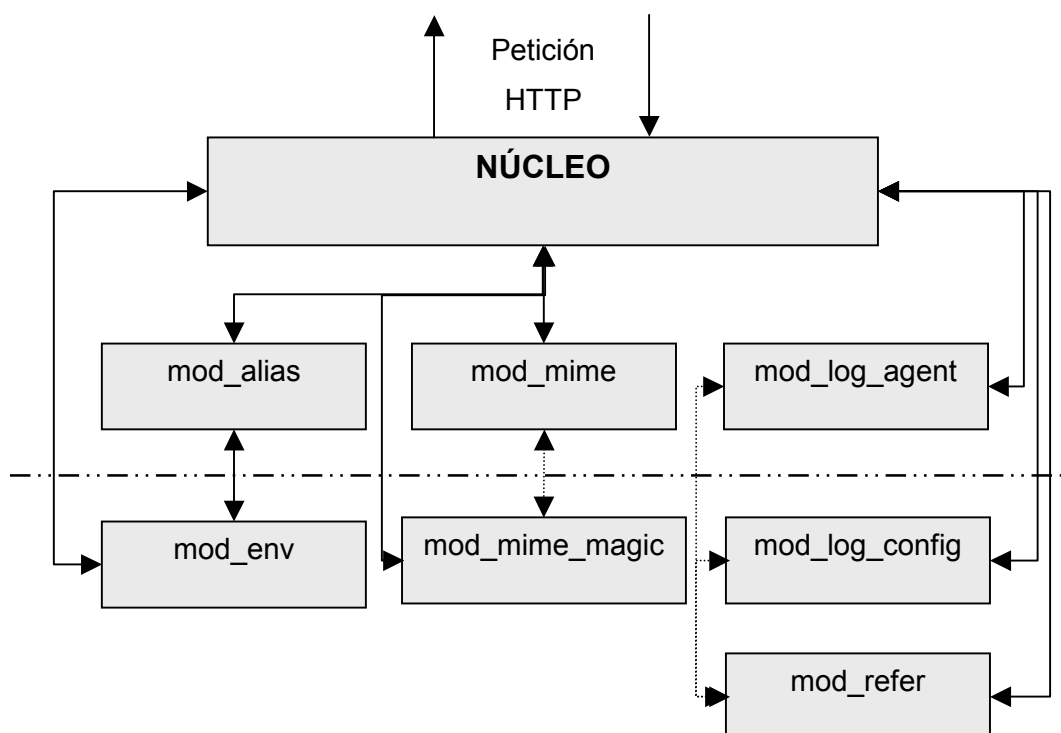
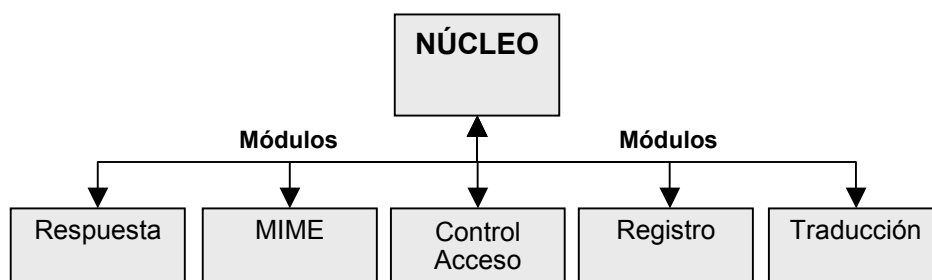


Figura 2.5 - Procesos a seguir por la petición de una dirección

El Núcleo del servidor web Apache está dividido en componentes, cada uno con una responsabilidad específica en el ciclo del tratamiento de peticiones. Una petición de un cliente primeramente contacta con *HTTP_PROTOCOL*, componente responsable de todas las conexiones con los clientes. El servidor Apache gestiona la petición transfiriéndola entre los diferentes módulos, en donde cada módulo es responsable de una funcionalidad específica, y por lo tanto el núcleo debe de asegurarse de que la petición es reenviada por el módulo apropiado para el procesado necesario. Cada módulo es independiente de los demás, y por lo tanto nunca interactúan entre ellos directamente, sólo a través del núcleo. El componente *HTTP_REQUEST* es el responsable del intercambio de peticiones entre núcleo y módulos. Por lo tanto, una petición es recibida por *HTTP_PROTOCOL*, *HTTP_REQUEST* toma el control y comienza el ciclo típico de ejecución.

Primero tiene lugar la traducción de la dirección (URL) y extracción del nombre del archivo solicitado en la petición. En el transcurso de esta etapa el módulo *mod_userdir* recibe la petición y resuelve la ruta de dirección. A continuación, se

autentica la petición. El componente *HTTP_REQUEST* envía la petición a los módulos *mod_auth*, *mod_auth_anon* y *mod_auth_db* y *mod_auth_dbm*. Esta etapa corresponde a la fase de autenticación anteriormente mencionada.

Una vez que el nivel de acceso es el correcto y se permite este, se entrega la petición al sistema de archivos o al script CGI. Finalmente el componente *HTTP_REQUEST* permite el tipo MIME necesario. El tipo MIME es el tipo de contenido que será enviado por el flujo de salida de datos hacia el cliente. Este determina el contenido, la codificación en este y el lenguaje empleado para entregar la salida al cliente. Al final del proceso, el resultado es colocado en el flujo de salida con la ruta y direcciones reales, las cuales serán des-traducidas. Así que por último, el componente *HTTP_PROTOCOL* comunicará al cliente la respuesta a su petición, y al final de esta labor la petición será registrada en el registro de peticiones del servidor.

2.3.3.4 Instalación y puesta en marcha

Al igual que para la instalación de MySQL, para instalar Apache se parte de las provechosas ventajas que presenta un sistema Debian GNU/Linux, como es el caso del ordenador en el que se instalará esta aplicación. Para ello se ha de seguir estos breves pasos:

1. Mediante la herramienta *apt-get* se instalan los paquetes necesarios:

```
#apt-get install apache-common apache
```

2. Para un correcto direccionamiento de los archivos que Apache sirve es necesario configurar las siguiente línea del archivo */etc/apache/httpd.conf*:

```
ServerName <host/ip con la que se accede desde el exterior>
DocumentRoot <ruta raíz de los documentos a "servir" en el
servidor>
DirectoryIndex index.html index.php index.htm index.shtml
index.cgi
Default.htm default.htm index.php
```

3. A continuación es necesario reiniciar Apache:

```
#/etc/init.d/apache restart
```

4. Por último puede resultar conveniente, siempre y cuando lo permita la directiva *AllowOverride* de *httpd.conf* lo permita, para dotar de una mayor seguridad a la aplicación, la creación de un archivo *.htaccess* dentro de los directorios que se deseen proteger de indexación de contenidos, introduciendo en este archivo la siguiente línea:

```
Options -Indexes
```

2.4 Lenguaje Interpretado

2.4.1. Introducción

En el apartado dedicado al Servidor Web se explicó de la existencia de aplicaciones en el lado del cliente y aplicaciones en el lado del servidor. Para el desarrollo de esta aplicación se ha decidido por el empleo, principalmente, de una aplicación en el lado del servidor. Esta decisión se adopta por varios motivos:

- En los clientes se requieren unos requisitos mínimos para que la aplicación web sea correctamente visible, tanto de software como de hardware.
- Al utilizar una base de datos, los tiempos de comunicación con ella serán mucho menores si es el propio ordenador que lleva a cabo las tareas y no tienen que realizar el acceso a ella ordenadores desde el exterior.
- Es más eficiente que los clientes soliciten información al servidor, y que este consulte los datos y construya la página de respuesta, a que sea el propio cliente quien tenga que construir la página para visualizar al usuario con la información obtenida sin formato visual.
- Resulta más seguro que los datos de autenticación frente al directorio LDAP y base de datos los almacene el servidor, que si este los tiene que enviar a los clientes para que ellos se conecten a estos servicios.
- <mas motivos que se me puedan ocurrir>

2.4.2. Lenguajes Interpretados

Los lenguajes interpretados forman un subconjunto de los lenguajes de programación, el cual incluye a aquellos lenguajes cuyos programas son habitualmente ejecutados en un intérprete, en lugar de compilados. También suelen llamarse *script*.

A continuación se mencionan y se explican muy brevemente detalles de los lenguajes interpretados más conocidos:

- ActionScript: es un lenguaje de programación Orientado a Objetos (OOP), basado en las mismas especificaciones que *JavaScript*, que se emplea especialmente en aplicaciones web vectoriales animadas, realizadas en entornos Flash, normalmente de Adobe, aunque también hay aplicaciones como Swish que trabajan con este lenguaje. Existen tres versiones, 1, 2 y 3, de las que actualmente la más utilizada es la versión 2.0. Sus scripts se ejecutan en el lado del cliente.
- ASP: *Action Server Pages*, consiste en una tecnología del lado del servidor de Microsoft para páginas web generadas dinámicamente, que actualmente se comercializa como un anexo a ISS. Tiene una variedad de versiones, de las que actualmente la más utilizada es ASP 3.0, aunque desde 2002 está siendo reemplazado por ASP.NET, que reemplaza los lenguajes interpretados por lenguajes compilados a código intermedio. Su gran ventaja es que las páginas web pueden ser generadas mezclando su código (por ejemplo de acceso a bases de datos o conexión a directorios LDAP) con HTML.

- JavaScript: se utiliza principalmente en páginas web y tiene una sintaxis muy parecida a Java y C, aunque no es Orientado a Objetos propiamente dicho, al no disponer de Herencia como Java, se basa en prototipos, mediante la clonación de clases base y extendiendo su funcionalidad. Su código es interpretado por la mayoría de navegadores y se ejecuta en el agente de usuario del navegador al mismo tiempo que HTML. Sus scripts por lo tanto son ejecutados en el lado del cliente.
- Perl: *Practical Extraction and Report Language*, o Lenguaje Práctico para la Extracción e Informe, se trata de un lenguaje que toma características de C y otros muchos lenguajes de programación. Su estructura está basada en bloques como C, y fue muy adoptado por no tener limitaciones que otros lenguajes de script sí que tienen y por su destreza en el procesado de texto. Resulta curioso distinguir que la mayoría de distribuciones GNU/Linux cuentan nativamente con este lenguaje. Es ejecutado en el lado del servidor.
- PHP: *Personal Home Page Tools*, se utiliza generalmente para la creación de contenido para sitios web y últimamente también para cualquier otro tipo de programas, incluyendo aplicaciones con interfaz gráfica usando la biblioteca GTK+. Tiene mucha similitud con otros lenguajes como C y Perl y sus scripts son ejecutados en el lado del servidor.

