

SOFTWARE LIBRE Y EDUCACIÓN:
SERVICIOS DE RED, GESTORES DE
CONTENIDOS Y SEGURIDAD

Contenido dinámico



José Ángel Bernal, Fernando Gordillo, Hugo Santander y Paco Villegas

22 de abril de 2005

Índice general

1. Páginas PHP	5
1.1. Introducción	5
1.2. Instalación	8
1.2.1. Configuración	10
1.3. Primeros pasos con php	18
1.3.1. Estructuras condicionales	22
1.3.2. Bucles	24
1.4. Ejemplos	25
1.4.1. Un ejemplo de Web con PHP	25
1.4.2. Representación gráfica de funciones con PHP	31
2. MySQL	37
2.1. Introducción a las BDR	37
2.2. Instalación	40
2.2.1. Configuración del servidor	41
2.3. Inicio de MySQL	43
2.3.1. Aseguremos el servidor	43
2.3.2. Un poco de comandos	44
2.4. PHPMyAdmin	47
2.4.1. Instalación	47
2.4.2. ➔Base de datos cursolinux	49
2.5. PHP y MySQL: páginas web dinámicas.	59
2.5.1. Más sentencias de PHP	59
2.5.2. Un ejemplo	63
3. Moodle y PHP-Nuke	73
3.1. Entorno virtual de aprendizaje: Moodle	73
3.1.1. Introducción.	73
3.1.2. Instalación	74
3.1.3. Primeros pasos en la administración.	80
3.1.4. Nuestro primer curso	82
3.1.5. Más configuración	85
3.2. PHP-Nuke	91
3.2.1. Introducción	91
3.2.2. Instalación de PHP-Nuke	91
3.2.3. Configuración básica del portal.	95
3.2.4. Coppermine	104

Capítulo 1

Páginas PHP

PHP, acrónimo de “PHP: Hypertext Preprocessor”, es un lenguaje “Open Source” interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl, y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP. <http://www.php.net/manual/es/preface.php>

1.1. Introducción

Hace algún tiempo¹, la información de la Web era relativamente estática. El cliente/navegador hacía una petición HTTP al servidor y éste le servía las páginas HTML que alguien había elaborado.

Este modelo se queda pequeño y tiene varios inconvenientes:

1. El mantenimiento de los enlaces y la información es tedioso. Cuando cambia algún dato, debemos recorrer todas las páginas para actualizar todos los lugares en los que aparece.
2. No enlaza con las bases de datos y aplicaciones que podamos tener en nuestra organización, con lo que perdemos una gran cantidad de información o lo que es peor, debemos duplicarla con el riesgo de desactualización.
3. El usuario se encuentra atado a la hora de seleccionar la información que desea. Solamente puede acceder a la información de la forma en la que ha previsto quien ha realizado la página web, pero no puede pedir por ejemplo que se ordene por edad, si está ordenada por apellidos.

En definitiva, podemos añadir más valor a la información si ésta es dinámica y se genera a petición del usuario.

Con el paso del tiempo, se han inventado varios métodos para conseguir que la información se genere de forma dinámica.

Los primeros en nacer fueron los *Server Side Includes* (SSI) y el interfaz *Common Gateway Interface* (CGI).

Una página SSI es una página HTML con comandos incluidos que se ejecutan en el servidor web. Por ejemplo, integrado con el código HTML se encuentra el SSI que nos detecta una cadena e imprimirá un valor dependiendo de ésta:

```
<html>
...
<body>
```

¹En la escala temporal de Internet, varios años dan para mucho.

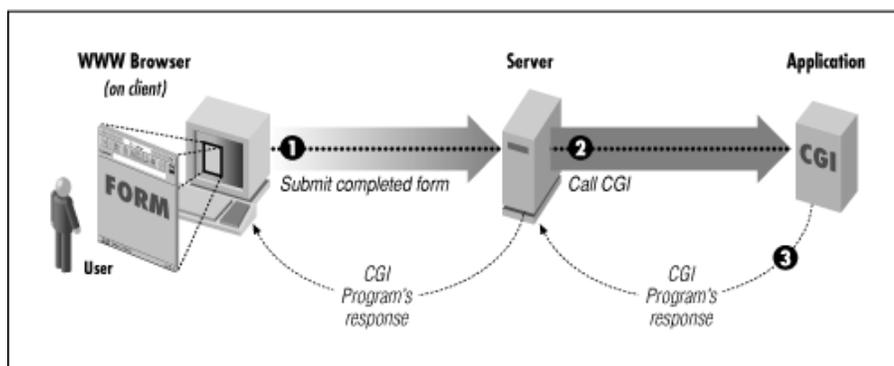
```

...
<!--#if expr="\$DOCUMENT_URI" = \"/linux/file.html\" -->
En Linux
<!--#elif expr="\$DOCUMENT_URI" = \"/windows/file.html\" -->
En Windows
<!--#else -->
Ni en Windows ni en Linux
<!--#endif -->
...
</body>
</html>

```

Es un avance, pero su potencia es muy limitada y complejo para sitios medianos y grandes.

CGI es un mecanismo estándar de comunicación entre un programa o script² y un servidor web. La limitación es que deben comunicarse de una manera determinada y poco eficiente. La entrada del cliente web se pasa al programa a través del servidor web, según el mecanismo establecido. El programa recibe la información, la procesa y devuelve el resultado al servidor web, que compone la página definitiva. El mecanismo se presenta en la siguiente imagen:



Sin embargo, este mecanismo es muy rígido y al tener que invocar a un programa externo al servidor web, es lento y consume muchos recursos tanto de memoria como de proceso.

Existen otros mecanismos para generar contenido dinámico, como los servidores de aplicaciones Java³, ColdFusion o ASP. Nuestro objetivo será conocer PHP, un lenguaje especialmente diseñado para generar páginas web dinámicas, y fácil de aprender y ejecutar.

Haciendo un poco de historia⁴, PHP comenzó en 1994 cuando RASMUS LERDORF quiso dar un paso más allá de los CGI. En los siguientes años evolucionó y comenzó a utilizarse en muchos sitios web. Un gran salto se produjo en 1997 cuando ZEEV SURASKI y ANDI GUTMANS lo reescribieron dando lugar a PHP 3.

En la actualidad disponemos de dos versiones estables de PHP, la 4 y la 5⁵. Ambas se basan en un motor denominado Zend (la 5 en Zend II). La principal diferencia entre ambas es que la versión 4 basa su programación en procedimientos y la 5, en cambio, está basada en objetos (POO <http://es.wikipedia.org/wiki/POO>). En esta entrega, así como el curso trabajaremos con la versión 4.

PHP no solamente podemos utilizarlo en sistemas Linux/Unix, sino que está disponible en multitud de plataformas, incluidas las Windows, siendo un gran competidor de ASP.

Datos recientes de Netcraft, estiman su uso en 18.455.683 dominios y 1.317.871 direcciones IP⁶.

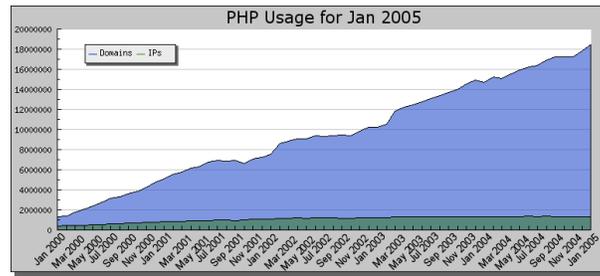
²Puede estar escrito en varios lenguajes: C, Perl...

³El estándar J2EE, está alcanzando su madurez y es adoptado por muchas empresas y organismos, pero su complejidad es alta.

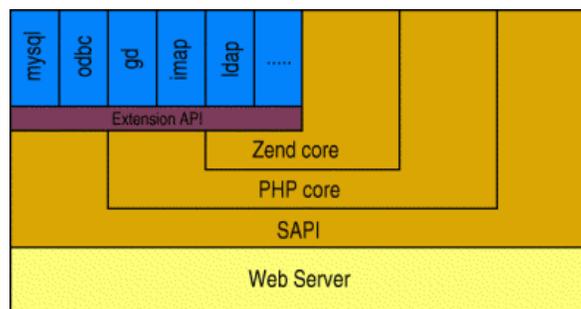
⁴Para conocer mejor esta historia se puede consultar <http://es.wikipedia.org/wiki/PHP>

⁵La primera versión estable vio la luz en julio de 2004

⁶Seguramente os llamen la atención estos datos. Como ya sabéis, una IP puede dar soporte a múltiples dominios virtuales.



La siguiente figura presenta la arquitectura de PHP. El interfaz SAPI permite integrarlo con la mayoría de los servidores web existentes. Por supuesto con Apache, pero también con IIS, Zeus, Netscape iPlanet, Java servlet, AOLServer, o Roxen.



Se pueden integrar módulos para conectarlo con bases de datos, servidores de directorios, servidores de correo...

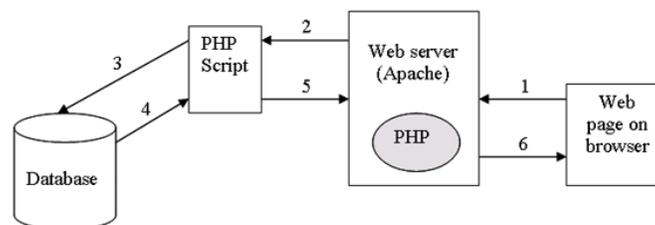
La sintaxis de PHP es similar a C, aunque hay características que ha tomado “prestadas” de Perl, C++ y Java.

PHP se ejecuta totalmente por el servidor web y al navegador le llega solamente código HTML. Aunque es compatible con lenguajes de script como javascript que se ejecutan en el navegador, con lo que podemos combinar ambas técnicas, si lo deseamos.

El siguiente código es una mezcla de HTML y PHP.

```
<html>
  <? echo date("M d, Y H:i:s", time()); ?>
</html>
```

El circuito que se forma viene dado por la siguiente figura. El navegador (1) solicita una página al servidor web. Éste ve que es una página con código PHP y la pasa al motor de PHP (2). Si es necesario, interacciona con bases de datos u otros módulos (3 y 4). Cuando la página está generada, se pasa al servidor web (5), que la traslada al cliente (6). Todo ello de una forma muy eficiente.



La salida HTML que obtendrá el navegador será la siguiente:

```
<html>
Apr 1, 2005 23:00:05
</html>
```

Resumiendo, PHP es un lenguaje de script que se ejecuta en el servidor web y permite que las páginas sean dinámicas. Nos permite conexiones a bases de datos y muchas utilidades más. La comparación con ASP de Microsoft es inmediata, pero, PHP es mucho mejor.

- El manual por excelencia: <http://www.php.net/manual/es/>
- Páginas interesantes:
 - <http://www.rinconastur.com/php/>
 - <http://jips.bankhacker.com/linux/apache/ssl/urlsphp.phtml>
 - <http://www.desarrolloweb.com/directorio/programacion/php/>

1.2. Instalación

Debian

Nos centraremos sólo en el módulo de php4 para Apache²⁷: `libapache2-mod-php4`

```
# apt-get install libapache2-mod-php4 php4-pear php4-cgi phpdoc
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Se instalarán los siguientes paquetes extras:
  php4-cli php4-common
Paquetes sugeridos:
  php4-dev
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-php4 php4-cgi php4-cli php4-common php4-pear phpdoc
0 actualizados, 6 se instalarán, 0 para eliminar y 119 no actualizados.
Necesito descargar 6876kB de archivos.
Se utilizarán 31,3MB de espacio de disco adicional después de desempaquetar.
¿Desea continuar? [S/n]
```

De los tres paquetes instalados, sólo el primero es imprescindible, los otros son:

`php4-pear` contiene archivos PEAR para php4. Esto es, numerosas clases para manejo más sencillo y limpio de, p.ej., bases de datos.

`php4-cgi` suministra CGI. Los módulos compilados son `bcmath`, `calendar`, `curl`, `dba`, `exif`, `filepro`, `ftp`, `mm`, `sockets`, `wddx`, `xml`, `yp` y `zlib`.

`phpdoc` documentación para PHP4.

Una vez instalado, reiniciamos el servidor con:

```
#apache2ctl restart
```

7

- Es el paquete “importante” ya que nos va a permitir trabajar con código php embebido en html. No podremos realizar scripts en línea de comandos con php, para eso es necesario instalar el intérprete “completo”

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

pero en este caso, nos instalará a su vez Apache 1.3. Así que no os lo recomendamos.

Red Hat/Fedora

Necesitamos el paquete⁸ `php`.

```
# apt-get install php
Leyendo listas de paquetes... Done
Construyendo árbol de dependencias... Done
Se instalarán los siguientes paquetes extras:
 php-pear
Se instalarán los paquetes NUEVOS siguientes:
 php php-pear
0 upgraded, 2 newly installed, 0 removed and 178 not upgraded.
Need to get 1642kB of archives.
After unpacking 4652kB of additional disk space will be used.
Quiere continuar? [S/n]
```

y

```
#apachectl restart
```

Hola Mundo

Ya tenemos `php` instalado, ejecutemos nuestro primer script:

- Creemos un fichero de contenido

```
# cat prueba.php
<?php
//Esto es un comentario de una línea
//Iniciamos un script php con <?php
//Mostramos el texto encerrado entre comillas
//con \n introducimos un retorno de línea
echo "Hola Mundo \n";
// La función getcwd() nos devuelve el directorio desde
// donde se ejecuta
echo getcwd(), "\n";
// Fin del script
?>
```

- Ejecutemos el intérprete con⁹

```
# php -q prueba.php
Hola Mundo
/root
```

⁸Viene en los CDs de Fedora

⁹

- Con `-q` se suprimen las cabeceras HTTP:

```
Content-type: text/html
X-Powered-By: PHP/4.3.4
```

- Otra opción para ejecutarlo es dotar al archivo de los permisos adecuados (`chmod a+x prueba.php`) y añadir al fichero la línea

```
#!/usr/bin/php -q
```

En este caso sólo hay que escribir:

```
./prueba.php
```

Pero no es la mejor forma de usarlo, ¿verdad?, veamos cómo usarlo dentro de una página web:

➔ Para practicar: Debian y Fedora

Podemos probar que nuestro servidor web funciona con el módulo PHP poniendo una página llamada `prueba.php` en el raíz del servidor Web con el contenido siguiente:

```
<html>
<head>
  <title>PHP Test</title >
</head>
<body>
  <?php echo "Prueba Curso Linux<P>"; ?>
  <?php phpinfo (); ?>
</body>
</html>
```

y apuntando nuestro navegador a `http://localhost/prueba.php`



Merece la pena pararse un poco en la salida del fichero anterior. En ella se nos informa del estado del intérprete y del valor de las distintas directivas de configuración.

1.2.1. Configuración

php.ini

El fichero de configuración de php es

Fedora `/etc/php.ini`

Debian `/etc/php4/apache2/php.ini`

con él podemos controlar el comportamiento por defecto del intérprete. Nos centraremos en analizar el que se instala por defecto en Debian. A partir de él, comprender el de Fedora no presenta dificultad.

Las directivas de este fichero mantienen las convenciones en cuanto a la sintaxis:

- Las directivas de configuración son de la forma:

`directiva = valor`

donde *valor* puede ser una cadena (`error_log=syslog`), un número (`precision=14`), una constante de PHP (`error_reporting=E_ALL`), una de las constantes¹⁰ INI (`On`, `Off`, `True`, `False`, `Yes`, `No` y `None`) o una expresión de constantes INI operadas con `|` (`OR`), `&` (`AND`), `~` (`NOT`), `!` (`FALSE`) o paréntesis `()`.

- Para introducir un comentario o comentar una directiva de configuración se le antepone un punto y coma.
- Los nombres de las directivas son sensibles a las mayúsculas y minúsculas.
- Las cabeceras de sección se indican encerrando el texto entre corchetes (por ejemplo `[MySQL]`).



No es necesario conocer todas las directivas de este fichero ni para escribir script PHP ni para seguir el resto de la entrega. El motivo de que se incluya es que os sirva de referencia por si hay que ajustar el intérprete a casos particulares. De hecho, casi con toda seguridad no tendremos que cambiar prácticamente ninguna directiva del fichero de configuración en un uso normal de PHP. Por tanto: desde este punto hasta la sección **Apache y php** (1.2.1 en la página 16) sólo habría que dar una lectura por encima para conocer las posibilidades de configuración.

Analicemos algunas directivas “importantes”. El fichero se inicia con la sección principal de configuración del intérprete

```
[PHP]
```

en ella merece la pena comentar:

Directivas generales

`engine = On` activa el intérprete de PHP como módulo de Apache.

`short_open_tag = On` si está en `On`, activa la posibilidad de que los scripts php se delimiten usando etiquetas abreviadas `<? ... ?>`. Si es `Off` hay que usar `<?php ... ?>` o `<script>`.

`asp_tags = Off` si está en `On` permite el uso de etiquetas estilo ASP `<% ... %>`.

`precision = 12` número de dígitos significativos mostrados cuando trabajamos con números en coma flotante.

`y2k_compliance = On` hace cumplir la conformidad con el año 2000 (Causará problemas con navegadores que no cumplan con esto).

...

`allow_call_time_pass_reference = On` fuerza a que se pasen las variables de las funciones por referencia

`safe_mode = Off` si está en `On`, activa el modo seguro de PHP¹¹.

...

`;open_basedir =` limita las operaciones que php puede realizar sobre archivos a los directorios especificados (separados por `:`). Tal cual está, permite abrir todos los archivos. No depende del valor de `safe_mode`

¹⁰Para activar una directiva booleana es equivalente usar: 1, `On`, `True` o `Yes`.

Para desactivarla, podemos usar: 0, `Off`, `False` o `No`

¹¹Para ampliar sobre este tema: <http://www.phpbuilder.com/manual2/manual/es/features.safe-mode.php>

`disable_functions` = permite desactivar determinadas funciones de PHP (separadas por ;).
Tampoco depende de si está activo el modo seguro.

`;highlight.string` = #DD0000 colores para resaltar la sintaxis de PHP. Los valores aceptables para la etiqueta `` se pueden poner aquí.

`;highlight.comment` = #FF9900

`;highlight.keyword` = #007700

`;highlight.bg` = #FFFFFF

`;highlight.default` = #0000BB

`;highlight.html` = #000000

`expose_php` = `On` en `On` se incluye una cadena en la cabecera http del servidor que indica qué versión de PHP está instalada.

Límites de recursos

`max_execution_time` = 30 tiempo máximo (en segundos) de ejecución de un script.

`max_input_time` = 60 tiempo máximo (en segundos) que un script puede invertir en analizar los datos requeridos.

`memory_limit` = 8M tamaño máximo de memoria que puede consumir un script¹².

Gestión y registro de errores

Podemos trabajar con varios niveles de registro de errores con

`error_reporting` = `E_ALL & ~E_NOTICE` muestra todos los errores y avisos (`E_ALL`), excluyendo (~) las advertencias en tiempo de ejecución (`E_NOTICE`) del intérprete. Podemos usar los parámetros recogidos en la tabla:

Cuadro 1.1: Errores

Nombre	Significado
<code>E_ALL</code>	todos los errores y avisos
<code>E_ERROR</code>	errores fatales en tiempo de ejecución
<code>E_WARNING</code>	avisos en tiempo de ejecución
<code>E_PARSE</code>	errores en tiempo de compilación
<code>E_NOTICE</code>	advertencias en tiempo de ejecución
<code>E_CORE_ERROR</code>	errores fatales ocurridos al iniciarse PHP
<code>E_CORE_WARNING</code>	avisos ocurridos al iniciarse PHP
<code>E_COMPILE_ERROR</code>	errores fatales en tiempo de compilación
<code>E_COMPILE_WARNING</code>	avisos en tiempo de compilación
<code>E_USER_ERROR</code>	mensajes de error generados por el usuario
<code>E_USER_WARNING</code>	avisos de error generados por el usuario
<code>E_USER_NOTICE</code>	advertencia generada por el usuario

↪ Ejemplo

```
error_reporting = E_COMPILE_ERROR|E_ERROR|E_CORE_ERROR
```

¹²Si se piensa instalar Moodle (ya lo conocéis, ¿verdad?) es mejor poner este valor a 16M.

muestra sólo errores.

`display_errors = On` en `On` muestra los errores en pantalla como parte de la salida HTML generada.

`display_startup_errors = Off` si se cambia a `On` muestra los errores al iniciar el intérprete. No le afecta que la anterior esté en `On`.

`log_errors = Off` no guardamos los log de error de PHP en un archivo de registro del servidor.

`log_errors_max_len = 1024` tamaño máximo del archivo `error_log`. Por defecto es de 1024. Si se pone a 0 no se impone restricción de tamaño.

Manipulación de datos

`;arg_separator.output = "&";` separador usado por PHP para separar argumentos en las urls generadas. Por defecto es `&`.

`;arg_separator.input = ";&";` lista de separadores usados por PHP para analizar una URL y obtener las variables pasadas en ella. Por defecto es `&`.

`variables_order = "EGPCS"` establecemos el orden (de izquierda a derecha) en que PHP registra las variables: **G**ET, **P**OST, **C**ookie, **E**ntorno¹³ y variables internas (**S**). Tal cual está, sería: Entorno, Get, Post, Cookie y Server.

`register_globals = Off` si se mantiene en `Off` impedimos que se creen automáticamente variables globales con los nombres de las variables pasadas como argumentos al script. Por defecto está desactivada.

`register_argc_argv = On` le dice a PHP que declare las variables¹⁴ `argv` y `argc`.

`post_max_size = 8M` tamaño máximo permitido de los datos usando el método POST

`magic_quotes_gpc = On` el estar en `On`, añade una barra invertida antes de la comilla sencilla (`'`), comilla doble (`"`), barra invertida (`\`) y los NUL en operaciones entrantes GET, POST o Cookie.

`magic_quotes_runtime = Off` si se pone en `On`, se activa la conversión automática de comillas para los datos generados en tiempo de ejecución (datos generados por una base datos externa, lectura de archivos de texto, etc).

