



ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

Curso Académico 2008/2009

Proyecto de Fin de Carrera

**Generación de una Web dinámica
para el Laboratorio de Tecnologías de la
Información en la Educación LITE**

**Autor: Juan Ramón Prats Yusty
Tutor: Jaime Urquiza Fuentes**

Resumen:

El objetivo del siguiente proyecto es la transformación de la actual Web estática del laboratorio LITE en una Web dinámica conservando en todo momento la apariencia.

Como se indica más adelante en los requisitos, esta nueva Web funciona con Apache como servidor Web y con MySQL para la gestión de los datos.

A continuación se describen los pasos que se definieron durante el desarrollo del proyecto

Inicialmente se procede al análisis y recogida de datos para poder conocer el ámbito del proyecto, efectuar una evaluación del esfuerzo asociado y conocer las necesidades del personal que va a usar esta Web.

Tras obtener dichos datos, se procede a la instalación de un entorno que permita el desarrollo del proyecto de una forma cómoda, segura y estable.

Una vez instalado el entorno de desarrollo, el siguiente paso es efectuar una Web de prueba para aprender a usar el entorno y probar el resultado previo al desarrollo final.

Con los datos obtenidos en el proceso de análisis y recogida de datos, se procede a modelar la base de datos.

Una vez disponemos del modelo de la base de datos, se procede a generar al código (PHP) que sustente dicha Web.

Un paso decisivo, una vez la Web está operativa en el entorno de desarrollo y los datos cargados en la base de datos, es la ejecución de una batería de pruebas para comprobar que el resultado es el deseado. Se complementan con pruebas en otros entornos, para poder efectuar la depuración y corrección de errores.

Finalmente, se procede a generar la documentación asociada al proyecto, así como explicaciones sobre el código, el manual de operación para el funcionamiento durante el día a día en producción y la descripción de las funcionalidades, para facilitar su despliegue y permitir su ampliación en un futuro.

Índice

CAPÍTULO 1. INTRODUCCIÓN Y OBJETIVOS	4
1.1 OBJETIVOS	4
CAPÍTULO 2. ANÁLISIS DE REQUISITOS	6
CAPÍTULO 3. INSTALACIÓN DEL ENTORNO DE DESARROLLO.	9
3.1 SELECCIÓN DEL ENTORNO DE DESARROLLO	9
3.2 INSTALACIÓN Y MANEJO DE LA PLATAFORMA APACHE/MYSQL/PHP	10
3.3 APRENDIZAJE DEL ENTORNO DE DESARROLLO	13
3.3.1 Base de datos	13
3.3.2 Esquema de página de prueba.....	19
CAPÍTULO 4. DISEÑO.....	22
4.1 MODELO DE DATOS DE LA BASE DE DATOS.....	22
4.1.1 Diseño de la Base de Datos	22
4.1.2 Creación de la base de datos.....	26
4.1.3 Creación del documento de conexión	26
4.2 IDENTIFICACIÓN DE LAS OPERACIONES	27
CAPÍTULO 5. IMPLEMENTACIÓN.....	29
5.1 DESARROLLO EL WEB SITE PARTIENDO DEL EXISTENTE.....	29
5.2 GENERACIÓN DEL WEB SITE DE ADMINISTRACIÓN	31
5.2.1 Acceso a la Web y funciones varias.....	31
5.2.2 Altas, bajas, modificaciones y listados	32
5.2.3 Apartado de líneas de investigación.....	33
5.2.4 Apartado de publicaciones	34
CAPÍTULO 6. CONCLUSIONES	35
6.1 HITOS	35
6.2 POSIBLES TRABAJOS FUTUROS	35
BIBLIOGRAFÍA Y REFERENCIAS.....	37
ANEXO A. MANUAL DE OPERACIÓN	39
INSTALACIÓN Y PUESTA EN MARCHA DEL ENTORNO (BAJO WINDOWS)	39
WEB DEL LABORATORIO	41
PORTAL DE ADMINISTRACIÓN.....	42
ANEXO B. CÓDIGO ASOCIADO A LA BASE DE DATOS	45
GENERACIÓN DE LA BASE DE DATOS	45
CREACIÓN DE LAS TABLAS	45
CARGA INICIAL DE DATOS	51
ANEXO C. CÓDIGO ASOCIADO A LA WEB	52
A.1 WEB PÚBLICA DEL LABORATORIO	52
A.1 PORTAL DE ADMINISTRACIÓN	54
ANEXO D. INSTALACIÓN DEL ENTORNO PARA LAS PRUEBAS FINALES	57
ANEXO E. CONTENIDO DE LA MEMORIA USB. PUESTA EN MARCHA DEL ENTORNO DE PRUEBA	60
ESQUEMA DEL CONTENIDO DE LA MEMORIA USB	60
PUESTA EN MARCHA DEL ENTORNO CONTENIDO EN LA MEMORIA USB FACILITADA, PARA ENTORNOS WINDOWS.....	61
ANEXO F. CONFIGURACIÓN DEL SERVIDOR APACHE	62

Capítulo 1. Introducción y objetivos

Este proyecto surgió a raíz de la necesidad de simplificar la actualización de contenidos en la Web del laboratorio LITE.

El funcionamiento actual obligaba a modificar el mismo contenido en diversas partes de la Web, lo que generaba un mayor trabajo administrativo y una mayor posibilidad de errores en la transcripción. Así mismo, datos como el número de proyectos, debían ser indicados manualmente dependiendo del contenido de otras páginas.

Cambiando a una Web dinámica y mediante el uso de un portal de administración, se puede modificar dicho contenido de una manera centralizada, lo que implica un menor esfuerzo, así como un mayor control gracias a la existencia de una única entrada o modificación por cada conjunto de datos. Esto, para un volumen alto de datos o modificaciones, reduce enormemente la posibilidad de errores y el trabajo de edición.

Por otro lado, el uso de MySQL [1] y PHP [2], ambos desarrollados como software libre [3], permite una sencilla administración y facilita posibles modificaciones en el futuro que respondan a nuevas necesidades del laboratorio.

1.1 Objetivos

Tras algunas reuniones, el objetivo de este proyecto de fin de carrera se fijó en la modificación de la actual Web del laboratorio de LITE, que era una Web estática, para generar una nueva Web. Esta nueva Web integra el contenido dinámico obtenido de una base de datos con el fin de facilitar la actualización de contenidos pero que para el usuario final sea idéntica a la Web anterior.

Así mismo, es necesaria la creación de una segunda Web mediante la cual, a través de una página de administración, los responsables de los contenidos de la Web del laboratorio puedan, entre otras acciones, efectuar altas, bajas, modificaciones y listados de diversos datos, como por ejemplo, de los propios usuarios para la gestión de dicha Web, personal asociado al departamento, tanto profesores como alumnos, proyectos, publicaciones, etc.

Tras las primeras reuniones, se fijaron los siguientes pasos para el desarrollo del proyecto.

1. Análisis de requisitos y recogida de información.
2. Plataforma de desarrollo:
 - Selección de la plataforma.
 - Instalación y manejo de la plataforma Apache/MySQL/PHP.
 - Aprendizaje. Realización de una página de prueba para:
 - i. Insertar información nueva o eliminarla en la base de datos.
 - ii. Consultar información de una base de datos.
 - iii. Modificar información existente en una base de datos.

3. Diseño del sistema:

- Modelo de datos de la base de datos. Modelar la información contenida en la base de datos. Esta base de datos tendrá los campos identificados en el punto 1 (Análisis de requisitos y recogida de información).
- Identificar las operaciones (inserción, eliminación, modificación y/o consulta) sobre la base de datos según el modelo de datos.
- Seleccionar qué información se mostrará en cada página.
- Identificar las páginas para insertar/modificar/eliminar contenidos.

4. Implementación:

- Desarrollar el Web site partiendo del existente: <http://lite.etsii.urjc.es>
- Generación de Web site de administración del entorno.

Capítulo 2. Análisis de requisitos

Tal y como se ha indicado en los objetivos, tras revisar los requisitos, se llegó a la conclusión de que el objetivo final para este proyecto no sería la creación de una nueva Web para la gestión de proyectos de fin de carrera, sino que el trabajo se centraría en la modificación de la actual Web del laboratorio de LITE para que su contenido fuera dinámico y se obtuviera de una base de datos, que a su vez se gestionaría desde una Web de administración.

Los requisitos funcionales (RFXX) recogidos durante el arranque del proyecto fueron los siguientes:

RF01: Formato de la Web del laboratorio.

- El diseño gráfico de la Web ha de ser idéntico al actual del laboratorio; en principio se trabajará sobre la actual Web, efectuando las modificaciones pertinentes para poder usar los datos contenidos en la base de datos.

RF02: Datos de personal.

- En el listado de detalles del personal asociado al departamento han de aparecer únicamente los datos de cada persona seleccionada, a diferencia de la Web actual, en la que aparecen los datos de todos los usuarios.
- En los proyectos de fin de carrera, cuando se muestran los tutores, ha de aparecer un enlace a su información personal.

RF03: Campos vacíos.

- No han de aparecer secciones vacías, esto es:
 - En personal, categorías que no tengan asignado personal.
 - En publicaciones, sublíneas de investigación que no tengan publicaciones asociadas.

RF04: Navegación.

- El enlace de cada evento debe abrirse en una ventana nueva.
- El enlace provisto por cada publicación debe abrirse en una ventana nueva.
- Los enlaces de las columnas laterales han de abrirse en la ventana original.

RF05: Orden en el listado de eventos.

- Los eventos han de ser mostrados por cercanía a la fecha actual, primero los actuales, después los futuros y por último los pasados.

RF06: Jerarquía de las líneas de investigación.

- Las líneas de investigación están divididas en sublíneas, éstas en apartados y estos a su vez en subapartados.

RF07: Datos gestionados desde la Web de administración.

- El contenido que ha de ser modificado desde la Web de administración para poder ser usado posteriormente en la Web pública está formado por:

- Usuarios de gestión de la propia base de datos (no usado en la Web pública).
- Personal (siendo estos miembros tanto profesores como becarios).
- Proyectos de investigación.
- Publicaciones.
- Líneas de investigación.
- Herramientas.
- Noticias.
- Eventos.
- Proyectos de fin de carrera.

RF08: Mapa Web.

- En la medida de lo posible el contenido del mapa Web ha de ser dinámico, esto es, el mapa Web tiene que generarse a partir de los datos necesarios contenidos en la base de datos.

RF09: Imágenes de herramientas.

- Se han de mostrar un máximo de dos o tres imágenes por herramienta. Finalmente se deja en cinco imágenes por si en un futuro fuera necesario mostrar más.

RF10: Datos a incluir.

- Los datos a incluir son los mostrados en la web actual. Su estructura y detalle se pueden encontrar en el Capítulo 4.

Al tratarse de una modificación de la Web existente, los requisitos detallan principalmente las modificaciones a realizar sobre dicha Web, así como características del modelo de datos a generar.

El proceso de desarrollo cíclico ha dado lugar a varias fases de pruebas en la que se han generado nuevos requisitos y cambios. A continuación detallamos las solicitudes de cambio (SCXX) que surgieron durante dicho desarrollo:

SC01: Solución de problemas de apariencia debido al uso de diferentes navegadores.

- Debido a que se ha mantenido el diseño original de la Web, que venía definido por plantillas, han surgido problemas como encuadre de texto. Estos problemas se han ido solventando con soluciones como, por ejemplo, la generación de un campo nombre abreviado para ciertos datos. De esta manera, en ciertos campos en los que, debido a su longitud los datos no aparecen correctamente, se puede proporcionar un texto alternativo, más corto, que evita que la página se muestre incorrectamente (ver SC05). Otros ejemplos han sido la revisión de etiquetas <DIV>.

SC02: Navegación.

- En los proyectos de fin de carrera (en la página *jobs.php*, que lleva a proyectos fin de carrera), la página ha de abrirse en la ventana original.

SC03: Confirmación de acciones en la Web privada.

- Cuando se solicite el borrado de algún dato, es necesario que primero pida confirmación, mostrando los datos del elemento que se va a borrar. Una vez borrado, dar mensaje de notificación.
- También aplica la aparición de mensajes de notificación de acción completada cuando se modifica o crea uno nuevo.

SC04: Ficheros adjuntos en apartado Publicaciones.

- La publicación ha de poder subirse a la Web y este fichero debe ser descargable (es necesario poder adjuntar fichero en la parte Publicación).

SC05: Nombre abreviado para menús.

- Dado que en algunos casos en el menú izquierdo podrían aparecer nombres excesivamente largos que pudieran no caber, es necesario insertar un campo nuevo que contenga el nombre abreviado en las tablas correspondientes (categorías, líneas y sublíneas de investigación, herramientas). Este será el campo que se muestre en los menús del lateral izquierdo. Esto se puede también aplicar al mapa Web. Se comprobó que el comportamiento para los nombres demasiado largos era distinto según el navegador empleado, en unos casos el texto no aparecía completo mientras que en otros sí. Mediante el uso del nuevo campo abreviado, al poder usar un texto más corto, estos problemas desaparecen y el contenido se muestra correctamente.

SC06: Gestión de usuarios.

- Se pide que se retire la contraseña del listado de usuarios, para que no sea visible al generar dichos listados. Solo será visible en el momento de cambiarla.

SC07: Categorías.

- Se considera conveniente permitir una gestión de posibles titulaciones similar a las categorías en el personal.

SC08: Notificaciones en el eliminado de categorías.

- Si la categoría tiene personal asignado, ha de generar un mensaje indicando que, para eliminarla, hace falta previamente eliminar o reasignar el personal asociado.

Capítulo 3. Instalación del entorno de desarrollo.

En el siguiente capítulo se detalla el entorno y las versiones de cada producto seleccionado para el desarrollo del proyecto y el motivo de dicha elección. Posteriormente, se describe la instalación y el manejo del entorno seleccionado previamente. Por último, se describen las fases que se han seguido para el aprendizaje y manejo del entorno.

3.1 Selección del entorno de desarrollo

Para el desarrollo del proyecto y para una mayor comodidad se ha usado la suite Portable Apps. [4]

Este paquete permite utilizar distintos programas, tales como editor y procesador de texto o navegadores para la edición y navegación (comprobación del correcto funcionamiento durante el desarrollo), así como el paquete XAMPP, que lleva incluido un servidor Web (Apache) y un servidor de base de datos (MySQL)¹.

Todo esto se realiza mediante la ejecución desde dispositivos de almacenamiento extraíbles, tales como memorias USB (pen drive) sin necesidad de instalación, por lo que es posible usarlo en cualquier PC basado en Windows sin tener que instalar ni modificar ninguna configuración en el sistema, lo que ha permitido una gran flexibilidad a la hora de trabajar.

Así mismo, el uso del servidor MySQL y la programación en PHP, requisitos imprescindibles del proyecto, permiten, por un lado una gran portabilidad y, por otro lado, no requieren el uso de licencias comerciales, (MS SQL Server, Oracle, etc). Además, se indicó que la plataforma final sobre la que se publicará esta Web, una vez entre en producción, sea un servidor que incluye Apache y MySQL.

Durante la preparación del entorno de desarrollo, se barajó la posibilidad del uso de un Live CD de Linux, como por ejemplo Knoppix [5], que permite arrancar un entorno Linux sin la necesidad de efectuar la instalación sobre el disco duro, lo que permite la utilización de un PC que no esté dedicado a este fin.

Finalmente, se descartó el uso del Live CD por la facilidad de manejo, portabilidad y comodidad de la suite Portable Apps. Al igual que Knoppix, admite el uso de distintas estaciones de trabajo y evita la necesidad de configurar distintas máquinas, pero además facilita el guardar los datos en el mismo dispositivo. Más importante aún es la posibilidad de trabajar simultáneamente con otras aplicaciones bajo Windows sin necesidad de iniciar la máquina con el LiveCD.