Para el funcionamiento de esta aplicación se ha decidido por el empleo de PHP, cuyos motivos se describen a continuación:

- Anteriormente se ha explicado del por qué utilizar aplicaciones del lado del servidor y elección de estas, por lo que lenguajes como ActionScript y JavaScript quedan descartados. Aunque este último será empleado para algunas ligeras utilidades, como la comprobación previa de relleno correcto de datos por parte del cliente.
- El servidor en el que se va a alojar la aplicación cuenta con Sistema Operativo Debian GNU/Linux y servidor web Apache, por lo que el empleo de ASP se hace algo complicado y poco adecuado al ser una tecnología propia de ISS de Microsoft. Aunque se podrían emplear servidores web como Chilisoft ASP o InstantASP para Linux que permiten el uso de esta tecnología.
- Se persigue una fácil integración entre código y HTML, y PHP permite ser embebido en una página HTML de una manera muy fácil, y permitiendo una clara separación de códigos y datos, lo cual también supone una fácil cooperación entre diseñadores con HTML y programadores con PHP. Y por último es importante tener en cuenta que el empleo de Perl como CGI, y no con *mod_perl*, hace que este sea interpretado por un intérprete externo y no por un módulo de Apache (como ocurre con PHP en el caso del modo de instalación de este proyecto), lo que supone una mayor sobrecarga al tener que hacer uso de un fork/exec para ejecutar el intérprete.

2.4.3. PHP

2.4.3.1 Breve reseña histórica

PHP fue concebido en 1994 por Rasmus Lerdorf. Curiosamente sus primeras versiones no distribuidas al público fueron usadas en sus páginas web para mantener un control de quien consultaba su Currículum Vitae y ciertos datos como la cantidad de tráfico que la página recibía. Al año siguiente, 1995, se publicó la primera versión disponible al público, que se conoció como “Herramientas para páginas web personales”, o *Personal Home Page Tools*. Estas herramientas consistían en un analizador sintáctico muy sencillo y unas utilidades de libro de visitas, contador de estas y otras de menor tamaño. Meses después el analizador sintáctico fue reescrito y nombrado a PHP/FI versión 2, (FI era otro programa del desarrollador que procesaba datos de formularios), por lo que ya la herramienta para páginas web también contaba con un intérprete de formularios, además de soporte para mSQL. PHP/FI fue creciendo con gran rapidez y mucha gente comenzó a contribuir en el desarrollo de este pequeño proyecto inicial.

En 1997 el desarrollo del proyecto sufrió un profundo cambio al pasar a ser un proyecto mucho más organizado, reescribiéndose por ejemplo el analizador sintáctico, y creando las bases para la versión 3, la cual fue lanzada oficialmente en 1998. En 1999 se volvió a reescribir gran parte del código produciéndose el motor *Zend*, motor de procesamiento para la compilación y cifrado del código, desarrollado por Zend Technologies. Un año después, en Mayo de 2000, fue lanzado PHP 4, en Julio de 2004 se lanzó PHP 5 y en breve se espera el lanzamiento de PHP 6.

Actualmente están activos las actualizaciones y soportes para las versiones 4 y 5, y para cuando haga aparición la versión 6 se espera que entonces este soporte se amplíe a las tres.

2.4.3.2 Características

Cabe destacar que de las dos versiones actualmente activas, las mayores ventajas de la versión 5 frente a la versión 4 son el mejor soporte para Programación Orientada a Objetos, mejoras de rendimiento, mejor soporte para MySQL y XML, iteradores de datos y excepción de errores. Dado que el servidor en el que se va a instalar este proyecto de repositorio de artículos cuenta con la versión 4.4.4, las características se centrarán en esta versión:

- Hace posible la utilización de páginas web dinámicas en combinación con el motor de base de datos MySQL, aunque cuenta con soporte para una gran variedad de motores de bases de datos.
- Con él se pueden crear aplicaciones gráficas independientes del navegador, por medio de PHP y GTK, permitiendo desarrollar aplicaciones de escritorio si el Sistema Operativo en cuestión lo soporta.
- Es multiplataforma, pudiéndose compilar su código en diferentes arquitecturas como AMD64, i386, SPARC, etc.
- Permite la lectura y manipulación de datos desde diversas fuentes, como datos ingresados en formularios HTML.
- Tiene capacidad para expandir su utilidad con una gran variedad de módulos, como el empleado en este proyecto para la autenticación mediante directorio LDAP.
- Amplia documentación y soporte en su página web.

- Es libre, resultando una alternativa de fácil acceso para todos.
- Soporta técnicas de Programación Orientada a Objetos.
- Tiene una biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere de la definición de tipos de variables ni manejo detallado del bajo nivel.

2.4.3.3 Funcionamiento del Lenguaje Interpretado (PHP)

PHP se puede entender también como una extensión para servidores web, poniéndose entre cliente y servidor, lo que para este proyecto se podría representar mediante el siguiente esquema:

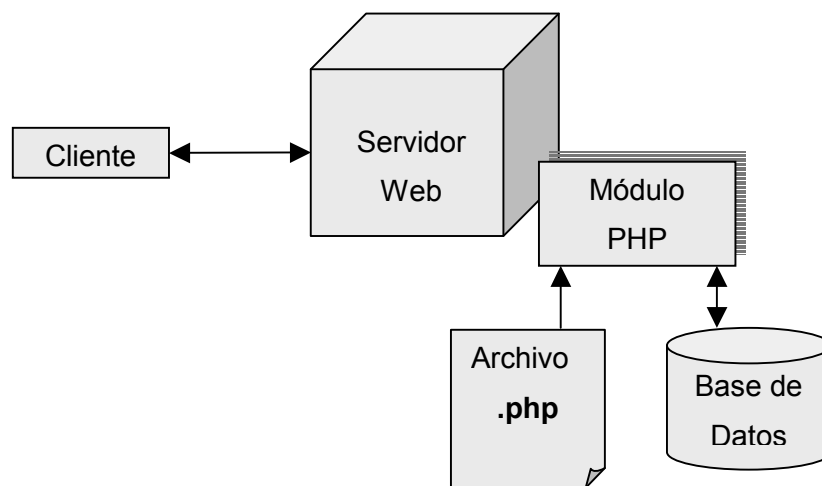


Figura 2.6 - Resumen de la Arquitectura utilizada

PHP se comporta como un módulo de Apache, que extrae código dentro de las páginas, lo ejecuta en el servidor y envía el resultado al cliente. El cliente no puede visualizar el código del programa, solamente su resultado. Además es importante destacar que al ser un lenguaje de script, los programas en PHP no se compilan, sino que únicamente se interpretan, por lo que es más lento en ejecutarse como programas hechos en C, que requieren estar compilados. No obstante, su gran ventaja es que un cambio en el código en PHP tiene el efecto inmediato, y en C por ejemplo, es necesario volver a compilarlo.

Resulta importante aclarar que sólo son interpretados por el módulo PHP, es decir pasados por parte del Núcleo de Apache a él, los archivos que hayan sido definidos en la configuración del servidor como tales, dado que la interpretación de código es un proceso que necesita un tiempo mayor que el empleado en enviar una página tal como esté. Normalmente se emplea la extensión *.php*, aunque también se utilizan otras como *.php3*, *.php4*, y *phtml*.

2.4.3.4 Instalación y puesta en marcha

Para instalar PHP en el servidor en el que se va a alojar la aplicación sobre la que versa este proyecto, y en concreto la versión que actualmente este tiene, sería necesario llevar a cabo los siguientes pasos:

1. Instalar el paquete de PHP versión 4 más reciente de los repositorios de Debian:

```
#apt-get install php4
```

2. A continuación también es necesario, al menos si se está usando Apache, instalar los módulos PHP para que Apache sea capaz de convertir el lenguaje PHP en HTML:

```
#apt-get install libapache-mod-php4 (para Apache 2 sería libapache2-mod-php4 y para PHP 5 libapache-mod-php5)
```

3. En muchas ocasiones suele ser necesario también la instalación de los módulos PHP para MySQL y LDAP (necesarios para este proyecto):

```
#apt-get install php4-mysql php4-ldap
```

4. Hay que editar el archivo php.ini. En este proyecto debiera de estar alojado en /etc/php.ini, descomentando las líneas, si es que no lo estuvieran:

```
extension=php_mysql.so  
extension=php_ldap.so
```

5. Por último, es necesario también editar el archivo /etc/apache/httpd.conf, descomentando las siguientes líneas (puede que no sean idénticas a las que contenta el archivo):

```
LoadModule php4_module /<la ruta que aparezca>/libphp4.so  
Addtype application/x-httpd-php .php  
Addtype application/x-httpd-php-source .phps
```


Capítulo 3.

Descripción de la aplicación

3.1 Introducción

A nivel físico, la aplicación está compuesta por archivos almacenados directamente en disco, tanto de contenido dinámico como estático, tales como scripts de PHP y JavaScript, documentos en HTML, hojas de estilo e imágenes, y los almacenados de manera indirecta a través de la Base de Datos.

El compendio de estos almacenes de información y manera de modelarla son los que harán posible el funcionamiento de la aplicación como a continuación se detalla.

3.2 Arquitectura de la aplicación

Como se ha explicado en el anterior capítulo, la aplicación esta compuesta por un servidor web, una base de datos y un módulo de lenguaje interpretado, cuyos detalles y funciones principales se detallan en los sucesivos apartados.

3.2.1 Servidor Web

Administra y atiende las peticiones de los clientes, enviándolas al módulo del lenguaje interpretado cuando sea necesario. Los archivos de la aplicación han de estar ubicados según donde indique la configuración del servidor, siendo imprescindible que el usuario correspondiente a Apache (o el servidor web empleado) tenga permisos de ejecución, escritura y lectura.

Desde la raíz de la arquitectura de directorios de la aplicación, los archivos están ubicados de manera agrupada según el tipo y función que desempeñan, distinguiendo:

1. Artículos: alberga los artículos en formato pdf, distribuidos estos a su vez en directorios internos a este, según el campo de investigación de cada uno.
2. Auxiliar: hojas de estilo, scripts de JavaScript y demás archivos de carácter secundario.
3. Imágenes: imágenes utilizadas, tanto para embellecer el aspecto de la aplicación como para facilitar su usabilidad.
4. Includes: scripts de PHP para ser incluidos a su vez en otros scripts, ahorrando espacio y tiempo ante modificaciones. Un claro ejemplo es el archivo de configuración de la aplicación, del que se comentará su función de manera detallada más adelante.