`auto_prepend_file` = permite añadir el archivo especificado al final de todo fichero php ejecutado. Se incluye como si fuese invocado por una llamada mediante la función `include()`. Si se ponen a `none`, se desactiva la adición automática de archivos.

`auto_append_file` = permite añadir el archivo especificado al principio de todo fichero php ejecutado. Se incluye como si fuese invocado por una llamada mediante la función `include()`. Si se ponen a `none`, se desactiva la adición automática de archivos.

`default_mimetype = "text/html"` tipo mime por defecto para la salida de datos.

`;default_charset = "iso-8859-1"` juego de caracteres usado por defecto.

`;always_populate_raw_post_data = On` si está en `On` se crea la variable `$HTTP_RAW_POST_DATA` aunque el tipo MIME sea conocido (por defecto sólo se crea cuando es desconocido).

¹³ *Environment*

¹⁴

`argv` array de argumentos pasados en el script

`argc` número de argumentos (parámetros) pasados en el script

Rutas y directorios

`;include_path = ".:\php\includes"` lista de directorios (separados por `:`) en los que las funciones `require()`, `include()` y `fopen_with_path()` buscarán los archivos requeridos. Por defecto es el directorio "actual" (`.`).

`doc_root` = el raíz desde donde se sirven las páginas. Usado si no está vacío.

`user_dir` = directorio desde el que php abre los scripts para un usuario (`/~usuario`). Si está vacío, no se usa.

`extension_dir =/usr/lip/php4` directorio desde donde cargar las extensiones dinámicas de los módulos.

`enable_dl = On` activa la posibilidad de emplear la carga dinámica de extensiones de php, útil si se trabaja con PHP como módulo de Apache.

`;cgi.force_redirect = 1` necesario para proporcionar seguridad cuando ejecutamos PHP como un CGI. Por defecto está en `on`. Puedes ponerlo en `off` "bajo tu responsabilidad" (para IIS sí puede estar en `off`).

`;cgi.redirect_status_env = ;` si el anterior está en `on` y no estamos trabajando bajo servidores web Apache o Netscape, podemos necesitar un nombre de variable de entorno donde PHP comprobaría si está OK para continuar la ejecución. Puede originar problemas de seguridad.

`;cgi.rfc2616_headers = 0` informa a PHP del tipo de encabezados a usar cuando envía código de respuesta HTTP. Si está a 0 (valor por defecto) envía *Status*: encabezado soportado por Apache. Si está a 1, PHP enviará un encabezado RFC2616.

Subir ficheros

`file_uploads = On` permite subir archivos HTTP

`upload_tmp_dir` = directorio temporal para los archivos HTTP que hemos subido (si no se especifica, usará el del sistema por defecto)

`upload_max_filesize = 2M` tamaño máximo permitido a los archivos a subir

Directivas relacionadas con fopen

`allow_url_fopen = On` permite tratar URLs (como `http://` o `ftp://`) como archivos.

`;from="john@doe.com"` define la contraseña para ftp anónimo (su e-mail)

`;user_agent="PHP"` define la cadena `User-Agent`

`default_socket_timeout = 60` tiempo por defecto para la disponibilidad de socket (en segundos)

Extensiones dinámicas

`;extension=modulename.extension` debemos usarlas si deseamos cargar extensiones automáticamente. Por ejemplo:

`;extension=mysql.so` sólo debe estar el nombre del módulo, no es necesaria ninguna información del directorio, ya que debemos especificar la localización de la extensión con la directiva `extension_dir`. **Nota:** para Red Hat/Fedora la extensión de paquete de los módulos se carga a través de los archivos `ini` en el directorio `/etc/php.d`

Configuración de módulos de PHP

Sólo vamos a comentar las directamente relacionadas con el curso:

[Syslog]

`define_syslog_variables = Off` para definir o no las diversas variables `syslog` (ej. `$LOG_PID`, `$LOG_CRON`). Es una buena idea ponerlo en `off`. En tiempo de ejecución podemos definir estas variables con la llamada `define_syslog_variables()`.

[MySQL]

`mysql.allow_persistent = On` permite o previene enlaces persistentes.

`mysql.allow_persistent = -1` número máximo de enlaces persistentes (-1 equivale a: sin límite).

`mysql.max_links = -1` número máximo de enlaces (+número persistentes, -1 es sin límite)

`mysql.default_port =` número de puerto por defecto para `mysql_connect()`. Si no se determina, usará `$MYSQL_TCP_PORT` o la entrada en `/etc/services` o el valor de tiempo de compilación definido en `MYSQL_PORT` (en ese orden).

`mysql.default_socket =` nombre de socket por defecto para las conexiones locales MySQL. Si no se especifica, usa el que tiene por defecto MySQL.

`mysql.default_host =` host por defecto para `mysql_connect()` (no se aplica en modo "safe")

`mysql.default_user =` usuario por defecto para `mysql_connect()` (no se aplica en modo "safe")

[bcmath]

`bcmath.scale = 0` número de cifras decimales para todas las funciones `bcmath`.

[browscap]

`;browscap = extra/browscap.ini` contiene información sobre las cadenas de identificación que usa cada navegador.

[Session]

`session.save_handler = files` manipulador usado para guardar/recuperar datos.

`;session.save_path = "N;/path"` argumento pasado a `save_handler`. En el caso de archivos, éste es el path donde se guardan los archivos de datos. Desde PHP 4.0.1 se define como "N;/path" donde N es un entero. En lugar de guardar todos los archivos de sesión en `/path` guardará los datos en subdirectorios de N niveles de profundidad (útil para servidores que manejan gran cantidad de sesiones).

Nota: Php no crea esta estructura de directorios automáticamente. Para esto se puede usar el script en `ext/session` dir. En Fedora está descomentada y vale `/var/lib/php/session`

`;session.save_path = /var/lib/php4`

`session.use_cookies = 1` para usar cookies

`session.use_only_cookies = 1` esta opción permite a los administradores proteger a los usuarios de ataques derivados de pasar la identidad de sesión en URLs, por defecto es 0.

`session.name = PHPSESSID` nombre de la sesión (usado como nombre de cookie).

`session.auto_start = 0` inicia la sesión ante peticiones de arranque.

`session.cookie_lifetime = 0` tiempo de vida, en segundos, de las cookies o, si es 0, hasta que el navegador es reiniciado.

`session.cookie_path = /` path para el que es válida la cookie.

`session.cookie_domain =` dominio para el que la cookie es válida.

`session.serialize_handler = php` manipulador para serializar los datos (php es el estándar para PHP).

`;session.gc_probability = 1` probabilidad de que el proceso de “recolección de basura” comience en cada inicialización de sesión. Se calcula usando `gc_probability/gc_divisor` (ej. 1/100 significa que hay un 1% de posibilidades de que el proceso arranque en cada petición).

`session.gc_divisor = 100`

`session.gc_maxlifetime = 1440` tiempo en segundos tras el que los datos pueden ser limpiados por el proceso de recolección de basura. **Nota:** si está usando la opción `subdirectory` para grabar los archivos de sesión la recolección de basura no se hará de forma automática, necesitará un shell script, entrada `cron` u otro método.

`session.bug_compat_42 = 1`

`session.bug_compat_warm = 1` PHP 4.2 y versiones anteriores tienen un bug no documentado que permite inicializar una variable de sesión de alcance global aunque `register_globals` está deshabilitado. PHP 4.3 y posteriores avisan si se usa esta característica. Puede deshabilitar la característica y el aviso de forma separada. Por ahora, el aviso sólo aparece si `bug_compat_42` está habilitado.

`session.referer_check =` chequea HTTP Referer para invalidar las URLs externas que contienen identidades.

`session.entropy_length = 0` bytes a leer desde el archivo.

`session.entropy_file =` para especificar la session id.

`session.cache_limiter = nocache` poner en {`nocache`, `private`, `public`,} para determinar los aspectos de caché HTTP, o dejarlo vacío para permitir enviar encabezados anti-caché.

`session.cache_expire = 180` el documento expira después de `n` minutos.

`session.use_trans_sid = 0` desactivado por defecto. Su uso puede poner en peligro la seguridad de sus usuarios. Se debe usar esta opción con precaución.

Por último, comentar que podemos enviar correo a través de php.

Apache y php

La interacción entre Apache y PHP se configura a través de:

Fedora `/etc/httpd/conf.d/php.conf` en él hay una serie de directivas que tienen que ver con su instalación, y que comentaremos. En la siguiente línea se carga el módulo de php con la orden `LoadModule`:

```
LoadModule php4_module          modules/libphp4.so
```

Con

```
AddType application/x-httpd-php .php
```

conseguiamos que los archivos de extensión `.php` sean manejados por el intérprete de PHP, y con la siguiente directiva, decimos qué páginas de un directorio¹⁵ pueden ser consideradas de inicio (si no se especifica una página concreta). En este caso: `index.php` se añade a la lista que ya existiera (`index.html`, `index.htm`, `index.shtml`...).

```
DirectoryIndex index.php
```

Debian el cometido del fichero anterior lo comparten los archivos:

```
/etc/apache2/mods-available/php4.conf
/etc/apache2/mods-available/php4.load
```

y los enlaces simbólicos a ellos (para que se active) del directorio `/etc/apache2/mods-enabled/`

Con `php4.load` cargamos el módulo y con el fichero

```
$cat /etc/apache2/mods-available/php4.conf
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

conseguiamos que los archivos de extensiones `.php` o `.phps` sean manejados por el intérprete de PHP.

➔ Para practicar

Un ejemplo que simplemente esboza lo que podemos hacer con PHP es el siguiente, formado por los ficheros `prueba.html` y `accion.php`

```
$cat prueba.html
<html>
<body>
<form action="accion.php" method="POST">
  Su nombre: <input type="text" name="nombre">
  Su edad: <input type="text" name="edad">
  <input type="submit">
</form>
</body>
</html>
```

```
$cat accion.php
Hola <?php echo $nombre?>. Tienes <?php echo $edad?>años.
```



Para poder trabajar con PHP tal cual aparece en el ejemplo hay que modificar una directiva del fichero de configuración de php: `php.ini`. Esta directiva cambió su valor por defecto a partir de la versión 4.2.0 de php (con versiones anteriores estaba en ON). Se trata de hacer que `register_globals = ON`, además hay que reiniciar apache:

Debian `apache2ctl restart`

Fedora `apachectl restart`

para que los cambios sean efectivos. Si se mantiene en OFF impedimos que se creen variables globales con los nombres de las variables pasadas como argumentos al script.

Si ponemos esos ficheros en el raíz del servidor Web y desde el navegador llamamos a `prueba.html` veremos el resultado de mezclar formularios y php. ■

¹⁵Cuando especificamos un directorio sin decir la página concreta (`http://localhost` o `http://localhost/pruebas`, siendo el primer caso el directorio raíz de los documentos y el segundo, un directorio llamado `pruebas`, dentro de él)

1.3. Primeros pasos con php

En este apartado, vamos a introducir algunos conceptos básicos de PHP de forma resumida, ya que este apartado por sí sólo podría ser un curso.

PHP y HTML

Como ya hemos visto en los ejemplos anteriores el modo estándar¹⁶ de incrustar código PHP en html es

```
<?php ... ?>
```

Por ejemplo con

```
<html>
<body>
  <?php echo "Hola Mundo"; ?>
</body>
</html>
```

obtendríamos de salida el texto encerrado entre comillas (es que el dichoso `Hola Mundo` no puede faltar)



Como es el servidor el que ejecuta el código PHP, el usuario nunca llega a ver el código PHP, solamente el resultado en formato HTML, como podremos observar si en el navegador seleccionamos la opción de “Ver código fuente”.



- Como ya hemos dicho, el código PHP se inserta en las páginas HTML entre las etiquetas `<?php ?>`; para insertar instrucciones HTML dentro de estas etiquetas se utiliza la instrucción `echo`. Hace que la cadena entrecomillada (`echo "Hola Mundo";`) o variable (`echo $variable;`) escrita a continuación se escriba en el documento HTML.
- Las instrucciones de PHP se cierran con un punto y coma¹⁷. Uno de los errores más comunes (y que más dolores de cabeza da) es que se nos olvide poner el punto y coma final.

¹⁶En realidad existen cuatro estilos diferentes de etiquetas para hacerlo:

Estilo XML el que hemos comentado.

Estilo corto `<? ... ?>`

Estilo script `<script language='php'>... </script>`

Estilo ASP `<% ... %>`

¹⁷Si el código PHP consta de una sola instrucción, no es necesario ponerlo.

Cuadro 1.2: Tipos de datos

Tipo	Descripción
Integer	Números enteros
Double	Números en coma flotante
String	Cadenas de caracteres
Boolean	valor lógico: TRUE o FALSE
Array	Matrices de datos del mismo tipo
Object	Tipo especial de dato complejo

Comentarios

Disponemos de varias formas de introducir comentarios que aclaren el código, se trata de

```
// Esto es un comentario de una sola línea

# este es otro comentario de una sola línea
/*Comentario de varias lineas,
varias líneas,
varias líneas */
```

Variables

Se pueden definir las variables como puntos de almacenamiento en la memoria del ordenador. Durante la "vida útil" de una variable siempre se puede acceder a los valores que ésta almacena, siendo posible modificarlos.

Con PHP no es necesario declarar las variables (aunque es conveniente hacerlo), cuando se utiliza una variable por primera vez, ésta se considera declarada. El nombre de cualquier variable ha de empezar siempre con un signo de dólar (\$). Además, la primera letra del nombre de la variable no puede ser un número.

PHP soporta varios tipos de datos (véase la tabla 1.2) que se reconocen y asignan automáticamente. Los básicos son: **string** (texto), **integer** (número entero) y **double** (número de coma flotante).

```
$cadena="Hola Mundo"; //string
$entero = 8; //entero
$decimal = 8.0; #float
$decimal = $entero; //ahora la variable decimal es un entero
```

Cuidado, que PHP discrimina entre mayúsculas y minúsculas en los identificadores de las variables. De esa forma, las dos variables siguientes son diferentes:

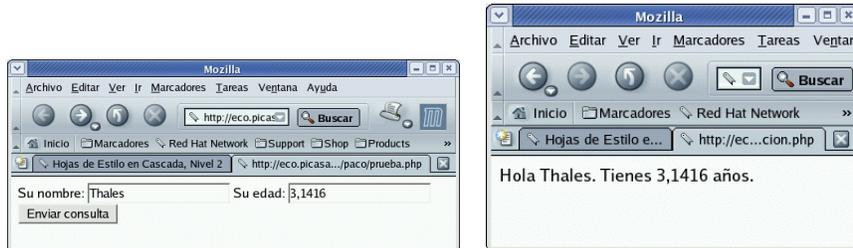
```
$mivariable=1;
$Mivariable='uno';
```

Se pueden concatenar *strings* y variables, bien escribiéndolas directamente unas a continuación de otras o, mediante el operador de concatenación que es el punto (.). Es más aconsejable este último método.

```
echo $cadena.' cruel';
```

¿Qué se obtendría?

Variables de formularios Si la variable `register_globals` del archivo de configuración de PHP está en `On`, desde PHP tendremos acceso directo a los valores introducidos en los formularios de una página Web (véase la práctica en la página 17). El resultado que obtenemos al ejecutar esa práctica es:



Es decir, hemos pasado al script `accion.php`, de forma directa, las variables `$nombre` y `$edad`, que provienen de los campos de tipo texto del formulario definido en el fichero `prueba`.

Como ya comentábamos en el ejemplo, para que eso sea posible, hemos tenido que modificar el fichero de configuración de PHP. Pero podemos hacerlo de otra forma.

Hay tres formas de acceder a las variables de un formulario:

- Estilo corto, es el comentado antes. Para eso es necesario poner `register_globals` en¹⁸ `On`.
- Estilo intermedio¹⁹. Surge a partir de la versión 4.1 de PHP. Consiste en recuperar las variables del formulario en alguna de las matrices `$_POST` (si hemos usado el método POST), `$_GET` (si lo enviamos por el método GET) y `$_REQUEST` (para ambos).

↔ Con este método el fichero `accion.php` sería de la forma:

```
Hola <?php echo $_POST['nombre'] ?>. Tienes <?php echo
$_POST['edad']?> años.
```

- Estilo largo. Es el más “engorroso” pero funciona siempre. En este caso, las variables del formulario las podemos recuperar a través de las matrices `$HTTP_POST_VARS` y `$HTTP_GET_VARS`

↔ Con este método el fichero `accion.php` sería de la forma:

```
Hola <?php echo $HTTP_POST_VARS['nombre'] ?>. Tienes <?php echo
$HTTP_POST_VARS['edad']?> años.
```

Arrays

“Un array es un conjunto de variables o registros del mismo tipo que puede estar almacenados en memoria principal o en memoria auxiliar. Los arrays de 1 dimensión se denominan también vectores, los de 2 o más dimensiones se denominan matrices. La forma de acceder a los elementos del array es directo, es decir, el elemento deseado es obtenido a partir de su índice” <http://es.wikipedia.org/wiki/Array>

A “grosso modo”, y sin entrar en más tecnicismos, podemos entender un array (table o matriz) como un nombre que permite almacenar varios datos del mismo tipo, que son accesibles usando el lugar que ocupan (comenzando desde el cero) o mediante una clave que los identifica (son los arrays asociativos). En la página 30 vemos más ejemplos sobre su uso.

↔ Por ejemplo,

- Un array que almacena los días de la semana se puede definir con

¹⁸En los dos que siguen no es necesario hacerlo.

¹⁹Es con el que trabajaremos en el ejemplo en la página 25.

```
$semana= array("Lunes", "Martes", "Miércoles", "Jueves", "Sábado", "Domingo")
```

o equivalentemente,

```
$semana[0]="Lunes";
$semana[1]="Martes";
$semana[2]="Miércoles";
$semana[3]="Jueves";
$semana[4]="Viernes";
$semana[5]="Sábado";
$semana[6]="Domingo";
```

Por ejemplo, para visualizar el 4º dato almacenado escribiremos

```
echo $semana[3]
```

- Podemos usar un array asociativo para almacenar los datos de una persona²⁰:

```
$alumno = array(nombre=>"Thales", apellido=>"de Mile-
to", edad=>"16");
```

y accederemos a los datos con:

```
echo $alumno["nombre"];
echo $alumno["apellido"];
echo $alumno["edad"];
```

Constantes

Una constante (como su nombre indica) no puede modificar su valor, se definen usando la palabra *define*

```
define['iva',16];
```

Operadores

Para conocer los operadores de PHP, os remitimos a la información disponible en la página <http://www.php.net/manual/es/language.operators.php>. Son los habituales en cualquier lenguaje de programación. Aquí tenéis un pequeño ejemplo de algunos de ellos.

```
$numero = 8 + 2; //tomaría el valor 10
$cuadrado = $numero * $numero; //valdría 100
$numero += 1; //equivale a $numero = $numero + 1;
```

Funciones

PHP trae una gran cantidad de funciones que nos pueden facilitar de forma considerable la realización de scripts (véase <http://es2.php.net/manual/es/funcref.php>). Además, podemos definir funciones. La idea de función surge de la necesidad de no repetir el mismo código²¹ varias veces. La forma general de definir una función es:

```
<?php
function foo ($arg_1, $arg_2, ..., $arg_n)
{
    echo "Función de ejemplo.\n";
```

²⁰Es equivalente a usar:

```
$alumno[nombre]="Thales";
```

y así, con el resto de valores

²¹También para facilitar su legibilidad



```
    return $retval;
}
?>
```

↔ Por ejemplo, veamos una función que nos calcula el iva del valor pasado como argumento²²:

```
function iva ($cantidad, $iva)
{
    //cálculo del iva, el valor del iva se pasa en%
    return $cantidad*$iva/100;
}
```

Para ejecutarla sólo hemos de llamarla con los parámetros adecuados, por ejemplo, con

```
echo iva(50,16);
```

obtendríamos de salida 8.

Básicamente, PHP permite trabajar con dos tipos de variables: *locales* y *globales*. Una variable es *local* cuando su uso está restringido al ámbito²³ de una función, es decir, sólo son accesibles dentro del cuerpo de la función. Una variable global es visible en toda la secuencia de comandos. Las variables globales se declaran anteponiendo la palabra `global`. Por ejemplo:

```
global $mysql_server;
```

↔ En la sección 2.5.2 en la página 65 sobre PHP y MySQL hay un script en el que es necesario usar estas últimas (en él, además, hacemos uso de un par de funciones).

Reutilización de código

PHP dispone de dos instrucciones que permiten la reutilización del código, se trata de

include cuando el intérprete llega a esta instrucción, sustituye el nombre del fichero que aparece entrecomillado por el código contenido en él. Existe otra instrucción parecida, que es **require**.

require se evalúa antes de procesarse el fichero. Es decir, PHP recorre el fichero antes de ejecutarlo, y donde encuentra un **require**, lo sustituye por el fichero "requerido" en cuestión, ejecutándose luego el código. **Include** en cambio, sólo se sustituye cuando el intérprete llega a la instrucción.

↔ En la sección 1.4.1 en la página 25 veremos un ejemplo sobre su uso.

1.3.1. Estructuras condicionales

La instrucción If

Se utiliza para tomar una decisión. La sintaxis es la siguiente:

```
if (condicion) {comandos_si_verdadera_condicion;}
else {comandos_falsa_condicion;}
```

Se evalúa la condición y en caso de ser verdadera, se ejecutan `comandos_si_verdadera_condicion`. Si la condición no es verdadera y, existe la cláusula `else`, se ejecutarían los `comandos_falsa_condicion`. Por ejemplo²⁴:

²²En PHP5 todas las variables se pasan por referencia. Para conseguir que sea así en PHP4, hay que añadirles el símbolo `&`

²³Con ámbito nos referimos a las zonas de código en las que es visible (y utilizable) una variable dada.

²⁴Se analiza un ejemplo un poco más complejo en la página 29.



