



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# BASE DE DADES DE PROBLEMES PER A UNA ASSIGNATURA

PROYECTO FINAL DE CARRERA

Ingeniería técnica en Informática de Sistemas

*Autor:* Esther Claudia Costa Segarra

*Director:* Ramón Esteban Romero

14 de diciembre de 2012

## Resumen

La finalidad de este proyecto es la creación de una base de datos de problemas generados con  $\text{\LaTeX}$  utilizando software libre, que nos permita generar listados de problemas ya sean elegidos o aleatorios, y poder modificar sus variables, si es necesario, para realizar diferentes modelos de exámenes o algún otro uso particular. Por ello, vamos a explicar la metodología a llevar a cabo para conseguir esta finalidad.

*Palabras clave:*  $\text{\LaTeX}$ , base de datos, software libre, archivos

# Índice general

# Capítulo 1

## Bases de datos

En este primer capítulo vamos a ver lo que son las bases de datos y luego nos centraremos en la que utilizaremos para esta tarea. Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente. Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro.

Se define una base de datos como una serie de datos organizados y relacionados entre sí, la cual ha de tener unas características principales, entre las que podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.

- Acceso a través de lenguajes de programación estándar.

Por otra parte, para su correcto funcionamiento ha de tener unos ciertos componentes:

- Hardware: constituido por un dispositivo de almacenamiento como discos, tambores, cintas...
- Software: que es el DBMS (*Data Management System* (Sistema de Gestión de Base de Datos)).
- Datos: los cuales están almacenados de acuerdo a la estructura externa y van a ser procesados para convertirse en información.

Los Sistemas de Gestión de Base de Datos (DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos (DDL), de un lenguaje de manipulación de datos (DML) y de un lenguaje de consulta (SQL). Por ello, hay que mencionar la importancia de una apropiada elección de DBMS, ya que se tiene que elegir el que tenga la mejor interfaz a las necesidades del trabajo y sobre todo que este apto para la base de datos que vayamos a utilizar.

## 1.1. MySQL: código abierto

MySQL es la DBMS *open source* (código abierto) más popular, la condición de *open source* de MySQL, hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores.

MySQL es un sistema de administración de bases de datos para bases de datos relacionales. Siendo una aplicación que permite gestionar archivos llamados de bases de datos.

MySQL fue escrito en C y C++ (Hay un documento que describe algunas de sus estructuras internas en <http://dev.mysql.com/doc/internals/en>) y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.

Las razones que nos han hecho decidir por este DBMS empiezan con la ventaja de que es *open source*, por lo que podemos manejarlo con total libertad, por otra parte su disponibilidad en gran cantidad de plataformas y sistemas, la posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación, soporte físico, capacidad, distribución geográfica..., facilidad en transacciones y claves foráneas, búsqueda e indexación de campos de texto y por supuesto la fiabilidad con su conectividad segura.

# Capítulo 2

## WampServer

### 2.1. Wamp

WAMP es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- Windows, como sistema operativo;
- Apache, como servidor web;
- MySQL, como gestor de bases de datos;
- PHP (generalmente), Perl, o Python, como lenguajes de programación.

El uso de un WAMP permite servir páginas html a internet, además de poder gestionar datos en ellas, al mismo tiempo un WAMP, proporciona lenguajes de programación para desarrollar aplicaciones web. En nuestro caso es perfecto ya que utilizamos el entorno Windows, pero:

LAMP es el sistema análogo que corre bajo ambiente Linux.

MAMP es el sistema análogo que corre bajo ambiente Macintosh.

### 2.2. ¿Qué es WampServer?

WampServer es un entorno de desarrollo web para Windows con el que podemos crear aplicaciones web con Apache, PHP y MySQL. También incluye PHPMyAdmis y SQLiteManager para manejar las bases de datos con facilidad.

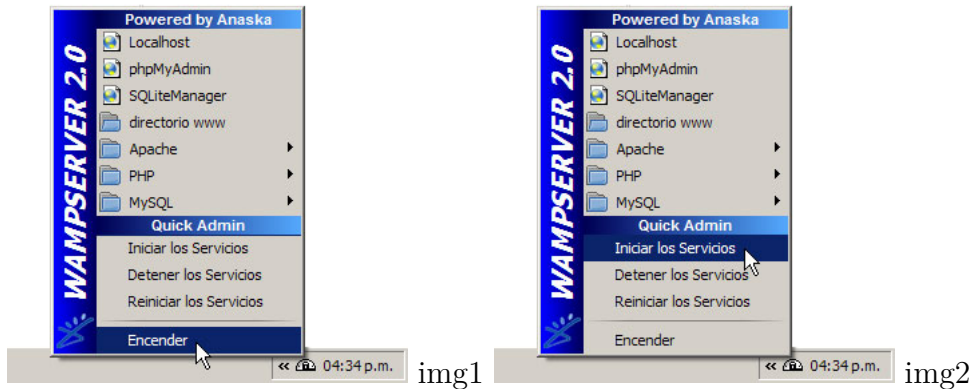
Este entorno provee a los desarrolladores con todos los elementos necesarios para gestionar un servidor web: un manejador de bases de datos (MySQL), un software para servidor web (Apache) y un software de programación *script* Web (generalmente PHP, Python o PERL).