3.2.2 Base de Datos

Almacena la información referente a los artículos insertados en la aplicación, dotando de robustez y seguridad a la misma. A su vez posibilita la consulta de la información de una manera rápida y eficiente mediante el uso de sentencias en lenguaje SQL.

Se ha dotado de 3 tablas:

1. *Artículos*: contiene la información referente a cada entrada de la aplicación, tal como Título, Autor, Año, etc, y el idioma y campo correspondiente conforme a las tablas campos e idiomas.
2. *Campos*: contiene los campos de investigación sobre los que tratan los artículos almacenados, permitiendo con rapidez y sencillez el añadido de nuevos campos de investigación a la aplicación.
3. *Idiomas*: contiene los idiomas de los artículos, permitiendo de manera análoga a los campos la sencilla inserción de nuevos idiomas.

Los índices de las tablas campos e idiomas actúan como llaves foráneas de artículos, en sus columnas correspondientes al campo e idioma de cada entrada.

La modelación y relación entre las tablas se representa con el siguiente diagrama:

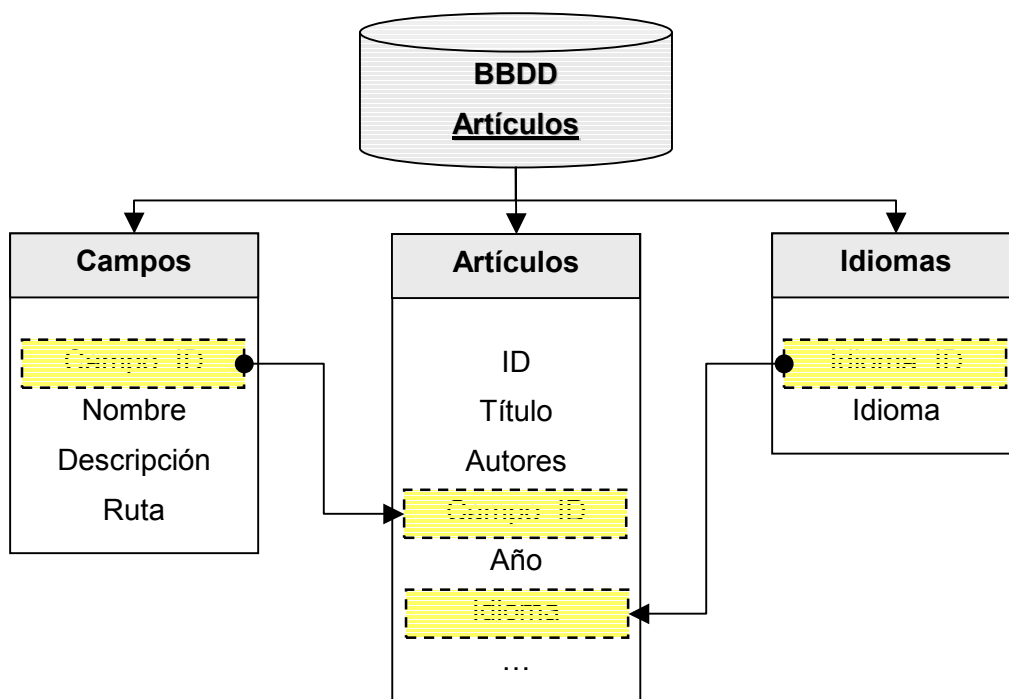


Figura 3.1 - Modelación y relación entre tablas de la Base de Datos

Es importante notar sobre la Base de Datos el uso de la tecnología **InnoDB** para la construcción de las tablas. Con ello se consigue integridad referencial. Esto permite, por ejemplo, borrar todos los artículos pertenecientes a un campo de investigación de la Base de Datos con tan solo borrar el campo en cuestión de la tabla *campos*. De manera análoga este ejemplo se puede observar con los idiomas.

El empleo de InnoDB supone que la rapidez de acceso a la información se pueda ver mermada. Se ha decidido su utilización, prefiriendo que fiabilidad e integridad de la información prevalezcan sobre velocidad, dado que para la finalidad de esta aplicación la demora es difícilmente apreciable desde el punto de vista del usuario normal.

3.2.3 Módulo de Lenguaje Interpretado

Se encarga de atender las peticiones por parte del Servidor Web, interpretándolas y construyendo la respuesta que va a ser destinada al cliente, llevando a cabo accesos a Base de Datos y a disco si es necesario.

Las funciones más generales que desempeña en la aplicación el módulo de lenguaje interpretado son:

1. Autenticación:

- Comprueba que se tengan permisos para el acceso a determinadas secciones, verificando que el usuario ha iniciado sesión.
- Se encarga de presentar en pantalla al cliente la necesidad de autenticarse así como también del proceso de inicio de la misma si no la hubiese y fuera necesaria, mediante el empleo del directorio LDAP de la Universidad.
- Gestiona las labores de cerrado y destrucción de las sesiones de usuarios.

2. Búsqueda:

- Requiere por pantalla el relleno de los parámetros de búsqueda y muestra los resultados.
- Permite visualizar los resultados tanto en formato normal como en referencia.
- Los resultados de la búsqueda se presentan de forma resumida con la información principal de cada entrada, pudiendo entrar en detalles si se está empleado formato normal. Con ello se ahorra gran cantidad de tráfico innecesario salvo que el usuario desee ver en profundidad los detalles de una entrada.
- Si el cliente está autenticado y fue aportador de la entrada, se ofrece la posibilidad de edición de la misma, y en caso de que pertenezca a los administradores de la aplicación se posibilita también la eliminación de la entrada en cuestión.

3. Borrado, edición y envío de nuevas entradas:

- Comprueba previamente que se esté autenticado de manera que permita:
 - a. Editar si el usuario fue aportador de la entrada.
 - b. Editar y borrar si el usuario es administrador de la aplicación.
- Edición y envío son dos acciones unificadas en una sola, con breves diferencias a nivel de introducir nuevos datos en la base de datos o actualizar los ya almacenados.

Las funciones de autenticación, búsqueda y editar/enviar cuentan con un archivo principal para cada una, que indica la presentación en pantalla de rasgos comunes, además de otros archivos que se le incluyen por medio de la función de PHP *include*. Los archivos que se les incluyen almacenan el código de las diversas funciones además de invocar al archivo de configuración de la aplicación.

Especial mención merece el archivo de configuración. En él se almacenan:

- Juego de caracteres actual, para un trabajo correcto con fechas
- Datos de conexión a directorio LDAP y Base de Datos
- Función de conexión a la Base de Datos que devuelve el resultado de la consulta que se le envíe como argumento.
- Función de conexión temporal a la Base de Datos para comprobar que la consulta que se realice no contenga un ataque [SQL Injection]
- Directorio en el que se van a almacenar los archivos PDF (en esta aplicación se le ha llamado *artículos*), tipos MIME aceptados así como el tamaño máximo permitido para el archivo que vaya a ser enviado al servidor.
- Número de entradas para cada paginación de los resultados de búsqueda, separados para formato de resultados normal y referencial.
- Nombre del archivo principal encargado de la autenticación así como el nombre de la sesión de usuario que el servidor crea para cada cliente. Con ello se consigue mayor dinamismo para enlazar desde cualquier otra sección a la de autenticación así como evitar conflictos entre sesiones desde un mismo navegador hacia el mismo subdominio.
- Lista de usuarios con privilegios de administradores, identificados por NIF.
- Año inicial desde el que se fechan los artículos, con ello se permite buscar por año sin que se puedan introducir años incoherentes ni haya que obtener de la Base de Datos un listado de los años en los que están fechados todos los artículos.

También es importante destacar el uso de la función GZIP de PHP (ver Apéndice A), incluida en todos los scripts de PHP de la aplicación. Con ella se consigue reducir el tamaño de la respuesta del servidor a los clientes, ya que se comprimen los documentos de hipertexto, y está ampliamente soportada por la gran mayoría de navegadores Web.

3.3 Aplicación en el lado cliente

La parte de la aplicación del lado del cliente ha sido orientada hacia una gran compatibilidad con todo tipo de navegadores:

- Cumpliendo el estándar del tipo de documento acorde al DTD empleado (XHTML 1.0 Transitional) dado por el consorcio [W3C] (*Word Wide Web Consortium*).
- Cumpliendo las reglas de uso de las Hojas de Estilo en Cascada versión 2.1 (CSS 2.1).
- Cumpliendo en todo lo posible los consejos de [TAW] (*Test de Accesibilidad Web*).
- Evitando emplear imágenes en las zonas que mayor accesibilidad requieren, como el título de la página en la parte superior, el nombre del área y menús. Con ello se fomenta también una mejor indexación por parte de los buscadores.

El cumplimiento de las recomendaciones impuestas por el consorcio W3C y el test TAW implica:

- Mejoras de cara a todo tipo de usuarios y navegadores Web
- Empleo de hojas de estilo en cascada así y código JavaScript, sin restar accesibilidad a usuarios que no dispongan de cualquiera de las dos funcionalidades.

Con ello se persigue garantizar en lo máximo posible una aplicación ampliamente compatible con multitud de plataformas, sistemas operativos y navegadores Web.

El cumplimiento de las recomendaciones de diseño también incluye el uso de colores agradables con el entorno y el usuario final. Para esta aplicación se ha decidido emplear un juego de colores similar al de la página Web del área de Ingeniería Telemática (<http://ait.upct.es>).

Además se ha dotado la parte superior de la página Web con dos botones para aumentar y reducir el tamaño de letra, sin tener que modificarla en las opciones del propio navegador Web. Con ello se ahorra en comodidad al cambiar el tamaño únicamente del documento de hipertexto que se esté visionando en ese momento.

Capítulo 4.

Puesta en marcha de la aplicación.

Manual del usuario.

4.1 Instalación

En el capítulo 2 se describió el procedimiento de instalación de Base de Datos, Servidor Web y Módulo de Lenguaje Interpretado. Partiendo de ello, el primer paso es la importación de la Base de Datos que se suministra con la aplicación.

4.2 Base de Datos

Con la aplicación se suministra un archivo de extensión SQL. Éste contiene la descripción de las tablas que forman la Base de Datos, con sus campos, tipos, índices, etc. La importación del archivo ha de llevarse acabo como a continuación se detalla:

A.1 Crear una Base de Datos:

```
#mysql -u usuario -p CREATE DATABASE `nombre_base_de_datos`  
DEFAULT CHARACTER SET latin1 COLLATE latin1_spanish_ci;
```

Es aconsejable emplear un Cotejamiento en español y CI (*Case Insensitive*), independiente de mayúsculas y minúsculas, como en el comando anterior se ha indicado.