```
if ($numero==10) && ($nombre=="pepe") {echo 'Pepe tiene 10 años' }
```



Un par de notas:

- Se pueden anidar.
- También existe la estructura `if` y `elseif`:

```
if(condición){si ocurre condición;}
elseif(otracondicion){si ocurre otracondición;}
else{si no ocurre ninguna de las dos anteriores;}
```

- Notar que las expresiones relacionadas con estructuras `if`, funciones o bucles se suelen agrupar mediante un par de llaves `{}`.

switch

Permite que se ejecute un bloque de instrucciones en función de un valor que tome una expresión. Es similar a una instrucción `if`. pero permitiendo que la condición tome más de dos valores.

La sintaxis es la siguiente:

```
switch (expresión)
{
  case resultado1:
    instrucciones;
    break;
  case resultado2:
    instrucciones2;
    break;
  ....
  default:
    instrucciones por defecto;
}
```

La sentencia `switch` ejecutará selectivamente los comandos que correspondan con la primera coincidencia. Esto se ve mejor con un sencillo ejemplo:

Si el valor de `$vehiculo` coincide con `automóvil`, se ejecutará el comando `echo "cuatro ruedas"` e irá al final. El otro patrón es similar, pero en el caso de que no coincida con ninguno, se ejecutarán los comandos de la entrada `default`.

```
echo "El ".$vehiculo." tiene ";
switch ($vehiculo)
{
  case 'automovil':
    echo "cuatro ruedas";
    break;
  case 'motocicleta':
    echo "dos ruedas";
    break;
  default:
    echo "no es un vehículo válido";
}
```

1.3.2. Bucles

for (y foreach)

Su sintaxis es:

```
for (inicialización; condición; actualización)
{
    instrucciones;
}
```

Su funcionamiento es como sigue: se evalúa la expresión *inicialización* (en este parámetro se suele establecer el valor inicial del contador). Después se evalúa *condición*. Si el resultado es falso, se abandona el bucle. Si es cierto, se ejecuta el bloque de *instrucciones* y se evalúa la expresión *actualización* (en esta expresión se suele ajustar el valor del contador).

↪ Por ejemplo

```
<?php
for ($i = 1; $i <=6; $i++ )
{
    echo "El número es: ".$i."<br>";
}
?>
```



La sentencia `foreach` permite recorrer todos los elementos de una matriz de una forma sencilla, su sintaxis es

```
foreach (NombreArray as Variable)
{
    instrucciones;
}
```

se puede ver un ejemplo de su uso en la página 30.

while y do...while

Usando un bucle `while` podemos ejecutar un bloque de instrucciones, mientras que la condición sea verdadera. Se usan cuando no se sabe de antemano el número de iteraciones que se deben ejecutar, sino bajo qué condición se ejecutan. La sintaxis básica es

```
while (condición) {
    instrucciones;
}
```

↪ El mismo ejemplo de la instrucción `for` quedaría

```
<?php
$i=1;
while ($i <=6)
{
    echo "El número es: ".$i."<br>";
    $i++;
}
?>
```

El bucle `do...while` es similar al generado con la instrucción `while`. La diferencia estriba en que si la condición es falsa, con el bucle `while` no ejecutamos ninguna instrucción, mientras que con el `do...while` siempre se ejecuta al menos una vez.

↔ De nuevo el ejemplo anterior, pero con `do...while`

```
<?php
$i=1;
do
{
    echo "El número es: ".$i."<br>";
    $i++;
}while ($i <=6);
?>
```

1.4. Ejemplos

1.4.1. Un ejemplo de Web con PHP

Como hemos comentado en la introducción, PHP es un sencillo lenguaje de secuencias de comandos. Normalmente se escribe directamente en las páginas HTML y es el servidor Web quien lo interpreta.

Para ver de una forma muy sencilla algunas de las cosas que se pueden hacer con PHP analizaremos el código de los siguientes ejemplos que componen un sitio web²⁵:

- `index.php`
- `pagina2.php`
- `inicio.php`
- `cabecera.php`
- `pagina1.php`
- `menu-izq.php`

Además, vamos a usar la siguiente hoja de estilo (`estilo.css`) que se lista a continuación.

```
1 H2 {text-align: center}
  H3 {text-align: center}
#cabecera{    margin-bottom: 10px;
              background-color: #000099;
              color: #ffcc99;
6             padding: 3px;
              text-align: center;
              border-bottom-style: solid;
              border-top-style: solid;
              }
11 #izquierda{ padding: 5px;
              margin: 0px;
```

²⁵Pondremos todos los ficheros en el servidor, en un fichero comprimido de nombre `ejemplo-php.tgz`

```

16         width: 10%;
           float: left;
           border-color: #000099;
           border-style: solid;
           border-width: 1px;
           background-color: #ffcc99;
           }
21 #contenido{
           padding: 10px;
           margin-left: 8%;
           float: left;
           width: 70%;
26         border-color: #000099;
           border-style: solid;
           border-width: 1px;
           }

```

Listado 1.1: estilo.css

Su interpretación no es demasiado compleja. Con ella redefinimos los encabezados de nivel 2 y 3 para que nos centre el texto y definimos tres estilos que nos van a permitir que las tres partes (cabecera, izquierda y derecha) que constituyen nuestra Web, tengan el aspecto deseado.

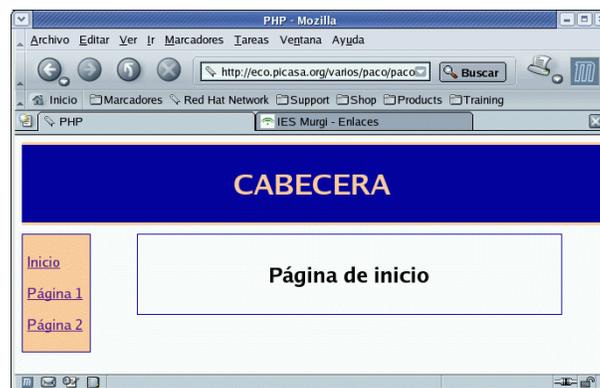
Cabecera	
Izquierda	Contenido



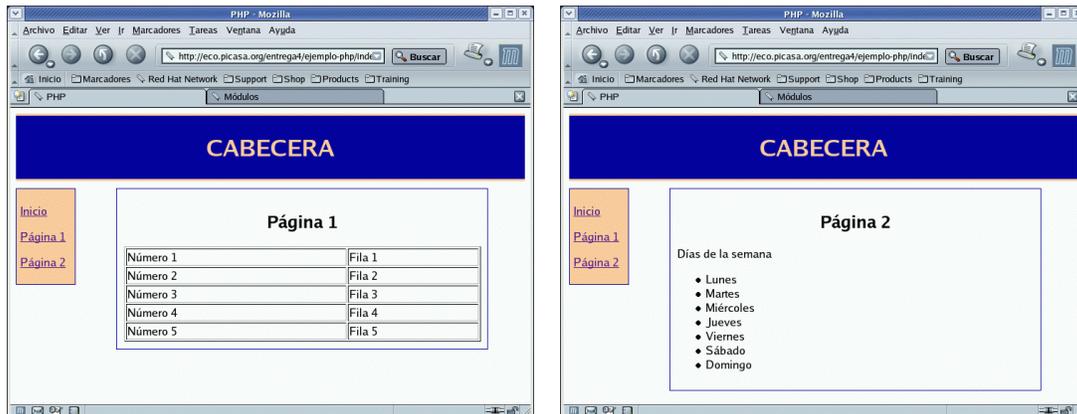
Si no conocemos las hojas de estilo y estamos interesados en ellas, en internet existen multitud de manuales que nos pueden ayudar sobre este tema. Una de las más interesantes es la Web de la especificación de las hojas de estilo <http://www.sidar.org/recur/desdi/traduc/es/css/cover.html>

Antes de analizar el código PHP de estas páginas, observaremos la página de inicio (`index.php`) a través de un navegador web donde podemos ver las distintas partes que componen la página Web y que se podrían reducir a:

- cabecera
- menú lateral izquierdo
- contenido de la página



Si accedemos a los distintos enlaces del sitio desde el menú lateral izquierdo, podemos observar que tanto éste, la cabecera y el estilo se mantienen, variando sólo el contenido central de las distintas páginas.



Estaría muy bien, por lo tanto, no tener que escribir en cada página los elementos que se repiten y escribir sólo lo que varía, que es el contenido central. Para ello podríamos pensar en generar una especie de esqueleto e insertar en él los distintos elementos de la página. Lo podemos hacer creando una página llamada `index.php`.

El código de las páginas php es el siguiente:

```

1 <html>
  <head>
    <title>PHP</title>
    <meta http-equiv="Content-Type" content="text/html;_charset=iso-8859-1"
      >
    <link rel="stylesheet" type="text/css" href="estilo.css">
6  </head>
  <body>
    <div id="cabecera"><?php include "cabecera.php"; ?></div>

    <div id="izquierda"><?php include "menu-izq.php"; ?></div>
11  <div id="contenido">
    <?php
      /*Definimos la variable 'contenido' que contendrá el nombre de
        la página
        a incluir en esta parte de la web. En caso de que no se pase
        ninguna variable
16     con la URL, la página a cargar será la de inicio.
      */
      $contenido;
      if (isset($_GET['pag'])){ $contenido = $_GET['pag'].'.php';}
      else {
21         $contenido = "inicio.php";
      }
      include $contenido;
    ?>
  </div>
26 </body>
</html>

```

Listado 1.2: `index.php`

```

<h2>
Página de inicio
3 </h2>

```

Listado 1.3: `inicio.php`

```
<h1>CABECERA</h1>
```

Listado 1.4: cabecera.php

```
<p>
  <a href="index.php" title="Ir_a_la_página_principal">Inicio </a>
</p>
4 <p>
  <a href="index.php?pag=pagina1" title="Ir_a_la_página_1">Página 1</a>
</p>
  <p>
9 <a href="index.php?pag=pagina2" title="Ir_a_la_página_2">Página 2</a>
</p>
```

Listado 1.5: menu-izq.php

```
1 <h2>
  Página 1
</h2>

<?php
6 //Se crea una tabla
  echo "<table_summary=''_border='1'_width='100%'">";

  for ($i=1; $i<6; $i++){
    /*Se van añadiendo filas mientras que el valor de la variable
11 $i sea menor que el de la variable $conta (6), incrementándose
    el valor de $i en una unidad ($i++) cada vez que se repite el bucle
    */
    echo "<tr><td>Número_". $i. "</td><td>Fila_". $i. "</td></tr>\n";
  }
16 echo "</table>"; //Se finaliza la tabla
?>
```

Listado 1.6: pagina1.php

```
<h2>
2  Página 2
</h2>

Días de la semana

7 <?php
  $semana = array("Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "
    Sábado", "Domingo");

  echo "<ul>\n";
  for ($i=0; $i<count($semana); $i++){
12 /*Se van añadiendo items de menú mientras que el valor de la variable
    $i sea menor que el número de elementos del array $semana,
    incrementándose
    el valor de $i en una unidad ($i++) cada vez que se repite el bucle
    */
    echo "<li>". $semana[ $i ]. "</li>\n";
17 }
  echo "</ul>\n";
?>
```

Listado 1.7: pagina2.php

Como se puede observar, el código de esta página es similar al de una página HTML excepto por las etiquetas `<?php ?>` que aparecen en él²⁶.

Podemos ver que hay definidos tres elementos DIV que es donde van las distintas partes de la página.

Como primera parte del código a destacar, parémonos en los comentarios²⁷. Recordemos que se pueden insertar comentarios de (al menos) dos formas:

```
// Esto es un comentario de una sola línea

/*Comentario de varias líneas,
varias líneas,
varias líneas */
```

En `index.php` podemos observar el siguiente código PHP entre el código HTML:

```
<?php
$contenido;
if (isset($_GET['pag'])){ $contenido = $_GET['pag'].".php"; }
else { $contenido = "inicio.php"; }
include $contenido;
?>
```

- Si analizamos esta parte del código, vemos que en primer lugar se ha definido la variable `$contenido`, que almacena el valor de la página a incluir en este elemento de la web.
- Después utilizamos la estructura condicional, `if else`, para determinar el valor de la variable `$contenido`. Si se cumple una condición, la variable toma un valor. Si no se cumple (`else`) el valor que toma es `inicio.php`.

En nuestro ejemplo la condición a cumplir es que esté definida la variable `$_GET['pag']`; para ello utilizamos la función `isset()`.

`isset($variable)` comprueba si la variable se ha definido, devuelve la respuesta `true` (verdadero) en este caso o `false` en caso contrario.

Veamos ahora de dónde sale la variable `$_GET['pag']`. Si observamos los enlaces de `menu-izq.php` nos encontramos:

```
<a href="index.php?pag=pagina1" title="Ir a la página 1">Página 1</a>
```

Junto al nombre de la página introducimos una variable (`pag`) con valor = `pagina1`.²⁸ En nuestro caso definimos una variable `$contenido`. La variable `$contenido` puede ser:

- igual al valor de `pag` (que tomará los valores `pagina1` o `pagina2`) junto a la cadena `".php"` en caso de estar definida `$_GET['pag']`,
- o igual a `"inicio.php"` si dicha variable no está definida (como ocurre en el primer enlace del menú)²⁹.

²⁶El código incluido en las etiquetas `<?php ?>` es el código PHP que el servidor web interpreta, terminando todas las instrucciones que aparecen entre estas dos etiquetas, en punto y coma.

²⁷Los comentarios no afectan a la ejecución del programa.

²⁸Se pueden adjuntar datos mediante la URL de esta forma, (`method="GET"`) o bien mediante un formulario (`method="POST"`). Se debe acceder a los datos adjuntos con la URL a través de la matriz `$_GET`.

²⁹Recordemos que se pueden concatenar *strings* y variables, bien escribiéndolas directamente unas a continuación de otras, o mediante el operador de concatenación que es el punto (`.`)

A continuación incluimos el código de la página php mediante la instrucción `include`, el código incluido será distinto en función del valor de `$contenido`: puede ser `pagina1.php`, `pagina2.php` o `inicio.php`.

Si observamos el código de las páginas `pagina1.php` y `pagina2.php` veremos que hemos utilizado la instrucción php: `echo`.

```
echo "<table summary="" border='1' width='100%'>";
echo "<tr><td>Número ".$i."</td><td>Fila ".$i."</td></tr>\n";
echo "</table>";
```

Hemos añadido `\n` al final de cada fila para añadir un salto de línea y facilitar así la lectura del código HTML generado.

En la página `pagina2.php` definimos otro tipo de variables, los `array` o matrices, que como hemos visto, pueden almacenar un conjunto de valores.

```
$semana = array("Lunes", "Martes", "Miércoles", "Jueves", "Viernes",
               "Sábado", "Domingo");
```

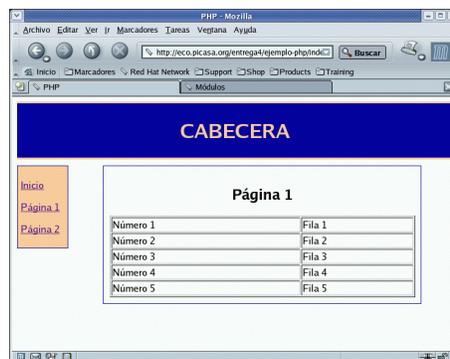
Se puede acceder a cualquiera de los elementos de una matriz³⁰ mediante `$semana[numero_orden]`. El primer elemento de una matriz ocupa el lugar 0 (`$semana[1]="Martes"`).

Tanto en `pagina1.php` como en `pagina2.php` utilizamos otro tipo de estructura PHP: el bucle FOR.

```
for ($i=1; $i<6; $i++){
echo "<tr><td>Número ".$i."</td><td>Fila ".$i."</td></tr>";
}
```

Se define una variable (`$i=1`), una condición (`$i<6`) y un patrón contador (`$i++`).

Las instrucciones que hay dentro del bucle se siguen realizando mientras que el valor de `$i` sea menor que 6, sumando 1 a la variable `$i` cada vez que se realiza el bucle (`$i++`), de esa forma construimos la tabla de la página 1.



Observar que en el fichero `pagina2.php`, para ejecutar el bucle, necesitamos contar primero el número de elementos del array usando la función `count($semana)` (daría como resultado 7)

```
for ($i=0; $i<count($semana); $i++){
}
```

Para recorrer los elementos del array en el que se almacenan los días de la semana, también podemos usar la sentencia `foreach`, en ese caso quedaría:

³⁰Se puede calcular el número de elementos de una matriz mediante la función `count()`.

```
1 <h2>
    Página 2
</h2>

Días de la semana

6 <?php
    $semana = array("Lunes", "Martes", "Miércoles", "Jueves", "
        Viernes", "Sábado", "Domingo");

    echo "<ul>\n";

11    foreach ($semana as $dia){
        echo "<li>".$dia."</li>\n";
    }

16    echo "</ul>\n";
?>
```

Listado 1.8: pagina2bis.php

1.4.2. Representación gráfica de funciones con PHP

Una de las ventajas que presenta el software libre es la cantidad de recursos que tenemos a nuestra disposición en Internet, y esto es más que manifiesto si se trata de usar aplicaciones (clases) creadas con PHP³¹. Vamos a ver en este ejemplo cómo usar una librería orientada a la creación de gráficos matemáticos usando PHP, se trata de `phpplot` <http://www.phpplot.com/>. Nos centraremos en un uso básico de ella.

Para trabajar con ella, bajamos la última versión de la Web del programa y la descomprimos en el raíz de Apache

```
root@eco:/var/www# tar -xzvf phplot-5.0rc2.tar.gz
```

Para facilitar las rutas, vamos a trabajar en el directorio creado (aunque no es obligatorio).

```
$ cd phplot
$ chmod 644 phplot.php
```

y nos garantizamos que los permisos sean los adecuados.

En él tenemos la clase `phpplot`, la documentación del programa y ejemplo de uso.

Además de tener Apache y php, para que podamos trabajar adecuadamente con él necesitamos tener instaladas las librerías `gd`³² y el módulo que permite trabajar con ellas con Apache³³:

```
# apt-get install libgd2-xpm php4-gd
```

³¹Por ejemplo:

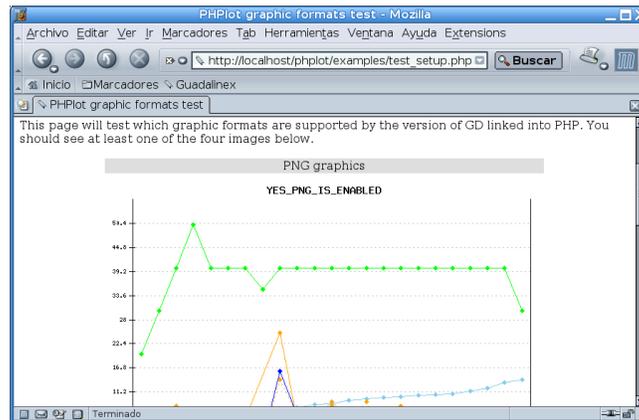
`phppdflib` <http://www.potentialtech.com/pp1.php> para generar pdfs dinámicamente en nuestras páginas Web

`phpmailer` <http://phpmailer.sourceforge.net/> se trata de una clase php para enviar emails

³²Para poder generar los gráficos

³³En Fedora:

```
#apt-get install php-gd
```



Documentación:

- La disponible en la Web del programa y que se instala al descomprimir el programa.
- Un tutorial sobre su uso http://www.programacion.net/php/tutorial/phpsol_phpplot/

En nuestro ejemplo, vamos a crear una Web que nos va a permitir representar parábolas. Para conseguirlo necesitamos dos ficheros:

`formulario.html` lo usaremos para introducir los coeficientes de la parábola, dominio de definición y resolución gráfica.

`parabola.php` para obtener los datos de la parábola y pasar los datos a la librería `phpplot`

Analicemos su contenido

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML_4.01_Transitional//EN" "http://www.
w3.org/TR/html4/loose.dtd">
<html>
3  <head>
    <title></title>
  </head>
  <body>
8    <form action="parabola.php" method="post">
      Coeficientes de la parábola
      <br>
      a= <input type="text" name="a" size="3" maxlength="3">
      b= <input type="text" name="b" size="3" maxlength="3">
      c= <input type="text" name="c" size="3" maxlength="3">
13     <br>
      Dominio de definición:
      <br>
      Inicio= <input type="text" name="x_ini" size="3" maxlength="3">
      Final= <input type="text" name="x_fin" size="3" maxlength="3">
18     <br>
      Número de puntos a evaluar (por defecto 50): <input type="text"
        name="evalua" value="50" size="3" maxlength="3">
      <br>
      <center>
23     <input type="submit" name="Representar">
      </center>
    </form>
  </body>
</html>

```

Listado 1.9: formulario.html



Del formulario hay poco que comentar (salvo que el formato de salida es manifiestamente mejorable :-). Lo utilizamos para introducir las 6 variables que necesitamos:

- 3 para los coeficientes (a, b y c)
- 2 para el dominio de definición (x_ini y x_fin). Deberíamos usar JavaScript para controlar su valor y obligar a que no se puedan pasar valores incorrectos.
- 1 para el número de puntos a evaluar (evalua)

Y el script en php

```
<?php
//incluimos la clase phplot
include_once("phplot.php");
4
//valor de inicio del eje x
$inicio=$_POST["x_ini"];

//valor final del eje x
9 $final=$_POST["x_fin"];

//numero de puntos a evaluar
$evalua=$_POST["evalua"];

14 //Incremento en cada paso.
//Lo obtenemos dividiendo la amplitud del intervalo
//entre el número de puntos a evaluar
//No se controla su valor
$passo=($final-$inicio)/$evalua;

19 //Parámetros de la parábola. No se testea el valor
//de a, si es cero se pintará una recta
$a=$_POST["a"];
$b=$_POST["b"];
24 $c=$_POST["c"];

//Valor inicial del contador
$i=$inicio;

29 //Definimos el array que contiene los valores a representar
//el formato de cada dato es (etiqueta ,x,y)
while ($i<=$final){
    $datos=array("",$i,$a*$i*$i+$b*$i+$c);
    $data[]=$datos;
34    $i=$i+$passo;
}

//Creamos una instancia de la clase PHPlot
$grafico = new PHPlot;

39 //Tipo de datos con los que se trabaja
$grafico->SetDataType("data-data");

//Título del gráfico
44 //es mejorable el formato de salida
$grafico->SetTitle("Gráfica_de_y=( ".$a." )x^2+( ".$b." )x+( ".$c." )");

//Tipo del gráfico, en este caso "lines": une
```



```
49 //los puntos con una línea
   $grafico->SetPlotType("lines");

   //Grosor de la línea
   $grafico->SetLineWidth(3);

54 //Longitud de las rayitas de los ejes en píxel
   $grafico->SetTickLength(1);

   //Incremento del eje y
   //En este ejemplo no lo usamos y dejamos que lo
59 //represente de forma automática
   // $grafico->SetVertTickIncrement(10);

   //Incremento del eje X
   //En este caso numeramos el eje X de uno en uno
64 //a partir del valor inicial
   $grafico->SetHorizTickIncrement(1);

   //Etiquetas de los ejes
   $grafico->SetXLabel("Eje_X");
69 $grafico->SetYLabel("Eje_Y");

   //Pasamos a PHPlot los datos del gráfico
   $grafico->SetDataValues($data);

74 //Abre la gráfica en el navegador Web
   $grafico->DrawGraph();

?>
```

Listado 1.10: parabola.php

Comentemos mejor algunas líneas del fichero:

`include_once("phpplot.php");` con esta línea incluimos la librería `phpplot.php` en nuestro script. Es similar a la ya estudiada `include()`, sólo se diferencia en que si el código ha sido ya incluido, no se volverá a incluir. Notar que tal cual está, obligamos a que se encuentre en el mismo directorio que `parabola.php`.

```
while ($i<=$final){

    $datos=array("", $i, $a*$i*$i+$b*$i+$c);
    $data[]=$datos;
    $i=$i+$paso;
}
```

Con este bucle construimos la matriz de datos que usaremos para representar nuestra parábola. Los datos que usa `phpplot` son un array cuyos datos son a su vez otro array. Cada uno de los datos tiene la forma (etiqueta, valor_eje_x, valor_eje_y). Por eso, lo que hacemos es recorrer el dominio de definición de la función (`$i<=$final`) avanzando en cada iteración el valor de paso (obtenido al dividir la amplitud del intervalo entre el número de puntos a evaluar).