¹ El paquete XAMPP finalmente se descargó por separado, ya que en la Web de los desarrolladores se encontraba una versión más actualizada [6].

Para las pruebas finales, el entorno que se utilizó fue una instalación de Linux Ubuntu Server, instalando el módulo LAMP (Linux, Apache, MySQL & PHP). En el Anexo D se detalla el montaje del entorno de pruebas.

3.2 Instalación y manejo de la plataforma Apache/MySQL/PHP

Como se ha comentado al principio del documento, la opción seleccionada ha sido la distribución de PortableApps [4].

Pero dado que el paquete XAMPP está publicado por Apache Friends [6], se procedió a la descarga en dicha Web del Xampp, que contiene el servidor Web (Apache), Base de datos (MySQL) y PHP.

Ya que en principio no hacen falta más módulos, fue seleccionada la versión **XAMPP 1.6.6.a Lite**, una versión más reducida que la XAMPP, que únicamente contiene:

- Apache 2.2.8.
- PHP 5.2.5.
- MySQL 5.0.51a.
- phpMyAdmin 2.11.4.
- OpenSSL 0.9.8g.
- SQLite 2.8.15.

A fecha de la redacción de esta documentación, la versión actual de XAMPP y de XAMPP Lite es la 1.7. Las versiones de los productos contenidas en esta versión Lite son:

- Apache 2.2.11.
- PHP 5.2.8.
- MySQL 5.1.30.
- phpMyAdmin 3.1.1.
- OpenSSL 0.9.8i.
- SQLite 2.8.15.

La Web ha sido probada en ambas versiones, así como en el entorno final de pruebas sobre el servidor basado en Ubuntu.

También ha sido probado en distintos navegadores, (Google Chrome, Mozilla FireFox, Microsoft Internet Explorer versiones 6 y 7). [ver SC01 y SC05 en Análisis de Requerimientos]

XAMPP está disponible para plataformas Windows, Linux (distribuciones SuSE, RedHat, Mandrake y Debian), Mac OS X y Solaris. Para este proyecto se ha utilizado la versión para el sistema operativo Windows.

Una vez descargada la versión en fichero zip con los binarios (no la versión instalable), se procede a su decompresión en la ruta adecuada. En nuestro caso, en una memoria USB a la que previamente hemos asignado la unidad E: → E:\Xampplite

Para configurar la ruta de ejecución de los distintos servidores, ejecutaremos el fichero `setup-xampp.bat`, esto modifica los ficheros de configuración actualizando dichas rutas y permitiendo el correcto funcionamiento de las aplicaciones.

Según está indicado en la documentación del producto, en el caso de hacerlo en la raíz de un disco (C:\xampp, E:\xampp,...), no es necesario ejecutar la actualización de rutas (salvo que ya haya sido hecho anteriormente).

Una vez descomprimido, se configura el servidor Apache (ver anexos A y F)

Para arrancar los servidores Apache y MySQL tendremos que ejecutar los siguientes scripts:

Arranque de los servidores Apache y MySQL: `.\xampp\xampp_start.exe`

Arranque de los servidores Apache y MySQL: `.\xampp\xampp_stop.exe`

Arranque del servidor Apache: `.\xampp\apache_start.bat`

Parada del servidor Apache: `.\xampp\apache_stop.bat`

Arranque del servidor MySQL: `.\xampp\mysql_start.bat`

Parada del servidor MySQL: `.\xampp\mysql_stop.bat`

Es importante que no haya otros programas o servicios escuchando por los puertos 80 (Apache) ni 3306 (MySQL), ya que si fuera así, estos servidores no funcionarían correctamente.

Una vez arrancado, abriremos un navegador y accederemos a la URL <http://localhost> (o <http://127.0.0.1>) para comprobar el correcto funcionamiento.

Si fuéramos a tener un servidor dedicado, podríamos instalar estas aplicaciones como servicio, mediante los siguientes scripts:

Instalación del servicio Apache: `.\xampp\apache\apache_installservice.bat`

Desinstalación del servicio Apache: `.\xampp\apache\apache_uninstallservice.bat`

Instalación del servicio MySQL: `.\xampp\mysql\mysql_installservice.bat`

Desinstalación del servicio MySQL: `.\xampp\mysql\mysql_uninstallservice.bat`

El siguiente paso es asegurar el entorno, ya que por defecto no tienen usuarios.

Entrando en la URL <http://localhost/security> (o <http://127.0.0.1/security>), ésta nos muestra los puntos de instalación no seguros, como por ejemplo, falta de contraseñas en el usuario root, etc.

Address <http://127.0.0.1/security/index.php> Go Google Configura

XAMPP for Windows

XAMPP
[PHP: 5.2.5]
Chequeo de seguridad

Lenguajes
Deutsch
English
Español
Français
Italiano
Nederlands
Norsk
Polski
Português
Slovenian
中文

©2002/2005
...APACHE
FRIENDS...

XAMPP-Seguridad [Security Check 1.0]

Por medio de este resumen puede verse que puntos de la instalación aún son inseguros y tendrían que ser controlados. (Siga leyendo debajo de la tabla.)

Concerniente a	Estado
Estas paginas XAMPP se visualizan a través de la red Todo lo que puedes ver aquí (estas paginas, este texto), puede verlas potencialmente cualquier otro, que puede conectar con tu ordenador por la red. Si por ejemplo conectas con este ordenador Internet, entonces tendría acceso a estas paginas cualquiera en Internet, que conociera tu dirección IP o la adivinara.	INSEGURO
MySQL-root NO tiene clave de acceso Al MySQL-root aún NO se le ha asignado clave de acceso. Cada usuario del ordenador podrá así usar de forma indiscriminada la base de datos MySQL. Al MySQL-root se le debiera asignar de todas formas una clave de acceso.	INSEGURO
PhpMyAdmin is free accessible by network PhpMyAdmin is accessible by network without password. The configuration 'httpd' or 'cookie' in the "config.inc.php" can help.	INSEGURO
PHP is NOT running in "safe mode" If do you want to offer PHP executions for outside persons, please think about a "safe mode" configuration. But for standalone developer we recommend NOT the "safe mode" configuration because some important functions will not working then. More Info	INSEGURO

Los puntos marcados en verde estan seguros; los puntos en rojo son definitivamente inseguros y en los amarillos no se pudo comprobar la seguridad (por ejemplo porque el programa a comprobar no estaba en marcha).

Para solucionar estos agujeros en la seguridad llame simplemente al siguiente comando:

```
=> http://localhost/security/xamppsecurity.php <= [allowed only for localhost]
```

De esta manera se inicia un programa interactivo, que cerrará todos estos agujeros de seguridad.

Please consider this: With more XAMPP security some examples will NOT execute error free. If you use PHP in "safe mode" for example some functions of this security frontend will not working anymore. Often even more security means less functionality at the same time.

The XAMPP default ports:

```
ftp      21/tcp    # File Transfer [Control] (XAMPP: FTP Default Port)
smtp     25/tcp    mail # Simple Mail Transfer (XAMPP: SMTP Default Port)
http     80/tcp    # World Wide Web HTTP (XAMPP: Apache Default Port)
pop3     110/tcp   # Post Office Protocol - Version 3 (XAMPP: POP3 Default Port)
imap     143/tcp   # Internet Message Access Protocol (XAMPP: IMAP Default Port)
https    443/tcp   # http protocol over TLS/SSL (XAMPP: Apache SSL Port)
mysql    3306/tcp  # MySQL (XAMPP: MySQL Default Port)
ajp/1.3  8009     # AJP/1.3 (XAMPP: Tomcat AJP/1.3 Port)
http-alt 8080/tcp  # HTTP Alternate (see port 80) (XAMPP: Tomcat Default Port)
```

Imagen 1. Página de configuración de seguridad

Son modificables accediendo al vínculo <http://localhost/security/xamppsecurity.php>

En este punto ya tendríamos el entorno preparado para trabajar con la Web.

3.3 Aprendizaje del entorno de desarrollo

El primer objetivo fijado en este proyecto fue el aprendizaje relacionado con la plataforma de desarrollo, esto es, el lenguaje de programación PHP y el uso del servidor de base de datos MySQL.

Para ello se usaron diferentes fuentes de formación y consulta, tales como manuales de referencia [7][8][9], cursos online [10] y distintos foros [11][12].

Establecido este objetivo y antes de acometer la implementación de la página Web del laboratorio, como paso previo se fijó la realización de una sencilla página Web de pruebas.

Una vez preparado el entorno, según lo indicado anteriormente, se crearon dos formularios, una base de datos con una única tabla para poder realizar cargas y borrados simples de datos y, por último, se integraron ambos componentes. Los requisitos para esta primera prueba eran:

Creación de una base de datos de prueba (con una única tabla “miembros”), en esta base de datos estaban incluidos los siguientes campos.

- Foto (en principio, simplemente un enlace a una foto).
- Nombre.
- Descripción (html con algún enlace).

Creación de formularios

- Uno para el alta de personas.
- Un segundo formulario para efectuar bajas de personas.

Creación de una sencilla Web

- Esta Web consta de un número reducido de páginas que generen listado de miembros y que permita mostrar la información almacenada en la base de datos (ficha de personal). Las diversas páginas realizan las siguientes acciones.
 - Insertar información nueva.
 - Eliminarla de una base de datos.
 - Consultar información de una base de datos.
 - Modificar información existente en una base de datos.

3.3.1 Base de datos

Para la puesta en marcha de la base de datos se procede primero al diseño y su posterior generación. Una vez disponemos de esta base de datos, se procede a la generación de un documento de conexión para el acceso a dicha base de datos.

Por último, se integra este documento de conexión en las distintas páginas web con formularios o código que acceden a la base de datos.

Diseño de la base de Datos

Lo primero es crear una base de datos de prueba.

Diseñamos la base de datos según los requisitos para la página de prueba citados anteriormente. En principio, generamos una única tabla, con un campo ID como clave principal, un campo para el nombre, uno para los apellidos y un último campo para la descripción de la persona.

Una vez diseñada, tenemos que generar la Base de Datos en el servidor MySQL

Generación de la base de Datos

Para ello primero tenemos que conectarnos al servidor MySQL mediante el comando:

```
mysql -u <usuario> -p <contraseña>
```

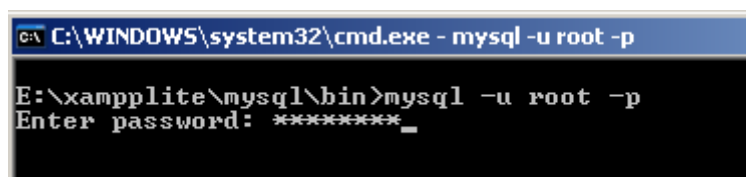


Imagen 2. Conexión al servidor MySQL

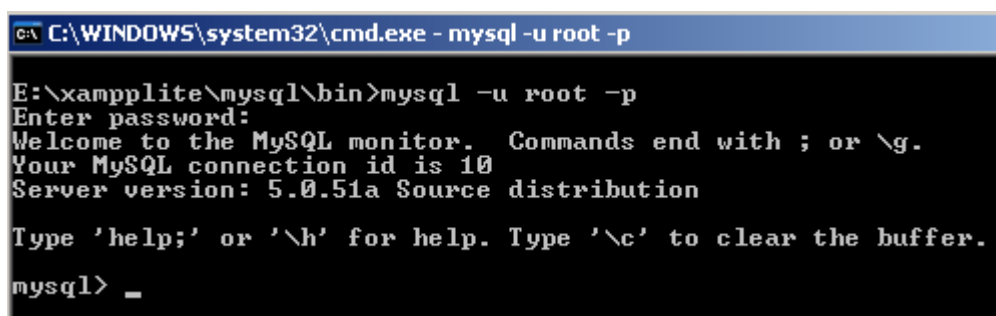


Imagen 3. Línea de comandos de la conexión al servidor MySQL

Después creamos la Base de datos

```
create database <nombre>;
```

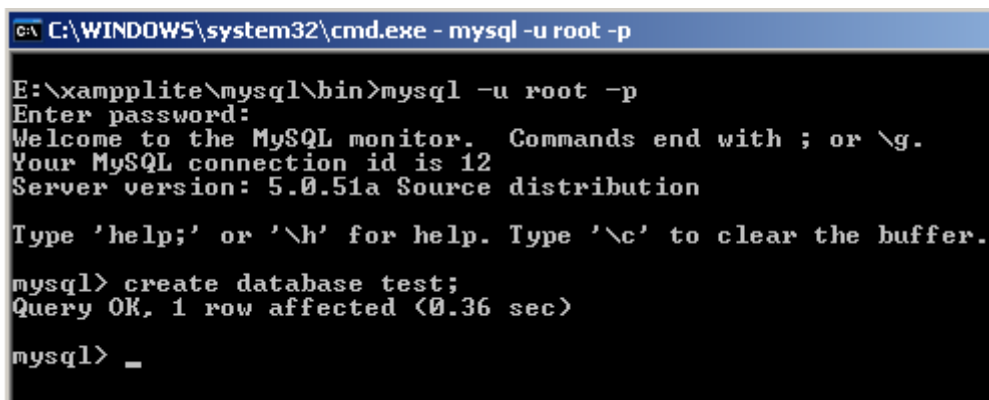
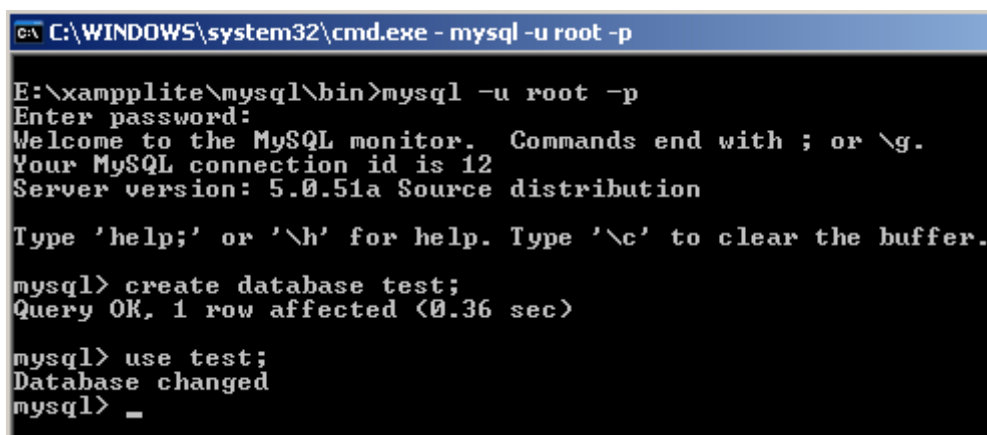


Imagen 4. Creación base de datos de prueba

Seleccionamos la base de datos en la cual vamos a trabajar

```
use <bddd>
```



```
C:\WINDOWS\system32\cmd.exe - mysql -u root -p

E:\xampplite\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.0.51a Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database test;
Query OK, 1 row affected (0.36 sec)

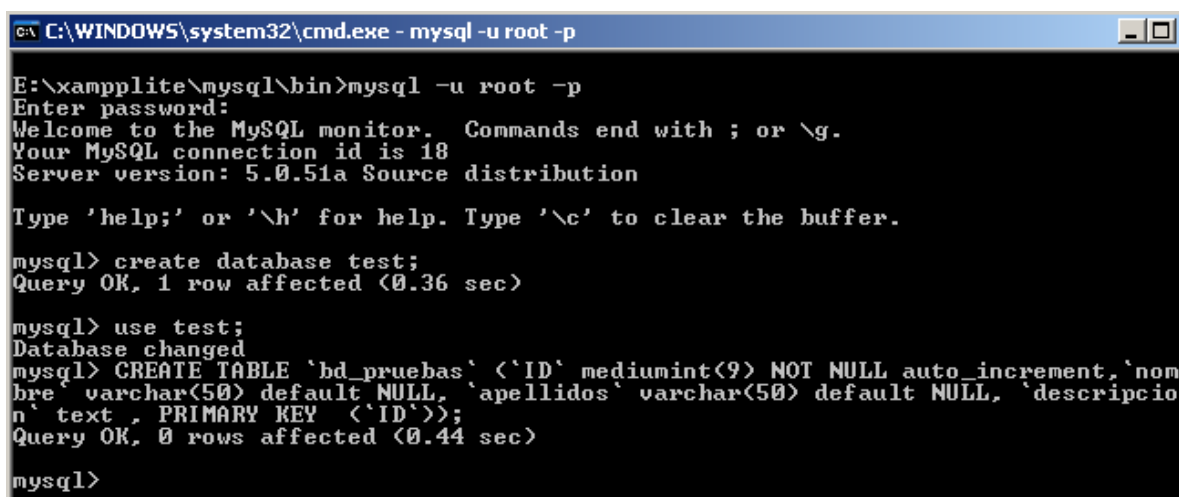
mysql> use test;
Database changed
mysql> _
```

Imagen 5. Selección de la base de datos

En caso de no existir todavía ninguna tabla o para generar una tabla que todavía no existe, podemos seleccionar la tabla de sistema *mysql*.