A.2 Importar las tablas desde el archivo adjuntado con la aplicación:

```
mysql - usuario -p nombre_base_de_datos < archivo.sql
```

El Cotejamiento de las tablas tomará por defecto el empleado por la Base de Datos.

4.3 Servidor Web y Módulo de Lenguaje Interpretado

En lo referente a estas dos partes es necesario ubicar los archivos y directorios suministrados dentro del directorio que se haya elegido para contener la aplicación. Se han de tener en cuenta los siguientes aspectos:

- El usuario Apache, normalmente llamado *www-data*, ha de tener acceso de lectura y ejecución a todos los ficheros y directorios de la aplicación.

- El directorio en el que se albergarán los archivos PDF, y sus subdirectorios a su vez, tienen que permitir la escritura por parte del usuario empleado por Apache. Con ello se permite el almacenamiento de los archivos enviados, así como la creación de un directorio para almacenar un campo de investigación que sólo figurase en la tabla *campos* de la Base de Datos.
- Modificar el archivo de configuración de PHP (*config.inc.php*) de la aplicación indicando:
 - Datos de conexión a la Base de Datos y *nombre* de la misma.
 - Ruta en la que está ubicado el directorio que contiene a todos los artículos, originariamente llamado *artículos*.
 - Tamaño máximo permitido de los ficheros PDF a insertar junto con cada nueva entrada en la aplicación. Por defecto son 2MBytes.
- Tanto el archivo de configuración, como gran variedad de detalles de la aplicación, han sido orientados a GNU/Linux, dadas las exigencias de las bases de este proyecto.

Una vez tenidas en cuenta las recomendaciones anteriores, la aplicación se encuentra lista para ser utilizada.

4.4 Mantenimiento

En principio, salvo errores de funcionamiento inesperados, la aplicación se automantiene, salvo para las siguientes situaciones, para las que es necesario la intervención de los administradores:

- Añadir un nuevo campo de investigación, para lo que es necesario insertar un nuevo registro en la tabla *campos* de la Base de Datos.
- Eliminar una entrada, para lo cual se busca a través del formulario de búsqueda, estando identificado como usuario con permisos de administración, se pulsa sobre el botón de *borrar* correspondiente a la misma y se confirma la intención de borrado.
- Borrar un campo de investigación, para lo que se requiere borrar la entrada correspondiente al campo en cuestión de la tabla *campos*. Con ello se eliminarán automáticamente todas las entradas en la tabla *articulos* asociadas a dicho campo de investigación. Seguidamente será necesario borrar del árbol de directorios el directorio correspondiente al campo eliminado.

El control de acceso se regula mediante una consulta al directorio LDAP de la Universidad con el NIF y PIN de usuario al contenedor de Ingeniería Telemática. Con ello se ahorra espacio en la Base de Datos y la labor de mantener actualizado un listado de personal autorizado así como sus datos, como por ejemplo email.

No menos importante es mantener actualizados los tres componentes principales que hacen posible el funcionamiento de la aplicación (Apache, MySQL y PHP). Resulta muy aconsejable, sobre todo en materia de seguridad ante fallos e

intentos de acciones malintencionadas contra la aplicación y el servidor en su totalidad.

Esta labor se deja en manos del administrador del servidor donde se instale la aplicación, que no ha de coincidir con los administradores de la aplicación presentada en este documento. No obstante, para el caso del Sistema Operativo encargado del alojamiento que se ha presumido, Debian GNU/Linux, con la ejecución de los comandos siguientes debiera ser suficiente:

- `#apt-get update`
- `#apt-get upgrade`

Por último es aconsejable que con una cierta periodicidad los administradores comprueben que no hay un número de archivos PDF mayor a la cantidad de entradas almacenadas en la Base de Datos. Esta situación podría darse, debida a un fallo fortuito en las acciones de borrado, edición o envío, y que cuanto antes debería de ser corregida por si se estuviese imposibilitando el acceso a alguna información.

Capítulo 5.

Inserción y búsqueda de documentación. Manual del usuario

5.1 Introducción

En este capítulo se pretende contemplar el total de funciones que ofrece la aplicación de cara al usuario habitual, tanto autenticado como anónimo. Por motivos de seguridad y condiciones de este proyecto, la accesibilidad a las funcionalidades se divide en tres grupos, ordenados de menor a mayor grado de seguridad:

- Accesibilidad de usuarios **no autenticados**:
 - Página principal
 - Búsqueda
- Accesibilidad de usuarios **autenticados**
 - Búsqueda
 - Inserción
 - Edición (si el usuario es el autor de la entrada a modificar)
- Accesibilidad de usuarios **administradores**
 - Todas las opciones anteriores
 - Eliminación

Mientras que los usuarios no autenticados únicamente poseen accesibilidad de lectura a los contenidos, los autenticados pueden insertar nuevos contenidos además de editarlos si fueron publicados por ellos, y los administradores disponen de total funcionalidad de la aplicación, tanto la ya comentada, como borrar y editar cualquier contenido.

En la figura 5.1 se muestra un diagrama de bloques general de las distintas funcionalidades de la aplicación diseñada y la utilidad de las mismas.

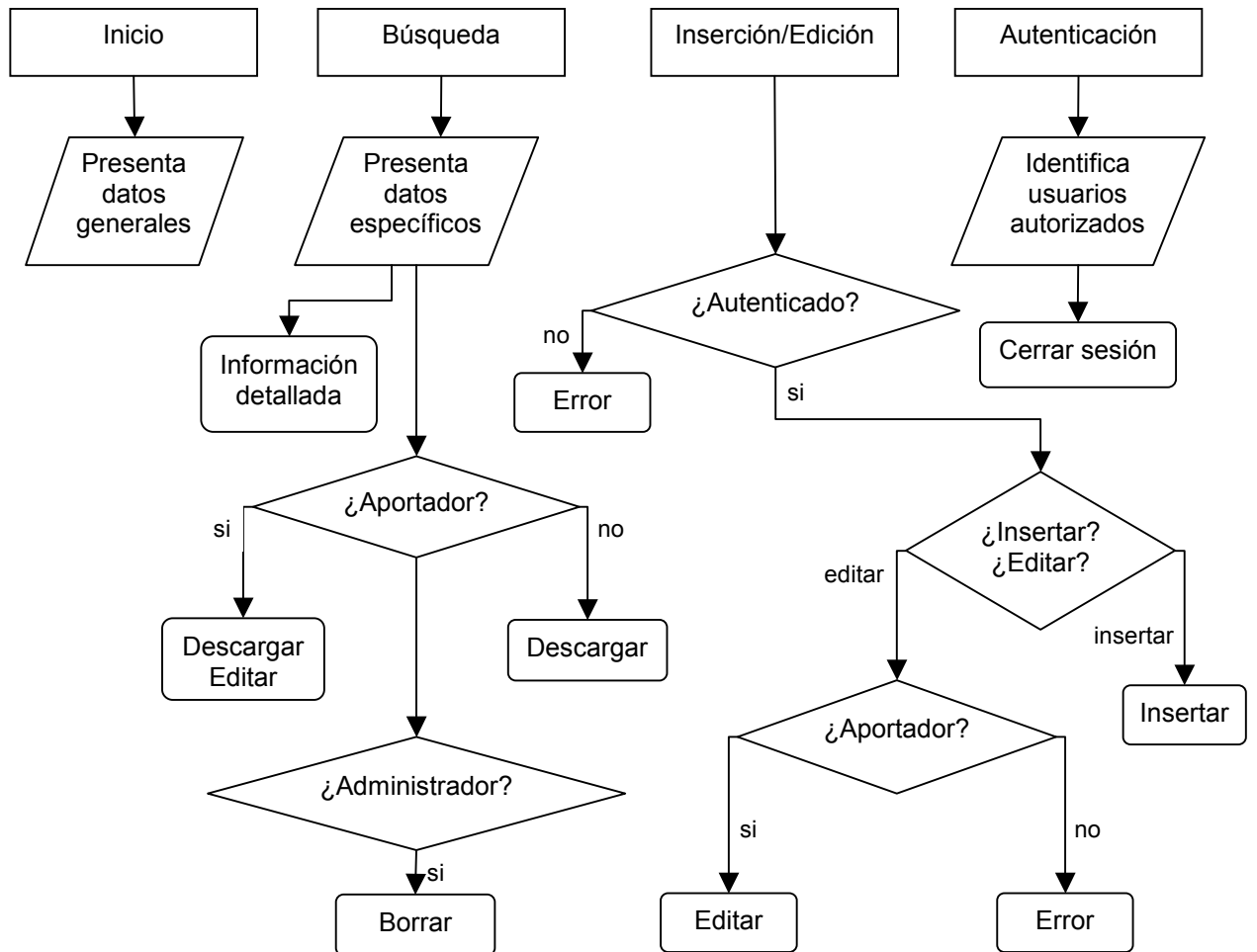


Figura 5.1 - Funcionalidades de la aplicación

Como se muestra en la figura 5.1, en la sección de inicio se puede visualizar información general acerca de la documentación almacenada, como son el total de registros así como campos de investigación tratados.



Figura 5.2 - Página principal de la aplicación

En el resto de capturas de la aplicación se van a obviar el menú y pié de página por no registrar cambios significativos a lo largo de las distintas secciones.

5.2 Autenticación

Una de las principales funciones de la aplicación consiste en la autenticación de usuarios. Para ello se dispone de una sección en la que se requieren al usuario NIF y PIN. En el caso de que la identificación sea satisfactoria al usuario se le otorgan acceso a funciones de uso restringido enviar y editar. Es muy importante destacar que para esta funcionalidad se requiere que el navegador Web acepte [cookies], (ver Apéndice A)

Figura 5.3 - Formulario de autenticación de usuarios

Los datos de identificación se cotejan con los alojados en el contenedor del área de Ingeniería Telemática del directorio LDAP de la Universidad. Por tanto, en principio la información a facilitar a la aplicación para la autenticación son NIF y PIN de usuario, (ver Apéndice A)

Una autenticación exitosa supone por tanto que durante 180 minutos el usuario disponga de las funcionalidades citadas con anterioridad, aunque cierre el navegador inclusive, siempre y cuando éste conserve las cookies. (ver Apéndice A).