En cada punto a evaluar creamos un array de nombre datos en el que almacenamos los valores

- etiqueta= "" es decir, nada de etiqueta

- `valor_eje_x=$i`
- `valor_eje_y=f($i)=$a*$i*$i+$b*$i+$c` es decir, la imagen de la función para el valor de x (que es i)

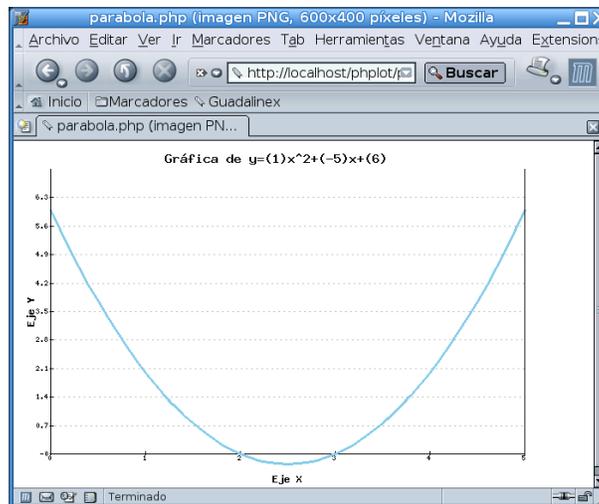
Después, añadimos al array de datos³⁴ (`$data`) el array creado y aumentamos el valor de x (`$i=$i+$paso;`)

`$grafico->SetDataType("data-data");` cuando trabajamos con programación orientada a objetos, la forma de apuntar a un método de una clase se consigue mediante el operador `"->"`. En este caso, lo que hacemos es definir el tipo de gráfico con el que vamos a trabajar en nuestro caso `data-data`³⁵: cada punto consta de una etiqueta, el valor de la x y el valor de y .

`$grafico->SetPlotType("lines");` nos permite definir el tipo de gráfico, permite: `bars`, `lines`, `linepoints`, `area`, `points` y `pie`.

El resto de opciones están comentadas en el propio fichero. Os remitimos a la documentación antes comentada para ampliar sobre su significado o posibilidades de mejora.

Para representar la gráfica de $f(x) = x^2 - 5x + 6$, escribiremos `http://localhost/phplot/formulario.html` en nuestro navegador y tras rellenar el formulario con los datos adecuados³⁶ obtendremos:



³⁴Notar que podíamos usar una sola línea de la forma

```
$data[]=array("$i",$a*$i*$i+$b*$i+$c);
```

³⁵Las opciones son `text-data`, `data-data` y `data-data-error`. Para saber su significado: http://www.phplot.com/doc/user_functions.html

³⁶Como dominio se ha tomado `[0, 5]`

Capítulo 2

MySQL

A menudo la gente se pregunta “¿Cómo es que tengo que obtener una licencia para el servidor de MySQL simplemente porque estoy ejecutando Windows?” Es una pregunta razonable y tiene una respuesta razonable.

...

Todo esto quiere decir que mientras los costes de desarrollo de software en UNIX son prácticamente cero, en WINDOWS pueden ser considerables. A los desarrolladores de MySQL les gusta trabajar en MySQL, pero no tanto como para que quieran pagar por ese privilegio. Los costes de desarrollo en Windows deben recuperarse de algún modo y el coste de la licencia es el método empleado.

Además, los desarrolladores se han encontrado con que necesitan emplear mucho más tiempo para el desarrollo en Windows que en UNIX. (*MySQL*, PAUL DUBOIS, Edit. Prentice Hall)

2.1. Introducción a las Bases de Datos Relacionales

MySQL es un Sistema Gestor de Bases de Datos Relacional (SGBDR¹). Los Sistemas de Gestión de Bases de Datos nos aíslan de la complejidad del almacenamiento de los datos, sin importarnos en dónde se almacenan físicamente.

Hay tres características importantes inherentes a los SGBD: la separación entre los programas de aplicación y los propios datos, el manejo de múltiples vistas por parte de los usuarios y el uso de un catálogo para almacenar el esquema de la base de datos.

En 1975, el comité ANSI-SPARC (*American National Standard Institute - Standards Planning and Requirements Committee*) propuso una arquitectura de tres niveles para los SGBD, que resulta muy útil a la hora de conseguir estas tres características.

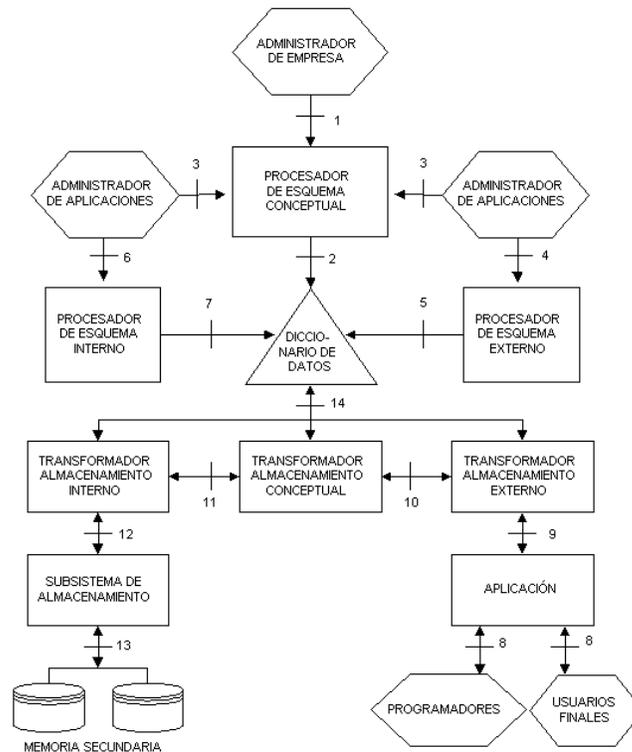
El objetivo de la arquitectura de tres niveles, es el de separar los programas de aplicación de la base de datos física. En esta arquitectura, el esquema de una base de datos se define en tres niveles de abstracción distintos:

1. En el *nivel interno* se describe la estructura física de la base de datos mediante un esquema interno. Este esquema se especifica mediante un modelo físico y describe todos los detalles para el almacenamiento de la base de datos, así como los métodos de acceso.
2. En el *nivel conceptual* se describe la estructura de toda la base de datos para una comunidad de usuarios (todos los de una empresa u organización), mediante un esquema conceptual. Este esquema oculta los detalles de las estructuras de almacenamiento y se concentra en describir entidades, atributos, relaciones, operaciones de los usuarios y restricciones. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar el esquema.

¹ *Relational DataBase Management System* (RDBMS) en la lengua de Chespír ;-).

3. En el *nivel externo* se describen varios esquemas externos o vistas de usuario. Cada esquema externo describe la parte de la base de datos que interesa a un grupo de usuarios determinado y oculta a ese grupo el resto de la base de datos. En este nivel se puede utilizar un modelo conceptual o un modelo lógico para especificar los esquemas.

El siguiente gráfico muestra la arquitectura ANSI/X3/SPARC.



El modelo relacional fue propuesto originariamente por E.F. CODD en 1970 y se basa en la teoría de conjuntos y la lógica de predicados de primer orden. Sin entrar en demasiados formalismos, significa que un sistema relacional almacena y recupera datos que están en forma de relaciones, que más gráficamente las vemos como tablas.

A muchos os parecerá que la tabla es la manera natural de guardar los datos, pero esto no ha sido siempre así, ni dentro de algún tiempo será lo común. Estructuras en forma de árbol, jerárquicas o en red se han usado para representar los datos, y la tendencia actual es a guardar "objetos".

Un Sistema Gestor de Bases de Datos Relacionales se encarga de administrar múltiples Bases de Datos Relacionales. Una Base de Datos Relacional almacena los datos en tablas separadas, pero que se pueden relacionar entre sí.

Las siglas SQL que en parte dan nombre a MySQL, provienen de "*Structured Query Language*"². SQL es un lenguaje de definición y acceso a las Bases de Datos Relacionales y está definido por el estándar ANSI/ISO SQL. Desde 1986 han surgido varias versiones del estándar. Las de uso más común son "SQL-92", referida a la versión surgida en 1992, "SQL:1999" y "SQL:2003" que es la versión actual del estándar.

Cada tabla representa a una entidad del universo que estamos modelando y consiste en una serie de filas (o tuplas) y de columnas (o atributos). Cada fila de una tabla tiene el mismo número de columnas y representa una *instancia* de esa entidad. Cada columna representa un atributo o *propiedad* de la entidad y es de un tipo determinado.

²Lenguaje de Consultas Estructurado, pero el peso de la historia hace que siempre hablemos de Ese-ku-ele.

La siguiente tabla, de nombre "CURSO", posee cuatro columnas denominadas: Número, Nombre, Apellidos y Fecha Nacimiento. Existen tres filas de datos, cada una correspondiente a un alumno de dicho curso, con sus atributos: número en la clase, nombre, apellidos y fecha de nacimiento. Cada atributo será de un tipo; por ejemplo, Nombre y Apellidos serán cadenas de caracteres, Número será un valor entero y Fecha Nacimiento será de un tipo fecha, si existe en nuestro sistema. La primera fila que representamos es una cabecera que contiene los nombres de las columnas y no son datos de la tabla.

Número	Nombre	Apellidos	Fecha Nacimiento
1	Juan	Pérez Gil	07/01/1992
2	Dolores	Fuertes Cabeza	12/03/1991
3	Carmelo	Cotón Rojo	22/10/1991

Una de las grandes ventajas del modelo relacional es que define un método de manipulación de los datos mediante el "álgebra relacional". Todas las manipulaciones posibles sobre las relaciones se obtienen gracias a la combinación de tan sólo cinco operadores: **SELECT**, **PROJECT**, **TIMES**, **UNION** y **MINUS**. Por comodidad, se han definido también tres operadores adicionales que de todos modos se pueden obtener aplicando los cinco fundamentales: **JOIN**, **INTERSECT** y **DIVIDE**. Los operadores relacionales reciben como argumento una relación o un conjunto de relaciones y devuelven una única relación como resultado.

Veamos brevemente estos ocho operadores:

SELECT: devuelve una relación que contiene un subconjunto de las tuplas de la relación a la que se aplica. Los atributos se quedan como estaban. También se denomina **RESTRICT**.

PROJECT: devuelve una relación con un subconjunto de los atributos de la relación a la que se ha aplicado. Las tuplas de la relación resultado se componen de las tuplas de la relación original, de manera que siguen siendo un conjunto en sentido matemático.

TIMES: se aplica a dos relaciones y efectúa el producto cartesiano de las tuplas. Cada tupla de la primera relación está concatenada con cada tupla de la segunda.

JOIN: se concatenan las tuplas de dos relaciones de acuerdo con el valor de un conjunto de sus atributos.

UNION: aplicando este operador a dos relaciones compatibles, se obtiene una que contiene las tuplas de ambas relaciones. Dos relaciones son compatibles si tienen el mismo número de atributos y los atributos correspondientes en las dos relaciones tienen el mismo dominio.

MINUS: aplicado a dos relaciones compatibles, devuelve una tercera que contiene las tuplas que se encuentran sólo en la primera relación.

INTERSECT: aplicado a dos relaciones compatibles restituye una relación que contiene las tuplas que existen en ambas.

DIVIDE: aplicado a dos relaciones que tengan atributos comunes, responde con una tercera que contiene todas las tuplas de la primera relación que se puede hacer que correspondan con todos los valores de la segunda relación.

Por ejemplo, la operación

```
SELECT CURSO donde FechaNacimiento > 05/05/1991
```

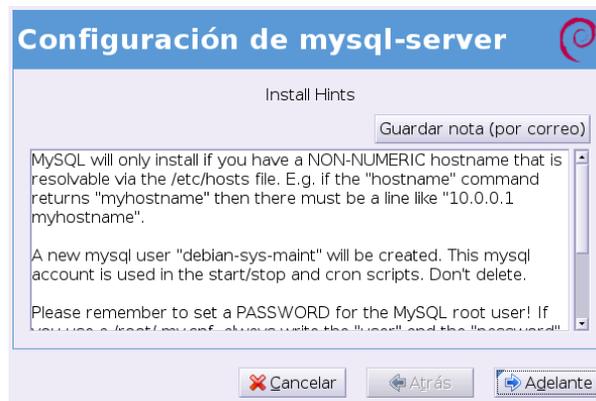
devolverá la relación:

Número	Nombre	Apellidos	Fecha Nacimiento
1	Juan	Pérez Gil	07/01/1992
3	Carmelo	Cotón Rojo	22/10/1991

en la que no aparece la tupla que no cumplía la condición de una fecha de nacimiento mayor del 05/05/1991. Al ataque con MySQL.


```
dbshell libterm-readline-gnu-perl libterm-readline-perl-perl
Paquetes recomendados
perl-doc
Se instalarán los siguientes paquetes NUEVOS:
libdbd-mysql-perl libdbi-perl libnet-daemon-perl libplrpc-perl mysql-
client mysql-doc mysql-server
Se actualizarán los siguientes paquetes:
libmysqlclient12 libperl5.8 mysql-common perl perl-base perl-modules
6 actualizados, 7 se instalarán, 0 para eliminar y 740 no actualizados.
Necesito descargar 15,7MB de archivos.
Se utilizarán 19,8MB de espacio de disco adicional después de desempaq-
uetar.
¿Desea continuar? [S/n]
```

En el proceso de instalación se nos avisará de una serie de cuestiones: información sobre la necesidad de que nuestra máquina tenga un nombre en `/etc/hosts`, que se ha creado una nueva cuenta de usuario⁴ y nos recuerda la necesidad de proteger el servidor asignando una password al root. También nos informa de que los accesos vía red están deshabilitados por defecto y que para activarlos hay que descomentar la opción `skip-networking` del fichero de configuración del servidor de bases de datos (`/etc/mysql/my.cnf`)⁵



Como nuestra intención es trabajar con PHP y MySQL instalemos, además, el módulo adecuado⁶:

```
#apt-get install php4-mysql
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Se instalarán los siguientes paquetes NUEVOS:
php4-mysql
0 actualizados, 1 se instalarán, 0 para eliminar y 730 no actualizados.
Necesito descargar 21,2kB de archivos.
Se utilizarán 119kB de espacio de disco adicional después de desempaq-
uetar.
```

2.2.1. Configuración del servidor

El servidor de bases de datos se configura mediante el fichero⁷:

⁴Este usuario especial (`debian-sys-maint`) tiene privilegios similares a root de mysql.

⁵Podemos reconfigurarlo con `#dpkg-reconfigure mysql-server`

⁶Depende, entre otros de `apache2` y `libapache2-mod-php4`, así que si no están instalados, los instalará ahora.

⁷En Debian, en el directorio `/etc/mysql` hay varios ficheros más, usados en la configuración del servidor (`debian.cnf` y `debian-log-rotate.conf`). Para conocer su significado, mirar la documentación que acompaña al programa.

Debian /etc/mysql/my.cnf

Fedora /etc/my.cnf

Está dividido en secciones y su sintaxis es la habitual⁸:

indica que se trata de un comentario

[sección] indica que se inicia una sección. Los parámetros incluidos dentro de una sección afectan sólo a ésta.

↪ [mysqld] Opciones para el servidor

opcion = valor

↪ port=3306 puerto en el que escucha el servidor

opcion se trata de parámetros booleanos que se establecen al estar presentes en el fichero de configuración.

↪ log-bin si lo escribimos (descomentamos) registraremos las actualizaciones que realicemos en todas las tablas.

set-variable = variable = valor para establecer los valores de las variables

↪ set-variable=write_buffer=2M establecemos en 2MB el bufer de escritura.



En⁹ /usr/share/doc/mysql-server/examples tenemos varios ficheros de configuración de ejemplo adecuados a la memoria dedicada (ojo, que no se trata de la memoria RAM instalada) a MySQL en nuestro ordenador. En la columna memoria hemos puesto los valores indicativos que no se deben interpretar de forma exacta, sino más bien como valores de referencia.

Fichero	Memoria para MySQL
my-huge.cnf	1 a 2 GB
my-large.cnf	512MB
my-medium.cnf	32MB
my-small.cnf	Sistemas con poca memoria RAM

Usando este fichero podremos, por ejemplo, optar por el tipo de tabla predeterminado o que los mensajes de error aparezcan en castellano. En general, para un trabajo normal no es necesario modificar nada en este fichero, salvo conseguir que los mensajes de error se muestren en castellano¹⁰

```
[mysqld]
....
language = /usr/share/mysql/spanish
....
```

⁸Para ampliar y si hemos instalado el paquete mysql-doc:

file:///usr/share/doc/mysql-doc/manual/Using_MySQL_Programs.html#Program_Options

⁹/usr/share/doc/mysql-server-x-x-x/ en Fedora

¹⁰Y, en su caso, permitir accesos vía red. Por defecto, sólo se permite accesos desde el bucle local:

```
bind-address = 127.0.0.1
```

Si deseamos conectar desde otra máquina, habrá que permitirlo haciendo que el servidor escuche en el interfaz de red. Por ejemplo, si nuestra IP local es 192.168.0.1 escribiremos

```
bind-address = 192.168.0.1
```

2.3. Inicio de MySQL

2.3.1. Aseguremos el servidor

Inicialmente, cualquiera puede conectarse al servidor MySQL y crear o modificar las bases de datos. Así que lo primero que deberíamos hacer es asignar al root una contraseña de acceso. Podemos conseguirlo de varias formas, pero para ejemplificar la forma de conectar al servidor lo haremos con:

```
root@guada04:~# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5 to server version: 4.0.24_Debian-5-log
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

y nos aparecerá el prompt de introducción de comandos. Si no deseamos hacer nada y deseamos salir, podemos escribir¹¹

```
mysql> quit
```

Pero todavía no vamos a salir. Veamos antes un par de ejemplos de cómo trabajar con él. En primer lugar listaremos las dos bases de datos que ya hay creadas, se consigue con el comando

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| mysql    |
| test     |
+-----+
2 rows in set (0.00 sec)
```

si nos fijamos un poco en él podemos constatar que:

- En las sentencias, da igual que los escribamos en mayúsculas o no. No es así en los nombres de los campos, tablas, bases de datos, etc.
- Las sentencias se han de terminar en ;
- Se mantiene un historial de comandos al que podemos acceder de la forma habitual, es decir, con los cursores. Pongamos la password del root¹²:

```
mysql> SET PASSWORD=PASSWORD('palabra_acceso');
Query OK, 0 rows affected (0.01 sec)
```

y salgamos del servidor

```
mysql> exit
Bye
```

¹¹MySQL no es casesensitive (sensible a mayúsculas y minúsculas) en cuanto a los comandos, es decir, que podemos escribir QUIT, Quit, ...