El siguiente paso es proceder a la creación de la tabla mediante el comando.

```
CREATE TABLE `bd_pruebas` (`ID` mediumint(9) NOT NULL auto_increment, `nombre` varchar(50) default NULL, `apellidos` varchar(50) default NULL, `descripcion` text, PRIMARY KEY (`ID`));
```



```
C:\WINDOWS\system32\cmd.exe - mysql -u root -p

E:\xampplite\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 5.0.51a Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database test;
Query OK, 1 row affected (0.36 sec)

mysql> use test;
Database changed
mysql> CREATE TABLE `bd_pruebas` (`ID` mediumint(9) NOT NULL auto_increment, `nombre` varchar(50) default NULL, `apellidos` varchar(50) default NULL, `descripcion` text, PRIMARY KEY (`ID`));
Query OK, 0 rows affected (0.44 sec)

mysql>
```

Imagen 6. Creación de tabla

Para comprobar si se ha creado la tabla correctamente, ejecutamos el comando:

show tables;

```
mysql> show tables;
+-----+
| Tables_in_test |
+-----+
| bd_pruebas     |
+-----+
1 row in set (0.00 sec)
```

Imagen 7. Listado de tablas en la base de datos actual

Y para ver el contenido:

desc <tabla>;

```
mysql> desc bd_pruebas;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| ID         | mediumint(9) | NO   | PRI | NULL    | auto_increment |
| nombre    | varchar(50)  | YES  |     | NULL    |                |
| apellidos | varchar(50)  | YES  |     | NULL    |                |
| descripcion | text         | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.36 sec)
mysql> _
```

Imagen 8. Listado de campos en tabla actual

En caso de que aparecieran warnings al importar el fichero, podríamos ver el error ejecutando el comando:

SHOW WARNINGS\G

```
mysql> show warnings\g
Empty set (0.00 sec)
mysql> _
```

Imagen 9. Listado de errores en el proceso del último comando

Habría que dar permisos a los usuarios que accedan a las diversas tablas

Por ejemplo, para que un usuario tuviera todos los permisos en todas las tablas sería:

GRANT ALL PRIVILEGES ON *.* TO 'root'@'127.0.0.1' WITH GRANT OPTION;

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'127.0.0.1' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)
mysql> _
```

Imagen 10. Asignación de permisos

Si quisiéramos dar permisos más restringidos, por ejemplo de sólo consulta, el comando sería:

```
GRANT SELECT, RELOAD, PROCESS, FILE, REFERENCES, INDEX, SHOW DATABASES, SUPER, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER ON *.* TO 'usr_limitado@'127.0.0.1';
```

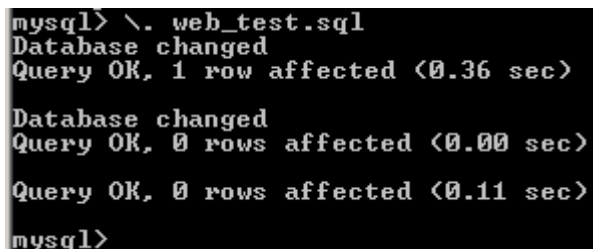


```
mysql> GRANT SELECT, RELOAD, PROCESS, FILE, REFERENCES, INDEX, SHOW DATABASES, SUPER, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER ON *.* TO 'usr_limitado'@'127.0.0.1';
Query OK, 0 rows affected (0.00 sec)
mysql>
```

Imagen 11. Asignación de permisos restringidos

Una segunda opción más cómoda para poder crear la base de datos y la tabla, es guardar toda la definición anterior (código asociado a la creación de la base de datos, tabla y usuarios) en un fichero de texto plano con extensión .sql e importarla mediante el comando

`\. <fichero.sql>`



```
mysql> \. web_test.sql
Database changed
Query OK, 1 row affected (0.36 sec)

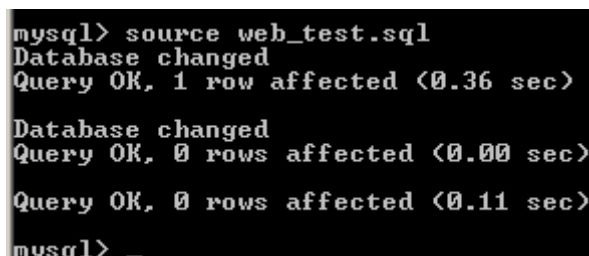
Database changed
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.11 sec)
mysql>
```

Imagen 12. Importación fichero de definición

o el comando

`source <fichero.sql>`



```
mysql> source web_test.sql
Database changed
Query OK, 1 row affected (0.36 sec)

Database changed
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.11 sec)
mysql>
```

Imagen 13. Importación fichero definición, forma alternativa

El código asociado a dicho fichero es el siguiente:

web_pruebas.sql

```
USE mysql;
CREATE DATABASE test;
USE test;

GRANT ALL PRIVILEGES ON *.* TO 'root'@'127.0.0.1' WITH GRANT OPTION;

DROP TABLE IF EXISTS test.bd_pruebas;

CREATE TABLE `bd_pruebas` (
  `id` mediumint(9) NOT NULL auto_increment,
  `nombre` varchar(50) default NULL,
  `apellidos` varchar(50) default NULL,
  `descripcion` text ,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT;
```

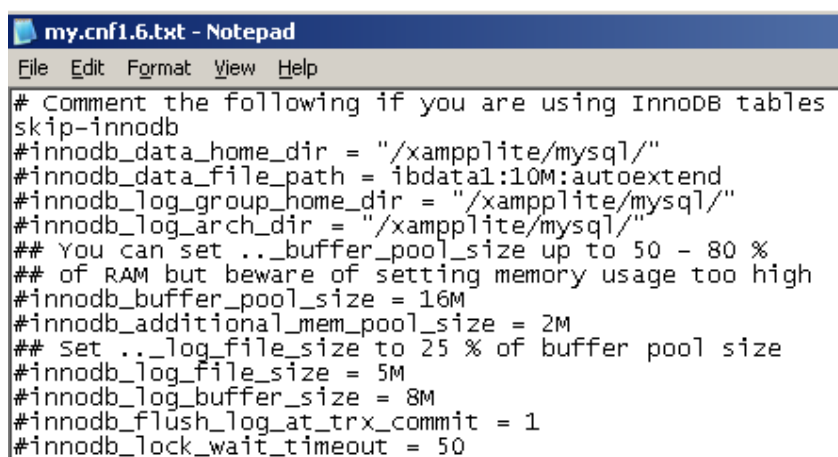
El fichero adjunto en la memoria USB para la generación de la base de datos de pruebas difiere ligeramente, ya que además tiene una carga de datos para poder mostrar las distintas funcionalidades en la página Web.

Uso del motor de BBDD INNODB

En caso de que existan tablas referenciadas (Foreign Keys), es necesario tener habilitado en el fichero de configuración del servidor MySQL (...\\xampplite\\mysql\\bin\\my.cnf) el soporte para el motor InnoDB, ya que el motor por defecto MyISAM no permite tablas referenciadas

Para la versión 1.6.6a se debe editar dicho fichero y descomentar la línea skip-innodb.²

Eliminando la almohadilla y dejando la línea de la siguiente manera:



```
my.cnf1.6.txt - Notepad
File Edit Format View Help
# Comment the following if you are using InnoDB tables
skip-innodb
#innodb_data_home_dir = "/xampplite/mysql/"
#innodb_data_file_path = ibdata1:10M:autoextend
#innodb_log_group_home_dir = "/xampplite/mysql/"
#innodb_log_arch_dir = "/xampplite/mysql/"
## You can set ..buffer_pool_size up to 50 - 80 %
## of RAM but beware of setting memory usage too high
#innodb_buffer_pool_size = 16M
#innodb_additional_mem_pool_size = 2M
## Set ..log_file_size to 25 % of buffer pool size
#innodb_log_file_size = 5M
#innodb_log_buffer_size = 8M
#innodb_flush_log_at_trx_commit = 1
#innodb_lock_wait_timeout = 50
```

Imagen 14. Fichero configuración MySQL v1.6.6a

² En la Release 1.7 ya viene habilitado por defecto.

Documento de conexión

El siguiente paso es la generación de un documento de conexión. Este es el código que va a usar la página en PHP para validarse contra el servidor MySQL. El uso de este documento de conexión en una página separada, permite que posteriormente pueda ser usado en el resto de páginas mediante una sentencia INCLUDE. Esto facilita enormemente la administración, ya que en caso de cambiar de servidor, de usuario o de contraseña, solo es necesario cambiarlo en un único fichero.

Un ejemplo básico de conexión a la base de datos con PHP sería el siguiente:

conexionbd.php

```
<?php
$Servidor = "127.0.0.1";
$Usuario_BD = "juanra";
$Clave = "pfctest";
$NombreBD = "db_pruebas";

$IdConexion = mysql_connect($Servidor, $Usuario_BD, $Clave);

mysql_select_db($NombreBD, $IdConexion);
?>
```

Dado que por defecto la instalación viene sin contraseña, y que en este caso, con el objetivo de facilitar el desarrollo y la transición a esta nueva Web dinámica, la conexión podría hacerse mediante la siguiente línea de código:

```
mysql_connect("localhost","root","");
```

Esto establece la conexión con el servidor de base de datos.

Generación de las páginas Web

El último paso es crear las páginas que realmente verá el usuario. Como se ha indicado en el apartado anterior, en las páginas en las que hay acceso a la base de datos se incluye el documento de conexión mediante la sentencia:

```
<?php
include('conexionbd.php');
?>
```

3.3.2 Esquema de página de prueba.

La Web realizada como prueba inicial para el aprendizaje de MySQL y PHP es sencilla. Consta de una página de inicio con enlaces a las páginas que realizan las funciones básicas de insertar, modificar, listar y eliminar un conjunto reducido de datos. A continuación se indica brevemente el funcionamiento de esta Web de pruebas

Insertar información nueva o eliminarla en la base de datos

A través del formulario de la página *alta_usuario.php* se recoge la información básica a dar de alta en la base de datos (nombre, apellidos y descripción). Una vez se envían, los datos llegan a la página *insertar_usuario.php*, que es la página encargada de realizar la inserción en la base de datos. Una vez realizada, se vuelve a la página principal

Listar información de la base de datos

En esta página se efectúa un listado de los datos contenidos en la base de datos. Además de existir un vínculo para volver a la página principal.

Modificar información existente en la base de datos

El vínculo de la página principal al apartado de modificación lleva a la página *modificar_usuario.php*, en la que existe un menú desplegable para seleccionar el registro a modificar.

Una vez seleccionado y tras dar al botón de envío, la siguiente página, *modif_usuario.php*, muestra un formulario en el que aparecen los datos existentes para el registro seleccionado permitiendo su modificación.

Una vez se envían estas modificaciones, los datos llegan a la página *update_usuario.php*, que es la página encargada de realizar la actualización en la base de datos del registro seleccionado inicialmente. Una vez realizada esta actualización se vuelve a la página principal

Eliminar información existente de la base de datos

En la página *baja_usuario.php*, al igual que en el apartado anterior, existe un desplegable para seleccionar el registro a eliminar. Esta página pasa el registro a eliminar a la página *eliminar_usuario.php*, que es la encargada de eliminar el registro y devolver a la página principal.

Un pequeño esquema de esta Web de pruebas se muestra en el siguiente diagrama:

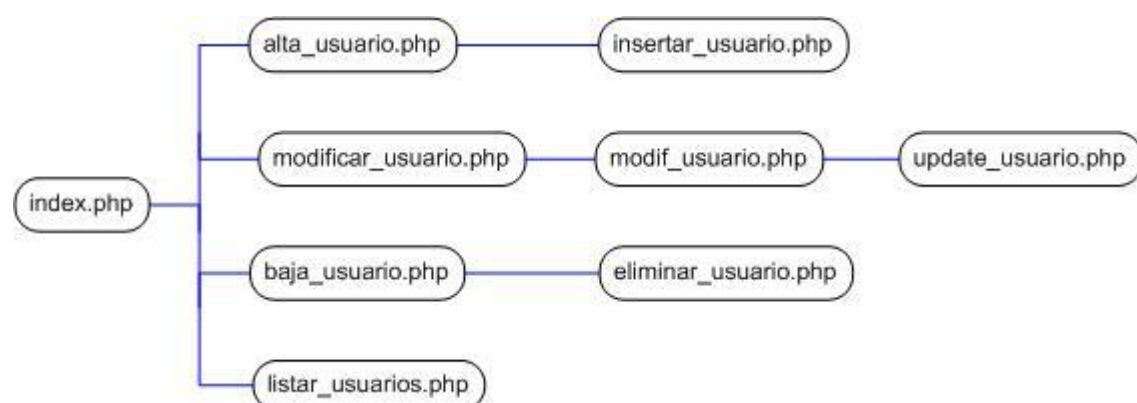


Imagen 15. Diagrama Web de prueba

Esta pequeña Web de pruebas ésta disponible en la memoria USB facilitada. En el anexo A se indica cómo preparar el entorno y la URL para acceder a esta Web de pruebas.

Capítulo 4. Diseño

En el siguiente capítulo, se indica cómo, partiendo de la obtención de los requisitos, se procede al diseño e implementación de la base de datos mediante el modelo entidad relación. En siguientes fases se procede a la creación del documento de conexión para integrar las distintas páginas web con la base de datos. Por último, se identifican las distintas operaciones que esta web va a realizar contra dicha base de datos.

4.1 Modelo de datos de la Base de Datos

4.1.1 Diseño de la Base de Datos

Tras el análisis realizado en el capítulo 2 de recogida de información, se obtienen las entidades con sus correspondientes atributos. Estos son:

- Usuarios para el acceso a la Web de administración:
 - Nombre del usuario.
 - Contraseña del usuario.
 - Descripción.

- Personal:
 - Nombre y Apellidos.
 - Despacho.
 - Teléfono y Fax.
 - Dirección.
 - Dirección de correo electrónico.
 - Web personal.
 - Foto.
 - Categoría profesional.
 - Descripción.
 - Estado (activo / antiguo), esto indica si se ve en la Web pública o no.

- Categoría:
 - Nombre de la categoría.
 - Nombre abreviado de la categoría.
 - Jornada (si es a tiempo completo o a tiempo parcial).

- Proyectos investigación:
 - Nombre del proyecto.
 - Resumen.
 - Descripción.
 - Fecha inicio.
 - Fecha fin.
 - Entidad financiadora.
 - Referencia del proyecto.
 - Director del proyecto.

- Apartado de investigación en el marco del cual está contenido el proyecto (Sublínea de investigación asociada).
- Lista participantes del proyecto.
- Palabras claves (para su búsqueda).
- Vínculo externo.
- Indicación si el proyecto está liderado por el departamento o es una colaboración.
- Campo Activo, que permitirá que se muestre o no el proyecto en la Web pública.

- Publicaciones:
 - Título de la publicación.
 - Título abreviado.
 - Sublínea de investigación asociada (la línea de investigación está implícita).
 - Texto de la referencia.
 - Vínculo externo a la publicación.
 - Fichero con la publicación.

- Líneas investigación, estructuradas en una organización jerárquica formada por líneas, sublíneas, apartados y subapartados de investigación.
 - Líneas de investigación:
 - Nombre de la línea de investigación.
 - Nombre abreviado de la línea de investigación.
 - Descripción de la línea de investigación.

 - Sublíneas de investigación:
 - Nombre de la sublínea de investigación.
 - Nombre abreviado de la sublínea de investigación.
 - Descripción de la sublínea de investigación.

 - Apartados de las sublíneas de investigación:
 - Nombre del apartado.

 - Subapartados:
 - Nombre del subapartado.
 - Texto asociado al subapartado.
 - Personal de contacto responsable.
 - Herramienta asociada.
 - Vínculo asociado.

- Herramientas:
 - Nombre de la herramienta.
 - Nombre abreviado.
 - Descripción.
 - Imágenes (un máximo de 5, ya que se indicó que con 3 sería suficiente).
 - Enlace a la Web de la herramienta.

- Noticias:
 - Título de la noticia.
 - Texto asociado.
 - Fecha de publicación.
 - Enlace.
 - Si la noticia está activa o archivada (si se ve en la Web pública o no).

- Eventos:
 - Nombre del evento.
 - Descripción del evento.
 - Fecha de inicio.
 - Fecha de finalización del evento.
 - Enlace al evento.

- PFC:
 - Nombre del proyecto.
 - Descripción.
 - Tutor-tutores (1 o 2).
 - Titulación.
 - Vínculo externo a la Web de dicho proyecto.

Una vez definidos los atributos necesarios para cada entidad, se procede al diseño de las relaciones. Se han identificado las siguientes:

- Un profesor pertenece a una categoría.
- Un profesor lidera uno o más proyectos.
- Un profesor es persona de contacto de uno o más subapartados de líneas de investigación.
- Uno o dos profesores supervisan uno o más proyectos de fin de carrera.
- Un proyecto de fin de carrera puede estar asociado a una o más titulaciones.
- Una publicación puede estar asociada a una sublínea de investigación.
- Un proyecto está asociado a una sublínea de investigación.
- Una sublínea de investigación puede tener asociadas ninguna, una o varias publicaciones.
- Una sublínea de investigación puede tener asociadas ninguno, uno o varios proyectos.
- Una sublínea de investigación pertenece a una línea de investigación.
- Un apartado investigación pertenece a una sublínea de investigación.
- Un subapartado investigación pertenece a un apartado de investigación.
- Un subapartado investigación puede tener una herramienta asociada.
- Un subapartado investigación puede tener un profesor como persona de contacto.

En el siguiente esquema se puede comprobar el diseño final de la base de datos.

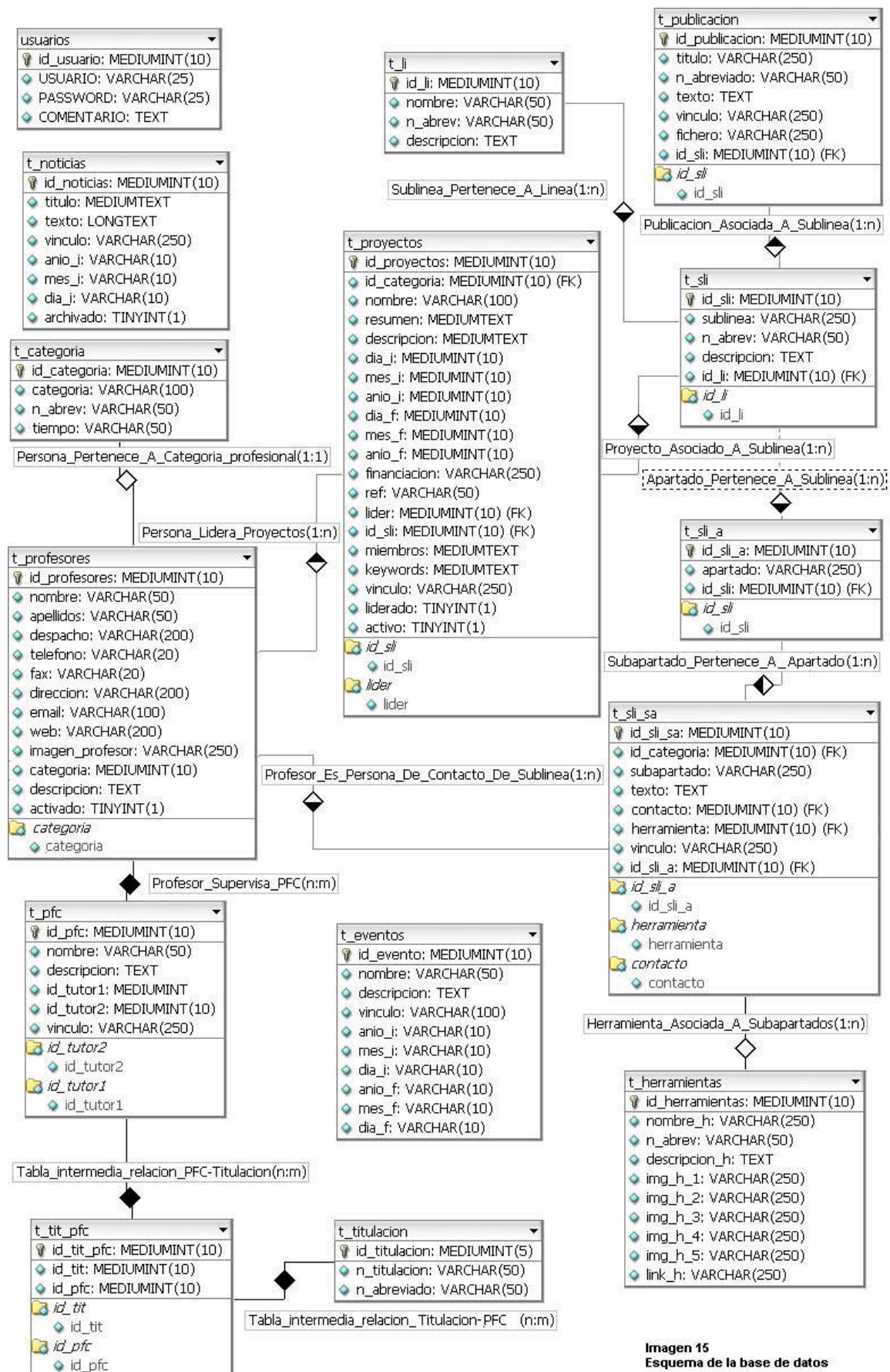


Imagen 15
Esquema de la base de datos

4.1.2 Creación de la base de datos

El siguiente paso es crear la base de datos y el documento de conexión. Ya que existen tablas referenciadas (Foreign Keys), es necesario tener habilitado el motor InnoDB. En el capítulo 3 de instalación y configuración del entorno se describe cómo habilitar dicho soporte.

Una vez habilitado el uso de dicho motor de base de datos, es necesario generar la base de datos. En este caso se ha procedido de la misma manera que para la Web de prueba, importando el fichero de definición con extensión sql con la configuración:

Brevemente y como resumen, los pasos a efectuar y comandos a ejecutar son los siguientes:

Conexión al servidor MySQL: `mysql -u <usuario> -p <password>`

Importar fichero: `source <fichero.sql>`

En caso de que aparecieran warnings al importar el fichero, para el caso de bases con motor InnoDB, a diferencia de las bases de datos MyISAM, el comando que nos muestra el estado y posibles errores es:

```
SHOW ENGINE INNODB STATUS;
```

No hace falta efectuar los pasos previos de creación y selección de la base de datos, ya que en este caso esos comandos ya están contenidos en el fichero de definición con extensión sql que se utiliza a tal efecto.

Una vez importado dicho fichero de extensión sql con la configuración de la base de datos, podemos proceder a una carga de datos masiva, en vez de tener que dar de alta todos los datos dentro de la Web.

En el anexo B se puede ver el contenido de los ficheros con extensión sql tanto para generación de la base de datos y tablas, como para la carga de datos inicial.

4.1.3 Creación del documento de conexión

Debido a que están diferenciadas la Web de administración y la Web pública del laboratorio, se han creado dos documentos distintos de conexión. Cada documento de conexión utiliza un usuario distinto. El usuario de la Web pública sólo tiene permisos de lectura en la base de datos, mientras que el usuario de la Web de administración tiene permisos de lectura y escritura.

Los documentos de conexión son los siguientes:

Conexión Web de administración:

```
<?php
```

```
$Servidor = "127.0.0.1";
$Usuario_BD = "adm2008";
$Clave = "JuanRa2008";
$NombreBD = "web_lite";

$IdConexion = mysql_connect($Servidor, $Usuario_BD, $Clave);

mysql_select_db($NombreBD, $IdConexion);
?>
```

Conexión Web del laboratorio (pública)

```
<?php
$Servidor = "127.0.0.1";

//conexión con usuario de solo lectura

$Usuario_BD = "pfc2008";
$Clave = "m4tr1cul1ll4";
$NombreBD = "web_lite";

$IdConexion = mysql_connect($Servidor, $Usuario_BD, $Clave);