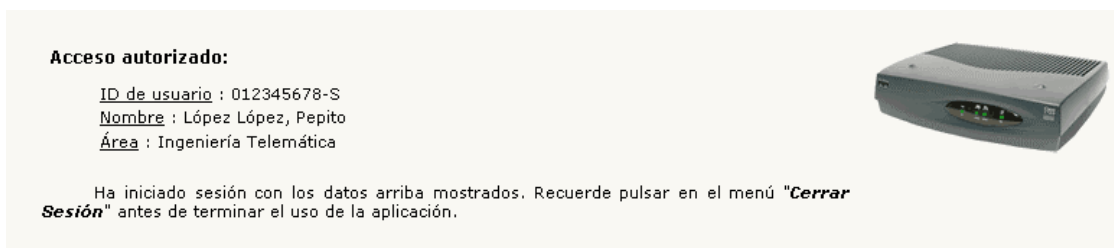


Figura 5.4 - Datos del usuario autenticado correctamente



Figura 5.5 - Verificación de sesión de usuario cerrada correctamente

Se ha dotado a la aplicación de una cuenta alternativa de administración, independiente del directorio LDAP y sólo utilizable en caso de fallo de éste. (ver Apéndice A).



Figura 5.6 - Indicación de error en la conexión con directorio LDAP

5.3 Búsqueda de documentación

La parte principal de las que comprenden esta aplicación es la búsqueda. En ella se muestran los resultados acordes a los parámetros de búsqueda introducidos por los usuarios y el formato elegido.

Los parámetros de búsqueda son:

- Título.
- Autor/es.
- Campo de investigación.
- Año de publicación del artículo.
- Palabras clave, o *keywords*.

Parámetros de búsqueda

Título:

Autor/es:

Campo:

Año:

Keywords:

MODO de mostrar resultados:
[Normal](#) • [Referencia](#)

ORDENACIÓN:
[Campo](#) • [Título](#) • [Autor](#) • [Año](#) • [Fecha de aportación](#)

Encontrados **1** resultado(s) ordenado(s) por **Fecha de aportación**

Figura 5.7 - Formulario de búsqueda de documentación

Cabe destacar:

- No es necesario indicar los campos con exactitud, se busca por palabras contenidas en los campos rellenos. Es decir, para encontrar un artículo titulado *A Real Time MAC Protocol for Wireless Sensor Networks: Virtual TDMA for Sensors(VTS)*, es suficiente con indicar en Título las palabras *wireless sensors*.
- Los resultados no son sensibles al orden de escritura de las palabras, por lo que se obtienen igual número de resultados al buscar, por ejemplo, por Título de *wireless sensors*, que *sensors wireless*.
- El motor de búsqueda no distingue entre caracteres acentuados y no acentuados, ni entre mayúsculas y minúsculas.
- Dejar uno o varios campos en blanco, y en el caso de los menús de selección en la opción *cualquiera*, supone siempre obtener unos resultados lo menos específicos posibles. A mayor cantidad de parámetros se obtendrán resultados más específicos a lo buscado.
- Los parámetros disponibles para búsquedas han sido utilizados por ser las mejores señas de identidad, y más fáciles de recordar, de cada artículo.
- La búsqueda con todos los campos sin rellenar, y los menús en la opción *cualquiera*, no se considera válida. Para ello se ha de emplear el botón Mostrar TODO.

De cada resultado se muestra (si fue relleno el campo), de manera común para los dos formatos de visualización:

- Primer autor, y resto de autores, según el artículo.
- Título
- Campo de investigación
- Congreso de publicación y las siglas del mismo, así como también y/o revista o libro, según el artículo.
- ISSN y ISBN
- Mes y Año de publicación.

Por otro lado, se han ideado dos formatos para mostrar los resultados:

- Normal: empleado por defecto, en el que los resultados procuran mostrarse simulando como si de fichas se trataran. Este formato permite

descargar el documento PDF que contiene el artículo publicado, así como editarlo y/o borrarlo (estas dos últimas prestaciones se muestran acorde a los permisos explicados en el anterior apartado). Al pie de cada ficha se muestra un enlace llamado *más detalles*, que muestra información adicional de cada artículo. En caso de no tener activado *JavaScript* (ver Apéndice B), esta última información será mostrada en una nueva ventana del navegador. Se muestran los iconos de edición y borrado porque el usuario actual tiene permisos de administración.

Esteban Egea-López, Javier Vales-Alonso, Alejandro S. Martínez-Sala, Joan García-Haro, Pablo Pavón Mariffo, M. Victoria Bueno-Delgado

WSN

A Real Time MAC Protocol for Wireless Sensor Networks: Virtual TDMA for Sensors(VTS)

Architecture of Computing Systems (ARCS) • Computer Science (Springer Verlag GmbH)

ISSN: 0302-9743 ISBN: 3-540-32765-7 Marzo 2006

« más detalles »

Figura 5.8 - Información general de una entrada

« menos detalles »

Abstract: Wireless Sensor Networks (WSN) are designed for data gathering and processing, with particular requirements and constraints: low hardware complexity, low energy consumption, special traffic pattern support, scalability, and in some cases, real-time operation. In this paper we present the Virtual TDMA for Sensors (VTS) MAC protocol, which intends to support the previous features, focusing particularly on real-time operation. VTS adaptively creates a TDMA arrangement with a number of timeslots equal to the actual number of nodes in range. Thus, VTS achieves an optimal throughput performance compared to TDMA protocols with fixed size of frame. The TDMA frame is set up and maintained by a distributed procedure, which allows sensors to asynchronously join and leave the frame. A major advantage of VTS is that it guarantees a bounded latency, which allows soft real-time applications. An expression for the upper latency bound is also provided in this paper. VTS performance is evaluated by simulation. Results show less power consumption than other proposals in the field. We also introduce a novel multi-hop operation by coordinated sleep/awake cycles among clusters.

Keywords: MAC, WSN, TDMA

Idioma: Inglés

Volumen: 127

Figura 5.9 - Información específica de una entrada (abstract, keywords e idioma)

« menos detalles »

Idioma: inglés

Volumen: 127

Páginas: 382

Lugar de Congreso/Simposium: Frankfurt (Germany)

Área: Ingeniería Telemática

Artículo propio del Área: si

Aportador/a: Bueno Delgado, M^a Victoria • MVictoria.Bueno@upct.es

Última edición por Bueno Delgado, M^a Victoria

Fecha de última modificación: 27/04/2007

Figura 5.10 - Información específica de una entrada (volumen, lugar, aportador, etc)

- Referencia: muestra prácticamente la misma información que el modo predeterminado, pero acorde al formato empleado en las referencias bibliográficas.

Esteban Egea-López, Javier Vales-Alonso, Alejandro S. Martínez-Sala, Joan García-Haro, Pablo Pavón Mariño, M. Victoria Bueno-Delgado, "A Real Time MAC Protocol for Wireless Sensor Networks: Virtual TDMA for Sensors(VTS)", *In Proc. Architecture of Computing Systems, (ARCS)*. Computer Science (Springer Verlag GmbH), Frankfurt (Germany), vol. 127, pp. 382, February 2006. ISSN: 0302-9743. ISBN: 3-540-32765-7.

Figura 5.11 - Resultado de búsqueda en modo Referencias

Los resultados de este tipo de búsqueda se organizan en páginas por comodidad del usuario y para evitar tener que recibir páginas de pesado tamaño, las cuales coinciden en numeración tanto para un formato como otro. Se puede estar viendo una determinada página de los resultados de una búsqueda, en modo normal, y si se conmuta a modo referencia los resultados mostrados serán los mismos y en el mismo número de página que en el anterior formato, y viceversa.

primera « 5 6 7 8 » última(16)

Figura 5.12 - Paginación de resultados

Así mismo, también se pueden visualizar los resultados acorde al orden que se escoja (como se ha mostrado anteriormente en la figura 4.7):

- Alfabético, para título, autor y campo de investigación.
- Por fecha de aportación, del más recientemente insertado al mas antiguo.
- Por año de publicación, de los más antiguos a los más modernos teniendo a su vez en cuenta los meses.

5.4 Inserción y mantenimiento de documentación

En este apartado se abordan las tareas de inserción, edición y borrado, de cara al usuario tanto normal como administrador, sin olvidar que requiere estar autenticado como persona autorizada para tales efectos.

Hay que tener en cuenta que los procesos de escritura, tanto en Base de Datos como en disco, sólo son llevados una vez se han comprobado satisfactoriamente todos los posibles errores que pudieran ocurrir.

5.4.1 Inserción

Cualquier usuario debidamente autenticado tiene permisos para insertar nueva documentación. Tras acceder por medio del menú a la sección de inserción (llamada *Enviar* en la aplicación), se encuentra con el siguiente formulario, cuyos aspectos más relevantes se explicarán a posteriori:

Formulario de envío

Consejos:

- debido a los diversos exploradores, evite en la medida de lo posible el uso de comillas, tanto simples como dobles
- tamaño máximo permitido para el archivo PDF: 2MBytes
- * : campos obligatorios
- ** : necesario rellenar al menos uno
- ***: rellenar sólo si se ha rellenado su campo desarrollado correspondiente

Datos generales

Título*:

Autor/es*:

Archivo (PDF)* Examinar...

Campo*: Idioma*:

Mes publicación: Año* publicación:

Datos revista/libro

Nombre Rev/Lib***:

Siglas Rev/Lib***: Editorial:

Volumen: Número: Página(s):

Datos congreso/simposium

Nombre Cong/Simp***:

Siglas Cong/Simp***: Lugar:

Datos de publicación

ISSN: ISBN: Procceding:

Keywords*:

Abstract:

Información adicional

Resumen:

Artículo del Área*: Si No

Figura 5.13 - Formulario de inserción de nuevos documentos

Debido al gran número de campos para rellenar que posee el formulario, se dan unos consejos sobre:

- Campos **obligatorios**, indistintamente del medio de publicación, como Título, Autor, Campo de investigación, Año, Archivo, etc. Acompañados de un asterisco (*).
- Campos **obligatorios en función de otros**, como son el caso de los campos correspondientes a una Revista o Libro y los correspondientes a un Congreso o Simposium. Es posible un artículo que cuente con información para ambos grupos de campos, pero el cumplimiento de

uno de estos grupos de campos, al menos, es obligatoria. Acompañados de dos asteriscos (**).