¹²Podemos hacer esto desde la línea de comandos, ejecutando:

```
#mysqladmin password palabra_acceso
```

2.3.2. Un poco de comandos

Si ahora intentamos conectar como antes, comprobaremos que no es posible

```
sh-2.05b$ mysql
ERROR 1045: Access denied for user: 'root@localhost' (Using pass-
word: NO)
```

así que, avisemos al servidor MySQL de quién somos (-u root, es decir el usuario root) y que, además, vamos a introducir la contraseña (-p)

```
sh-2.05b$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8 to server version: 4.0.24_Debian-5-log
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

Pero sigamos. Demos un breve repaso a algunos comandos básicos. Creemos una base de datos¹³ (que luego borraremos)

```
mysql> CREATE DATABASE prueba;
Query OK, 1 row affected (0.05 sec)
```

y comprobemos los cambios:

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| mysql    |
| prueba   |
| test     |
+-----+
3 rows in set (0.00 sec)
```

Pero sólo hemos creado la base de datos, así que está vacía. Creemos una tabla (muy simple) que va a contener sólo dos campos: nombre y apellidos, serán de tipo texto de un máximo de 20 caracteres, longitud variable (VARCHAR) y no permitimos que ninguno de los dos quede vacío¹⁴ (not null)

```
mysql> USE prueba;
Database changed
mysql> CREATE TABLE datos(nombre VARCHAR(20) not null, apellidos VAR-
CHAR(30) not null);
Query OK, 0 rows affected (0.06 sec)
```

En primer lugar, usamos la base de datos en la que crearemos la tabla, para después crearla. Comprobemos las tablas creadas con¹⁵:

¹³De forma equivalente, podemos usar:

```
#mysqladmin -u root -p create prueba
```

¹⁴Formalmente hablando, si especificamos que una columna es NOT NULL obligamos a que la columna no pueda contener valores NULL. Pero didácticamente es más sencilla y efectiva la idea de campo vacío o no vacío.

¹⁵Con:

SHOW STATUS se nos muestran las variables de estado del servidor

SHOW VARIABLES lista las variables del servidor

```
mysql> SHOW TABLES;
+-----+
| Tables_in_prueba |
+-----+
| datos             |
+-----+
1 row in set (0.00 sec)
```

y analicemos su estructura con:

```
mysql> DESCRIBE datos;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nombre     | varchar(20)   |      |     |         |       |
| apellidos  | varchar(30)   |      |     |         |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Es el momento de introducir un par de datos

```
mysql> INSERT INTO datos VALUES ("Pepe","Pinto Gorgorito"),("Pilar","Pan y Agua");
Query OK, 2 rows affected (0.00 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

y mostrarlos por pantalla:

```
mysql> SELECT * FROM datos;
+-----+-----+
| nombre | apellidos      |
+-----+-----+
| Pepe   | Pinto Gorgorito |
| Pilar  | Pan y Agua      |
+-----+-----+
2 rows in set (0.02 sec)
```

Pero deseamos que el usuario Thales pueda manejar esta base de datos (tal cual está, sólo el root tiene acceso a ella). Así que lo autorizamos con:

```
mysql> GRANT ALL ON prueba.* TO thales@localhost IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec)
```

➔ Para practicar

1. Comprobar que ahora el usuario al que hayamos dado permiso con la línea anterior, puede acceder a esta base de datos. Habrá que conectarse al servidor con:

```
$mysql -u usuario -p
```

e introducir la contraseña "password"

2. Si hemos permitido accesos vía red y deseamos que nuestro usuario Thales pueda conectar de forma remota desde cualquier IP (eso significa el%) escribiremos

```
mysql> GRANT ALL ON prueba.* TO thales@%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec)
```

Si nuestro servidor de bases de datos está a la escucha en la IP 192.168.0.1, para conectar, usaremos el comando

```
$mysql -u thales -h 192.168.0.1 -p
```



Usando GRANT¹⁶ y REVOKE podemos conceder y retirar, respectivamente, derechos a los usuarios¹⁷ de cuatro maneras distintas: a nivel global, de base de datos, de tablas o de columnas.

Los objetos a los que se puede conceder cada privilegio son:

Cuadro 2.1: Privilegios para usuarios

Privilegios	Se aplica a	Permite al usuario
SELECT	tablas y columnas	seleccionar filas (registros)
INSERT	tablas y columnas	insertar filas
UPDATE	tablas y columnas	modificar valores de filas
DELETE	tablas	eliminar filas
INDEX	tablas	crear y eliminar índices
ALTER	tablas	modificar su estructura
CREATE	bases y tablas	crear bases de datos o tablas
DROP	bases y tablas	eliminar bases de datos o tablas

La sintaxis genérica es¹⁸:

```
GRANT privilegios [columnas] ON elementos TO usuario
[IDENTIFIED BY 'contraseña'] [WITH GRANT OPTION]
```

donde

privilegios son los que aparecen en la primera columna de la tabla 2.1.

columnas es opcional, se trata de una lista de columnas separadas por comas a las que aplicar los privilegios.

elemento se trata de la(s) base(s) de datos o tabla(s) a las que aplicar los privilegios. Es posible usar comodines:

. todas las bases y tablas

prueba.* todas las tablas de la base de datos *prueba*

prueba.datos a la tabla *datos* de la base de datos *prueba*

usuario [IDENTIFIED BY 'contraseña'] usuario al que se conceden los privilegios y en su caso contraseña de acceso

WITH GRANT OPTION si se pone, permite que el usuario delegue sus privilegios en otros usuarios.

Sin que nuestro sufrido usuario se entere de que podía conectar, nos arrepentimos y le revocamos los privilegios concedidos con:

```
mysql> REVOKE ALL ON * FROM thales@localhost;
```

Listemos ahora sólo los apellidos con:

¹⁶En la sección 3.1.2 en la página 76, en la explicación sobre la instalación de Moodle, se ejemplifica la forma de trabajar con esta sentencia en modo gráfico usando phpMyAdmin.

¹⁷No vamos a analizar ni los privilegios para administradores, ni privilegios especiales.

¹⁸Como es usual, las opciones incluidas entre corchetes son opcionales

```
mysql> SELECT apellidos FROM datos;
+-----+
| apellidos      |
+-----+
| Pinto Gorgorito |
| Pan y Agua     |
+-----+
2 rows in set (0.00 sec)
```

mejor los ponemos ordenados alfabéticamente:

```
mysql> SELECT apellidos FROM datos ORDER BY apellidos;
+-----+
| apellidos      |
+-----+
| Pan y Agua     |
| Pinto Gorgorito |
+-----+
2 rows in set (0.00 sec)
```



El comando `SELECT` es quizás el más utilizado cuando trabajamos con MySQL. Sirve para recuperar las filas de una base de datos que cumplan con los criterios especificados. Su sintaxis es “potente y compleja”, permite usar operadores de comparación y recuperar datos combinados de varias tablas. Más adelante, cuando estudiemos el programa phpMyAdmin (2.4) y la integración de PHP y MySQL ampliaremos sobre su uso.

Tras esta pequeña introducción dejemos las cosas (casi) como estaban, así que antes de seguir borremos la base datos¹⁹:

```
mysql> DROP DATABASE prueba;
```

El número de sentencias y posibilidades es muy grande, así que pensando en gente “desmemoriada” como nosotros, mejor si usamos: PHPMyAdmin

2.4. PHPMyAdmin

Se trata de un conjunto de scripts de php que permiten gestionar bases de datos MySQL usando un navegador web. Usándola, y sin conocer las sentencias de MySQL, es posible (entre otras cosas) gestionar los permisos de acceso de los distintos usuarios a las bases de datos o tablas, así como consultar, crear, borrar, modificar bases de datos o tablas.

La web principal es http://www.phpmyadmin.net/home_page/ y la última versión estable la podemos bajar de http://www.phpmyadmin.net/home_page/downloads.php

2.4.1. Instalación



Si usamos Debian, podemos instalarlo con:

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

¹⁹Con este comando podemos borrar, a su vez, una tabla.

El que para Fedora no exista el paquete en los repositorios oficiales, añadido a la facilidad de instalación si usamos el paquete comprimido, nos ha hecho optar por el método general común a ambas distribuciones.

Una vez descargado, lo instalamos con²⁰:

```
# cp phpMyAdmin-2.6.1-pl3.tar.bz2 /var/www/
# cd /var/www/
# tar -xjvf phpMyAdmin-2.6.1-pl3.tar.bz2
```

Una vez desempaquetado, os recomendamos que le cambiéis el nombre al directorio o que, usando un enlace simbólico, facilitéis el acceso a él desde el navegador web tecleando menos

```
#ln -s /var/www/html/phpMyAdmin-2.6.1-pl3 /var/www/html/phpMyAdmin
```

En el directorio creado, tenemos dos ficheros que nos explican la forma de finalizar la instalación, se trata de `Documentation.txt` o `Documentation.html`. En resumen, los pasos a seguir son:

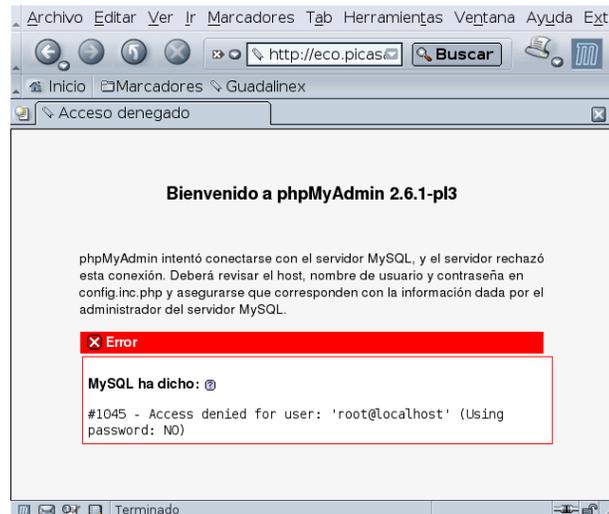
1. Editar el fichero `config.inc.php` y adecuarlo a nuestra máquina, los parámetros importantes a cambiar son²¹:

```
$cfg['PmaAbsoluteUri'] = 'http://localhost/phpMyAdmin/';
```

aquí escribiremos la ruta completa para acceder a phpMyAdmin, por ejemplo

`http://localhost/phpMyAdmin` o `http://www.midominio.org/phpMyAdmin`

Para acceder al programa, sólo es necesario ejecutar²² `http://localhost/phpMyAdmin`



Si hemos puesto contraseña para acceder al motor de base de datos, nos aparecerá el error y tendremos que configurar adecuadamente las variables:

²⁰En Fedora hay que adecuar lo que sigue al `DocumentRoot` de Apache: `/var/www/html`

²¹Si no tenemos un nombre de máquina completamente cualificado, podemos optar por escribir `localhost`.

²²

- Si no hemos puesto contraseña de acceso al gestor de base de datos, se nos avisará además de que: “Su archivo de configuración contiene parámetros (root sin contraseña) que corresponden a la cuenta privilegiada predeterminada de MySQL. Su servidor de MySQL está usando estos valores, que constituyen una vulnerabilidad. Se le recomienda corregir esta brecha de seguridad.”
- Si nos aparece en inglés sólo hemos de optar por el idioma adecuado.

```
$cfgServers[$i]['user']= 'root';
$config['Servers'][$i]['password'] = 'contraseña';
```

con ellas, definimos el usuario y contraseña que usará el programa para iniciar la conexión con el servidor MySQL.



2. Crear un archivo `.htaccess` en el directorio `/var/www/phpMyAdmin-2.6.1-p13` para que sólo nosotros podamos gestionar nuestras bases de datos (véase la entrega anterior para saber qué pasos hay que seguir). Por ejemplo, para restringir el acceso al usuario²³ `thales`

```
# cat .htaccess
AuthType Basic
AuthName "phpMyAdmin"
AuthUserFile /var/www/passwd/.htpasswd
AuthGroupFile /dev/null
require user thales

#htpasswd /var/www/passwd/.htpasswd thales
```



Actualizar: Para actualizar a una versión más moderna, sólo hay que descomprimir la nueva versión en el lugar adecuado (raíz de Apache) y copiar el fichero viejo `config.php.inc` a la nueva situación.

2.4.2. ↪ Base de datos *cursolinux*

Vamos a ejemplificar el trabajo con una base de datos de nombre *cursolinux* en la que vamos a controlar las faltas de nuestros alumnos. El proceso lo vamos a comentar de dos formas:

- Con phpMyAdmin
- Usando comandos de MySQL: los hemos puesto a continuación y en recuadros de texto²⁴.

Mediante este ejemplo, introduciremos los conceptos básicos para trabajar con bases de datos.

La base de datos de ejemplo va a constar inicialmente de dos tablas.

²³En Fedora, `htpasswd` y ajustar el `DocumentRoot`

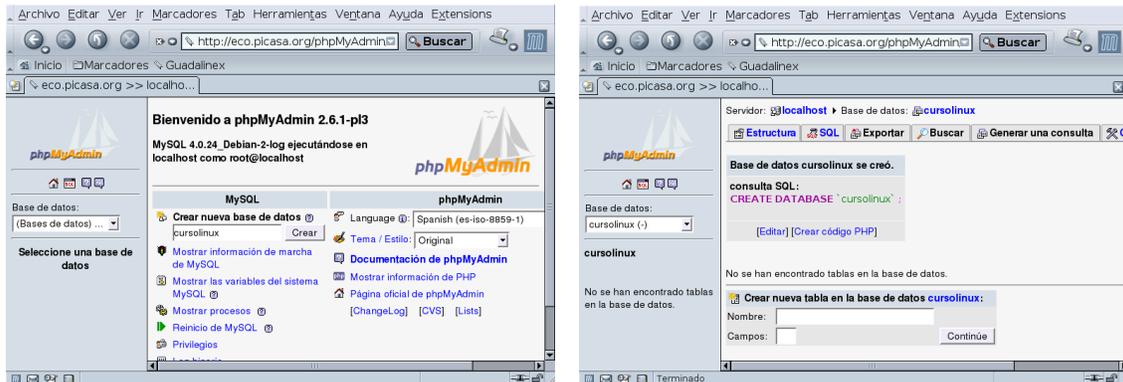
²⁴Si bien aparecen en la capturas gráficas, hemos optado “por sacarlos” de ellas para facilitar su lectura y no tener que agrandar en exceso los gráficos.

- La tabla `alumnos`, que constará de los campos: `codigo`, `nombre`, `apellido1`, `apellido2`, `curso` y `dni`

El único campo extraño es `codigo`, va a ser la clave principal, la generará el propio MySQL y nos va a permitir (después) establecer una relación uno varios (un alumno puede tener varias faltas) entre ambas bases de datos.

- La tabla `faltas`, de estructura: `codigo`, `fecha`

Lo primero será crear la base de datos. Iniciamos la aplicación, escribimos el nombre de nuestra base de datos y pulsamos sobre **[Crear]**



De esta forma, acabamos de crear un directorio en `/var/lib/mysql` de nombre `cursolinux`. El código equivalente en modo comando sería:

```
# mysql -u root -p
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 18 to server version: 4.0.18-log

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

mysql> CREATE DATABASE cursolinux;
```

Es el momento de añadir las tablas. En primer lugar introducimos el nombre de la tabla y número de campos:



para, a continuación, establecer el nombre y tipo de cada uno de ellos:

Servidor: localhost Base de datos: cursolinux Tabla: alumnos

Campo	Tipo	Longitud/Valores*	Atributos	Nulo	Predeterminado**	Extra
codigo	TINYINT		UNSIGNED	not null		auto_increment
nombre	VARCHAR	20		not null		
apellido1	VARCHAR	15		not null		
apellido2	VARCHAR	15		null		
curso	ENUM	'1','2','3','4'		not null	1	
dni	CHAR	8		not null		

Comentarios de la tabla:

Tipo de tabla: Predeterminado

Añada campo(s)

* Si el tipo de campo es "enum" o "set", por favor ingrese los valores usando este formato: 'a','b','c'...
 Si alguna vez necesita poner una barra invertida("\") o una comilla simple ("") entre esos valores, siempre ponga una barra invertida. (Por ejemplo "\xyz" o 'a\b').

** Para valores predeterminados, por favor ingrese solamente un valor, sin "escaping backslash" ni comillas, usando este formato: a

Los valores posibles son:

Campo se trata del nombre del campo en el que vamos a almacenar los datos.

Tipo básicamente disponemos de tres tipos de datos en MySQL: numéricos, de cadena y fechas. Seleccionar el tipo y tamaño que mejor se adecúen a nuestra base de datos no es tarea sencilla, aunque para comenzar podemos permitirnos ciertos lujos que seguro que en la “facultad” serían gravemente penalizados. Las tablas 2.2 en la página siguiente, 2.4 en la página 53, 2.6 en la página 54 y 2.7 en la página 54, resumen los tipos soportados²⁵.

Longitud/Valores aquí introduciremos los tamaños máximos para nuestros campos o, en el caso de enumeraciones o conjuntos, los posibles valores a tomar.

↔ Por ejemplo, el campo `codigo` lo dejamos con el valor por defecto (de 0 a 255). El campo `nombre` tendrá una anchura máxima de 20 caracteres que si no rellenamos en su totalidad, no ocuparán espacio²⁶ (tipo `VARCHAR`). Sin embargo, el campo `dni` está limitado a 8 pero de tipo `CHAR`: aunque un `dni` tenga 5 dígitos se completará con espacios en blanco hasta llegar a los 8. El campo `curso` es una enumeración de posibles valores 1, 2, 3 y 4.

Atributos se puede dejar vacío o ser²⁷: `BINARY` (si se opta por este atributo, hacemos distinción entre mayúsculas y minúsculas), `UNSIGNED` o `UNSIGNED ZEROFILL`.

Nulo si optamos por `null`, permitimos que ese campo se pueda dejar vacío. Si optamos por `not null` obligamos a que se tenga que rellenar de forma obligatoria.

↔ Observar que salvo la columna en donde se almacenará el segundo apellido (en previsión de inmigrantes que provengan de países en el que sólo usan el primer apellido) el resto de campos hay que rellenarlos de forma obligatoria.

Predeterminado nos permite introducir un valor predeterminado.

²⁵No es una referencia exhaustiva. Para eso, consultar: http://dev.mysql.com/doc/mysql/en/Column_types.html

²⁶Sólo se almacenan los caracteres introducidos más un byte para almacenar la longitud de la cadena. El tipo `CHAR`, pese a ocupar más espacio de almacenamiento es, en general, más eficiente.

²⁷Para entender su significado, véanse las notas que siguen a las tablas con los tipos de datos

Cuadro 2.2: Tipos Numéricos

Tipo	Valor	Rango	Bytes
TINYINT [(M)]	Entero pequeño	-128 a 127 (-2^7 a $2^7 - 1$)	1
SMALLINT [(M)]	Entero pequeño	-32.768 a 32.767	2
MEDIUMINT [(M)]	Entero mediano	-8.388.608 a 8.388.607	3
INT [(M)]	Entero	-2.147.483.648 a 2.147.483.647	4
BIGINT [(M)]	Entero grande	-2^{63} a $2^{63} - 1$	8
FLOAT [(M,D)]	Decimal de precisión simple	3,402823466E+38 a -1,175494351E-38 y de 1,175494351E-38 a 3,402823466E+38	3
DOUBLE [(M, D)]	Decimal de precisión doble	-1,7976931348623157E+308 a -2,225073855072014+E308 y de -2,225073855072014+E308 a 1,7976931348623157E+308	4
DECIMAL [(M [, D])]	Decimal almacenado como cadena	-1,7976931348623157E+308 a -2,225073855072014+E308 y de -2,225073855072014+E308 a 1,7976931348623157E+308	M+2

Notas: Todos los tipos admiten, como parámetros opcionales:

M indica el tamaño máximo mostrado

UNSIGNED en el caso de los tipos enteros, el rango es de 0 a $|\text{valormínimo}| + \text{valormáximo}$ (es decir, $2^{8 \cdot \text{bytes} - 1}$). Por ejemplo el tipo **TINYINT UNSIGNED** tendrá de rango de 0 a 255. Si se trata de un número de coma flotante (**FLOAT** o **DOUBLE**) se mantiene el valor máximo positivo y se impiden los valores negativos. Si el tipo es **DECIMAL**, **M** se usa para indicar el número total de dígitos (sin el signo ni punto decimal) y **D** es el número de decimales (por defecto 0).

ZEROFILL el resultado sería que rellenamos con ceros y no con espacios, hasta completar el valor.

↔ En el ejemplo, y como este año tenemos muchos primeros de ESO, hemos optado por poner un 1. Un caso un poco más real²⁸ puede ser: una base de datos en la que almacenar el domicilio de nuestros alumnos y optar por poner como valor predeterminado del campo “provincia” aquella en la que se sitúe el centro de enseñanza.

Extra si deseamos que el campo se autoincremente cada vez que introduzcamos un registro, optaremos por marcar **auto_increment**. Sólo puede usarse con tipos enteros. Los campos de este tipo se usan en general para las claves principales y sólo puede existir uno por tabla. Este tipo de columnas debe estar indexadas.



Primaria si marcamos esta casilla, la columna se tratará como clave principal de la tabla y se indexará de forma automática. Las entradas en esta columna han de ser únicas.



Índice usar índices es el mejor método para hacer las consultas más rápidas.



Único con **UNIQUE** obligamos a que en esa columna no puedan existir datos repetidos.

↔ El **dni** es un buen ejemplo de ello. Podemos tener varias personas con el mismo nombre y apellidos pero no deberían existir dos con el mismo **dni**

²⁸O al menos eso os deseamos -;)

Cuadro 2.4: Tipos cadena

Tipo	Descripción	Máx. caracteres	Bytes
CHAR (M) [BINARY]	Cadena de caracteres de longitud fija	De 0 a 255	M
VARCHAR (M) [BINARY]	Cadena de caracteres de longitud variable	De 0 a 255	L+1
TINYBLOB	Objeto binario largo pequeño	255 ($2^8 - 1$)	L+1
TINYTEXT	Objeto largo pequeño	255 ($2^8 - 1$)	L+1
BLOB	Objeto binario largo	65.535 ($2^{16} - 1$)	L+2
TEXT	Objeto largo	65.535 ($2^{16} - 1$)	L+2
MEDIUMBLOB	Objeto binario largo mediano	16.777.215 ($2^{24} - 1$)	L+3
MEDIUMTEXT	Objeto largo mediano	16.777.215 ($2^{24} - 1$)	L+3
LOBLOB	Objeto binario largo grande	4.294.967.295 ($2^{32} - 1$)	L+4
LONGTEXT	Objeto largo grande	4.294.967.295 ($2^{32} - 1$)	L+4
ENUM('valor1', 'valor2')	Cadena de caracteres con uno solo de los valores especificados	65.535 valores	1 ó 2
SET('valor1', 'valor2')	Conjunto de caracteres formado por la unión de ninguno, uno o varios de los valores especificados.	64 valores	1,2,3,4 u 8

Nota: Los tipos binarios discriminan entre mayúsculas y minúsculas (CHAR y VARCHAR lo admiten como opción).