mysql_select_db($NombreBD, $IdConexion);
?>
```

4.2 Identificación de las operaciones

Al igual que en la Web de prueba, sólo existen cuatro operaciones básicas en la Web de administración: inserción, eliminación, modificación y consulta de datos de la base de datos asociada.

En la Web pública solo se efectúa la consulta de datos.

Dado que la Web actual procede de la página del laboratorio existente, en el siguiente apartado se explica más detenidamente que función realiza cada página.

El esquema, a grandes rasgos, es el siguiente:

- Página para el listado (consulta de datos), en el caso de la Web de administración.
- Páginas que presentan diversos datos según el apartado por el que naveguemos en el caso de la Web pública.
- Grupo de páginas para realizar el alta.
- Grupo de páginas para efectuar modificaciones sobre conjuntos de datos ya existentes.
- Grupo de páginas para eliminar conjuntos de datos.

La información que muestra o requiere cada grupo de páginas se definió en el requisito funcional 10 (RF10), que a su vez está definida en el modelo de datos del capítulo 4.

El funcionamiento de estas páginas se explica en el capítulo 5, cada grupo de páginas está identificado por los siguientes códigos:

CAT	Categorías profesionales.
EVT	Eventos.
NTC	Noticias.
PFC	Proyectos de fin de carrera.
PRF	Personal.
PROY	Proyectos.
TIT	Titulaciones.
TOOL	Herramientas.
USR	Usuarios.
LI	Líneas, sublíneas, apartados y subapartados de investigación.
PBL	Publicaciones.

Debido a que el funcionamiento de un gran número de páginas es similar, las páginas en que aparece la etiqueta XXX en el capítulo 5 han de ser sustituidas por el código respectivo.

Capítulo 5. Implementación

5.1 Desarrollo el Web site partiendo del existente

Como se ha indicado anteriormente, la Web actual procede de la página del laboratorio existente en <http://lite.etsii.urjc.es>.

Las páginas que han sido modificadas llevan extensión PHP, ya que llevan código php, las que no han sido modificadas conservan su extensión html.

Las modificaciones han sido las siguientes.

Columna derecha (*columnaderecha.php*, *columnaderecha-es.php*, *columnaderechaindex.php*, *columnaderechaindex-es.php*)

- Estas columnas eran una página en javascript, se cambió por una página en php. Las dos primeras conservan sus funcionalidades, mostrando vínculos de la propia Web (*columnaderecha.php*, *columnaderecha-es.php*) y las dos últimas (*columnaderechaindex.php*, *columnaderechaindex-es.php*) muestran vínculos y contenido dinámico (última noticia y eventos en curso).

Eventos (*events.php*)

- Esta página originalmente era una página html, se cambió a una página php. En ellas se genera el contenido dinámicamente, mostrando los eventos, primero los actuales, después los futuros y por último los antiguos, en orden de proximidad a la fecha actual.

Funciones (*funciones.php*)

- Esta página no existía previamente, contiene la función que permite procesar cadenas de texto para evitar errores al mostrarlos en el código html (comillas y dobles comillas, convirtiéndolas en las cadenas `'` y `"`; respectivamente).

Trabajo (*jobs.php*)

- Se mantiene el contenido de la página html original, añadiendo únicamente al final de la página un listado de las titulaciones que tienen proyectos de fin de carrera (en la base de datos) y el número de proyectos ofertados en cada titulación.

Noticias (*news.php*)

- Esta página genera el contenido dinámicamente, mostrando primero la última noticia, después las noticias del último mes y por último las noticias del último año.

Gente (*people.php*, *people-detail.php*)

- En la primera página (*people.php*) se muestra un listado del personal contenido en la base de datos. Solo se muestra si está marcado como activo, en caso contrario, no aparece en la Web pública. El personal aparece primero ordenado por tiempo completo o parcial, y dentro de esta clasificación, por su categoría profesional.

- En la segunda página (*people-detail.php*), accediendo a mostrar detalles en la página anterior o a través de algún enlace (tutor de proyecto, persona de contacto, etc.), se muestran los detalles relativos a dicha persona que estén contenidos en la base de datos.

Proyectos de fin de carrera (*pfcs.php*)

- En esta página se muestran los datos contenidos en la base de datos relativos a los proyectos de fin de carrera ordenados por titulaciones.

Proyectos (*projects.php*)

- En esta página se muestran los proyectos del laboratorio, diferenciando si se trata de proyectos liderados por personal del laboratorio o si son colaboraciones.

Publicaciones (*publications.php*)

- En esta página se muestran las publicaciones del laboratorio ordenadas por sublíneas de investigación. En caso de tener un fichero asociado, aparece un vínculo al fichero.

Líneas de investigación (*reslines.php*, *reslines-li.php*, *reslines-sli.php*)

- En este grupo de páginas, se muestran dinámicamente, en primer lugar las líneas de investigación (*reslines.php*); al seleccionar una de ellas, muestra las sublíneas asociadas (*reslines-li.php*), y seleccionando éstas, nos muestra los apartados y subapartados de dichas líneas, así como todos los datos contenidos en el subapartado, como persona de contacto, herramienta relacionada, etc.
- En el lateral izquierdo, en todos los casos muestra el nombre (abreviado, si lo tuviera, y si no, el nombre original) de la línea y las sublíneas que contiene cada uno (también con nombre abreviado si lo tuviera).

Mapa Web (*sistema.php*, *sitemap-es.php*)

- En esta página solo se generan dinámicamente las dos columnas centrales (existen cuatro en la página). Estas corresponden a líneas de investigación, miembros de LITE y proyectos (segunda columna) y eventos, publicaciones y herramientas (tercera columna), mostrando en todos los apartados el nombre abreviado si lo tuviera y si no el nombre original.

Herramientas (*tools.php*)

- Muestra un listado dinámico de las herramientas del laboratorio en el que puede mostrar el resumen, un máximo de cinco imágenes y un vínculo externo a la herramienta.

5.2 Generación del Web site de administración

Esta parte es de nueva creación y se trata de un portal para el acceso a la base de datos, que permite crear, modificar, listar y borrar datos asociados a la Web pública. Funcionalmente está dividido principalmente en dos partes:

- Acceso a la Web y funciones varias.
- Altas, bajas, modificaciones y listados. (directorio `.\adm\abm`)

5.2.1 Acceso a la Web y funciones varias

Estas páginas están situadas en el directorio `pfc\adm` (siendo `pfc` el directorio raíz de la Web)

Admin.php

- Página principal del portal de administración.

Conexionbd.php

- Documento de conexión a la base de datos. Este es el documento de conexión para la Web de administración. El usuario tiene privilegios de lectura y escritura en la base de datos.

Conexionbd_p.php

- Documento de conexión a la base de datos. Este es el documento de conexión para la Web pública. El usuario solo tiene privilegios de lectura en la base de datos.

Funciones.php

- Esta página contiene funciones como la que permite procesar cadenas de texto para evitar errores al mostrarlos en el código html (comillas y dobles comillas, convirtiéndolas en las cadenas `'` y `"` respectivamente), la función para subir ficheros y la función para subir imágenes.

Index.php

- Página de inicio del portal de administración. En esta página se solicita las credenciales de acceso al portal de administración.

No_autorizado.php

- Página que indica al usuario que la validación no ha sido correcta.

Salir.php

- Página para finalizar la navegación en la Web de administración. Cierra el navegador para mayor seguridad.

Sesiones.php

- Esta página comprueba si el usuario tiene una sesión válida (ha proporcionado un usuario y una clave correctas). En caso de que no esté validado, le envía a la página `no_authorized.php`.

Validalogin.php

- En esta página se comprueba si el usuario está registrado en la base de datos (si tiene permiso o no) y le redirige a la página correspondiente.

5.2.2 Altas, bajas, modificaciones y listados

Estas páginas están situadas en el directorio `pfc\adm\abm` (siendo `pfc` el directorio raíz de la Web). Salvo dos casos en los que la estructura es algo más compleja, la mayoría de los apartados (personal, proyectos, herramientas, etc) siguen la siguiente estructura:

xxx_alta.php:

- Página que contiene el formulario con los datos a rellenar.
- Esta página envía los datos del formulario a la página *xxx_alta_insertar.php*.

xxx_alta_insertar.php:

- Esta es la página que efectúa la inserción de los datos del formulario en la base de datos y muestra un resumen de la acción realizada.

xxx_baja.php:

- Página que contiene un desplegable para la selección del objeto a eliminar.
- Envía la selección a la página *xxx_baja_conf.php*.

xxx_baja_conf.php:

- Página que muestra los datos a eliminar y solicita la confirmación.
- Envía el objeto a eliminar a la página *xxx_baja_eliminar.php*.

xxx_baja_eliminar.php:

- Página que efectúa el borrado de los datos en la base de datos.

xxx_lista.php:

- Página que genera el listado de los datos relativos al apartado en cuestión.

xxx_modif.php:

- Página que contiene un desplegable para la selección del objeto a modificar.
- Envía la selección a la página *xxx_modif_xxx.php*.

xxx_modif_xxx.php:

- Página que contiene el formulario con los datos del objeto, permitiendo modificar los que sean necesarios.
- Solicita confirmación de la acción a realizar.
- Esta página envía los datos del formulario a la página *xxx_modif_update.php*.

xxx_modif_update.php:

- Esta es la página que efectúa la inserción de los datos del formulario en la base de datos y muestra un resumen de la acción realizada.

Las dos excepciones que se indicaban anteriormente son los apartados de líneas de investigación y publicaciones.

5.2.3 Apartado de líneas de investigación

li_alta.php:

- Página con distintos desplegados, para poder seleccionar el objeto a dar de alta (línea, sublínea, apartado o subapartado).
- Pasa como parámetro la selección a la página *li_alta_li.php*.

li_alta_li.php:

- Muestra un formulario con distintos campos. Estos dependen de la selección realizada en la página *li_alta.php*.
- Pasa los datos a la página *li_alta_insertar.php*.

li_alta_insertar.php:

- Inserta los datos del formulario mostrado en la página *li_alta_li.php* en la base de datos.

li_baja.php:

- Página con distintos desplegados, para poder seleccionar el objeto a eliminar.

li_baja_conf.php:

- Página que muestra los datos a eliminar y solicita la confirmación.
- Envía el objeto a eliminar a la página *li_baja_eliminar.php*.

li_baja_eliminar.php:

- Página que efectúa el borrado de los datos en la base de datos.
- Esta incluye la página *li_baja_inc.php*.

li_baja_inc.php:

- Página con código html para indicar que se ha borrado el registro seleccionado.
- Está incluida mediante una sentencia INCLUDE en la página *li_baja_eliminar.php*.

li_modif.php:

- Página que mediante un desplegable, permite la selección del objeto a modificar (línea, Sublínea, apartado o subapartado).
- Pasa como parámetro la selección a la página *li_modif_li.php*.

li_modif_li.php:

- Página que contiene el formulario con los datos del objeto seleccionado previamente en la página *li_modif.php*, permitiendo modificar los que sean necesarios.
- Dependiendo del caso, se envía de manera oculta (input type='hidden') el ID de la línea, Sublínea, apartado o subapartado.

- Este formulario varía en función del nivel que se haya seleccionado (línea, Sublínea, apartado o subapartado).
- Solicita confirmación de la acción a realizar.
- Esta página envía los datos del formulario a la página *li_modif_update.php*.

El funcionamiento del resto de las páginas asociadas a líneas de investigación es idéntico al de las páginas con prefijo *xxx* explicadas anteriormente (*li_lista.php* y *li_modif_update.php*).

5.2.4 Apartado de publicaciones

pbl_modif_publicacion.php:

- Página que contiene el formulario con los datos del objeto, permitiendo modificar los que sean necesarios.
- Solicita confirmación de la acción a realizar.
- Esta página envía los datos del formulario a la página *pbl_modif_update.php*.
- Tiene una casilla de verificación para modificar la sublínea de investigación.

pbl_modif_update.php:

- Esta es la página que efectúa la inserción de los datos del formulario en la base de datos y muestra un resumen de la acción realizada.
- En caso de haberse marcado la casilla de verificación en *pbl_modif_publicacion.php*, muestra un botón con un vínculo a la página *pbl_modif_pbl2.php*, que permite la modificación de la sublínea de investigación.

pbl_modif_pbl2.php:

- Esta página muestra dos campos desplegable que permiten la modificación de la sublínea de investigación a la que pertenece la publicación. En caso de cambiar la línea de investigación (primer desplegable), modifica el segundo desplegable mostrando todas las sublíneas asociadas a dicha línea.
- Esta página pasa los datos (publicación como parámetro oculto y sublínea de investigación) a la página *pbl_modif_update2.php*.

pbl_modif_update2.php:

- Esta es la página que efectúa la actualización en la base de datos de los datos proporcionados en el formulario de la página *pbl_modif_pbl2.php* y muestra un resumen de la acción realizada.

El funcionamiento del resto de las páginas asociadas a publicaciones es idéntico al de las páginas con prefijo *xxx* explicadas anteriormente (*pbl_alta.php*, *pbl_alta_insertar.php*, *pbl_baja.php*, *pbl_baja_conf.php*, *pbl_alta_eliminar.php*, *pbl_lista.php* y *pbl_modif.php*).

Capítulo 6. Conclusiones

En este capítulo se indican los hitos alcanzados y algunas posibles futuras líneas de desarrollo para la mejora de esta web. Debido a que el contenido es dinámico, más adelante pueden surgir nuevas líneas de trabajo o nuevas necesidades no contempladas actualmente.

6.1 *Hitos*

Se ha cumplido con el objetivo inicial de convertir la Web en una dinámica, permitiendo una administración más sencilla, así como la posibilidad de automatizar la carga de datos. Se ha respetado el formato de la Web inicial, solucionando los problemas que dicho formato implicaba, como en el caso de los nombres largos.

6.2 *Posibles Trabajos futuros*

Algunas de las mejoras que podrían realizarse como ampliación del presente proyecto serían:

Datos en formularios

Sería interesante que en los formularios, si se comete algún error al introducir datos, por ejemplo, que la imagen no sea gif o jpg, se guardara el estado, esto es, que al volver atrás se conservaran los datos y no fuera necesario volver a introducirlos de nuevo. Actualmente es un formulario estándar y no se guarda ningún estado.

Confirmaciones

Se comentó, aunque no llegó a formar parte de ningún requisito, la posibilidad de que al borrar un subapartado, si el apartado padre se queda sin subapartados, se preguntara si también se quiere borrar.

Tutores de PFCs

Podría ser conveniente que los estudiantes (de grado o máster) no aparecieran como posibles tutores de PFCs.

Inicialmente se consideró la creación de una tabla para estudiantes y otra para profesores, pero finalmente se vio conveniente el uso de una única tabla.

Para implementar esto se podría efectuar de dos maneras:

- Añadir una sentencia IF para no mostrar estudiantes. (solución óptima).
- Crear dos tablas separadas, una para profesores y una para estudiantes (llevaría más modificaciones de código).

Rendimiento

Se ha comprobado que la velocidad de acceso es menor con las tablas con motor INNODB, y debido a la manera en la que se ha creado la estructura de la Web y la base de datos, en la

que las restricciones son aplicadas desde los propios formularios en la Web y no en la base de datos, es posible usar tablas con motor MyISAM. En estas no existe un log de transacciones y su rendimiento durante las pruebas parece haber sido mayor.

Buscando recomendaciones sobre el uso de motores MyISAM o INNODB, parece que el rendimiento es mayor en donde priman sentencias SELECT sobre INSERT o UPDATES, como es el caso de esta Web, ya que se espera que sea muy superior el número de consultas que las modificaciones a realizar (relativamente pocas una vez hecha la carga inicial) [13], [14]. La siguiente matriz de decisión ayuda a la elección del tipo de motor [15]

¿Se van a realizar en las tablas inserciones, borrados y actualizaciones mucho más a menudo que selecciones?	InnoDB
Es necesario búsquedas en todo el texto	MyISAM
Es necesario/preferido el uso de una diseño de base de datos relacional	InnoDB
La memoria RAM o el disco es un problema	MyISAM
En caso de duda	MyISAM

Esta es una modificación sencilla, y solo requiere sustituir en el fichero fuente de carga de tablas el tipo de tabla. Al seleccionar como motor MyISAM automáticamente se descartarán las relaciones (FOREIGN KEYS), ya que no están soportadas.

Para el proyecto, por motivos académicos, se ha seleccionado el uso del motor de base de datos INNODB, ya que muestra más claramente las relaciones entre tablas, pero por motivos de rendimiento puede ser conveniente la evaluación, una vez la Web sea puesta en producción, del motor MyISAM, ya que como se ha explicado anteriormente, las restricciones ya están implementadas en la propia Web, por lo que no sería indispensable realizarlas en la base de datos.

Bibliografía y referencias

- [1] Wikipedia. *MySQL - Wikipedia, la enciclopedia libre*.
<http://es.wikipedia.org/wiki/MySQL>. Enero 2009.
- [2] Wikipedia. *PHP - Wikipedia, la enciclopedia libre*.
<http://es.wikipedia.org/wiki/.php>. Enero de 2009.
- [3] Wikipedia. *Software libre - Wikipedia, la enciclopedia libre*.
http://es.wikipedia.org/wiki/Software_libre. Enero de 2009.
- [4] Portable Apps – *Portable Software for USB Drives*.
<http://www.portableapps.com>. Diciembre 2008
- [5] Knoppix – *Live Linux Filesystem on CD*.
<http://www.knoppix.com/>. Diciembre 2008
- [6] Apachefriends – *Very easy apache, mysql, php and perl installation without hassles*.
<http://www.apachefriends.org/en/index.html>. Diciembre de 2008.
- [7] MySQL - *MySQL 5.0 Reference Manual*.
<http://dev.mysql.com/doc/refman/5.0/en/>. Diciembre de 2008.
- [8] Mehdi Achour, Friedhelm Betz, Antony Dovgal, Nuno Lopes, Philip Olson, Georg Richter, Damien Seguy, Jakub Vrana, And several others. *PHP Manual*. The PHP Documentation Group, 2007.
- [9] w3schools – *HTML 4.01 / XHTML 1.0 Reference*.
<http://www.w3schools.com/tags/>. Agosto de 2008.
- [10] Salvado Pozo - *Curso de MySQL*.
<http://mysql.conclase.net/curso/>. Agosto de 2008.
- [11] MySQL Hispano – *La comunidad de usuarios de MySQL*.
<http://www.mysql-hispano.org/>. Diciembre de 2008.
- [12]
Foros del Web – *PHP Foros del Web*.
<http://www.forosdelweb.com/f18/>. Septiembre de 2008.
- Apache Friends – *Apache Friends Forum*.
<http://www.apachefriends.org/f/>. Agosto de 2008.

- [13] Barrapunto - *MySQL: MyISAM vs InnoDB*.
<http://softlibre.barrapunto.com/article.pl?sid=07/11/13/0921211&from=rss>. Enero de 2009.
- [14] Tufuncion - *MyISAM vs InnoDB*.
<http://www.tufuncion.com/myisam-vs-innodb>. Enero de 2009.
- [15] Mikebertnat - *MyISAM vs InnoDB*.
<http://www.mikebernat.com/blog/MySQL - InnoDB vs MyISAM>. Enero de 2009.
- [16] Ubuntu – Download Ubuntu.
<http://www.ubuntu.com/getubuntu/download>. Diciembre 2008.
- [17] VMWare – VMWare Server, Free VMWare Download for Virtual Server Consolidation.
<http://www.vmware.com/products/server/>. Diciembre 2008
- [18] Apache Friends – XAMPP for windows.
<http://www.apachefriends.org/en/xampp-windows.html>. Agosto de 2008.

Anexo A. Manual de operación

Instalación y puesta en marcha del entorno (bajo Windows)

En este anexo se indica cómo arrancar el entorno bajo Windows usado para el desarrollo. Para ello, es necesario realizar los siguientes pasos:

- 1.- Descargar y descomprimir el paquete XAMPP.
- 2.- Será necesario copiar en la carpeta *htdocs\xampp* los ficheros que componen la Web.
- 3.- También será necesario importar la base de datos (podemos tanto importarla como copiar los ficheros que la componen al directorio *mysql\data*).

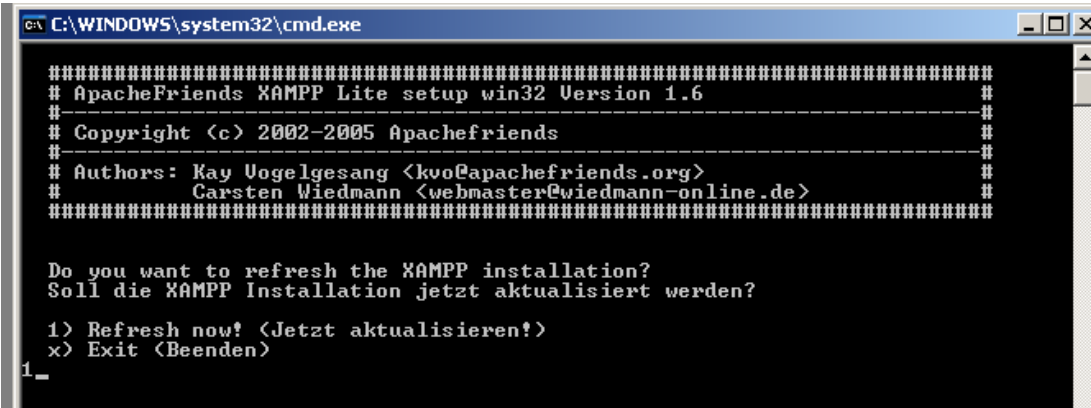
En la memoria USB adjunta se encuentra la versión lite de XAMPP, la Web y la base de datos, listo para ser usado.

Al usar la versión lite, el directorio de referencia será XAMPPLITE en vez de XAMPP. En caso de descargar la versión completa, habría que sustituir las referencias de *xampplite* a *xampp*.

Una vez descomprimida la aplicación y agregados los datos de la Web, continuaremos con los siguientes pasos para poder utilizar el entorno.

- 4.- Configuración del entorno.

Para ello se ejecuta dentro de la carpeta *xampplite* el fichero batch *setup_xampp.bat*, si el path (letra de unidad) es distinta a la configurada previamente, indicará que se pulse 1 y posteriormente la tecla Enter para efectuar la modificación de configuración de *xampp*, o cualquier otra tecla para omitir la configuración (véase la imagen 17).