- Campos **requeridos unos a otros**, en función del relleno o no de otros. Por ejemplo, en el caso de querer indicar las siglas de un congreso es también necesario indicar el nombre completo del mismo; en cambio la acción inversa no requiere de la indicación de las siglas. Acompañados de tres asteriscos (***).
- Evitar introducir comillas, tanto simples como dobles, en los nombres de Revistas/Libros y Congresos/Simposiums, para evitar:
 - No ser tenidas en cuenta, dado que la aplicación las elimina de los nombres de archivo, evitando incompatibilidades entre navegadores al leer los archivos PDF en las búsquedas.
 - Correcciones por parte del módulo del lenguaje interpretado (PHP) para prevenir posibles ataques *SQL Injection* a la Base de Datos.

En caso un cumplimiento incorrecto del formulario, al enviarlo se informará de los errores cometidos por una ventana emergente de *JavaScript*, o en ausencia de *JavaScript* se recargará el formulario con los datos introducidos junto con una lista a su comienzo de los errores detectados. En la mayoría de situaciones, estos errores serán debidos principalmente a:

- Haber superado el tamaño máximo permitido para el archivo PDF (por defecto 2MBytes).
- No haber relleno alguno de los campos obligatorios.
- Formato de archivo PDF incorrecto.
- Interrupción de la transmisión por fallo en la red de comunicación.

Una vez relleno correctamente y enviado el formulario, el servidor informará sobre el éxito del proceso, o del error en concreto que pudiera haber ocurrido.

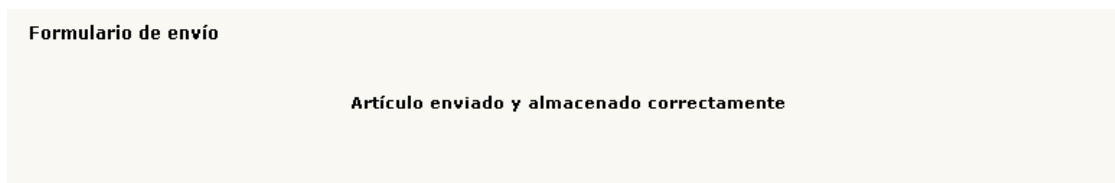



Figura 5.14 - Acuse de envío satisfactorio

5.4.2 Edición

El procedimiento de edición de la documentación existente es muy similar a la inserción de nuevas entradas, (ver Apéndice A). Para llevar a cabo esta labor, como se ha comentado con anterioridad, es necesario estar autenticado y ser aportador del documento y/o ser administrador de la Aplicación.

El formulario de edición aparece en una nueva ventana, y para acceder a él hay que pulsar sobre este botón en la ficha correspondiente al artículo que se 

desea editar, en la sección de resultados de las búsquedas.

La única diferencia entre el formulario de inserciones y el de edición radica en:

- El formulario de edición se muestra ya rellenado con los datos que se tienen almacenados de la entrada que se esté editando en cuestión.
- Junto al campo del archivo a introducir hay una casilla de verificación. De manera que, si se quiere reemplazar el archivo que actualmente hay en el Servidor Web, es necesario marcarla y seleccionar el nuevo archivo a enviar.

Los consejos sobre el cumplimentado del formulario y el archivo a enviar son similares a los tratados en el apartado anterior con la inserción de nuevas entradas.

Una vez se ha enviado la información modificada, si la actualización ha sido correcta se indicará por un mensaje en pantalla. En caso contrario se indicará el problema que concretamente ha ocurrido y las instrucciones a seguir. No obstante, en cualquiera de las dos situaciones podrá proceder a cerrar la ventana.




Figura 5.15 - Acuse de actualización satisfactoria

5.4.3 Borrado

Esta operación es, si cabe, la más peligrosa de todas las que puede llevar a cabo la aplicación. Esto es debido porque a diferencia de la edición, una vez borrada una entrada ya es irrecuperable, mientras que en la anterior si hay fallos en el proceso de actualización, el archivo anterior y los datos correspondientes en la Base de Datos seguirán íntegros.

Se recuerda que esta operación única y exclusivamente está disponible para usuarios con permisos de administración, obviamente autenticados previamente.

Para acceder al borrado, de manera similar a la edición, hay que pulsar  sobre este botón en la ficha correspondiente al artículo que se desea eliminar, de los resultados de la sección de búsqueda. Se abrirá entonces una ventana emergente de *JavaScript*, o una nueva página de navegación habitual en su defecto.

En dicha ventana se mostrarán Autor/es y Título del documento que se desea eliminar y un botón de verificación de dicha acción.

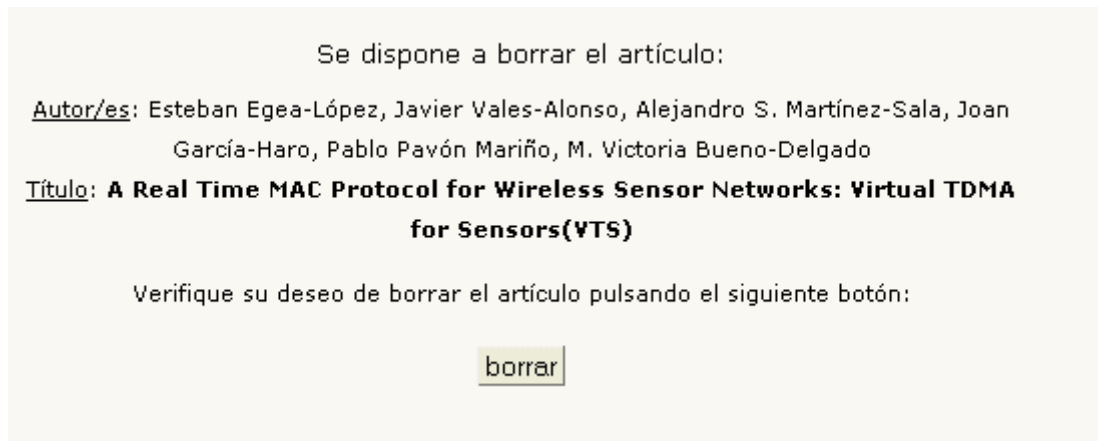


Figura 5.16 - Petición de confirmación de eliminación de un documento

Una vez se pulsa sobre el botón se mostrará un mensaje informando de si el documento ha sido borrado exitosamente o de lo contrario ha ocurrido algún error, en cuyo caso se darían indicaciones de las posibles soluciones. (ver Apéndice A).

Capítulo 6.

Conclusiones y líneas futuras.

6.1 Conclusiones

El desarrollo de la aplicación que se presenta en este proyecto surge de la necesidad de almacenamiento y organización de los recursos de investigación del Área de Ingeniería Telemática. Para ello, se propuso el diseño de un repositorio de recursos de investigación, centralizado y coordinado donde los investigadores del área pudieran almacenar en una base de datos los artículos de investigación generados así como realizar consultas, búsquedas y descargas de los mismos.

Se opta por el desarrollo de una aplicación que en un futuro próximo se instalará en un ordenador del área que realiza las tareas de servidor, concretamente en el servidor donde se aloja la Página Web del Área (<http://ait.upct.es>). Por tanto, esta aplicación estará orientada a dar mayor servicio y utilidad a la página principal de la mencionada área de investigación.

Para abordar el diseño de la aplicación propuesta, se han estudiado las posibles soluciones software, enumerando y destacando en el segundo capítulo de este documento aquellas que, por sus ventajas desde el punto de vista de la arquitectura, se han elegido para realizar la aplicación. En el mismo capítulo también se aborda, desde el punto de vista del diseño de la aplicación, los motivos que han llevado al autor a elaborar una aplicación desde el principio. Entre los motivos enumerados destaca la necesidad de obtener el máximo aprovechamiento de los recursos y servicios que ofrece el ordenador servidor del Área, así como optimizar las funcionalidades requeridas en este proyecto. Constando dicha aplicación por tanto de:

- Servidor Web Apache, encargado de atender y responder a las peticiones de los clientes, así como albergar y gestionar los documentos y demás archivos necesarios para hacer posible el uso de la aplicación.
- Base de Datos MySQL, cuya principal funcionalidad reside en almacenar la información de cada documento del área y realizar las tareas de motor de búsqueda.
- Módulo de Lenguaje Interpretado PHP, realiza las funcionalidades de intermediario entre Servidor Web y Base de Datos, además de otorgar dinamismo a la página al construir los documentos de hipertexto que van a ser enviados al cliente.

En el tercer capítulo se detalla la arquitectura de la aplicación desarrollada bajo las herramientas software elegidas en el capítulo anterior. La arquitectura de la aplicación se aborda tanto a nivel de servidor como cliente, destacando los siguientes aspectos:

- Secciones y principales funcionalidades
- Medidas de seguridad adoptadas, por parte de los contenidos almacenados y de los usuarios de la misma entre otras.

- Reducción en la medida de lo posible de los usos de recursos y ancho de banda disponibles del servidor Web.
- Compatibilidad con la mayor cantidad posible de navegadores Web así como de plataformas, en lo que al ámbito de los clientes se refiere.

En el cuarto capítulo se han tenido como principales objetivos la indicación de pasos a seguir, así como requisitos para una correcta instalación y configuración de la aplicación. Para el caso de la Base de Datos se puede utilizar phpMyAdmin, aunque por mayor compatibilidad se han indicado los pasos a seguir desde una consola de texto, y se adjunta con este proyecto un archivo con las tablas exportadas de la Base de Datos, apto para crear las tablas necesarias en la Base de Datos de destino y ponerla en marcha.

En el quinto capítulo se presenta un manual de usuario. En él se trata una descripción a nivel de usuario de la aplicación, sus secciones y procedimientos para las funcionalidades, tanto como usuario normal como autenticado.

Para abordar el desarrollo de este proyecto, han empleado las siguientes herramientas:

- UltraEdit, de IDM Computer Solutions, Inc. (editor de código)
- Adobe Dreamweaver (entorno IDE de edición Web)
- WinSCP (cliente SFTP)
- Nano (editor de texto, en el servidor)
- phpMyAdmin (herramienta para manejar la administración de MySQL)

6.2 Líneas futuras

La aplicación desarrollada pasará a formar parte de la Página Web principal del Área (<http://ait.upct.es>). Como principales líneas futuras se podrían contemplar:

- Implementación de un demonio de sistema encargado de la realización de un respaldo de la información en Base de Datos, así como contenidos almacenados en disco de la aplicación. Se podrían establecer estos respaldos con una periodicidad de cada siete días, por ejemplo.
- Ampliación completa de la aplicación para ser utilizada conjuntamente por otras áreas del departamento de Tecnologías de la Información y las Comunicaciones.
- Ampliación de los criterios de búsqueda, como pueden ser nombre de congreso o libro, agradecimientos, etc. Aunque el llevar acabo este cometido pueda suponer un mayor tiempo dedicado al relleno de campos en la inserción, o el ser capaz de desarrollar una aplicación que extraiga información contenida en los archivos en formato PDF.
- Posibilidad de poder marcar varios documentos conforme son visualizados como resultados de búsqueda, permitiendo posteriormente la descarga múltiple en un solo archivo de los resultados marcados.