### 2.2.1. Instalación WampServer

Si nos dirigimos a la página oficial <sup>1</sup> de WampServer, podemos ver que hay un enlace de descarga del paquete instalador, con todo lo necesario para facilitarnos su utilización. Para realizar su instalación tenemos que descargarnos el ejecutable y seguir los pasos de la instalación, en un momento, la instalación nos pedirá que definamos nuestro navegador por defecto, debemos buscar el archivo en nuestro disco (ej. C:\Archivos de Programas\Mozilla Firefox\firefox.exe, o el navegador que tu utilices). También nos pedirá que configuremos los datos de nuestro servidor SMTP, en caso de tener, ingresamos los datos, en caso contrario dejamos los datos que vienen por defecto y seguimos en “Next >”.

Una vez el programa esté instalado, lo ejecutamos y si queremos cambiar el idioma, ya que viene por defecto en inglés, clicleamos en el botón derecho sobre el nuevo icono que generó en la barra de tareas (al lado del reloj), y definimos *language >spanish*.

Para utilizarlo, lo primero es encender el servidor, para ello hacemos clic con el botón izquierdo sobre el icono, y le damos clic a “Encender” (ver img1). Y luego clic a “Iniciar los servicios” (ver img2).



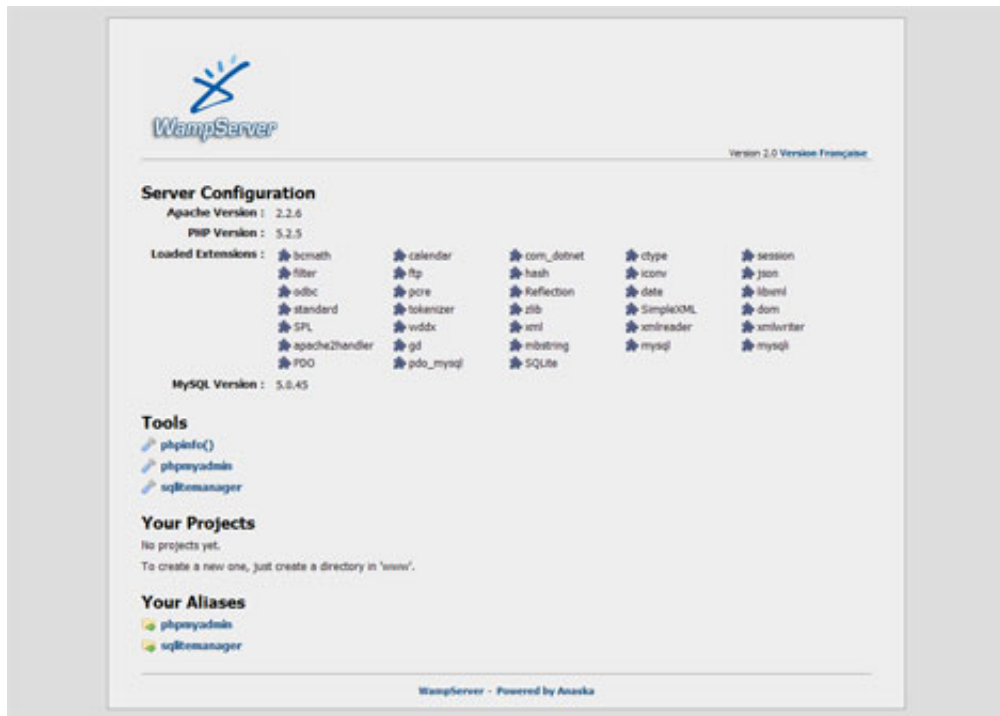
Ahora ya está instalado, y para comprobarlo accederemos a nuestro *localhost*, para ello, abrimos nuestro navegador favorito y tecleamos la siguiente url: <http://localhost>

Nos debería abrir una página como la siguiente:

---

<sup>1</sup>[www.wampserver.es](http://www.wampserver.es)





img3

Por último, recordar que para poder ejecutar nuestros scripts en nuestro servidor debemos guardarlos en la carpeta www dentro del directorio donde instalamos el wamp (por defecto: C:\wamp\www).