```
C:\WINDOWS\system32\cmd.exe
#####
# ApacheFriends XAMPP Lite setup win32 Version 1.6
#
# Copyright (c) 2002-2005 ApacheFriends
#
# Authors: Kay Vogelgesang <kvo@apachefriends.org>
#          Carsten Wiedmann <webmaster@wiedmann-online.de>
#####

Do you want to refresh the XAMPP installation?
Soll die XAMPP Installation jetzt aktualisiert werden?

1) Refresh now! <Jetzt aktualisieren!>
x) Exit <Beenden>
1_
```

Imagen 17. Modificación de variables de entorno

Tardará un tiempo e indicará que se han realizado los cambios, ver la imagen 18.

```

C:\WINDOWS\system32\cmd.exe

#####
# ApacheFriends XAMPP Lite setup win32 Version 1.6
#-----#
# Copyright (c) 2002-2005 ApacheFriends
#-----#
# Authors: Kay Vogelgesang <kvo@apachefriends.org>
#          Carsten Wiedmann <webmaster@wiedmann-online.de>
#####

Do you want to refresh the XAMPP installation?
Soll die XAMPP Installation jetzt aktualisiert werden?

1) Refresh now! (Jetzt aktualisieren?)
x) Exit (Beenden)
1

XAMPP is refreshing now ...
XAMPP wird nun aktualisiert ...

Refreshing all paths in config files ...

Configure XAMPP with awk for 'Windows_NT'
Please wait ... DONE!

##### Have fun with ApacheFriends XAMPP Lite! #####

Press any key to continue . . .
    
```

Imagen 18. Modificación de variables de entorno realizadas correctamente

En caso de que la ruta ya sea la correcta, indicará que no realizará ningún cambio, ver la imagen 19.

```

C:\WINDOWS\system32\cmd.exe

#####
# ApacheFriends XAMPP Lite setup win32 Version 1.6
#-----#
# Copyright (c) 2002-2005 ApacheFriends
#-----#
# Authors: Kay Vogelgesang <kvo@apachefriends.org>
#          Carsten Wiedmann <webmaster@wiedmann-online.de>
#####

Sorry, but ... nothing to do!

Press any key to continue . . . _
    
```

Imagen 19. Modificación de variables de entorno no realizadas

5.- Arranque de servidores MySQL y Apache.

Para ello se ejecutan los ficheros batch *apache_start.bat* y *mysql_start.bat*, localizados en el directorio xampp.

6.- Acceso a las distintas Webs

Una vez arrancados, podremos acceder a la Web del laboratorio y al portal de administración a través de un navegador, mediante los siguientes enlaces.

Web del laboratorio:

Desde la misma máquina: <http://localhost/xampp/pfc/>
Desde otra máquina: <http://<IP o nombre DNS>/xampp/pfc/>

Web de administración:
Desde la misma máquina: <http://localhost/xampp/pfc/adm/>
Desde otra máquina: <http://<IP o nombre DNS>/xampp/pfc/adm/>

Web inicial de prueba
Desde la misma máquina: <http://localhost/xampp/wprueba/>
Desde otra máquina: <http://<IP o nombre DNS>/xampp/wprueba/>

Web del laboratorio

En esta parte de la Web, los usuarios pueden navegar por los distintos apartados de la Web del laboratorio de LITE, como Research Lines, People, Projects, Publications, Tools, Events, etc.

Mapa de la Web pública

En la siguiente página podemos ver el diagrama asociado al mapa de la Web de pública.

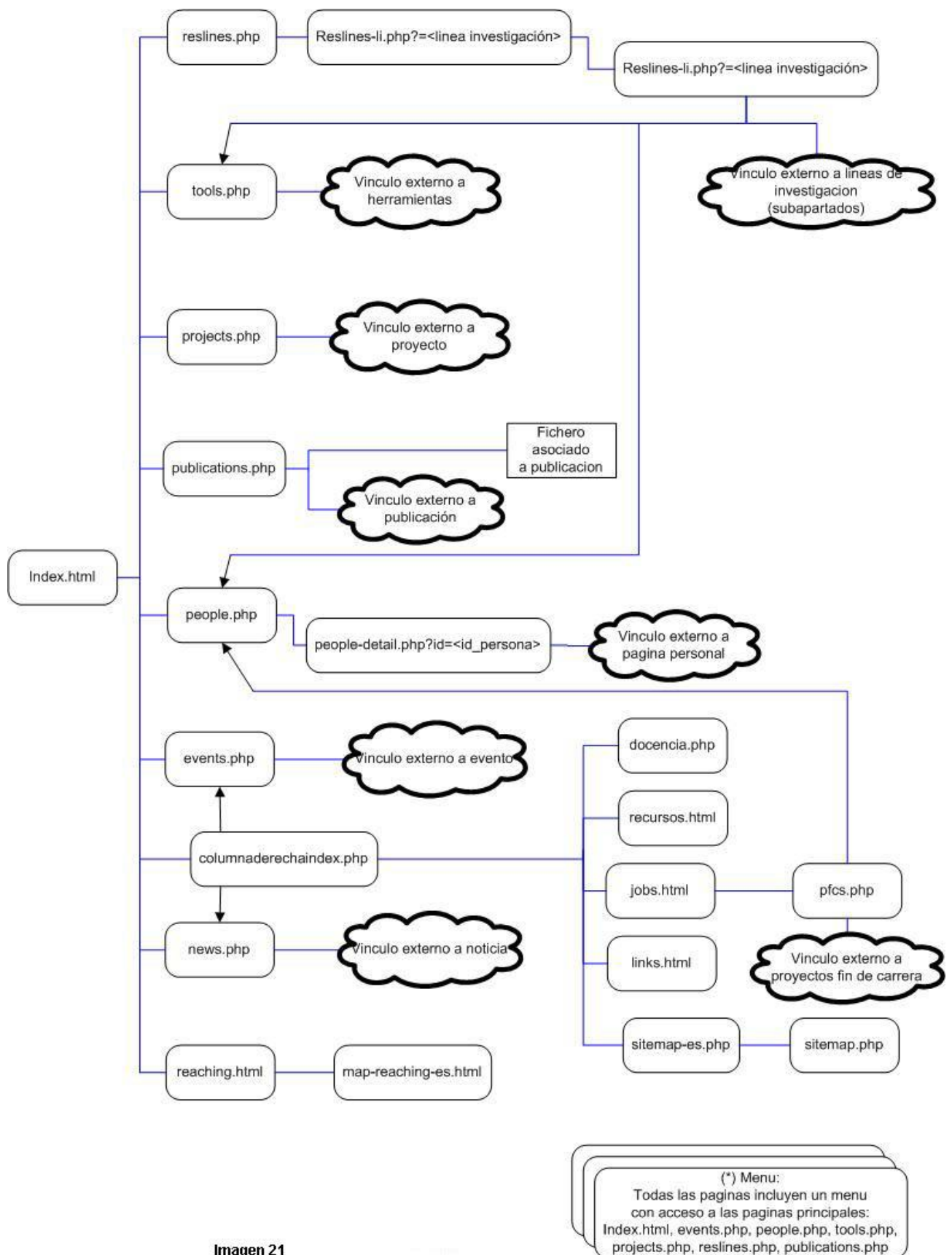


Imagen 21
 Mapa Web pública

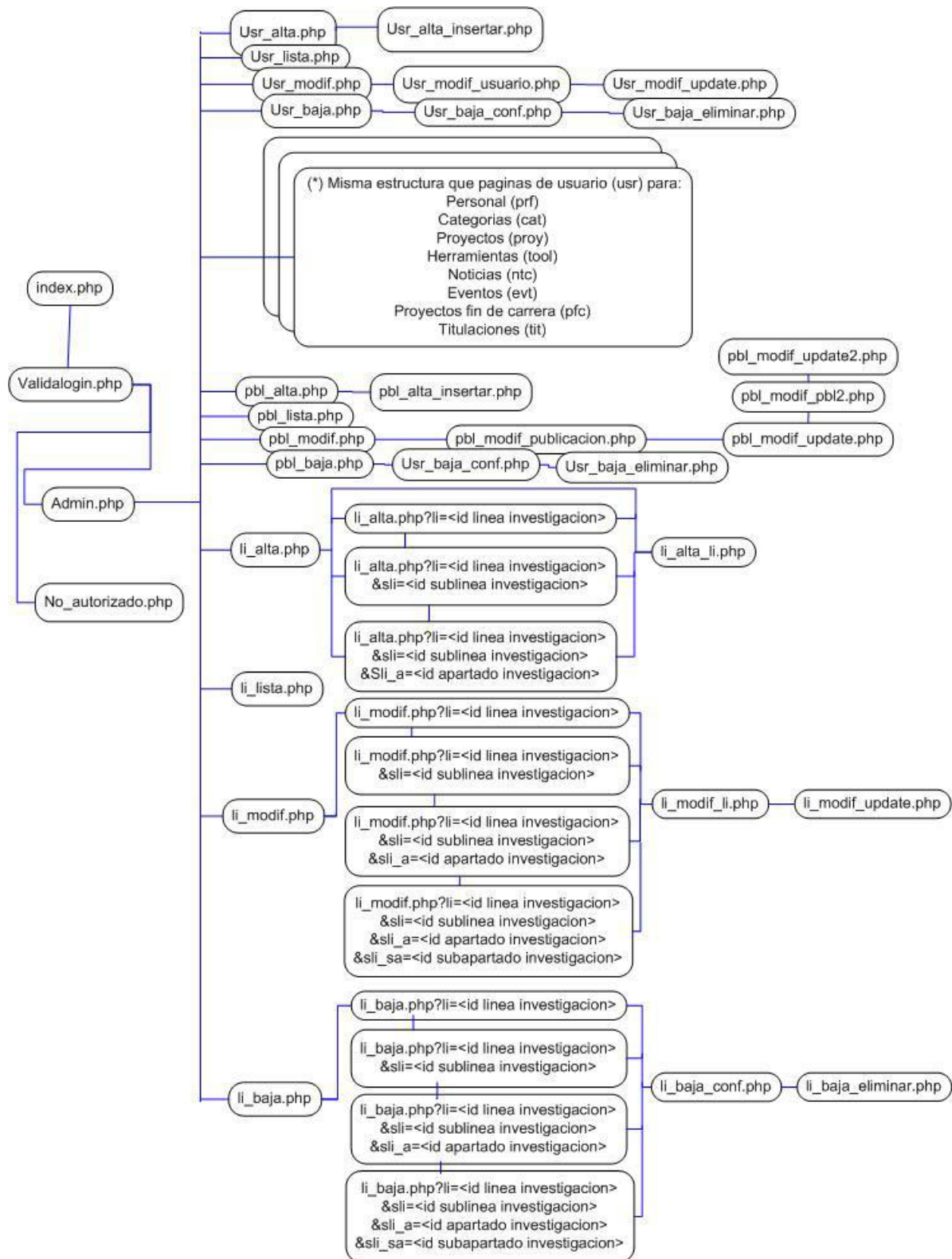
Portal de administración

En esta parte de la Web podremos efectuar altas, bajas, modificaciones y generar listados de los datos contenidos en la BBDD sobre los distintos apartados. Tras entrar por la página de validación, accederemos a la página principal. Estos apartados son:

- Usuarios.
- Personal Docente.
- Categorías de personal docente.
- Proyectos.
- Publicaciones.
- Líneas de investigación.
- Herramientas.
- Noticias.
- Eventos.
- Proyectos de Fin de Carrera.
- Titulaciones.

Mapa de la Web de administración

En la siguiente página podemos ver el diagrama asociado al mapa de la Web de administración



(*) Menu:
 Todas las paginas incluyen un menu con acceso a las paginas inicales de alta, baja, modificacion y listado: Xxx_alta.php, xxx_lista.php, xxx_modif.php & xxx_baja.php:

Imagen 22
 Mapa Web administración

Anexo B. Código asociado a la Base de Datos

Generación de la base de datos

```
USE mysql;
DROP DATABASE IF EXISTS web_lite;
CREATE DATABASE web_lite;
USE web_lite;

GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION;
GRANT SHUTDOWN ON *.* TO 'pma'@'localhost';
GRANT ALL PRIVILEGES ON *.* TO 'root'@'127.0.0.1' WITH GRANT OPTION;

GRANT SELECT, RELOAD, PROCESS, FILE, REFERENCES, INDEX, SHOW
DATABASES, SUPER, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE,
REPLICATION SLAVE, REPLICATION CLIENT, CREATE VIEW, SHOW VIEW,
CREATE ROUTINE, ALTER ROUTINE, CREATE USER ON *.* TO
'pfc2008'@'127.0.0.1';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS,
FILE, REFERENCES, INDEX, ALTER, SHOW DATABASES, SUPER, CREATE
TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE,
REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER
ROUTINE, CREATE USER ON *.* TO 'adm2008'@'127.0.0.1';

SET PASSWORD FOR 'pfc2008'@'127.0.0.1' = PASSWORD('m4tr1cul1ll4');
SET PASSWORD FOR 'adm2008'@'127.0.0.1' = PASSWORD('JuanRa2008');
```

Creación de las tablas

Tabla Usuarios

```
DROP TABLE IF EXISTS usuarios;
CREATE TABLE `usuarios` (
  `id_usuario` mediumint(10) NOT NULL auto_increment,
  `USUARIO` varchar(25) NOT NULL,
  `PASSWORD` varchar(25) default NULL,
  `COMENTARIO` text,
  PRIMARY KEY (`id_usuario`)
) ENGINE=MyISAM AUTO_INCREMENT=7 DEFAULT CHARSET=latin1;
```

Tabla Eventos

```
DROP TABLE IF EXISTS t_eventos;
CREATE TABLE `t_eventos` (
  `id_evento` mediumint(10) NOT NULL auto_increment,
  `nombre` varchar(50) default NULL,
  `descripcion` text,
  `vinculo` varchar(100) default NULL,
  `anio_i` varchar(10) default NULL,
```

```
`mes_i` varchar(10) default NULL,  
`dia_i` varchar(10) default NULL,  
`anio_f` varchar(10) default NULL,  
`mes_f` varchar(10) default NULL,  
`dia_f` varchar(10) default NULL,  
PRIMARY KEY (`id_evento`)  
) ENGINE=MyISAM AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;
```

Tabla Noticias

```
DROP TABLE IF EXISTS t_noticias;  
CREATE TABLE `t_noticias` (  
  `id_noticias` mediumint(10) NOT NULL auto_increment,  
  `titulo` mediumtext,  
  `texto` longtext,  
  `vinculo` varchar(250) default NULL,  
  `anio_i` varchar(10) default NULL,  
  `mes_i` varchar(10) default NULL,  
  `dia_i` varchar(10) default NULL,  
  `archivado` tinyint(1) default '0',  
  PRIMARY KEY (`id_noticias`)  
) ENGINE=MyISAM AUTO_INCREMENT=22 DEFAULT CHARSET=latin1;
```

Tabla Categorías Profesionales

```
DROP TABLE IF EXISTS t_categoria;  
CREATE TABLE `t_categoria` (  
  `id_categoria` mediumint(10) NOT NULL auto_increment,  
  `categoria` varchar(100) default NULL,  
  `n_abrev` varchar(50) default NULL,  
  `tiempo` varchar(50) default NULL,  
  PRIMARY KEY (`id_categoria`)  
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=latin1;
```

Tabla Herramientas