- Posibilidad de suscripción por parte del investigador a un boletín de novedades por cada campo de investigación, recibiendo notificaciones con cierta periodicidad de los nuevos documentos añadidos en el repositorio.
- Menús en búsqueda e inserción que muestren los autores que ya figuran en los documentos almacenados. Facilitando y agilizando el relleno de sendos formularios.

Apéndice A.

Detalles del lado del Servidor

En esta parte se detallan y explican de manera más amplia ciertos aspectos del servidor que se han mencionado en el manual de usuario.

- A.1** Uso de cookies: el SID (Session ID, Identificador de Sesión) de las sesiones utilizadas por PHP se puede propagar por cookie o por URL. Para el caso de este proyecto se utiliza la propagación por cookie debido a que así está configurado en el servidor, además de ser un método más seguro que su alternativo mediante URL.
- A.2** Identificación cotejada con directorio LDAP: los datos de inicio de sesión introducidos por los usuarios que quieren autenticarse son comprobados con los alojados en el directorio LDAP de la Universidad. Con este mecanismo se logra:
- Menor sobrecarga de procesos para la Base de Datos.
 - Menor espacio requerido en Base de Datos, dado que sólo se almacenan nombres y NIF cuando se insertan nuevas entradas, en lo que a datos de usuarios respecta.
 - No tener necesidad de mantenimiento de un listado de personal docente así como sus datos.
 - Mayor facilidad y rapidez para añadir autenticación y almacenamiento de otras áreas o departamentos a la aplicación, concatenando los contenedores del directorio LDAP consultados por la aplicación para cotejar los datos de inicio de sesión.

No obstante, existe un método alternativo de autenticación, únicamente para labores de administración cuando la autenticación con el directorio LDAP no está disponible.

- A.3** Tiempo de sesiones: el tiempo de caducidad de cada sesión es de 180 minutos. Es el valor predeterminado por la configuración del servidor y no ha sido objeto de sugerencia de modificación, dado que parece adecuado para los fines de esta aplicación. Sin embargo, no deja de ser recomendable que cada vez que un usuario autenticado abandone la aplicación, cierre debidamente su sesión sesión.
- A.4** Procedimientos de Enviar/Editar similares: existen dos scripts, enviar y editar, pero ambos a su vez incluyen a uno mismo con las funciones mostrar formulario, validar y procesar. Los pasos a seguir resumidamente son (una vez comprobado que se poseen los permisos pertinentes):
- A.4.1 Comprobación mediante el archivo invocador de si se pretende enviar una nueva entrada o editar una ya existente.
 - A.4.2 Si se va a enviar se muestra el formulario sin rellenar. Si se va a editar se muestra rellenado con los datos de la Base

de Datos, y con un verificador por si se precisa cambiar el archivo almacenado.

A.4.3 Una vez se envían los datos, si ha habido errores se notifica al usuario mediante una ventana emergente por *JavaScript*, o se recarga la página indicando los errores (sin perder los datos introducidos previamente). Si no ha habido errores, se informa del correcto envío o actualización del artículo.

A.5 Procedimientos de eliminación de entradas: una vez comprobado que el usuario tenga los permisos adecuados, los procedimientos para borrar son:

A.5.1 Mostrar los datos principales de la entrada a borrar y se pide verificación al usuario para eliminar la entrada.

A.5.2 Se intenta borrar el archivo almacenado en disco.

A.5.3 Si el borrado en disco se realiza de manera correcta, se procede al borrado de la entrada en Base de Datos, de lo contrario se aborta el procedimiento y se informa al usuario.

A.5.4 La lógica de este orden reside en resultan más críticas las operaciones en disco que las referentes a la propia Base de Datos. Por ello se intenta primero borrar la parte de la información que más posibilidades de fallo pudiera tener, así en caso de error en el primer procedimiento de borrado no se perdería información alguna.

A.6 Compresión GZIP: al comienzo e inicio de cada script de PHP destinado a construir una página se incluyen dos scripts a su vez, encargados de comprimir la página generada. Con ello se reduce en la medida de lo posible el peso de las respuestas del navegador al cliente Web, y el 99% de los navegadores soportan esta función y descomprimen las respuestas para interpretarlas y mostrarlas a los clientes.

A.7 Mayor eficacia con SQL: dado que la Base de Datos emplea código ya compilado, se ha procurado maximizar en la medida de lo posible la cantidad de procesado declinado a la misma, porque sus procesados son más rápidos que los del módulo del lenguaje interpretado (PHP). Con ello se logra ahorrar procesos al módulo de PHP y una respuesta más rápida en la construcción de las páginas de respuesta al cliente.

Apéndice B.

Detalles del lado del Cliente

En esta parte se detallan y explican de manera más amplia ciertos aspectos del cliente que se han mencionado en el manual de usuario.

- B.1. Uso de cookies:** para un correcto uso de la aplicación como usuario autenticado, el navegador Web debe ser compatible y tener activada la aceptación de cookies. Ello es debido a limitaciones de seguridad por parte del servidor y las sesiones de usuario (ver Apéndice A), además de permitir no tener que introducir los datos de inicio de sesión cada vez que se visita la aplicación.
- B.2. DTD e idioma de las páginas:** Todas las páginas de la aplicación indican la definición del tipo de documento empleado para las mismas, así como su idioma, siguiendo las recomendaciones del consorcio W3C. Así mismo el seguimiento de las recomendaciones impuestas por el DTD otorgan a las páginas una mayor compatibilidad con todos los navegadores Web, ya que éste indica cómo interpretar cada etiqueta.
- B.3. Elementos META:** se emplean para describir las características de las páginas Web generadas por la aplicación. Principalmente en esta aplicación se usan para otorgar una descripción, palabras clave y fecha de caducidad a sus páginas, de cara a los buscadores de Internet generalmente, además de la codificación de los caracteres utilizada. En el caso de palabras clave se incluyen los campos de investigación albergados en la aplicación, generándose por tanto acorde a los campos almacenados en la tabla campos de la Base de Datos.
- B.4.** Para el caso de las páginas de acceso restringido, como es el caso de formulario de envío, página de datos del usuario actual autenticado y borrado, se emplean elementos META para indicar a los navegadores que no cacheen (almacenen temporalmente), los contenidos de las mismas, por labores de privacidad y seguridad.
- B.5. Atributos section:** empleados entre los elementos <head> y </head> para referir al visitante a documentos que actúan como secciones en la aplicación. En esta aplicación se han empleado estos atributos para cada una de las secciones que componen el menú, por ser las secciones principales de la misma. Adicionalmente hay navegadores Web que poseen menús propios de navegación de la página que se está visualizando, cuyos datos se extraen de estos atributos.
- B.6. Scripts de JavaScript:** se han utilizado estos scripts para el manejo de ciertos eventos de teclado y ratón sobre algunos componentes, permitiendo la realización de funciones adicionales que ayudan a un uso más fácil de la aplicación. Principales ejemplos de estas funciones son el permitir agrandar/reducir el tamaño de la fuente de las páginas pulsando sobre los botones correspondientes sobre el menú, y el despliegue de información detallada de cada resultado de la sección de búsqueda. No obstante, se ha tenido en cuenta que todas las funciones de accesibilidad en las que interfiere JavaScript, sean también accesibles si éste no es permitido por el navegador Web.

Parámetros de búsqueda

Título:

Autor/es:

Campo:

Año:

Keywords:

MODO de mostrar resultados:
[Normal](#) • [Referencia](#)

ORDENACIÓN:
[Campo](#) • [Título](#) • [Autor](#) • [Año](#) • [Fecha de aportación](#)

Encontrados **8** resultado(s) ordenado(s) por **Fecha de aportación**

Su navegador **no soporta JavaScript**, los detalles serán mostrados en una página nueva

Figura B.1 - Formulario de búsqueda con aviso del no funcionamiento de JavaScript

B.7. Hojas de Estilo en Cascada (CSS): describen fundamentalmente cómo se va a mostrar un documento en la pantalla. La elaboración de las páginas Web se ha diseñado de manera que información y formato de la misma no vayan unidos directamente. Para el caso de esta aplicación, los documentos visualizados en los navegadores Web contienen la información entre elementos y etiquetas del lenguaje HTML, y a su vez incluyen enlaces a las hojas de estilo que dicen cómo debe ser mostrada. De esta manera la información es igualmente accesible y entendible desde un mayor número de navegadores, aunque éstos no las soporten.

Acrónimos y descripción de términos relativos a la aplicación

[Opensource]: es el término con el que se refiere al software distribuido y desarrollado libremente, en español Código Abierto, y además su código fuente es accesible. <http://www.opensource.org/>

[CMS]: del inglés *Content Management System*, se trata de un sistema de gestión de contenido, que permite la creación y administración de contenidos, que principalmente se utiliza en páginas web. Suele controlar una o varias bases de datos en donde se almacenan los contenidos, y permite manejar de una manera independiente el contenido por una parte y el diseño por otra, haciendo posible, entre otras cosas, rápidos cambios en el diseño sin alterar en nada el contenido.

[Pdf]: *Portable Document Format*, o Formato de Documento Portátil, consiste en un formato de almacenamiento de documentos desarrollado por *Adobe Systems*, especialmente pensando para documentos susceptibles de ser impresos o cuyo formato se quiere mantener allá donde sea trasladado. <http://www.adobe.com/>

[WYSIWYG]: *What You See Is What You Get*, quiere decir “lo que ves es lo que obtienes”, y se aplica a procesadores de texto y otros editores con formato, que permiten escribir un documento viendo directamente el resultado final. Para el caso de editores de HTML este concepto se aplica a los que posibilitan la escritura sobre una vista preliminar del resultado final, generando el propio programa de generar el código en HTML. Un claro ejemplo de esto último son los campos de texto en muchos foros actualmente.

[RSS]: conocido como redifusión, forma parte de la familia de los formatos XML, y está desarrollado específicamente para todo tipo de sitios que se actualicen con asiduidad, a través del cual se puede compartir la información y usarla en otros sitios web o programas. Su mayor uso es para redifundir contenidos a suscriptores de un sitio web.