Las funciones que es capaz de realizar WampServer las podemos ver si hacemos clic en el icono (antes mencionado), y nos permitirán:

- Gestionar los servicios de Apache y MySQL
- Cambiar de línea / fuera de línea
- Instalar y cambiar de Apache, PHP y MySQL emisiones
- Acceder a los registros
- Acceder a los archivos de configuración
- Crear alias
- Gestión de la configuración de sus servidores

# Capítulo 3

## Nuestra Base de datos

### 3.1. El campo BLOB y TEXT

Antes de crear nuestra base de datos, tenemos que saber, que lo estándar, no es guardar los archivos en la propia base de datos, es por ello, que necesitamos de un campo especial si queremos conseguirlo, y ese es el campo BLOB y el campo TEXT. Un BLOB es un objeto binario que puede tratar una cantidad de datos variables. Los cuatro tipos BLOB son TINYBLOB, BLOB, MEDIUMBLOB, y LONGBLOB. Difieren sólo en la longitud máxima de los valores que pueden tratar. Los cuatro tipos TEXT son TINYTEXT, TEXT, MEDIUMTEXT, y LONGTEXT. Se corresponden a los cuatro tipos BLOB y tienen las mismas longitudes y requerimientos de almacenamiento. Las columnas BLOB se tratan como cadenas de caracteres binarias (de bytes). Las columnas TEXT se tratan como cadenas de caracteres no binarias (de caracteres). Las columnas BLOB no tienen conjunto de caracteres, y la ordenación y la comparación se basan en los valores numéricos de los bytes. Las columnas TEXT tienen un conjunto de caracteres y se ordenan y comparan en base de la colación del conjunto de caracteres asignada a la columna desde MySQL 4.1.

No hay conversión de mayúsculas/minúsculas para columnas TEXT o BLOB durante el almacenamiento o la recuperación.

Si asigna un valor a una columna BLOB o TEXT que exceda la longitud máxima del tipo de la columna, el valor se trunca. Si los caracteres truncados no son espacios, aparece una advertencia. Puede hacer que aparezca un error en lugar de una advertencia usando el modo SQL estricto.

En la mayoría de aspectos, puede tratar una columna BLOB como VARBINARY que puede ser tan grande como desee. Similarmente, puede tratar columnas TEXT como VARCHAR. BLOB y TEXT difieren de VARBINARY

y VARCHAR en los siguientes aspectos:

- No se eliminan espacios al final para columnas BLOB y TEXT cuando los valores se almacenan o recuperan. Antes de MySQL 5.0.3, esto difiere de VARBINARY y VARCHAR, para los que se eliminaban los espacios al final cuando se almacenaban.

Tenga en cuenta que TEXT realiza comparación espacial extendida para coincidir con el objeto comparado, exactamente como CHAR y VARCHAR.

- Para índices en columnas BLOB y TEXT, debe especificar una longitud de prefijo para el índice. Para CHAR y VARCHAR, la longitud de prefijo es opcional.
- BLOB y TEXT no pueden tener valores DEFAULT

En MySQL 5.0, LONG y LONG VARCHAR se mapean con el tipo de datos MEDIUMTEXT. Esto existe por compatibilidad. Si usa el atributo BINARY con el tipo de columna TEXT, se asigna la colación binaria del conjunto de caracteres a la columna.

MySQL Connector/ODBC define los valores BLOB como LONGVARBINARY y valores TEXT como LONGVARCHAR.

Como los valores BLOB y TEXT pueden ser extremadamente grandes, puede encontrar algunas restricciones al usarlos:

- Sólo los primeros `max_sort_length` bytes de la columna se usan al ordenar. El valor por defecto de `max_sort_length` es 1024; este valor puede cambiarse usando la opción `-max_sort_length` al arrancar el servidor `mysqld`.

Puede hacer que haya más bytes significativos al ordenar o agrupar incrementando el valor de `max_sort_length` en tiempo de ejecución.

Cualquier cliente puede cambiar el valor de su variable de sesión `max_sort_length`:

```
mysql>SET max_sort_length = 2000;
mysql>SELECT id, comment FROM tbl_name
->ORDER BY comment;
```

Otra forma de usar GROUP BY o ORDER BY en una columna BLOB o TEXT conteniendo valores grandes cuando quiere que más de `max_sort_length` bytes sean significativos es convertir el valor de la columna en un objeto de longitud fija. La forma estándar de hacerlo es con la función SUBSTRING. Por ejemplo, el siguiente comando causa que 2000 bytes de la columna `comment` se tengan en cuenta para ordenación:

```
mysql>SELECT id, SUBSTRING(comment,1,2000) FROM tbl_name  
->ORDER BY SUBSTRING(comment,1,2000);
```

- El tamaño máximo de un objeto BLOB o TEXT se determina por su tipo, pero el valor máximo que puede transmitir entre el cliente y el servidor viene determinado por la cantidad de memoria disponible y el tamaño de los buffers de comunicación. Puede cambiar el tamaño de los buffers de comunicación cambiando el valor de la variable `max_allowed_packet`, pero debe hacerlo para el servidor y los clientes. Por ejemplo, `mysql` y `mysqldump` le permite cambiar el valor de la variable del cliente `max_allowed_packet`. Cada valor BLOB o TEXT se representa internamente como un objeto a parte. Esto se hace en contraste con todos los otros tipos de columnas, para los que el almacenamiento se hace una vez por columna cuando se abre la tabla.