```
DROP TABLE IF EXISTS t_herramientas;  
CREATE TABLE `t_herramientas` (  
  `id_herramientas` mediumint(10) NOT NULL auto_increment,  
  `nombre_h` varchar(250) default NULL,  
  `n_abrev` varchar(50) default NULL,  
  `descripcion_h` text,  
  `img_h_1` varchar(250) default NULL,  
  `img_h_2` varchar(250) default NULL,  
  `img_h_3` varchar(250) default NULL,  
  `img_h_4` varchar(250) default NULL,  
  `img_h_5` varchar(250) default NULL,  
  `link_h` varchar(250) default NULL,  
  PRIMARY KEY (`id_herramientas`)  
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=latin1;
```

Tabla Personal

```

DROP TABLE IF EXISTS t_profesores;
CREATE TABLE `t_profesores` (
  `id_profesores` mediumint(10) NOT NULL auto_increment,
  `nombre` varchar(50) default NULL,
  `apellidos` varchar(50) default NULL,
  `despacho` varchar(200) default NULL,
  `telefono` varchar(20) default '+34-91',
  `fax` varchar(20) default NULL,
  `direccion` varchar(200) default NULL,
  `email` varchar(100) default NULL,
  `web` varchar(200) default NULL,
  `imagen_profesor` varchar(250) default NULL,
  `categoria` mediumint(10) default NULL,
  `descripcion` text,
  `activado` tinyint(1) default '1',
  PRIMARY KEY (`id_profesores`),
  KEY `categoria` (`categoria`),
  CONSTRAINT `t_profesores_ibfk_1` FOREIGN KEY (`categoria`) REFERENCES
`t_categoria` (`id_categoria`) ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=34 DEFAULT CHARSET=latin1;

```

Tabla Proyectos de Fin de Carrera

```

DROP TABLE IF EXISTS t_pfc;
CREATE TABLE `t_pfc` (
  `id_pfc` mediumint(10) NOT NULL auto_increment,
  `nombre` varchar(50) default NULL,
  `descripcion` text,
  `id_tutor1` mediumint(10) default NULL,
  `id_tutor2` mediumint(10) default NULL,
  `vinculo` varchar(250) default NULL,
  PRIMARY KEY (`id_pfc`),
  KEY `id_tutor1` (`id_tutor1`),
  KEY `id_tutor2` (`id_tutor2`),
  CONSTRAINT `t_pfc_ibfk_1` FOREIGN KEY (`id_tutor1`) REFERENCES
`t_profesores` (`id_profesores`) ON DELETE CASCADE ON UPDATE NO
ACTION,
  CONSTRAINT `t_pfc_ibfk_2` FOREIGN KEY (`id_tutor2`) REFERENCES
`t_profesores` (`id_profesores`) ON DELETE NO ACTION ON UPDATE
CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=26 DEFAULT CHARSET=latin1;

```

Tabla Líneas de Investigación

```

DROP TABLE IF EXISTS t_li;
CREATE TABLE `t_li` (
  `id_li` mediumint(10) NOT NULL auto_increment,
  `nombre` varchar(50) default NULL,
  `n_abrev` varchar(50) default NULL,
  `descripcion` text,
  PRIMARY KEY (`id_li`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=latin1;

```

Tabla SubLíneas de Investigación

```

DROP TABLE IF EXISTS t_sli;
CREATE TABLE `t_sli` (
  `id_sli` mediumint(10) NOT NULL auto_increment,
  `sublinea` varchar(250) default NULL,
  `n_abrev` varchar(50) default NULL,
  `descripcion` text,
  `id_li` mediumint(10) default NULL,
  PRIMARY KEY (`id_sli`),
  KEY `id_li` (`id_li`),
  CONSTRAINT `t_sli_ibfk_1` FOREIGN KEY (`id_li`) REFERENCES `t_li`
(`id_li`) ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=latin1;

```

Tabla Apartados de Líneas de Investigación

```

DROP TABLE IF EXISTS t_sli_a;
CREATE TABLE `t_sli_a` (
  `id_sli_a` mediumint(10) NOT NULL auto_increment,
  `apartado` varchar(250) default NULL,
  `id_sli` mediumint(10) default NULL,
  PRIMARY KEY (`id_sli_a`),
  KEY `id_sli` (`id_sli`),
  CONSTRAINT `t_sli_a_ibfk_1` FOREIGN KEY (`id_sli`) REFERENCES `t_sli`
(`id_sli`) ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=18 DEFAULT CHARSET=latin1;

```

Tabla SubApartado de la Línea de Investigación

```

DROP TABLE IF EXISTS t_sli_sa;
CREATE TABLE `t_sli_sa` (
  `id_sli_sa` mediumint(10) NOT NULL auto_increment,
  `subapartado` varchar(250) default NULL,
  `texto` text,
  `contacto` mediumint(10) default NULL,
  `herramienta` mediumint(10) default NULL,
  `vinculo` varchar(250) default NULL,
  `id_sli_a` mediumint(10) NOT NULL,
  PRIMARY KEY (`id_sli_sa`),
  KEY `id_sli_a` (`id_sli_a`),
  KEY `herramienta` (`herramienta`),
  KEY `contacto` (`contacto`),
  CONSTRAINT `t_sli_sa_ibfk_1` FOREIGN KEY (`id_sli_a`) REFERENCES
`t_sli_a` (`id_sli_a`) ON DELETE NO ACTION ON UPDATE CASCADE,
  CONSTRAINT `t_sli_sa_ibfk_2` FOREIGN KEY (`herramienta`) REFERENCES
`t_herramientas` (`id_herramientas`) ON DELETE NO ACTION ON UPDATE
CASCADE,
  CONSTRAINT `t_sli_sa_ibfk_3` FOREIGN KEY (`contacto`) REFERENCES
`t_profesores` (`id_profesores`) ON DELETE NO ACTION ON UPDATE
CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=latin1;

```


Tabla Proyectos del Departamento

```

DROP TABLE IF EXISTS t_proyectos;
CREATE TABLE `t_proyectos` (
  `id_proyectos` mediumint(10) NOT NULL auto_increment,
  `nombre` varchar(100) default NULL,
  `resumen` mediumtext,
  `descripcion` mediumtext,
  `dia_i` mediumint(10) default NULL,
  `mes_i` mediumint(10) default NULL,
  `anio_i` mediumint(10) default NULL,
  `dia_f` mediumint(10) default NULL,
  `mes_f` mediumint(10) default NULL,
  `anio_f` mediumint(10) default NULL,
  `financiacion` varchar(250) default NULL,
  `ref` varchar(50) default NULL,
  `lider` mediumint(10) default NULL,
  `sli` mediumint(10) default NULL,
  `miembros` mediumtext,
  `keywords` mediumtext,
  `vinculo` varchar(250) default NULL,
  `liderado` tinyint(1) default '1',
  `activo` tinyint(1) default '1',
  PRIMARY KEY (`id_proyectos`),
  KEY `sli` (`sli`),
  KEY `lider` (`lider`),
  CONSTRAINT `t_proyectos_ibfk_1` FOREIGN KEY (`sli`) REFERENCES `t_sli`
(`id_sli`) ON DELETE NO ACTION ON UPDATE CASCADE,
  CONSTRAINT `t_proyectos_ibfk_2` FOREIGN KEY (`lider`) REFERENCES
`t_profesores` (`id_profesores`) ON DELETE NO ACTION ON UPDATE
CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=latin1;

```

Tabla Publicaciones

```

DROP TABLE IF EXISTS t_publicacion;
CREATE TABLE `t_publicacion` (
  `id_publicacion` mediumint(10) NOT NULL auto_increment,
  `titulo` varchar(250) default NULL,
  `n_abreviado` varchar(50) default NULL,
  `texto` text,
  `vinculo` varchar(250) default NULL,
  `fichero` varchar(250) default NULL,
  `id_sli` mediumint(10) default NULL,
  PRIMARY KEY (`id_publicacion`),
  KEY `id_sli` (`id_sli`),
  CONSTRAINT `t_publicacion_ibfk_1` FOREIGN KEY (`id_sli`) REFERENCES
`t_sli` (`id_sli`) ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=22 DEFAULT CHARSET=latin1;

```

Tabla intermedia asignación Titulaciones-Proyectos Fin de Carrera

```
DROP TABLE IF EXISTS t_tit_pfc;
CREATE TABLE `t_tit_pfc` (
  `id_tit_pfc` mediumint(10) NOT NULL auto_increment,
  `id_tit` mediumint(10) default NULL,
  `id_pfc` mediumint(10) default NULL,
  PRIMARY KEY (`id_tit_pfc`),
  CONSTRAINT `t_tit_pfc_ibfk_1` FOREIGN KEY (`id_tit`) REFERENCES
`t_titulacion` (`id_titulacion`) ON DELETE NO ACTION ON UPDATE CASCADE,
  CONSTRAINT `t_tit_pfc_ibfk_2` FOREIGN KEY (`id_pfc`) REFERENCES
`t_pfc` (`id_pfc`) ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=31 DEFAULT CHARSET=latin1;
```

Tabla titulaciones

```
DROP TABLE IF EXISTS t_titulacion;
CREATE TABLE `t_titulacion` (
  `id_titulacion` mediumint(5) NOT NULL auto_increment,
  `n_titulacion` varchar(50) default NULL,
  `n_abreviado` varchar(50) default NULL,
  PRIMARY KEY (`id_titulacion`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=latin1;
```

Carga inicial de datos

Debido a la cantidad de datos que tiene el fichero de carga de datos, no aporta nada reproducir en esta memoria la carga de datos completa. A continuación se muestra el ejemplo de la carga de la tabla de líneas de investigación. El fichero completo se encuentra en la memoria USB en el directorio *fuentes*.

Tabla Líneas de Investigación

```
LOCK TABLES t_li WRITE;
/*!40000 ALTER TABLE t_li DISABLE KEYS */;
INSERT INTO t_li VALUES (2, 'Educational Tecnology', 'that covers all the
software applications developed with educational purpouses.\r\n');
INSERT INTO t_li VALUES (3, 'Human-Computer Interacion', 'which is an
important area to take into account. Note that our clients, the students, will
interact with our educational software. We have to ensure that this interaction
does not disturbe, and enrichs if possible, the educational experience of our
students. \r\n');
INSERT INTO t_li VALUES (1, 'Computer Science Education', 'which is the field
where we apply most of our research results. This line overlaps with most of the
following lines.');
```

```
INSERT INTO t_li VALUES (7, 'Miscellaneous', 'branch, where some of us have
research interest.\r\n');
/*!40000 ALTER TABLE t_li ENABLE KEYS */;
UNLOCK TABLES;
```

Anexo C. Código asociado a la Web

Dado que hay una gran cantidad de código HTML y PHP, sólo se muestran brevemente algunos de los extractos de código que pudieran resultar más interesantes.

A.1 Web pública del laboratorio

En toda la Web pública solo se realizan consultas de datos, por lo que en todos los casos se realiza una consulta (SELECT o SELECT DISTINCT) a la base de datos, que en algunos casos puede llevar algún tipo de filtrado (WHERE), u orden (ORDER BY).

Para generar los listados, se cuenta el número de registros procedentes de la consulta y mediante el uso de bucles FOR se repite el código necesario para mostrar cada registro.

Estos bucles FOR pueden estar anidados para poder mostrar datos referenciados por estos registros (personas de contacto, herramientas asociadas, etc). Un ejemplo abreviado sería:

tools.php

```

$query = "SELECT * FROM t_herramientas ORDER BY nombre_h";
$idConsulta = mysql_query($query,$idConexion);
$numRegistros=mysql_num_rows($idConsulta);
for($i=0;$i<$numRegistros;$i++)
{
    $resultQuery = mysql_fetch_array($idConsulta);
    $id = $resultQuery['id_herramientas'];
    $nombre = $resultQuery['nombre_h'];
    $descripcion = $resultQuery['descripcion_h'];
    $i1 = $resultQuery['img_h_1'];
    $i2 = $resultQuery['img_h_2'];
    ...
    $enlace = $resultQuery['link_h'];
    $nombre = limpia_cadena ($nombre);
    $descripcion = limpia_cadena ($descripcion);
    if ($id<>'1')
    {
        echo "<div class='middle-column-box-white'>";
        echo "<div class='middle-column-box-title-grey'><a name='$nombre'
        id='$nombre'> </a>$nombre</div>";
        echo "$descripcion";
        echo "<div align='center'><br>";
        if ($i1<> ""){ echo "<img src='$i1' alt='$nombre'
        height='250' width='400'> <br> &nbsp;<br>";}
        if ($i2<> ""){ echo "<img src='$i2' alt='$nombre'
        height='250' width='400'> <br> &nbsp;<br>";}
        ...
        echo "</div>";
        echo "<p class='middle-column-linktop'>";
        if ($enlace!=""){echo "<a href='$enlace'>More
        Details</a> | ";};
        echo "<a href='#top'>Top of the page</a></p>";
        echo "</div>";
    }
}

```

En este ejemplo, cabe destacar el uso de la sentencia IF, que se usa en distintas partes de la Web para no mostrar campos vacíos o huecos en caso de que no existan datos.

También se usó la sentencia IF para el caso de los nombres abreviados en los que en los navegadores no aparecía correctamente. En el siguiente extracto de código se muestra el nombre corto si existe y si no el nombre normal:

```
if ($n_abrev<>"){echo "<li><a class='left-navheader' href='reslines-
li.php?id=$id_linea'>$n_abrev</a></li>";}
else {echo "<li><a class='left-navheader' href='reslines-
li.php?id=$id_linea'>$nombre_li</a></li>";}
```

Y otro uso de la sentencia IF que merece la pena remarcar es el utilizado en las páginas *events.php* y *news.php*. En ellas se compara la fecha de la noticia o del evento con la fecha actual para poder mostrarlas en diversos apartados.

Como ejemplo:

Los eventos están en curso (la fecha actual está entre el comienzo y el fin del evento)

```
if($fecha_act <= "$fechafin") { if ($fecha_act >= $fechainicio) {}};;
```

Es un evento futuro (fecha de comienzo posterior a la actual)

```
if($fecha_act < $fechainicio) {};
```

O un evento pasado (fecha de finalización anterior a la fecha actual)

```
if($fecha_act >= $fechafin) {};
```

Mediante el uso del WHERE podemos realizar filtrados, por ejemplo, sublíneas asociadas a cierta línea de investigación.

```
$Query2 = "SELECT DISTINCT * FROM t_sli WHERE id_li='$id_linea' ORDER BY
sublinea;";
```

A través de los distintos enlaces de cada persona de la página de personal (*people.php*), se pasa el parámetro ID, este es recogido en la página de detalles (*people-detail.php*) mediante una sentencia GET. Con dicho parámetro, se realiza la consulta a la base de datos usando, el modificador WHERE, lo que nos da los datos de la persona en cuestión

people.php

```
<a href='people-detail.php?id=$id'>More details</a>
```

people-detail.php

```
$id_pr=mysql_escape_string($_GET['id']);
$query = "SELECT * FROM t_profesores WHERE id_profesores='$id_pr';";
```

A.1 Portal de administración

En esta parte se usan inserciones, consultas, borrados y modificaciones en la base de datos.

Consultas:

Para la parte de consultas, se procede igual que en la Web pública, realizando una o varias consultas anidadas en bucles FOR.

Inserciones:

En la parte de inserciones, en la primera página (*xxx_alta.php*) existe un formulario, que mediante el método POST envía los datos a la página que realiza las inserciones en la base de datos (*xxx_alta_insertar.php*) mediante el uso de una sentencia INSERT.

Ej. *usr_alta_insertar.php*

```
"INSERT INTO usuarios (USUARIO,PASSWORD,COMENTARIO) VALUES
('$usuario','$password','$comentario');";
```

Es interesante el alta de *líneas de investigación*, ya que la página contiene una serie de desplegables, que aparecen en función de la acción realizada.

Si en vez de seleccionar “Alta Línea de Investigación” desplegamos y seleccionamos una línea existente, la página se recarga, pasándose como parámetro dicha línea. En ese momento aparecerá un segundo desplegable en el que se muestran las sublíneas asociadas a la línea seleccionada previamente.

Así mismo, si en vez de seleccionar “Alta Sublínea de Investigación” desplegamos y seleccionamos una sublínea existente, se procederá de la misma manera, recargando la página y pasando como parámetros la línea y sublínea seleccionadas, mostrando los apartados asociados a la selección anterior. Y esto mismo ocurre con los subapartados.

El ejemplo de código es el siguiente:

Tras realizar una selección (ONCHANGE):

```
<select name='sli' onchange='reload2(this.form)'><option value=''>--Alta
SubLinea de Investigacion--</option>
```

Se ejecuta la función RELOAD2, que vuelve a llamar a la página pasando, en este caso, como parámetro la línea y sublínea.