[GNU]: se trata de un proyecto con el objetivo de crear un sistema operativo completo libre, conocido como el sistema GNU. Este sistema fue diseñado para ser totalmente compatible con UNIX, lo que implica que ha de estar compuesto por pequeñas piezas de software individuales, de las cuales muchas ya estaban disponibles y otras tuvieron que ser reescritas. <http://www.gnu.org/>

[GNU GPL]: *General Public License*, una licencia pública general creada por *Free Software Foundation* en los años ochenta, y orientada principalmente a proteger la libre distribución, modificación y uso de software, es decir, proteger el software de apropiación que restrinja esas libertades a los usuarios. <http://www.gnu.org/licenses/gpl.html>

[ACID]:

- Atomicidad (*Atomicity*): cada transacción [transacción] debe tratarse de forma atómica, es decir, o se ejecuta todo o nada, ya que en todos los sistemas de la información una transacción no se puede realizar parcialmente.

- **Consistencia** (*Consistency*): para mantener la integridad y la coherencia de los datos, una transacción tiene que cumplir las restricciones que la base de datos posea, de manera que si las incumple no se permita la ejecución de ella.
- **Aislamiento** (*Isolation*): cada transacción ha de ser tratada como una unidad independiente, ejecutándola de forma concurrente con otras, por lo que dos transacciones sobre una misma información no generará error nunca.
- **Durabilidad** (*Durability*): una vez completada una transacción, sus resultados han de ser permanentes y persistir aunque falle el sistema.

[SQL]: *Structured Query Language*, se trata de un lenguaje de acceso a bases de datos que permite indicar las operaciones a realizar sobre la misma. Emplea características de álgebra y cálculo relacional, permitiendo llevar a cabo consultas de forma sencilla.

[Transacción]: conjunto de procesos ejecutados uno tras otro, que deben ejecutarse sólo una vez de forma completa, y que si alguno falla, lo realizado anteriormente debe reservarse para evitar la alteración de los datos.

[Bases de datos relacionales]: se trata del modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. En él no tienen relevancia el lugar ni la forma en que se almacenan los datos, lo que supone una mayor facilidad para entender y utilizar por parte de un usuario casual de la base de datos. Además, la información puede ser almacenada y/o extraída a través de “consultas” que ofrecen gran flexibilidad y poder para administrar la información.

[MySQL AB]: fundada en 1995, se trata de una compañía de software creadora del sistema administrador de bases de datos relacionales MySQL, además de una de las empresas más grandes de software libre a nivel mundial. <http://www.mysql.com/>

[mSQL]: también llamado *MiniSQL*, se trata de una base de datos para sistemas Unix de rápido acceso y almacenamiento simple, con unos requerimientos mínimos de memoria que la hace adecuada para publicar información en webs. Este sistema ofrece un subconjunto de sentencias SQL como interfaz de consulta. <http://www.hughes.com.au/>

[OpenGIS]: *Open Geospatial Consortium*, se trata de un consorcio creado en 1994 y que agrupa a organizaciones públicas y privadas. Su finalidad reside en conseguir acuerdos entre las diferentes empresas del sector, que permitan la interoperación de sus sistemas de geoprocésamiento y facilitar el intercambio de información geográfica en bien del usuario. <http://www.opengeospatial.org/>

[Debian]: se define a así mismo como un proyecto llevado a cabo por una asociación de personas que han hecho causa común para crear un Sistema Operativo *libre* (puede ser usado, copiado, estudiado, modificado y redistribuido libremente). Su nombre completo es *Debian GNU/Linux*, y nació como una apuesta por separar en sus versiones el software libre del no libre, con un modelo de desarrollo independiente de empresas, creado por los propios usuarios, sin depender de necesidades comerciales. Además cuenta con varias distribuciones, desde la más estable, llamada *stable*, hasta la más experimental que se trata más bien de un repositorio de paquetes, pasando por la inestable (*unstable*) y de pruebas (*testing*). <http://www.debian.org>

[phpMyAdmin]: se define a así mismo como una herramienta escrita en *PHP* para administrar MySQL a través de páginas Web, que actualmente permite crear, editar y eliminar bases de datos, tablas y registros, ejecutar cualquier sentencia SQL, administrar las llaves (*keys*) de los campos, privilegios y exportar datos a varios formatos. PhpMyAdmin está disponible en multitud de idiomas, entre ellos español, y además cuenta con licencia GPL. <http://www.phpmyadmin.net/>

[HTTP]: del inglés *HyperText Transfer Protocol*, sus creadores lo describen como un protocolo de nivel de aplicación orientado a sistemas distribuidos, para la colaboración e hypermedia. Un protocolo genérico, sin estado, orientado a objetos y que puede ser utilizado para muchas aplicaciones como servidores de nombres y sistemas de gestión de objetos distribuidos, a través de las extensiones de los métodos de petición. Una característica de este protocolo es la negociación de los tipos y representación de los datos, permitiendo que los sistemas no dependan del tipo de datos que se utilicen.

[plugins]: son aplicaciones que interactúan con una aplicación para aportarle una función o utilidad muy específica. Esta aplicación adicional es ejecutada por la aplicación principal, y las típicas tienen la función de reproducir determinados formatos de gráficos, datos multimedia, filtrar imágenes, ect. Otra manera de entender a los plugins consiste en considerarlos añadidos a la expansión de programas de forma modular, añadiendo nuevas funcionalidades sin afectar a las ya existentes ni complicar el desarrollo del programa principal.

[NCSA]: *Nacional Center for Supercomputing Applications* (Centro Nacional de Aplicaciones de Supercomputación). Se trata de un organismo de Estados Unidos relacionado con la investigación en el campo de la Informática y las Telecomunicaciones, el cual jugó un papel muy importante en el desarrollo de Word Wide Web. <http://www.ncsa.uiuc.edu/>

[parches]: consisten en modificaciones llevadas a cabo en programas y tienen el objetivo de sustituir una parte del código por otra, con el fin de eliminar un error anterior de programación.

[MIME]: *Multipurpose Internet Mail Extensions*, es decir, Extensiones de Correo Internet Multipropósito. Consiste en una serie de especificaciones dirigidas a que se puedan intercambiar todo tipo de archivos a través de Internet de una manera transparente para el usuario. Actualmente se considera incompleto cualquier programa de correo electrónico o navegador de Internet que no acepte MIME en sus facetas tanto de texto como de formatos de archivo.

[InnoDB]: tecnología de almacenamiento de datos para MySQL cuya principal característica consiste en el soporte de transacciones de tipo ACID y bloqueo de registros e integridad referencial. Con ello se posibilita una alta consistencia y fiabilidad mayor que la de la otra tecnología más común denominada MyISAM. <http://www.innodb.com/>

[SQL Injection]: Vulnerabilidad consistente en la inserción de una sentencia SQL dentro de otra sentencia SQL a su vez, por la cual un usuario malintencionado puede leer o modificar cualquier dato contenido en la Base de Datos.

[W3C]: *Word Wide Web Consortium*, consorcio internacional productor de estándares para los navegadores web. Estos estándares pasan por varios estados hasta que llegan a su aprobación y consecuente recomendación. <http://www.w3.org/>

[TAW]: *Test de Accesibilidad Web*, consiste en una herramienta desarrollada por CTIC (Centro Tecnológico de la Información y la Comunicación) para el análisis de la accesibilidad de los sitios web, alcanzando de una manera global e integral a todos los elementos y páginas que componen dicho sitio. Con ello se persigue permitir el acceso a todas las personas a ese sitio, independientemente de sus características diferenciadoras. <http://www.tawdis.net/>

[cookies]: Consisten en archivos de texto de un tamaño sumamente pequeño, que se almacenan en los clientes de las páginas webs mediante sus navegadores Web, por petición de los servidores Web visitados. La información guardada puede ser recuperada a posteriori en futuras visitas. Con ello sus mayores utilidades residen en guardar información como nombre de usuario, contraseña, preferencias de la página, etc, del cliente, que el servidor recuperará e interpretará de manera automática en sucesivas visitas.

Referencias

[Novell] Web de iFolder: <http://www.novell.com/products/ifolder/>

[XOOPS] Web oficial: <http://www.xoops.org>

[TYPO3] Web oficial: <http://typo3.com>

[Mambo] Web oficial: <http://www.mamboserver.com>

[PHP-Nuke] Web oficial: <http://phpnuke.org>

[Fundamentos de bases de datos]: Korth, Henry F.. "Fundamentos de bases de datos", 2ª edición en español. McGraw-Hill, D.L. 1995, Madrid.

[Oracle] Web principal: <http://www.oracle.com>

[MS SQL Server] Web principal en español:
<http://www.microsoft.com/spain/sql/default.mspix>

[PostgreSQL] Web principal: <http://www.postgresql.org>

[Apache] Web del proyecto Apache: <http://www.apache.org>

[ISS] Web del producto: <http://www.microsoft.com/spain/technet/productos/iis/default.mspix>

[Cherokee] Web del proyecto: <http://www.cherokee-project.com/>

[Lighttpd] Web oficial: <http://www.lighttpd.net/>

[Thttpd] Web oficial: <http://www.acme.com/software/thttpd/>

[ActionScript] Web dedicada a este lenguaje: <http://www.actionscript.org/>

[ASP] Web oficial: <http://msdn2.microsoft.com/en-us/library/ms644563.aspx>

[JavaScript] Manual de referencia por parte de SUN: <http://docs.sun.com/source/816-6408-10/contents.htm>

[Perl] Información completa acerca del mismo: <http://www.perl.org/>

[PHP] Desarrollo de PHP: <http://www.php.net>

No sé cómo denominarlos, porque son referencias bibliográficas que he ido consultando. En uno de los artículos he visto que se ponen numeradas, así que de momento las pongo así que algo es algo. (las webs como apache o mysql aquí también las pongo o no porque ya las he puesto en los enlaces de arriba???)

1. Foros del Web: <http://www.forosdelweb.com>
2. Desarrollo Web: <http://www.desarrolloweb.com>
3. MySQL Hispano: <http://www.mysql-hispano.org>
4. MySQL 5.0 Reference Manual: <http://dev.mysql.com/doc/refman/5.0/es/index.html>
5. Apache HTTP Server Version 1.3: <http://httpd.apache.org/docs/1.3>
6. Welling, Luke; Thomson, Laura. "Desarrollo Web con PHP y MySQL" Ed. Anaya Multimedia, 2003. ISBN: 8441515697

7. Revisando la Accesibilidad con Estilo: <http://www.sidar.org/hera>
8. Servicio de Validación de CSS del W3C: <http://jigsaw.w3.org/css-validator>