## 3.2. Creación de la base de datos

Nuestra base de datos contendrá los nuestros archivos `.tex`. Antes de abordar los scripts de nuestro caso, es importante crear la base de datos (BD) que va a ser utilizada por los mismos, para ello tenemos que seguir los siguientes pasos:

Creación de la base de datos Una vez dentro de MySQL, el primer paso es crear la base de datos:

```
Create database DB_problemas;
```

## 3.3. Crear y guardar la miniatura

La idea de nuestra base de datos de problemas es facilitar la creación de exámenes, por ello puede ser de gran utilidad ver una imagen miniatura del problema y saber a simple vista si es el que nos interesa o no. Por otra parte, los ejercicios guardados en la base de datos, son del tipo `.tex`, por lo que para poder crear la miniatura, debemos primero obtener el archivo pdf de cada uno de los problemas (compilándolo con PDF<sub>La</sub>TeX) y guardar una imagen de ello. Una vez acabado este proceso, podemos comenzar a construir nuestras miniaturas.

Para crear la miniatura necesitamos la Librería Gráfica GD, en este script estamos usando la versión 2.0.28 de esta librería. Sabremos si está activada, con el `phpinfo()`, que nos informará de la versión que estamos utilizando. Si no la tenemos activada solo tenemos que modificar el archivo `php.ini` que

se encuentra en C:\Windows (puede variar según la versión de Windows), y agregar la línea `extension=php_gd2.dll` en la sección "Dynamic Extensions". El archivo `php_gd2.dll` debe estar en la carpeta 'extensions' dentro del directorio donde se instaló el php, por ejemplo C:\php\extensions.

Empezemos con el formulario de upload y el script php:

### Php:

```
<?php
// Verificamos que el formulario no ha sido enviado aun
$postback = (isset($_POST["enviar"])) ? true : false;
if($postback){
    // Nivel de errores
    error_reporting(E_ALL);
    // Constantes
    # Altura de el thumbnail en píxeles
    define("ALTURA", 100);
    # Nombre del archivo temporal del thumbnail
    define("NAMETHUMB", "/tmp/thumbtemp"); //Esto en servidores Linux,
    en Windows podría ser:
// define("NAMETHUMB", "c:/windows/temp/thumbtemp");
    y no tendríamos los problemas de permisos
    # Servidor de base de datos
    define("DBHOST", "localhost");
    # nombre de la base de datos
    define("DBNAME", "test");
    # Usuario de base de datos
    define("DBUSER", "root");
    # Password de base de datos
    define("DBPASSWORD", "");
    // Mime types permitidos
    $mimetypes = array("image/jpeg", "image/pjpeg",
        "image/gif", "image/png");
    // Variables de la foto
    $name = $_FILES["foto"]["name"];
    $type = $_FILES["foto"]["type"];
    $tmp_name = $_FILES["foto"]["tmp_name"];
    $size = $_FILES["foto"]["size"];
    // Verificamos si el archivo es una imagen válida
    if(!in_array($type, $mimetypes))
```

```
    die("El archivo que subiste no es una imagen válida");
// Creando el thumbnail
switch($type) {
    case $mimetypes[0]:
    case $mimetypes[1]:
        $img = imagecreatefromjpeg($tmp_name);
        break;
    case $mimetypes[2]:
        $img = imagecreatefromgif($tmp_name);
        break;
    case $mimetypes[3]:
        $img = imagecreatefrompng($tmp_name);
        break;
}
$datos = getimagesize($tmp_name);
$ratio = ($datos[1]/ALTURA);
$ancho = round($datos[0]/$ratio);
$thumb = imagecreatetruecolor($ancho, ALTURA);
imagecopyresized($thumb, $img, 0, 0, 0, 0, $ancho,
    ALTURA, $datos[0], $datos[1]);
switch($type) {
    case $mimetypes[0]:
    case $mimetypes[1]:
        imagejpeg($thumb, NAMETHUMB);
        break;
    case $mimetypes[2]:
        imagegif($thumb, NAMETHUMB);
        break;
    case $mimetypes[3]:
        imagepng($thumb, NAMETHUMB);
        break;
}
// Extrae los contenidos de las fotos
# contenido de la foto original
$fp = fopen($tmp_name, "rb");
$tfoto = fread($fp, filesize($tmp_name));
$tfoto = addslashes($tfoto);
fclose($fp);
# contenido del thumbnail
$fp = fopen(NAMETHUMB, "rb");
$thumb = fread($fp, filesize(NAMETHUMB));
```

```

    $tthumb = addslashes($tthumb);
    fclose($fp);
    // Borra archivos temporales si es que existen
    @unlink($tmp_name);
    @unlink(NAMETHUMB);
    // Guardamos todo en la base de datos
    #nombre de la foto
    $nombre = $_POST["nombre"];
    $link = mysql_connect(DBHOST, DBUSER, DBPASSWORD)
    or die(mysql_error($link));;
    mysql_select_db(DBNAME, $link) or die(mysql_error($link));
    $sql = "INSERT INTO tabla(nombre, foto, thumb, mime)
    VALUES
    ('$nombre', '$tfoto', '$tthumb', '$type')";
    mysql_query($sql, $link) or die(mysql_error($link));
    echo "Fotos guardadas";
    exit();
}
?>