Los atributos NULL y NOT NULL se pueden especificar para cualquiera de los tipos (el predeterminado es NULL).

L es la longitud de la cadena y M el ancho fijo.

- - - Para desmarcar las tres últimas casillas.



Texto Completo si optamos por marcar esta casilla, crearemos un índice "especial" de tipo FULLTEXT, con un índice de este tipo (sólo se usa con tablas MyISAM²⁹) para columnas del tipo CHAR, VARCHAR o TEXT. Con bases de datos muy grandes es mejor no usarlos.³⁰

Al pulsar sobre [Grabar]³¹ obtendremos una nueva ventana desde la que podremos modificar los valores de todos los campos introducidos hasta ahora, añadir nuevos campos o borrar algunos de los existentes.

²⁹En realidad, esto no es una limitación seria ya que son las las tablas que usa MySQL desde la versión 3.23. Son las sustitutas de las tablas ISAM (Método de Acceso Secuencial Indexado).

³⁰Para ampliar: http://dev.mysql.com/doc/mysql/en/Fulltext_Search.html

³¹Acabamos de crear tres ficheros en /var/lib/mysql/cursolinux, se trata de

```
# ls -al /var/lib/mysql/cursolinux
total 16
-rw-rw---- 1 mysql mysql 8712 abr 1 17:58 alumnos.frm
-rw-rw---- 1 mysql mysql  0 abr 1 17:58 alumnos.MYD
-rw-rw---- 1 mysql mysql 1024 abr 1 17:58 alumnos.MYI
```

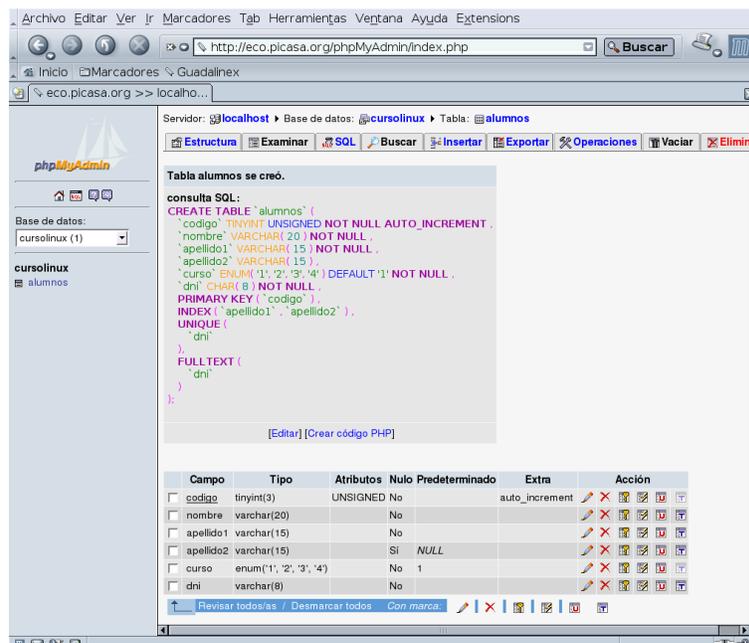
el primero (.frm) contiene la estructura de la tabla, el segundo (.MYD) contendrá los datos y el tercero (.MYI) los índices asociados a esa tabla.

Cuadro 2.6: Tipos fecha y hora

Tipo	Descripción	Rango	Bytes
DATETIME	AAAA-MM-DD HH:MM:SS	1000-01-01 00:00:00 a 9999-12-31 23:59:59	8
DATE	AAAA-MM-DD	1000-01-01 a 9999-12-31	3
TIMESTAMP	AAAAMMDDHHMMSS	19700101000000 a cualquier fecha del 2037	4
TIME	HH:MM:SS	-838:59:59 a 838:59:59	3
YEAR	AAAA	1901 a 2155	1

Cuadro 2.7: Tipos TIMESTAMP

Tipo	Formato pantalla
TIMESTAMP(14)	AAAAMMDDHHMMSS
TIMESTAMP(12)	AAMMDDHHMMSS
TIMESTAMP(10)	AAMMDDHHMM
TIMESTAMP(8)	AAAAMMDD
TIMESTAMP(6)	AAMMDD
TIMESTAMP(4)	AAMM
TIMESTAMP(2)	AA



```
mysql> CREATE TABLE 'alumnos' (
'codigo' TINYINT UNSIGNED NOT NULL AUTO_INCREMENT ,
'nombre' VARCHAR( 20 ) NOT NULL ,
'apellido1' VARCHAR( 15 ) NOT NULL ,
'apellido2' VARCHAR( 15 ) ,
'curso' ENUM( '1', '2', '3', '4' ) DEFAULT '1' NOT NULL ,
'dni' CHAR( 8 ) NOT NULL ,
PRIMARY KEY ( 'codigo' ) ,
INDEX ( 'apellido1' , 'apellido2' ) ,
UNIQUE (
'dni'
) ,
FULLTEXT (
'dni'
)
```

Creemos ahora la segunda tabla. Para ello, en la parte izquierda del navegador, pulsemos sobre la base de datos sobre la que vamos a crear la nueva tabla (**cursolinux**) y en la página mostrada introduciremos el nombre de la tabla (**faltas**) así como el número de campos (2)

y **[Continúe]**. Es el momento de introducir los nombres de los campos y su tipo

Para volver a la página inicial, pulsaremos sobre **Página de Inicio**  (parte superior derecha de la ventana)

```
mysql> CREATE TABLE 'faltas' (
  'codigo' TINYINT NOT NULL ,
  'fecha' TIMESTAMP( 6 ) NOT NULL ,
  INDEX ( 'codigo' )
);
```

En este caso, acabamos de usar un nuevo tipo de columna. Se trata del tipo fecha `timestamp` de longitud 6, eso significa que nuestras fechas serán del tipo AAMDD (véase la tabla 2.7 en la página 54). El resto no presenta problemas, indexamos por el código para optimizar las búsquedas y no permitimos campos vacíos.

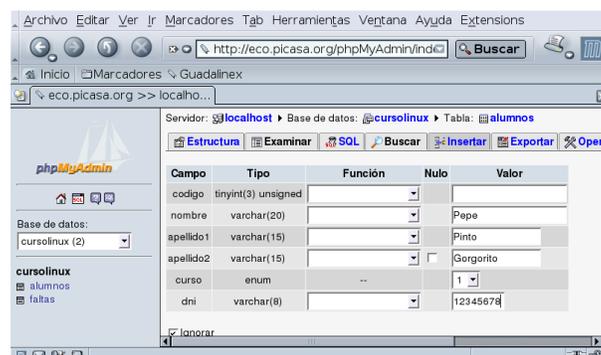
Ya vimos que usando el comando `INSERT` (en la página 45) podíamos hacerlo, pero en general y en espera de usar php y formularios Web, la mejor forma de hacerlo es usando un archivo auxiliar que contenga los datos. En primer lugar, vamos a usar el interfaz Web de phpMyAdmin, pulsemos sobre la tabla `alumnos` para acceder a



Figura 2.1: Tablas y phpMyAdmin

y pulsemos en la pestaña [Insertar] para introducir nuestros alumnos:

- Pepe Pinto Gorgorito de 1º de ESO y dni 12345678
- Pilar Pan y Agua de 2º de ESO y dni 87654321

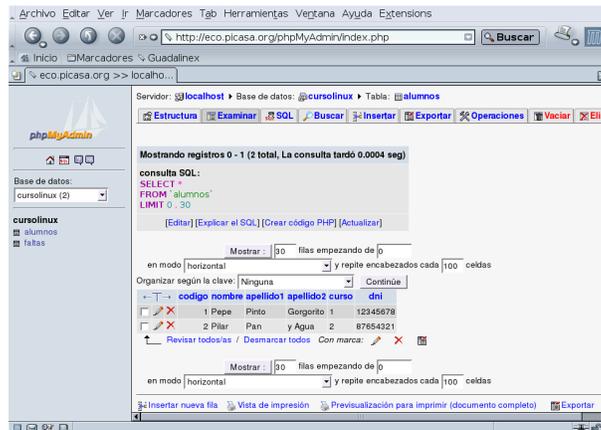


```
mysql> INSERT INTO 'alumnos' ( 'codigo' , 'nombre' ,
  'apellido1' , 'apellido2' , 'curso' , 'dni' )
VALUES (
  "", 'Pepe', 'Pinto', 'Gorgorito', '1', '12345678'
);
```

Notar que:

1. Dejamos el campo código sin introducir
2. Marcamos la opción de **Insertar un nuevo registro**.

Cuando tengamos nuestros alumnos introducidos, pulsando sobre la pestaña  **Examinar**] podremos visualizarlos. Desde aquí podemos ordenar los registros, editarlos, ...



Desde esta página y si introducimos algunos registros más, podemos comprobar la potencia y versatilidad del comando `SELECT`. Las pestañas del gráfico 2.1 en la página anterior no comentadas aún, nos permiten³²:



Estructura nos permite modificar la estructura de la tabla (es la captura gráfica 2.1)



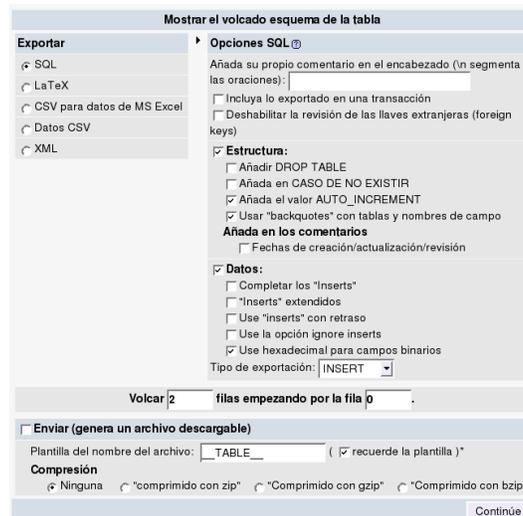
SQL nos permite ejecutar sentencias en modo comando



Buscar para realizar búsquedas en nuestras tablas.



Exportar nos permite exportar los datos y la estructura de nuestra tablas a varios formatos estándar.



³²Para bases de datos, las opciones son: **Estructura**, **SQL**, **Exportar**, **Buscar**, **Generar una consulta** y **Eliminar**. Su funcionalidad es similar a la que comentamos sobre tablas.

 **Operaciones** para realizar distintas operaciones sobre ella: cambiarle el nombre, el orden, copiarla a otra base de datos, tabla, etc

 **Vaciar** con esta opción podemos eliminar los registros de nuestra tabla.

 **Eliminar** para borrar la tabla.

➔ **Insertar registros desde un fichero.** Vamos a realizar algunas funciones más con el par de alumnos que hemos introducido

1. Desde la pestaña  **Exportar** exportemos los datos (no la estructura) a formato SQL enviándolos a un archivo descargable de nombre `datos_alumnos.sql`.

Si lo visualizáis después de exportarlos, su contenido³³ será similar a:

```
# phpMyAdmin SQL Dump
# version 2.6.1-pl3
# http://www.phpmyadmin.net
#
# Servidor: localhost
# Tiempo de generación: 10-04-2005 a las 08:51:42
# Versión del servidor: 3.23.58
# Versión de PHP: 4.3.11
#
# Base de datos : 'cursolinux'
#
#
# Volcar la base de datos para la tabla 'alumnos'
#
INSERT INTO 'alumnos' VALUES (1, 'Pepe', 'Pinto', 'Gorgori-
to', '1', '12345678');
INSERT INTO 'alumnos' VALUES (2, 'Pi-
lar', 'Pan', 'y Agua', '2', '87654321');
```

como siempre, lo importante son las dos últimas líneas, el resto son comentarios. Con este fichero podremos insertar datos en nuestra tabla `alumnos`.

2. Borremos los datos introducidos desde la pestaña  **Vaciar** (la pestaña  **Examinar** se pondrá “negra”)
3. Volvemos al modo comando, introduzcamos los datos anteriores ejecutando el comando

```
# mysql -u root -p cursolinux < datos_alumnos.sql
Enter password:
```

De esta forma, estamos introduciendo datos desde un fichero de texto que podría estar preparado por nosotros previamente. Podemos comprobar que todo ha ido bien recargando phpMyAdmin y tras llegar a la ventana  **Examinar**, ver que tenemos de nuevo a nuestros dos alumnos.

4. Exportemos ahora de nuevo la tabla, pero esta vez dejando marcada la opción de exportar la **Estructura**. El resto igual que antes. El resultado final del fichero de nombre `alumnos.sql` será³⁴

³³Si no se exporta a un fichero externo se muestra el resultado en el navegador.

³⁴Aquí hemos eliminado algunos comentarios.



```

CREATE TABLE 'alumnos' (
  'codigo' tinyint(3) unsigned NOT NULL auto_increment,
  'nombre' varchar(20) NOT NULL default "",
  'apellido1' varchar(15) NOT NULL default "",
  'apellido2' varchar(15) default NULL,
  'curso' enum('1','2','3','4') NOT NULL default '1',
  'dni' varchar(8) NOT NULL default "",
  PRIMARY KEY ('codigo'),
  UNIQUE KEY 'dni' ('dni'),
  KEY 'apellido1' ('apellido1','apellido2'),
  FULLTEXT KEY 'dni_2' ('dni')
) TYPE=MyISAM AUTO_INCREMENT=3 ;
#
# Volcar la base de datos para la tabla 'alumnos'
#
INSERT INTO 'alumnos' VALUES (1, 'Pepe', 'Pinto', 'Gorgorito', '1', '12345678');
INSERT INTO 'alumnos' VALUES (2, 'Pilar', 'Pan', 'y Agua', '2', '87654321');

```

Usando este fichero podremos crear la tabla alumnos e insertar datos, todo de una sola vez. Veamos cómo.

5. Borremos ahora la tabla entera, datos y estructura pulsando sobre  [Eliminar]. Tras recargar la aplicación, comprobaremos que sólo tenemos ya la tabla *faltas*.
6. Creemos la tabla *alumnos* e insertemos los dos datos usando el comando

```

# mysql -u root -p cursolinux < alumnos.sql
Enter password:

```

De nuevo, si recargamos phpMyAdmin comprobaremos que todo ha funcionado como debe.



2.5. PHP y MySQL: páginas web dinámicas.

2.5.1. Más sentencias de PHP

En este apartado, vamos a enumerar algunas de las sentencias de PHP que nos permiten conectar con MySQL. Sólo vamos a enumerar las que vamos a usar en el ejemplo que analizaremos después³⁵ (apartado 1.4.1 en la página 25). Os remitimos a la documentación añadida sobre PHP

³⁵De hecho la mayoría de los ejemplos se han tomado de él.



para ampliar sobre todas ellas (<http://es2.php.net/manual/es/ref.mysql.php>) así como para conocer todas las que hay.

Conexión y desconexión con la base de datos

`mysql_connect(host, usuario, password)` Abre una conexión a un servidor MySQL³⁶. Los tres argumentos son opcionales. Si falta alguno se toman los valores por defecto:

```
host      'localhost'
usuario   que conectará con la base de datos
password  vacía
```

↪ Ejemplo

```
<?php
//variables para almacenar los datos de la conexión
//se puede optar por declararlas o por escribir los datos di-
rectamente
$mysql_server="localhost";
$mysql_login="thales";
$mysql_pass="contraseña";
$c = mysql_connect($mysql_server,$mysql_login,$mysql_pass){
or die ("No he podido conectar");
}
```



Notar que hacemos uso de una función “añadida”, se trata de la función `die()`. Se ejecuta en el caso de que falle la conexión (`or`). En ese caso, finaliza el script y se muestra en el navegador el mensaje especificado.

`mysql_select_db(basededatos, conexión)` selecciona la base de datos sobre la que se va a trabajar, asociada con el identificador de enlace especificado³⁷.

↪ Ejemplo

```
// Nombre de la base de datos que contiene todos los datos
// necesarios para la práctica, usamos una constante
//También se podría escribir el nombre directamente
define("base_de_datos", "cursolinux");
//Seleccionamos la base de datos con la que trabajar
mysql_select_db(base_de_datos,$c) or die("No se puede selec-
cionar la
base de da-
tos");
```

`mysql_close(conexión)` cierra el enlace con MySQL con la conexión especificada³⁸.

↪ Ejemplo: `mysql_close($c);`

³⁶Devuelve un identificador de enlace positivo si tiene éxito, o falso si error.

³⁷Si no se especifica un identificador de enlace, se toma como identificador de enlace el último que se ha abierto.

³⁸Si no se especifica, se asume el último enlace.



Preparación de la consulta

`mysql_query(consulta, conexión)` ejecuta la sentencia de MySQL especificada en el primer parámetro, sobre el identificador de conexión del segundo parámetro³⁹.

↷ Ejemplo:

```
//usar las variables que siguen no es "obligatorio"
//pero no está mal parametrizar el código

//tabla en la que hacer la consulta
$tabla="alumnos";

//campos a mostrar, en este caso todos
$campos="*";

//cadena para la consulta
$consulta="SELECT ".$campos." FROM ".$tabla;

//ejecuta la consulta especificada
$resultado= mysql_query($consulta,$c);
```

Si no usamos variables, escribiríamos la sentencia equivalente⁴⁰

```
//ejecuta la consulta especificada
$resultado= mysql_query(SELECT * FROM alumnos,$c);
```

Recuperación de resultados

`mysql_num_fields(cursor)` devuelve el número de campos de un resultado

↷ Ejemplo:

```
//se almacena en la variable el
//número de campos solicitado
$numero_campos=mysql_num_fields($resultado);
```

`mysql_num_rows(cursor)` devuelve el número de filas de un resultado

↷ Ejemplo:

```
//se almacena en la variable el número
//de filas obtenido en el resultado
$numero_filas=mysql_num_rows($resultado);
```

`mysql_field_name(cursor, índice)` Devuelve el nombre del campo de índice especificado en un resultado.

↷ Ejemplo: para obtener el nombre del segundo campo de una consulta podemos usar

```
mysql_field_name($resultado,2)
```

`mysql_fetch_row(cursor)` devuelve un array enumerado (el primer elemento se enumera con 0) en que se selecciona una fila de datos del resultado, o falso si no quedan más líneas. Con esta función y un bucle es fácil mostrar todos los campos (y registros) de una consulta.

³⁹Si no se especifica un identificador de enlace, se trabaja con el último abierto.

⁴⁰Notar que no es necesario escribir el ; final.



↪ Ejemplo:

```
//creamos una matriz enumerada que contiene los datos
$datos=mysql_fetch_row($resultado);
//número de campos que contiene
$numero_campos=mysql_num_fields($resultado);
//Mostramos los datos
for ($i=0;$i<$numero_campos;$i++){
    echo $datos[$i]." ";
}
}
```

`mysql_fetch_array(cursor, tipo_resultado)` devuelve un array asociativo en que se selecciona una fila de datos del resultado o falso si no quedan más líneas. Con `mysql_fetch_array` y un bucle es fácil mostrar todos los campos (y registros) de una consulta. Se trata de una versión extendida de la anterior ya que “además de guardar los datos en el índice numérico de la matriz, guarda también los datos en los índices asociativos, usando el nombre de campo como clave” [12]. El segundo parámetro es opcional e indica el tipo de array que devuelve⁴¹.

↪ Ejemplo: Con el código siguiente, podemos recorrer varias filas de una consulta y mostrarlas en pantalla

```
# establecemos un bucle que recoge en un array
# cada una de las líneas del resultado de la
# usamos <<mysql_fetch_array>> y para evitar dupli-
# cados
# optamos por el parámetro MYSQL_ASSOC
while ($registro=
mysql_fetch_array($resultado, MYSQL_ASSOC)){

    // insertamos un salto de línea para cada fila
    echo "<br>";

    # establecemos el bucle de lectura del ARRAY
    # con los resultados de cada línea
    foreach($registro as $clave => $valor){
        echo $valor." ";
    }
}
}
```

Notar que la condición del bucle `while` es cierta hasta que lleguemos a la última línea.

`mysql_free_result(cursor)` cierra la conexión establecida con `mysql_connect()`. Sólo puede ser interesante utilizarla si nos preocupa la posibilidad de que usemos demasiada memoria en la ejecución del script PHP, ya que toda la memoria del resultado especificado en `cursor` será automáticamente liberada.

↪ Ejemplo:

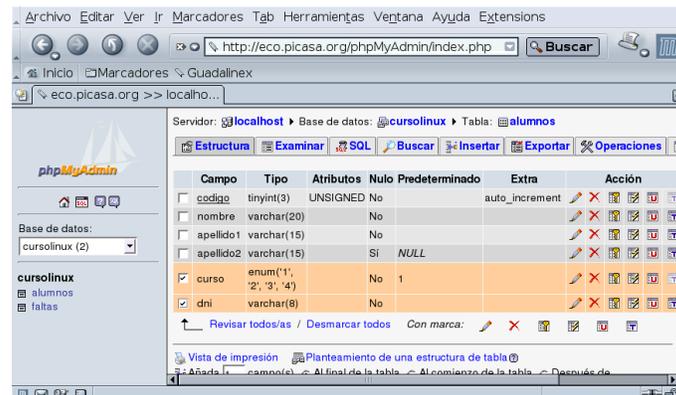
```
//libero los recursos de las consultas
mysql_free_result($resultado);
```

⁴¹ Puede tomar los valores: `MYSQL_NUM`, `MYSQL_ASSOC` y `MYSQL_BOTH`.

2.5.2. Un ejemplo

Vamos a ver, de forma bastante simplificada, cómo podemos usar conjuntamente MySQL y PHP. Para esto vamos a echar mano de la Web creada en el capítulo sobre PHP (apartado 1.4.1 en la página 25) y de la base de datos `cursolinux` que tenemos creada (suponiendo que hemos introducido los datos previamente). En primer lugar vamos a “simplificarla” un poco eliminando algunos campos (`grupo` y `dni`) para centrarnos sólo en las cuestiones más elementales.