```
function reload2(form)
{var v1=form.li.options[form.li.options.selectedIndex].value;
var v2=form.sli.options[form.sli.options.selectedIndex].value;
self.location='li_alta.php?li=' + v1 + '&sli=' + v2 ;}
```

Dichas variables son recogidas cada vez al comienzo de la página mediante el siguiente código.

```
$li=$HTTP_GET_VARS['li'];
$sli=$HTTP_GET_VARS['sli'];
```

Bajas:

Para el proceso de baja, en la primera página (*xxx_baja.php*) existe un desplegable para seleccionar el registro a eliminar. Los datos que se muestran en este desplegable se obtienen igual que las consultas, realizando la consulta dentro de un bucle FOR en el que se enumeran todos los registros.

Al seleccionar el registro y pulsar el botón correspondiente, se efectúa mediante el método POST, el envío del registro a eliminar a la página *xxx_baja_conf.php*. Una vez mostrados los datos (extraídos mediante una consulta a la base de datos con un filtro WHERE y pasando como parámetro el ID del registro a eliminar) y confirmada la acción de borrar el registro, es redirigido a la página *xxx_baja_eliminar.php*. En esta última página, además de proceder a la eliminación del registro en la base de datos y de mostrar un mensaje indicando que el registro se ha borrado satisfactoriamente, en caso de existir imágenes o ficheros anexos, estos son eliminados del directorio correspondiente (mediante el uso de la sentencia UNLINK).

Al igual que se ha comentado en el apartado de inserciones, en *líneas de investigación*, en la página *li_baja.php* existe un desplegable que varía en función de la selección para poder gestionar las bajas de líneas, sublíneas, apartados y subapartados de investigación.

Modificaciones:

El apartado de modificaciones es una mezcla del apartado de bajas y el de altas.

La primera página (*xxx_modif.php*) es parecida a la de bajas, ya que existe un desplegable para seleccionar el registro a modificar. Esto se aplica también a los desplegables de *líneas de investigación*, cuyo comportamiento ya ha sido explicado.

Una vez seleccionado y pasado como parámetro el ID del registro a modificar, el formulario de la página *xxx_modif_xxx.php* es similar a la de altas, salvo que los campos que contienen datos almacenados en la base de datos aparecen con dichos datos.

Si existen imágenes o ficheros en el registro mostrado, estos pueden ser respetados, eliminados (seleccionando una casilla de verificación puesta a tal efecto) o sustituidos por otros.

Una vez realizadas las modificaciones pertinentes y tras pulsar el botón correspondiente y confirmar el cambio, se envían el ID del registro a la página *xxx_modif_update.php*, en la que se efectúa la modificación de los registros correspondientes en la base de datos mediante el uso de una sentencia UPDATE

```
{ $sql="UPDATE t_li SET nombre='$nombreli";  
$sql.=",descripcion='$descripcionli',n_abrev='$n_abrev";  
$sql.=" WHERE id_li='$idli';" ; }
```

Dos grupos de páginas que merece la pena comentar son:

- Líneas de investigación
 - El formulario mostrado en la página *li_modif_li.php* presenta los campos correspondientes a dicha selección, además de permitir cambiar la pertenencia a su nivel jerárquico superior. Esto es, para sublíneas, además de cambiar el nombre de sublínea y descripción (campos propios de la sublínea), se puede cambiar a la línea de investigación a la que pertenece; para apartados, el nombre (campo propio del apartado) y la sublínea a la que pertenece, etc.

- Publicaciones
 - Como se comentó en la descripción de las páginas en el capítulo 5 tras comprobar en la página *pbl_modif_update.php* (mediante una sentencia IF) si en la página *pbl_modif_pbl.php* ha sido marcada la casilla de verificación, aparecerá un vínculo que permite modificar la línea y sublínea de investigación a la que está asociada una publicación. Esta pasa el ID de la publicación y la sublínea para poder realizar la actualización en la base de datos a la página *pbl_modif_pbl2.php*, que a su vez lo pasará posteriormente a la página *pbl_modif_update.php* para su actualización (UPDATE) en la base de datos.
 - El comportamiento de esta página es parecido al indicado en *li_modif_li.php*, ya que recarga la página en caso de seleccionar una línea de investigación distinta (lo que permite cambiar a cualquier sublínea de investigación existente).

Anexo D. Instalación del entorno para las pruebas finales

Para las pruebas en un entorno distinto al de desarrollo, se descargó la última distribución de Ubuntu versión Server para 32bit, en el momento de realizar las pruebas la versión era la 8.10. Este software se descargó de la Web de Ubuntu [16].

Debido a la falta de un PC de fácil acceso en el que montar dicho entorno, se instaló en una máquina virtual, para ello se eligió VMWare versión Server, ya que es gratuita y mucho más potente, estable y con mayor rendimiento que otras versiones también gratuitas, como VirtualPC. Este software, en el momento de la redacción de este documento, pudo descargarse en la página del fabricante [17].

Una vez instalado VMWare, se genera una nueva máquina virtual con Bridged Network (para que aparezca como una máquina más en la red), un pequeño disco (ya que únicamente va a llevar un pequeño servidor Web y una pequeña base de datos) y se asignan sólo 256Mb de memoria RAM, debido a que no se va a utilizar ningún entorno gráfico, únicamente el servidor Web y servidor de base de datos.

Para efectuar la instalación del servidor Ubuntu, se procede, a grandes rasgos de la siguiente manera:

- Se arranca la máquina desde la unidad de CD (asignando a la máquina virtual el fichero .ISO descargado de la Web de Ubuntu, no es necesario generar un CD).
- Se selecciona el idioma para el arranque del asistente de instalación idioma, en nuestro caso, seleccionamos el Español.
- En el menú, se selecciona “Instalar Ubuntu Server”.
- Se elige Idioma (país) para la instalación del servidor, en nuestro caso, de nuevo el Español (España).
- Se llega a la detección de la distribución del teclado, en el que se solicita que se presionen ciertas teclas.
- Se intenta configurar la red por DHCP; para el servidor es recomendable una IP fija, pero esta se configurara más adelante.
- Se introduce el nombre de máquina (en nuestro caso, el nombre asignado es Ulab)
- Se selecciona ubicación (península).
- Se efectúa el particionado de discos, en nuestro caso se selecciona la opción “Guiado, utilizar todo el disco” y aceptamos las indicaciones.
- Se crea cuenta de usuario (creamos la cuenta pfc2008, con la contraseña WebLite).
- Omitimos la creación de un directorio cifrado, así como la configuración de un servidor proxy.
- Se selecciona “Sin actualizaciones automáticas”, ya que en este entorno no se va a poner con conectividad a Internet.
- Se selecciona la colección de programas “LAMP Server”, lo que nos instalará de manera predeterminada un servidor Apache, MySQL y soporte para PHP.
- Por último, se solicita la configuración (optativa) de la contraseña de administrador (root) del servidor MySQL (en nuestro caso, introducimos la contraseña pfc2008j).

Con esto, finaliza la instalación del servidor Ubuntu. El siguiente paso es configurar la red del servidor. Para ello, editamos el fichero interfaces:

```
sudo vi /etc/network/interfaces
```

Y lo modificamos, cambiando los parámetros que quedaron durante la instalación (IP dinámica).

```
auto eth0
iface eth0 inet dhcp
```

Dejándolo de la siguiente manera:

```
auto eth0
iface eth0 inet static
address 192.168.1.85
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1
```

Para que surtan efecto los cambios, paramos y arrancamos la red

```
sudo /etc/init.d/networking restart
```

Una vez realizada la instalación, es necesaria la copia de los ficheros que componen la Web

1. Creamos un directorio “pen” para montar la memoria USB.
2. Realizamos el montaje de dicha memoria en el punto de montaje que definimos previamente.
3. Realizamos la copia del directorio de la memoria en el que esta contenida la Web al servidor Ubuntu.

```
sudo mkdir pen
sudo mount -t vfat /dev/sdb1 /mnt/pen
sudo cp /mnt/pen/xampplite/htdocs/xampp/pfc /var/www -R
```

El siguiente paso es la creación de la base de datos y la carga de los datos, tal como se explicaba en el capítulo 4 para el entorno de XAMPP. En este caso realizaremos los siguientes pasos:

1. arrancaremos el intérprete de comandos de MySQL con el usuario root

```
sudo mysql -u root -p
```


e introducimos la contraseña (pfc2008).
2. Cargamos el fichero con la definición de base de datos, usuarios y tablas.

```
source /mnt/pen/fuentes/tablas.sql
```
3. Cargamos el fichero con los datos

```
source /mnt/pen/fuentes/datos.sql
```

A partir de este punto ya se podría acceder a la Web desde otro puesto de usuario mediante un navegador en la URL adecuada. En nuestro caso, y siguiendo el procedimiento adjunto, sería: <http://192.168.1.85/pfc>

En caso de que quisiéramos que arrancara en la URL <http://192.168.1.85/> los datos se copiarían directamente en la ruta /var/www en vez de en la ruta /var/www/pfc.

Anexo E. Contenido de la memoria USB. Puesta en marcha del entorno de prueba

En la memoria USB facilitado se encuentra una copia en formato electrónico de este mismo documento, así como una copia del entorno XAMPP, la Web y la base de datos asociada al proyecto, para permitir la puesta en marcha de la Web.

NOTA:

Como ya se ha indicado anteriormente, es importante que no se esté ejecutando ningún servidor Web ni ningún servidor de SQL, ya que los puertos necesarios (80 y 3306 respectivamente) estarían bloqueados.

Esquema del contenido de la memoria USB

Por directorios:

- Fuentes:
 - Contiene los ficheros .sql para la generación de la base de datos y para la carga inicial de datos.
 - *web_lite.sql*: fichero para la generación de la base de datos.
 - *web_lite_datos_iniciales.sql*: fichero para la carga inicial de datos.
 - También está el fichero .sql para la generación de la base de datos de la Web de pruebas.
 - *web_pruebas.sql*: fichero para la generación de la base de datos y datos de prueba.
- Memoria:
 - Contiene la memoria del proyecto en formatos MS Word, PDF.
 - 2008-2009-PFC-Memoria_Web_LITE-JuanRPrats.doc.
 - 2008-2009-PFC-Memoria_Web_LITE-JuanRPrats.pdf.
- Programas:
 - Contiene algunos de los programas usados durante la generación del proyecto (XAMPP).
- Scripts:
 - Scripts de arranque del entorno, generados para mayor comodidad.
- Xampplite:
 - Directorio que contiene el entorno XAMPP versión lite (Servidores Apache y MySQL) descomprimido, preparado para ser puesto en marcha.
 - Contiene la base de datos con la carga inicial de datos hecha.

Puesta en marcha del entorno contenido en la memoria USB facilitada, para entornos Windows.

Aproximadamente son los mismos pasos que los indicados en el Anexo A, pero personalizados para la memoria USB facilitada.

1.- Tras introducir la memoria USB, hay que ejecutar el fichero “setup_xampp.bat”, éste modifica los path asociados a los servidores Apache y MySQL.

En caso de no necesitar modificaciones aparecerá un texto indicándolo (ver imagen 19):

Sorry, but ... nothing to do!
Presione una tecla para continuar . . .

En caso contrario, en la ventana que sale hay que elegir la primera opción (ver imagen 18):
“1) Refresh Now” (poner un 1 y pulsar la tecla Enter)

Al cabo de un tiempo, que dependerá de la velocidad de acceso del almacenamiento dedicado al entorno (mucho más rápido en disco que en la memoria) y del PC en el que se ejecute, indicará lo siguiente (ver imagen 19):

Press any key to continue

Presionar *Enter*, con lo que ya estarán modificadas las variables de entorno.

2.- Después es necesario arrancar el servidor Web y el servidor de base de datos, para ello hay que ejecutar los ficheros *apache_start.bat* y *mysql_start.bat* localizados en el directorio xampplite.

3.- Una vez que ambos servicios estén arrancados, podremos arrancar navegadores para acceder a las distintas páginas:

Web del laboratorio

En la Web pública, están los datos que se verían desde Internet.

<http://localhost/xampp/pfc/>

Web administración

En la Web de administración se encuentran las páginas de administración.

<http://localhost/xampp/pfc/adm/>

Web inicial de prueba

En esta memoria también está incluida la Web inicial de prueba

<http://localhost/xampp/wprueba/>

4.- Una vez se quiera dar por finalizado el acceso a estas Webs, ejecutar los ficheros *mysql_stop.bat* y *apache_stop.bat* para la parada ordenada de la base de datos y del servidor Web y, posteriormente, ir al icono de expulsar hardware para poder retirar de manera segura la memoria USB.

Anexo F. Configuración del servidor APACHE

El directorio de publicación raíz por defecto es `\xampp\htdocs`

Las páginas de arranque están definidas en el fichero `httpd.conf`, bajo “DirectoryIndex”

Si fuera necesario, se editaría el fichero `conf/httpd.conf` para modificar los parámetros.

Los ficheros de configuración están en:

- Configuración básica del servidor Apache: `.\xampp\apache\conf\httpd.conf`
- Configuración SSL: `.\xampp\apache\conf\ssl.conf`
- Configuración Perl (módulo): `.\xampp\apache\conf\perl.conf`
- Configuración Tomcat (módulo): `.\xampp\apache\conf\java.conf`
- Configuración Python (módulo): `.\xampp\apache\conf\python.conf`
- Configuración PHP: `.\xampp\apache\bin\php.ini`
- Configuración MySQL: `.\xampp\mysql\bin\my.cnf`
- Configuración phpMyAdmin: `.\xampp\phpMyAdmin\config.inc.php`

Directorio	Contenido
<code>\xampp\anonymous</code>	Directorio de ejemplo de FTP anónimo
<code>\xampp\apache</code>	Directorio del servidor Apache
<code>\xampp\cgi-bin</code>	Directorio para la ejecución de scripts cgi
<code>\xampp\FileZillaFTP</code>	Directorio del servidor FileZilla FTP
<code>\xampp\htdocs</code>	Directorio principal de documentos http
<code>\xampp\install</code>	Directorio de instalación y configuración de XAMPP (no borrar!)
<code>\xampp\licenses</code>	Directorio con los acuerdos de licencias
<code>\xampp\MercuryMail</code>	Directorio del servidor Mercury Mail SMTP POP3 IMAP.
<code>\xampp\mysql</code>	Directorio del servidor MySQL
<code>\xampp\perl</code>	Directorio de Perl
<code>\xampp\php</code>	Directorio de PHP (4+5)
<code>\xampp\phpmyadmin</code>	Directorio de phpMyAdmin
<code>\xampp\security</code>	Directorio extra para configuraciones de seguridad
<code>\xampp\tmp</code>	Directorio temporal
<code>\xampp\webalizer</code>	Directorio de estadísticas web Webalizer
<code>\xampp\webdav</code>	Directorio de ejemplo de WebDAV

Estos datos están extraídos de la Web de los desarrolladores [18]