```

**Html:**

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Imagen a Blob</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body>
<form name="frmimage" id="frmimage" method="post"
    enctype="multipart/form-data" action="<?php
    echo $_SERVER['PHP_SELF'];?>">
    Nombre: <input type="text" id="nombre" name="nombre" /><br />
    Imagen: <input type="file" id="foto" name="foto" /><br />
    <input type="submit" name="enviar" id="enviar" value="Guardar" />
</form>
</body>
</html>

```

Las primeras líneas controlan que el formulario aun no se ha enviado, después definimos constantes para guardar los archivos en nuestra base de datos, solo las cambiamos según nos convenga.

Después verificamos si el archivo que subimos es una imagen, para esto verificamos su tipo MIME. Los tipos que se permiten en este script son jpeg, gif y png. En caso contrario recibiremos el mensaje "El archivo que subimos no es una imagen válida".

Según el tipo MIME de la imagen, crearemos la miniatura con las funciones `imagecreatefromjpeg()`, `imagecreatefromgif()` o `imagecreatefrompng()`.

La función `imagecopyresized()` crea la miniatura de la imagen, aunque también podemos utilizar `imagecopyresampled()`.

Y por último falta el script de creación de la tabla de miniaturas:

**MySQL:**

```
CREATE TABLE 'tabla_mini' (  
  'idfoto' int(3) NOT NULL AUTO_INCREMENT,  
  'nombre' varchar(255) NOT NULL DEFAULT '',  
  'foto' blob NOT NULL,  
  'thumb' blob NOT NULL,  
  'mime' varchar(40) NOT NULL DEFAULT '',  
  PRIMARY KEY ('idfoto')  
) ;
```

## 3.4. Guardar archivos en nuestra base de datos

Una vez que ya hemos visto como guardar las miniaturas, guardar un archivo en la base de datos es muy parecido, veamos:

**Php:**

```
<?php  
// Verificamos que el formulario no ha sido enviado aun  
$postback = (isset($_POST["enviar"])) ? true : false;  
if($postback){  
  // Nivel de errores  
  error_reporting(E_ALL);  
  // Constantes  
  
  # Nombre del archivo temporal del thumbnail  
  define("NAMETHUMB", "/tmp/thumbtemp"); //Esto en servidores Linux,
```



```
    en Windows podría ser:
// define("NAMETHUMB", "c:/windows/temp/thumbtemp");
y no tendríamos los problemas de permisos
# Servidor de base de datos
define("DBHOST", "localhost");
# nombre de la base de datos
define("DBNAME", "test");
# Usuario de base de datos
define("DBUSER", "root");
# Password de base de datos
define("DBPASSWORD", "");
// Mime types permitidos
$mimetypes = "text/plain";
// Variables del archivo
$name = $_FILES["prob"]["name"];
$type = $_FILES["prob"]["type"];
$tmp_name = $_FILES["prob"]["tmp_name"];
$size = $_FILES["prob"]["size"];
// Verificamos si el archivo válido
if(!in_array($type, $mimetypes))
    die("El archivo que subiste no es imagen válido");
// Extrae los contenidos del archivo
# contenido original
$fp = fopen($tmp_name, "rb");
$tprob = fread($fp, filesize($tmp_name));
$tprob = addslashes($tprob);
fclose($fp);
// Borra archivos temporales si es que existen
@unlink($tmp_name);
// Guardamos todo en la base de datos
#nombre del archivo
$nombre = $_POST["nombre"];
$link = mysql_connect(DBHOST, DBUSER, DBPASSWORD)
or die(mysql_error($link));
mysql_select_db(DBNAME, $link) or die(mysql_error($link));
$sql = "INSERT INTO tabla_arch(nombre, prob, mime)
VALUES
    ('$nombre', '$tprob', '$type')";
mysql_query($sql, $link) or die(mysql_error($link));
echo "Archivos guardados";
exit();
```