Eliminemos pues, los campos. Desde phpMyAdmin, nos situamos sobre estructura y tras marcar las casillas de verificación que hay a la derecha de `grupo` pulsaremos sobre 



```
mysql> ALTER TABLE 'alumnos' DROP 'curso';
Query OK, 2 rows affected (0.01 sec)
Registros: 2 Duplicados: 0 Peligros: 0
mysql> ALTER TABLE 'alumnos' DROP 'dni';
Query OK, 2 rows affected (0.01 sec)
Registros: 2 Duplicados: 0 Peligros: 0
```

Después lo haremos con el campo `dni`.

Pretendemos que en la página principal se nos muestre un listado de los alumnos de un grupo y que, seleccionando el que deseemos, podamos introducir las faltas de ese alumno para la fecha en curso.

Realmente tendríamos que crear primero un formulario que nos permitiera ir introduciendo los datos de la base de datos, pero intentar abarcar todos los aspectos del funcionamiento conjunto de MySQL y PHP es algo que escapa de los objetivos de este curso y que sería materia suficiente para un curso específico.

Nuestra utilidad va a constar casi de los mismos ficheros⁴² que cuando se estudió en el tema anterior:

- `estilo.css` es la misma hoja de estilo.
- Los ficheros `index.php` (archivo principal de la aplicación), `cabecera.php`, `menu-izq.php` e `inicio.php` se han mantenido prácticamente iguales

⁴²Estarán a vuestra disposición en el servidor todos los ficheros, en un archivo comprimido de nombre `mysql-php.tgz`



sólo se han cambiado los textos que se muestran en pantalla para adecuarlos a la nueva situación. Si pulsamos sobre **Insertar** o **Listar** obtendremos una página de contenido similar, pero de funcionalidad final diferente.



- Con:

Insertar se carga el fichero `pagina1.php` y nos permite poner la falta del día en curso a uno de los alumnos.

Listar se carga `pagina2.php` y nos permite visualizar las faltas de asistencia del alumno seleccionado.

El contenido de ambos ficheros sí que cambia, es:

```

<h2>
  Introducir faltas a un alumno
</h2>
<?php
  //variable para la acción a realizar
  $accion="introducir-faltas";

  //listado común a la aplicación
  require "listar-grupo.php";
?>

```

Listado 2.1: pagina1.php

```

<h2>
  Listar las faltas de un alumno
</h2>

```



```
<?php
//variable para la acción a realizar
$accion="listar-faltas";

//listado común a la aplicación
require "listar-grupo.php";
?>
```

Listado 2.2: pagina2.php

Este par de ficheros tienen poco que comentar. Se declara la variable `$accion` que va a permitir el que con un solo fichero (`listar-grupo.php`) realicemos acciones diferentes dependiendo de la página desde la que se cargue (más adelante volveremos sobre ella). Es importante reseñar la ventaja que supone la posibilidad de reutilizar el código en ambos archivos.

Comencemos con el primer fichero "nuevo" y dejemos para después el fichero `listar-grupo.php`. Se trata del fichero que nos va a permitir conectar con la base de datos *cursorlinux* y obtener o modificar los datos de las tablas que la componen. Se tendrá que incluir en los script PHP en los que conectemos con la base de datos.

```
<?php
//Variables para almacenar los datos de la conexión
//se puede optar por declararlos o por escribir los
//datos directamente en los scripts php
4 $mysql_server="localhost";
  $mysql_login="thales"; //este valor hay que adecuarlo a vuestro sistema
  $mysql_pass="contraseña"; //poner aquí la contraseña de la base de datos

9 // Nombre de la base de datos que contiene todos los datos
  // necesarios para la práctica, usamos una constante
  define("base_de_datos", "cursorlinux");

//variable que recoge el identificador de conexión
14 $c;

// Función que realiza la conexión con el gestor de base de datos
// y selecciona la base de datos con la que vamos a trabajar
// Antepone & a la variable para indicar que se pasa
// por referencia
19 function conectar_bd(&$c)
  {
    //las definimos como globales para trabajar con ellas en la función
    global $mysql_server, $mysql_login, $mysql_pass;

24 // Nos conectamos con el gestor MySQL y seleccionamos una BD
    $c = mysql_connect($mysql_server, $mysql_login, $mysql_pass) or die("No se puede conectar");
    mysql_select_db(base_de_datos, $c) or die("No se puede seleccionar la base de datos");
    return $c;
29 }

// Función que nos desconecta del gestor de base de datos
function desconectar_bd(&$c)
34 {
  mysql_close($c);
}
?>
```

Listado 2.3: comun.inc



Todo el código que aparece en este fichero se ha analizado ya en la subsección 2.5.1 en la página 60. Llega el momento de la verdad, analicemos el fichero `listar-grupo.php`

```
<script type="text/javascript">
<!--
4 //código javascript necesario para la ventana emergente
function abrir(pagina) {
    window.open(pagina, 'ventanaFaltas', 'scrollbars=no, resizable=yes, width
        =200,height=280,status=no, location=no, toolbar=no');
    }
// -->
9 </script>

<?php
//datos comunes
require "comun.inc";

14 //establecemos la conexión con el servidor
conectar_bd($c);

//tabla en la que hacer la consulta
19 $tabla="alumnos";

//campos a listar, en este caso todos
$campos="*";

24 //cadena para la consulta
$query="SELECT_" . $campos . "_FROM_" . $tabla;

//ejecuta la consulta especificada
29 $resultado= mysql_query($query,$c);

//número de campos solicitados
$numero_campos=mysql_num_fields($resultado);

//Aviso sobre cómo proceder
34 echo "<h3>Marca_sobre_el_c&oacute;digo</h3>";

// Creamos una cabecera de una tabla (código HTML)
echo "<table_align=center_border=2>";

39 //comienzo en cero para que se muestre el código
for ($i = 0; $i < $numero_campos; $i++){
    echo "<td_<b_<center_&nbsp;";
    echo ucfirst(mysql_field_name($resultado,$i));
    echo "</b_</td_</center_&nbsp;";
44 }
echo "</tr>";

# establecemos un bucle que recoge en un array
# cada una de las líneas del resultado
# usamos mysql_fetch_array y para evitar duplicados
# optamos por el parámetro MYSQL_ASSOC
while ($registro= mysql_fetch_array($resultado, MYSQL_ASSOC)){

54 // insertamos un salto de línea en la tabla HTML
echo "<tr>";
```



```
59 # establecemos el bucle de lectura del ARRAY
# con los resultados de cada línea
# y encerramos cada valor en etiquetas <td></td>
# para que aparezcan en celdas distintas de la tabla
foreach($registro as $clave => $valor){

    if ( $clave == "codigo" ) {
        echo "<td><center><a href=javascript:abrir(' $accion.php?codigo=
64 $valor ');>". $valor . "</a></center></td>";
    }
    else
    {
        echo "<td><center>". $valor . "</center></td>";
    }
69 }
}

echo "</table>";

74 //libero los recursos de las consultas
mysql_free_result( $resultado);

// cerramos la conexión
desconectar_bd( $c);
79 ?>
```

Listado 2.4: listar-grupo.php

La primera parte es código javascript. Nos permite abrir una ventana emergente en la que se nos informa del resultado y se ha puesto sólo con la intención de que no se pierda de vista la página principal.

Aparece una función⁴³ nueva de PHP en la línea:

```
echo ucfirst(mysql_field_name($resultado,$i));
```

`ucfirst('cadena')` pone en mayúsculas el primer carácter la cadena si es un carácter alfabético. Por ejemplo `ucfirst('thales')` daría como resultado "Thales". El bucle contiene un `if`

```
if ( $clave == "codigo" ) {
    echo "<td><center><a href=javascript:abrir(
        '$accion.php?codigo=$valor');>". $valor . "</a></center></td>";
}else{
    echo "<td><center>". $valor . "</center></td>";
}
```

que nos permite diferenciar el primer campo del resto (campo código), de esa forma creamos el enlace sólo con él.

El enlace creado llama a la función javascript (`abrir`) que permite ver los resultados en la ventana emergente. Notar además que en el argumento de la llamada se introduce la variable `$accion`. De esa manera, cuando se ejecuta desde `pagina1.php`, `$accion=introducir-faltas` y sin embargo, si se hace desde `pagina2.php` su valor es `listar-faltas`. La consecuencia es que si bien el código es común, dependiendo de dónde se llame, ejecutaremos el script php adecuado. Es decir, si

`$accion=introducir-faltas` se ejecuta el script `introducir-faltas.php`

⁴³El número de funciones para tratar cadenas de PHP es muy amplio: <http://es2.php.net/manual/es/ref.strings.php>



`$accion=listar-faltas` se ejecuta el script `listar-faltas.php`

Además, en ambos casos le pasamos al script una variable (`$codigo`) cuyo valor coincide con el código del alumno sobre el que pulsemos. El resto se reduce a cuestiones ya estudiadas, o a crear la tabla en que se muestran nuestros alumnos.

El script que se lista a continuación nos permite poner la falta de asistencia (la fecha del día en curso) al alumno sobre cuyo código hayamos pulsado:

```
1 <?php
  //datos comunes
  require "comun.inc";

  //tabla en la que introducir las faltas
6 $tabla="faltas";

  //capturamos la fecha del sistema
  $fecha=date("ymd");

11 //almacenamos la variable código
  $codigo=$_GET['codigo'];

  //Añadimos el nuevo registro
  //Primero establecemos la conexión con el servidor
16 conectar_bd($c);
  $consulta="INSERT_". $tabla . "_("codigo , fecha)_VALUES_('$codigo ',' $fecha ')";
  mysql_query($consulta , $c);

  //comprobamos el resultado de la inserción
  //el código de error CERO significa NO ERROR
21 if (mysql_errno($c)==0){
    echo "<p><br><p><center><h2>Registro_a&ntilde;adido</b></H2></center>";
  } else {
    echo "<p><br><center><h2>Se_ha_producido_un_error<br></h2></center>";
26 }

  # cerramos la conexión
  desconectar_bd($c);
?>

31 <!-- Formulario que permite cerrar la ventana -->
<center>
  <form>
    <font size=2 face="arial">
36     <input type="button" value="Cerrar_ventana" name="B1" onclick="
        window.close()">
    </font>
  </form>
</center>
```

Listado 2.5: introducir-faltas.php

Una nueva función (`date()`) en la línea

```
$fecha=date("ymd");
```

con ella capturamos la fecha del sistema. Además, al pasarle los parámetros "ymd" la almacenamos en la variable `$fecha` de la forma adecuada (AAMDD) para insertarla en el campo `fecha` de la tabla `faltas`. Con



```
$consulta="INSERT ".$tabla." (codigo,fecha) VALUES ('$codi-  
go','$fecha)";  
mysql_query($consulta, $c);
```

creamos la cadena que contiene la sentencia de inserción. Recordar que el campo `codigo` es el que nos permite establecer la relación uno a muchos entre ambas tablas (`alumnos` y `faltas`). Como pasamos el código de alumno en la URL, sólo tenemos que recuperarlo en una variable y, junto a la fecha del sistema, insertamos ambos valores en la base de datos `faltas`.

Por último, otro añadido (que no es imprescindible):

```
//comprobamos el resultado de la inserción  
//el error CERO significa NO ERROR  
if (mysql_errno($c)==0){  
    echo "<p><br><p><br><center><h2>Registro a&ntilde;adido</b></H2></center>";  
}else{  
    echo "<p><br><center><h2>Se ha produci-  
do un error<br></h2></center>";  
}
```

La función `mysql_errno($c)` nos devuelve el código de error para la última función llamada. Si bien aquí no está muy “explícito”, su uso nos permite (usando ese código de error) detectar a qué puede ser debido el que un registro no se haya insertado correctamente.

Llegamos al último archivo, con él listamos todas las faltas del alumno seleccionado. De nuevo tendremos que pasar en la URL el valor del código de ese alumno que recuperaremos en una variable (`$codigo=$_GET['codigo'];`).

```
1 <?php  
    //datos comunes  
    require "comun.inc";  
  
    //establecemos la conexión con el servidor  
6 conectar_bd($c);  
  
    //almacenamos la variable código  
    $codigo=$_GET['codigo'];  
  
11 //tabla en la que hacer la consulta  
    $tabla="alumnos";  
  
    //campos a mostrar  
    $campos="nombre, apellido1 , apellido2";  
  
16 //cadena para la consulta  
    $consulta="SELECT_". $campos. "_FROM_". $tabla. "_WHERE_". $codigo;  
  
    //establecemos el criterio de selección  
21 $resultado= mysql_query($consulta, $c);  
  
    //creamos una matriz enumerada que contiene los datos  
    $datos=mysql_fetch_row($resultado);  
  
26 //tabla en la que hacer la consulta  
    $tabla="faltas";  
  
    //campos a mostrar, en este caso solo el campo fecha  
    $campos="fecha";  
  
31 //cadena para la consulta
```



```
$consulta="SELECT_UNIX_TIMESTAMP(".$campos.")_FROM_" . $tabla . "_WHERE_
    codigo=".$codigo;

//ejecutamos la consulta
36 $resultado= mysql_query($consulta,$c);

//número de faltas del alumno
$numero_filas=mysql_num_rows($resultado);

41 //mostramos el nombre de alumno (en negrita)
echo "<b>";
for ($i=0;$i<3;$i++){
    echo $datos[$i]."_";
}

46 //mostramos el número de faltas
echo "</b>_ha_faltado_" . $numero_filas . "_d&iacute;as ,_la_fechas_son:<br>
    <br>";

# establecemos un bucle que recoge en un array
# cada una de las líneas del resultado de la consulta
# como sólo constan de un campo será el de índice 0
51 while ($registro= mysql_fetch_row($resultado)){
    echo "_".date("d_m_Y",$registro[0]).";";
}

56 //retorno de línea
echo "<p><br>";

//libero los recursos de las consultas
61 mysql_free_result($resultado);

//cerramos la conexión
desconectar_bd($c);
?>

66 <!-- Formulario que permite cerrar la ventana -->
<center>
    <form>
        <font size=2 face="arial">
71         <input type="button" value="Cerrar_ventana" name="B1" onclick="
            window.close()">
        </font>
    </form>
</center>
```

Listado 2.6: listar-faltas.php

Cambia la cadena de consulta respecto a lo que habíamos usado hasta ahora

```
//cadena para la consulta
$consulta="SELECT ".$campos." FROM ".$tabla." WHERE codigo=".$codigo;
```

ya no nos interesan todos los registros, sino sólo aquellos cuyo código coincide con el de nuestro alumno. Por eso añadimos en ambas consultas la cláusula `WHERE codigo=".$codigo`. Además, tenemos que conseguir que la fecha salga en el formato al que estamos habituados (DD MM AA). Para eso hacemos un par de cambios:

- Con `UNIX_TIMESTAMP(fecha)` recuperamos la fecha en formato de tiempo Unix⁴⁴.

⁴⁴Número entero de 32 bits que contiene el número de segundos transcurridos desde la media noche del 1 de enero



- Después, con `date("d m Y", fecha)`, le damos el formato que deseamos (al poner Y en mayúsculas nos devolverá el año con 4 cifras y no con dos, que es como está almacenado)

El resto de sentencias del fichero se han analizado ya.



Capítulo 3

Moodle y PHP-Nuke

Aunque PHP compite con ASP de Microsoft, Cold Fusion de Allaire, JSP de Sun, e incluso un primo de código fuente abierto llamado `mod_perl`, realmente se encuentra por encima de prácticamente todos sus competidores por al menos un año. (*Servidor Apache 2*, MOHAMMED J. KABIR)

3.1. Entorno virtual de aprendizaje: Moodle

La palabra Moodle era al principio un acrónimo de *Modular Object-Oriented Dynamic Learning Environment* (Entorno de Aprendizaje Dinámico Orientado a Objetos y Modular), lo que resulta fundamentalmente útil para programadores y teóricos de la educación. También es un verbo que describe el proceso de deambular perezosamente a través de algo, y hacer las cosas cuando se te ocurre hacerlas, una placentera chapuza que a menudo te lleva a la visión y la creatividad. Las dos acepciones se aplican a la manera en que se desarrolló Moodle y a la manera en que un estudiante o profesor podría aproximarse al estudio o enseñanza de un curso en línea. Todo el que usa Moodle es un Moodler.

Ven y ¡moodlea con nosotros!

<http://moodle.org/doc/>

3.1.1. Introducción.

Las plataformas educativas o sistemas de gestión de aprendizaje son paquetes de software que permiten la educación a distancia. Esta fórmula educativa tiene cada vez mayor aceptación y se usa no sólo en educación a distancia sino también en la educación presencial. Ejemplos cercanos de esto son estos cursos y los centros TIC.

Dentro del software libre hay distintos proyectos donde poder elegir. Nosotros hemos optado por Moodle por su potencia y facilidad de uso, razones que determinaron así mismo su uso en estos cursos.

Para conocer qué software hay disponible para este cometido se puede consultar:

- <http://www.linuxjournal.com/article/7817>
- http://www.elearningworkshops.com/modules.php?name=Web_Links&l_op=MostPopular
- <http://www.uv.es/ticape/docs/sedelce/mem-sedelce.pdf>.

Además de Moodle, una plataforma que merece mención especial es *Ilias*, en la 2ª URL (documento en PDF) aparece justificada su valía y una guía de cómo instalarla. <http://www.gate.upm.es/>

Moodle es una “plataforma educativa” que se desarrolla bajo licencia GPL, se trata de “un paquete de software para la creación de cursos y sitios Web basados en Internet. Es un proyecto en desarrollo diseñado para dar soporte a un marco de educación social constructivista.”



Es sencillo de mantener y actualizar y, salvo el proceso de instalación, no necesita prácticamente de “mantenimiento” por parte del administrador.

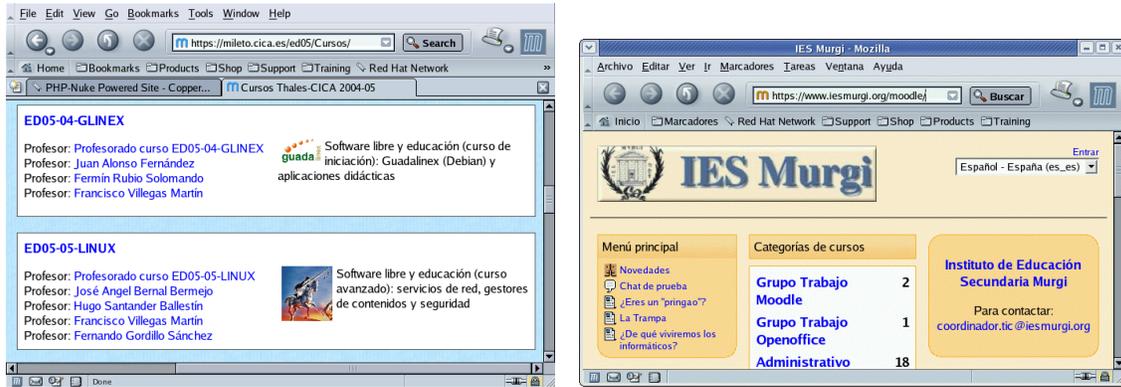


Figura 3.1: Moodle de Mileto e IES Murgi

3.1.2. Instalación



Con objeto de homogeneizar el proceso de instalación en ambas distribuciones antes de iniciar la instalación vamos a adecuar la configuración de Debian para que sea similar a la de Fedora. De esa forma todo lo que sigue será válido para ambas. Para eso, en Guadalinex crearemos un directorio de nombre html dentro del raíz de Apache2

```
#mkdir /var/www/html
```

Y modificaremos el fichero `/etc/apache2/sites-available/default` para que apunte a él, se trata de cambiar las líneas, ajustando las primeras y comentando la última:

```
DocumentRoot /var/www/html
<Directory /var/www/html>
#RedirectMatch ~/$ /apache2-default/
```

Tras guardar los cambios

```
# apache2ctl restart
```



Existe un paquete para Debian de nombre moodle que nos automatiza el proceso de instalación. Pero en general no es mucho más sencillo que el proceso manual. El hecho de que además el método “general” es válido para ambas distribuciones nos ha llevado a optar por él.

La Web principal del programa es <http://moodle.org/>. Desde allí (sección **Downloads**) podemos bajar la última versión estable (por ahora), se trata de

```
moodle-latest-14.tgz
```

Para instalar Moodle en nuestro sistema necesitamos tener en funcionamiento: Apache, php¹ y MySQL, y partiremos de esto.

¹Además es necesario que estén instalados los paquetes (se instalarán con ellos las librerías GD):

php-gd en Fedora

php4-gd en Debian



Tenemos que garantizarnos que la directiva de Apache `AcceptPathInfo` esté en `On`², así que revisemos el fichero de configuración de Apache³ y añadámosla:

```
AcceptPathInfo on
```

después de cambiar la línea tendremos que reiniciar el servidor.

Una vez en nuestra máquina, el proceso de instalación es muy simple y se encuentra bien guiado en las intrucciones de la web de moodle (<http://moodle.org/doc/?file=install.html>). Se resume en:

- Poner el paquete `moodle-latest-14.tgz` en el lugar adecuado, en general será `/var/www/html`, y desempaquetarlo

```
# cp moodle-latest-14.tgz /var/www/html; cd /var/www/html
# tar -xzvf moodle-latest-14.tgz
```

- Asignamos como dueño del directorio al usuario bajo el que se ejecuta el servidor Web⁴

```
# chown www-data moodle
```

- Crear el directorio `moodledata` y ajustarle los permisos de forma adecuada. Es preferible que no sea accesible directamente desde la web. Así que un lugar posible puede ser:

Debian

```
#mkdir /var/moodledata
#chown www-data /var/moodledata
```

Fedora

```
# mkdir /var/www/moodledata
# chown apache /var/www/moodledata
```

- Crear la base de datos⁵ moodle

```
# mysqladmin -u root -p create moodle
Enter password:
```

²Si no está así, no se pueden ver las fotos ni los ficheros que se suban.