```
}  
?>
```

#### Html:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<title>Archivo a Blob</title>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
</head>  
<body>  
<form name="frmimage" id="frmimage" method="post"  
    enctype="multipart/form-data" action="<?php  
    echo $_SERVER['PHP_SELF'];?>">  
    Nombre: <input type="text" id="nombre" name="nombre" /><br />  
    Archivo: <input type="file" id="prob" name="prob" /><br />  
    <input type="submit" name="enviar" id="enviar" value="Guardar" />  
</form>  
</body>  
</html>
```

E igual que en el apartado anterior, solo nos queda crear la tabla de archivos, que lo haremos según el siguiente script:

#### MySQL:

```
CREATE TABLE 'tabla_arch' (  
    'idprob' int(3) NOT NULL AUTO_INCREMENT,  
    'nombre' varchar(255) NOT NULL DEFAULT '',  
    'prob' blob NOT NULL,  
    'mime' varchar(40) NOT NULL DEFAULT '',  
    PRIMARY KEY ('idprob')  
) ;
```

### 3.5. Recuperar archivos de nuestra base de datos

Una vez hecho la selección de los problemas que necesitamos, vamos a recuperarlos, para ello haremos uso de unos scripts, que lo que nos devolverá

### 3.5. Recuperar archivos de nuestra base de datos

será el contenido de cada archivo, es decir, el código L<sup>A</sup>T<sub>E</sub>X del problema, que luego utilizaremos para completar el documento/examen final.

Para que el script se muestre como un texto necesitamos usar la cabecera Content-type y poner el tipo MIME del archivo que recuperaremos de la tabla, esto es:

#### PHP:

```
\begin{center}
header("Content-type: text/plain");
\end{center}
```

Donde text/plain es el el tipo MIME para un texto plano, como es nuestro caso.

Los archivos han sido guardados en la tabla y tienen un único identificador representado por el campo idProb, el archivo original se guarda en el campo prob y el tipo MIME en el campo mime, como podemos ver en la estructura de la tabla creada anteriormente:

#### MySQL:

```
CREATE TABLE 'tabla_arch' (
  'idprob' int(3) NOT NULL AUTO_INCREMENT,
  'nombre' varchar(255) NOT NULL DEFAULT '',
  'prob' blob NOT NULL,
  'mime' varchar(40) NOT NULL DEFAULT '',
  PRIMARY KEY ('idprob')
) ;
```

Necesitamos enviar parámetros al script para indicarle el id del archivo que queremos ver, en nuestro caso necesitamos del parámetro idprob que indica el id del archivo. El script trabaja de la siguiente manera:

#### PHP:

```
// Parámetros para recuperar el archivo
# Recuperamos el parámetro GET con el id único del archivo
  que queremos mostrar
$idprob = (isset($_GET["idprob"])) ? $_GET["idprob"] : exit();
```

Ya sabiendo el id del archivo que vamos a mostrar, hacemos la consulta a la base de datos:

**PHP:**

```
// Recuperamos el archivo de la tabla
$sql = "SELECT $campo, mime
        FROM tabla_arch
        WHERE idprob = $idprob";
# Conexión a la base de datos
$link = mysql_connect(DBHOST, DBUSER, DBPASSWORD)
or die(mysql_error($link));
mysql_select_db(DBNAME, $link) or die(mysql_error($link));
$conn = mysql_query($sql, $link) or die(mysql_error($link));
$datos = mysql_fetch_array($conn);
// El archivo
$miProb = $datos[0];
// El mime type del archivo
$mime = $datos[1];
// Gracias a esta cabecera, podemos ver el archivo
// que acabamos de recuperar del campo blob
header("Content-Type: $mime");
// Muestra el archivo
echo $miProb;
```

# Capítulo 4

## L<sup>A</sup>T<sub>E</sub>X

### 4.1. ¿Qué es L<sup>A</sup>T<sub>E</sub>X?

L<sup>A</sup>T<sub>E</sub>X es un sistema de creación de textos, especialmente orientado para cubrir las necesidades de los técnicos y científicos. Está basado en un lenguaje de composición de bajo nivel llamado T<sub>E</sub>X y fundamentalmente consiste en una colección de 'macros' que facilitan el uso de este potente lenguaje. A diferencia de otros sistemas para procesar textos no se obtiene el resultado final a medida que se va escribiendo sino que primero se crea un código fuente y luego se procesa para llegar al documento. En este sentido es similar a los lenguajes de 'marcas' como el HTML.

El creador de T<sub>E</sub>X es Donald E. Knuth. Su trabajo fue un encargo de la American Mathematical Society a principios de los años 70. Esta sociedad buscaba un lenguaje para formatear sus artículos llenos de teoremas y fórmulas matemáticas muy complejas. El resultado fue un lenguaje extremadamente potente y configurable pero también difícil de aprender y de usar. Para facilitar el trabajo con T<sub>E</sub>X fueron surgiendo "macros" que agrupaban diferentes instrucciones de T<sub>E</sub>X.

El responsable inicial del L<sup>A</sup>T<sub>E</sub>X es Leslie Lamport. Aunque en su página podemos encontrar algunos estilos de L<sup>A</sup>T<sub>E</sub>X y poco más.

Una de las grandes ventajas de L<sup>A</sup>T<sub>E</sub>X es la existencia de una gran cantidad de "paquetes. estándares pensados para dotar a los textos de toda la funcionalidad que se precise. Así hay paquetes para incluir gráficos, textos de lenguajes de programación, fórmulas físicas y químicas, diagramas matemáticos, etc.

El sistema L<sup>A</sup>T<sub>E</sub>X está incluido de serie en todas las distribuciones de LINUX. Esto es una razón más para instalarnos este magnífico sistema operativo. Sin embargo, para todos aquellos que estamos en el mundo Windows

también existe la posibilidad de usarlo. Desde hace unos años algunos programadores han creado distribuciones de calidad de L<sup>A</sup>T<sub>E</sub>X para Windows. Su trabajo ha tenido doble mérito: no sólo es bueno sino que también es gratis.

L<sup>A</sup>T<sub>E</sub>X presupone una filosofía de trabajo diferente a la de los procesadores de texto habituales (conocidos como WYSIWYG, es decir, «lo que ves es lo que obtienes») y se basa en comandos. Tradicionalmente, este aspecto se ha considerado una desventaja (probablemente la única). Sin embargo, L<sup>A</sup>T<sub>E</sub>X, a diferencia de los procesadores de texto de tipo WYSIWYG, permite a quien escribe un documento centrarse exclusivamente en el contenido, sin tener que preocuparse de los detalles del formato. Además de sus capacidades gráficas para representar ecuaciones, fórmulas complicadas, notación científica e incluso musical, permite estructurar fácilmente el documento (con capítulos, secciones, notas, bibliografía, índices analíticos, etc.), lo cual brinda comodidad y lo hace útil para artículos académicos y libros técnicos.

Con L<sup>A</sup>T<sub>E</sub>X, la elaboración del documento requiere normalmente de dos etapas: en la primera hay que crear mediante cualquier editor de texto llano un fichero fuente que, con las órdenes y comandos adecuados, contenga el texto que queramos imprimir. La segunda consiste en procesar este fichero; el procesador de textos interpreta las órdenes escritas en él y compila el documento, dejándolo preparado para que pueda ser enviado a la salida correspondiente, ya sea la pantalla o la impresora. Ahora bien, si se quiere añadir o cambiar algo en el documento, se deberá hacer los cambios en el fichero fuente y procesarlo de nuevo. Esta idea, que puede parecer poco práctica a priori, es conocida a los que están familiarizados con el proceso de compilación que se realiza con los lenguajes de programación de alto nivel (C, C++, etc.), ya que es completamente análogo.

El modo en que L<sup>A</sup>T<sub>E</sub>X interpreta la 'forma' que debe tener el documento es mediante etiquetas. Por ejemplo,

```
\documentclass{article}
```

le dice a L<sup>A</sup>T<sub>E</sub>X que el documento que va a procesar es un artículo. Puede resultar extraño que hoy en día se siga usando algo que no es WYSIWYG, pero las características de L<sup>A</sup>T<sub>E</sub>X siguen siendo muchas y muy variadas. También hay varias herramientas (aplicaciones) que ayudan a una persona a escribir estos documentos de una manera más visual (LyX, TeXmacs y otros). A estas herramientas se les llama WYSIWYM («lo que ves es lo que quieres decir»).

Una de las ventajas de L<sup>A</sup>T<sub>E</sub>X es que la salida que ofrece es siempre la misma, con independencia del dispositivo (impresora, pantalla, etc.) o el sistema operativo (MS Windows, MacOS, Unix, GNU/Linux, etc.) y puede ser exportado a partir de una misma fuente a numerosos formatos tales como Postscript, PDF, SGML, HTML, RTF, etc. Existen distribuciones e IDEs

de L<sup>A</sup>T<sub>E</sub>X para todos los sistemas operativos más extendidos, que incluyen todo lo necesario para trabajar. Hay, por ejemplo, programas para Windows como TeXnicCenter, para Linux como Kile, o para MacOS como TeXShop, todos liberados bajo la Licencia GPL. Existe además un editor multiplataforma (para MacOS, Windows y Unix) llamado Texmaker, que también tiene licencia GPL.

## 4.2. Creación y Compilación

En nuestro caso estamos utilizando el entorno Windows, por lo que tendremos especial atención a la hora de instalarnos L<sup>A</sup>T<sub>E</sub>X. Puesto que nuestro trabajo, es fundamentalmente la creación de una base de datos, simplemente haremos mención de la creación y compilación de un documento. Veamos como crear un documento L<sup>A</sup>T<sub>E</sub>X en un entorno Windows:

Lo primero es instalar el core de T<sub>E</sub>X. La implementación libre de T<sub>E</sub>X para Windows se llama MiKTeX. Así que tenemos que descargarnos e instalar MiKTeX<sup>1</sup>. Después de la instalación, tenemos que recurrir a un editor L<sup>A</sup>T<sub>E</sub>X que nos posibilite interactuar con el core. La oferta es amplia, pero en nuestro caso hemos elegido TeXmaker. Libre y gratuito. Ya casi estamos preparados para escribir en L<sup>A</sup>T<sub>E</sub>X. Otra de las ventajas de TeXmaker es que reconocerá que tenemos el core de MiKTeX instalado, y nos hará las asociaciones necesarias para que la totalidad de módulos esté operativa. Al menos, los módulos esenciales. En este momento, ya podemos redactar nuestro documento, siguiendo, claro está, con las pautas del código. Una vez finalizado, procedemos a su compilación, pudiendo exportarlo a una variedad de formatos, aunque el más común, es el pdf.

---

<sup>1</sup>en esta Web se encuentra todo lo necesario: <http://conocimientoadictivo.blogspot.com/2011/08/instalar-latex-en-windows.html>

# Capítulo 5

## Resultados

Llegados a este punto, ya tenemos:

- Creada una base de datos en un servidor local
- Guardado los archivos en la base de datos

Por lo que únicamente nos queda completar el resultado. En la base de datos, a parte de insertar los problemas con y sin solución, también insertamos una plantilla de examen vacía, con la finalidad de completarla con los diferentes problemas elegidos. El usuario tendrá que guardarse esta plantilla y copiar el resultado de la muestra por pantalla del problema elegido al cuerpo de la plantilla (correctamente indicado), y si quisiera modificar algún valor, podría perfectamente. Una vez realizados todos los cambios oportunos, se compila con PDFLaTeX y se obtiene el documento pdf final.

La plantilla sería un documento como este, muy simple:

```
\documentclass[12pt,a4paper]{article}
\usepackage[utf8]{inputenc}
\usepackage{amsmath}
\usepackage{makeidx}
\usepackage{graphicx}
\usepackage[left=2cm,right=2cm,top=2cm,bottom=2cm]{geometry}

\author{depMat}
\begin{document}
\begin{center}
  Examen de Matemática
\end{center}
```



```
\begin{enumerate}
\item %Insertar problema1
\item %Insertar problema2
\item %Insertar problema3
\item %Insertar problema4
\item %Insertar problema5
\item %Insertar problema6
\item %Insertar problema7
\item %Insertar problema8
\end{enumerate}
\end{document}
```

# Bibliografía

- [1] Video explicativo PHP, <http://www.youtube.com/watch?v=p1QgIhwDhcY>
- [2] Video explicativo PHP, <http://www.youtube.com/watch?v=87dHDBh7HPA>
- [3] Video explicativo MYSQL (Bases de datos), <http://www.youtube.com/watch?v=FBzyw9-HWkA>
- [4] <http://www.desarrolloweb.com/articulos/346.php>
- [5] Guardar un archivo con MySQL, <http://www.forosdelweb.com/f86/guardar-archivo-mysql-532952/>
- [6] Como guardar la ruta de un archivo en tabla MySQL, <http://gonzasilve.wordpress.com/2011/02/27/como-guardar-la-ruta-de-un-archivo-en-tabla-mysql-y-java/>
- [7] Tipos de extensiones mime, <http://www.iana.org/assignments/media-types/index.html>
- [8] Instalar L<sup>A</sup>T<sub>E</sub>X en Windows, <http://conocimientoadictivo.blogspot.com/2011/08/instalar-latex-en-windows.html>
- [9] Página de desarrolladores expertos en MySQL, <http://dev.mysql.com/>
- [10] Información sobre L<sup>A</sup>T<sub>E</sub>X, <http://es.wikipedia.org/wiki/LaTeX>
- [11] Información sobre Wamp, <http://es.wikipedia.org/wiki/WAMP>
- [12] Página oficial de WampServer, <http://www.wampserver.es>