³

Debian `/etc/apache2/apache2.conf`

Fedora `/etc/http/conf/httpd.conf`

⁴

- Es necesario para que el script de instalación pueda hacer los cambios por nosotros. Si no lo hacemos, en el proceso de instalación se nos indicará la forma de corregir este pequeño problema.
- En Fedora será:

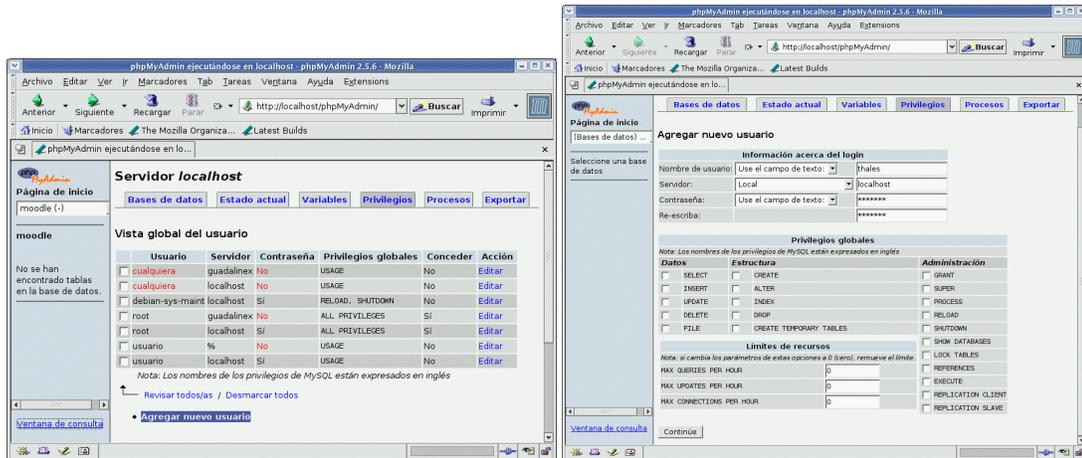
```
#chown apache moodle
```

⁵Si se desea se puede usar phpMyAdmin o ejecutar:

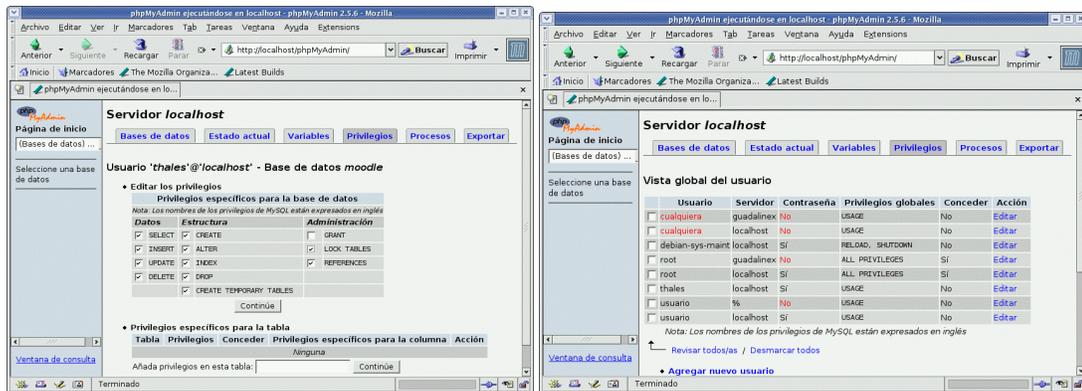
```
# mysqladmin -u root -p
mysql> CREATE DATABASE moodle;
mysql> quit
```



De forma opcional, podemos optar porque un usuario (Thales por ejemplo) se conecte a esa base de datos. Para eso en la ventana principal de phpMyAdmin pulsemos sobre **Privilegios**→**Agregar nuevo usuario**



Introducimos el nombre de usuario, optamos porque las conexiones se realicen sólo desde **Local** y contraseña de acceso, y pulsamos sobre **Continúe**. Se nos abren más posibilidades y optamos por moodle en la lista desplegable **Añada privilegios en esta base de datos**. Después, marcamos todos los privilegios para ella:

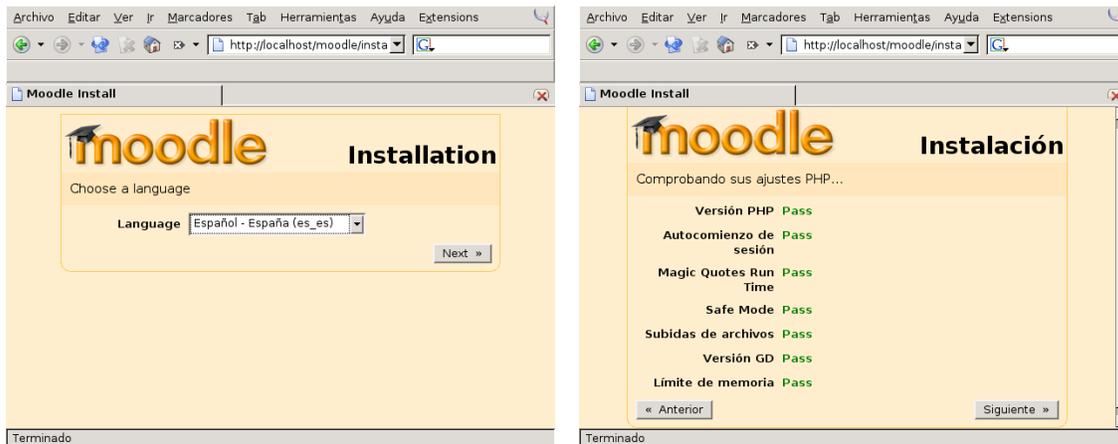


```
# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 55 to server version: 4.0.18-log

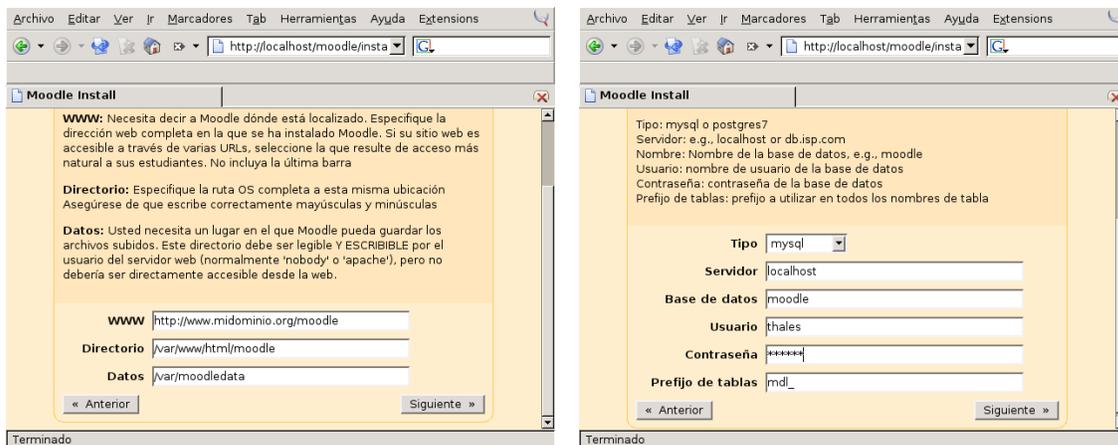
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>GRANT ALL PRIVILEGES ON moodle.* TO thales@localhost
IDENTIFIED BY 'contraseña';
Query OK, 0 rows affected (0.00 sec)
```

- Comienza la “moodlemanía”. Escribamos en un navegador web: <http://localhost/moodle>. Lo primero que debemos hacer será seleccionar el idioma adecuado, a renglón seguido se chequeará nuestro sistema para comprobar si reúne las condiciones necesarias para instalar Moodle



Ahora es el momento de introducir los datos adecuados a nuestro sistema. En la captura hemos optado por <http://www.midominio.org/moodle> pero si no disponemos de un dominio aquí escribiremos <http://localhost/moodle>. Además debemos adecuar los directorios de datos (moodledata) a la distribución con que trabajemos.



Después seleccionaremos el programa servidor de bases de datos y configuraremos de forma adecuada el resto de campos para que el script de instalación tenga permisos para crear las tablas de la base de datos moodle.

Si apache tiene permisos para escribir en el directorio `/var/www/html/moodle`, el script de instalación creará el fichero `config.php` con los datos que hemos introducido hasta ahora, si no es así nos avisará de que tenemos que hacerlo “a mano”⁶. Ya casi, aceptemos los términos de la licencia (GPL claro -:)

⁶Podemos crear ese fichero a partir del fichero `config-dist.php`

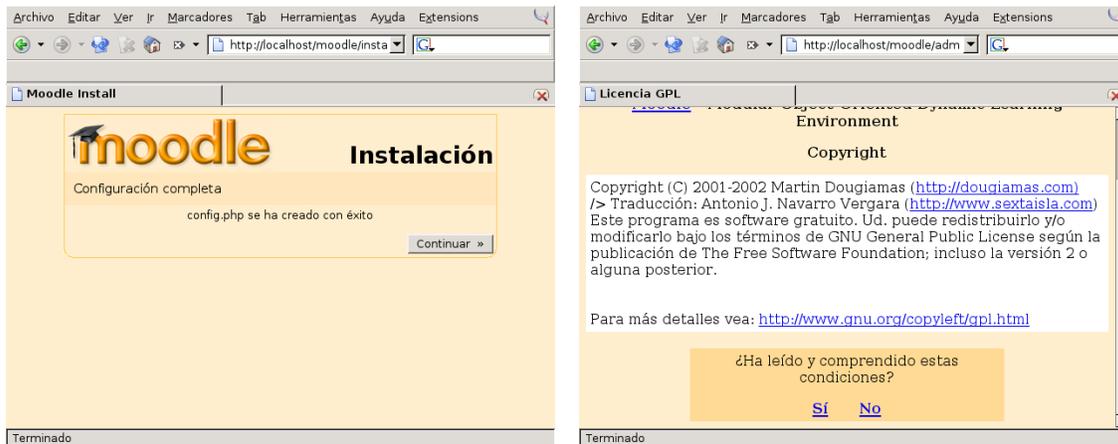
```
#cp config-dist.php config.php
```

Editamos el fichero y, además de ajustar los valores adecuados para poder conectar, hay que ajustar los paths (por ejemplo) y mejor si restringimos un poco los permisos del directorio de datos:

```
$CFG->dbuser      = 'thales';
$CFG->dbpass      = 'password';
$CFG->wwwroot     = 'http://localhost/moodle';
$CFG->dirroot     = '/var/www/html/moodle';
$CFG->dataroot    = '/var/www/moodledata';
$CFG->directorypermissions = 0750;
```

Si tenemos un dominio en vez de la línea anterior escribiremos

```
$CFG->wwwroot = 'http://www.midominio.org/moodle';
```



Después, cuando se han creado las tablas de la base de datos y directorios de datos, accederemos a la primera ventana de configuración propiamente dicha. Salvo que nos guste el inglés, lo mejor es seleccionar el castellano⁷. La ayuda de contexto es muy buena⁸, así que sólo comentaremos las variables susceptibles de ser cambiadas desde el principio:

lang Español-España (es_es)

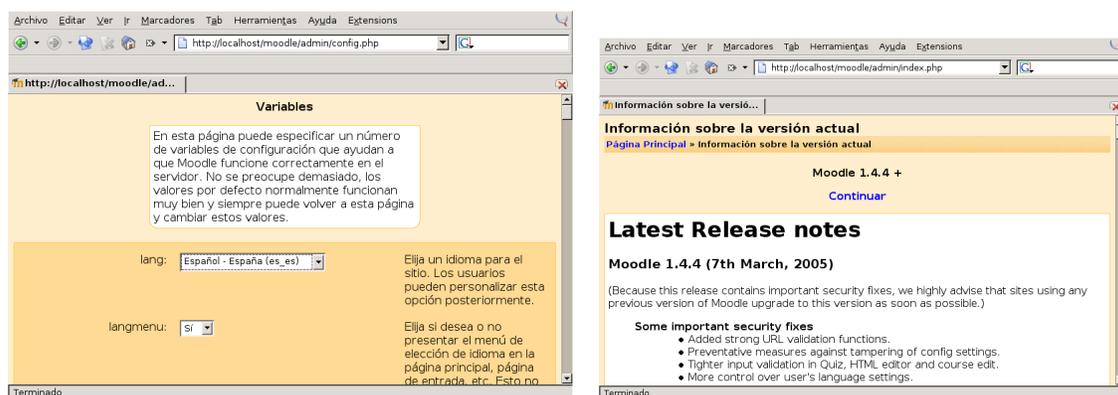
locale optaremos por escribir es_ES

zip en general será /usr/bin/zip

unzip en general será /usr/bin/unzip

country deberíamos elegir el país por defecto para los nuevos usuarios.

loginhttps Deberíamos activar esta opción si hemos montado un servidor seguro, de esa forma el nombre de usuario y contraseña de entrada no viajarán en texto plano. Se iniciará una conexión segura (https) para la página de entrada y, una vez autenticados, se volverá a trabajar con una conexión http normal. **Si no tenemos un servidor seguro en marcha y activamos esta opción no podremos acceder a Moodle.**



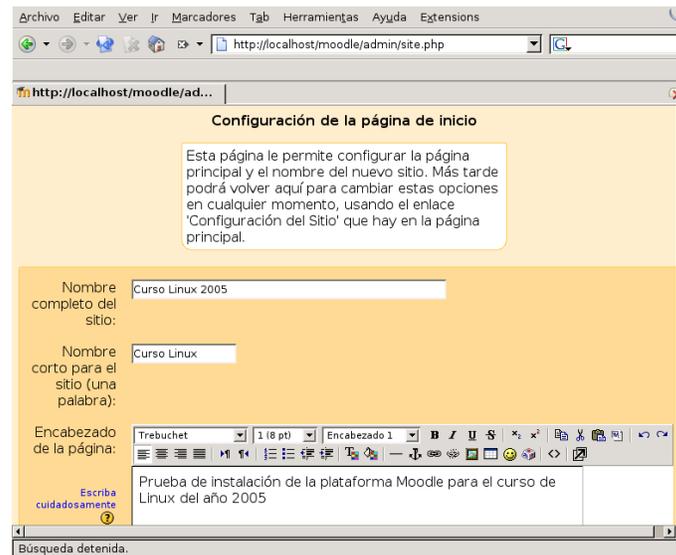
Merece la pena pararse en la que nos informa de los cambios surgidos en esta versión.

Sólo tenemos que ir aceptando en las distintas pantallas que nos van a ir saliendo, con ellas se van creando las tablas necesarias para los distintos módulos y bloques de la aplicación. Configuraremos la página de inicio

⁷Todo lo que cambiemos desde este momento podrá ser modificado después. Así que no hay ningún problema si nos equivocamos en algo ahora.

Aunque la captura aparece en castellano, inicialmente estará en inglés. Hasta que no se opte por el idioma y se guarden los cambios no la veréis así.

⁸Y en castellano, sólo hay que optar por la variable del idioma, guardar y retroceder luego en el navegador.



Con la lista de forma de encabezado podemos optar por la forma que tendrá la página inicial de la aplicación, podemos elegir entre **Mostrar items de noticias** (captura siguiente), **Mostrar un listado de cursos** o **Mostrar un listado de categorias** (véase la figura 3.1 en la página 74 para las dos últimas). Merece la pena pararse un poco en el magnífico editor de html que integra. Tiene de casi todo.

A continuación debemos configurar la cuenta para el administrador principal. Debemos asegurarnos de darle un nombre de usuario y contraseña seguras, completar adecuadamente los campos relativos a la Ciudad y País, y de una dirección de correo electrónico válida (y cómo no, la foto de rigor).

Posteriormente podremos crear más cuentas de administración.



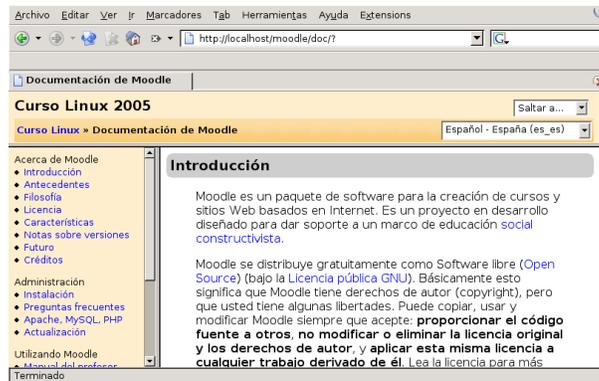
Figura 3.2: Inicio Moodle

Listo, ya tenemos nuestro Moodle en funcionamiento. Si pulsamos sobre **Admin**⁹, además de

⁹A la izquierda, en la zona de abajo del bloque de Administración.



poder modificar todas las variables que definen el sitio, podremos acceder a la magnífica ayuda (en castellano) que acompaña al programa.



Antes de seguir es importante tener en cuenta que algunos módulos de Moodle necesitan revisiones continuas para llevar a cabo tareas. Por ejemplo, Moodle necesita revisar los foros para poder enviar copias de los mensajes a las personas que están suscritas. Lo podemos hacer de forma manual usando

```
http://www.midominio.org/moodle/admin/cron.php
```

o bien seguir la indicaciones de la documentación de Moodle:

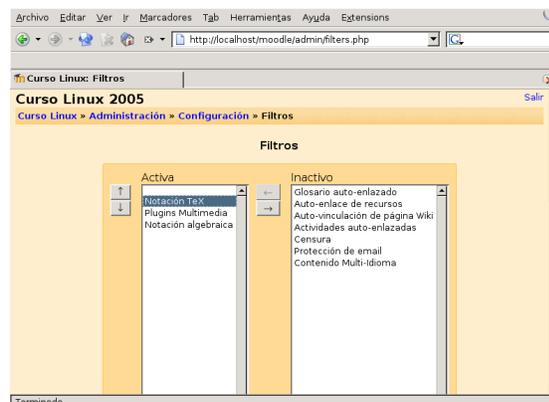
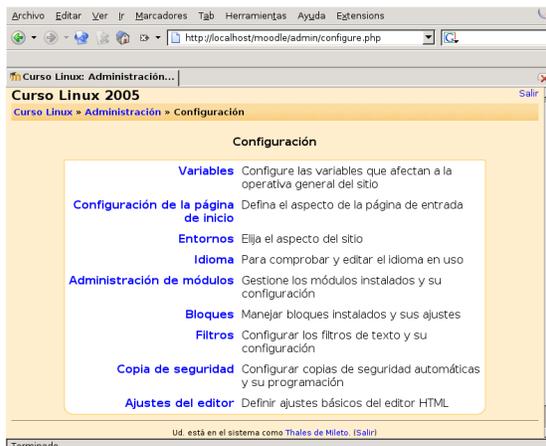
“En un sistema Unix: Use cron. Edite las opciones de cron desde la línea de comandos usando `crontab -e` y añada una línea como la siguiente:

```
*/5 * * * * wget -q -
0 /dev/null http://www.midominio.org/moodle/admin/cron.php
```

Normalmente, el comando `crontab` le enviará al editor `'vi'`. Se entra en `modo de inserción` presionando `”i”`, después teclee la línea de arriba, luego salga del modo de inserción presionando `[ESC]`. Se guardan los cambios y se sale tecleando `”:wq”`, se puede salir también sin guardar usando `”:q!”` (sin las comillas).”

3.1.3. Primeros pasos en la administración.

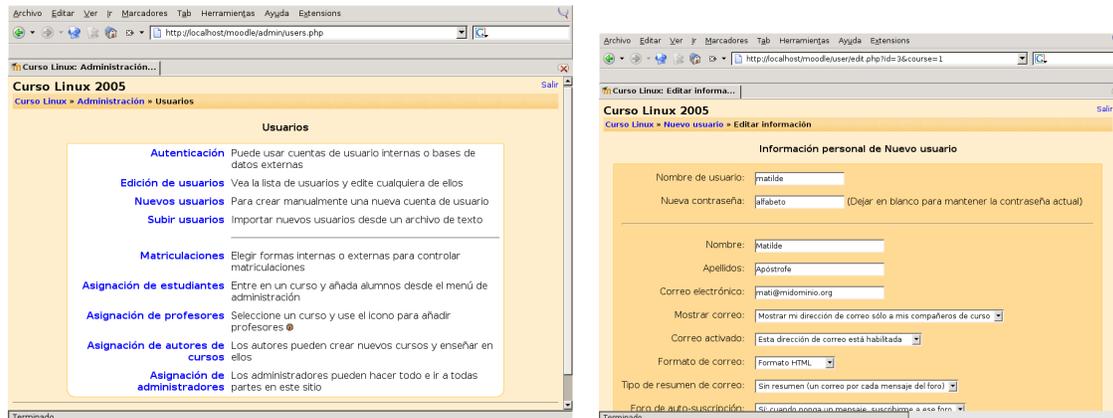
Ya tenemos nuestro entorno instalado. Para que algunas funcionalidades añadidas estén activas debemos hacerlo antes de continuar, así que entramos como administrador del sistema y en la página principal de la aplicación (gráfico 3.2 en la página anterior), pulsamos sobre **Configuración** y después sobre **Filtros**





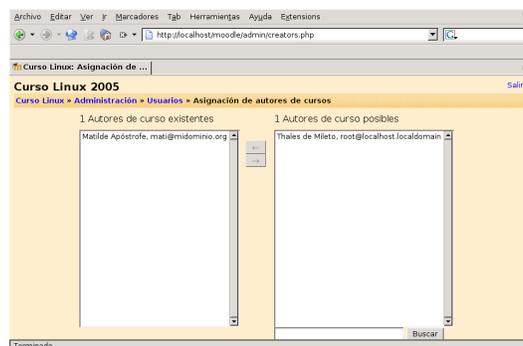
y activamos los que nos interesan (sobre todo Escritura $\text{T}_{\text{E}}\text{X}$).

Pero nuestra intención es la de “descargarnos” un poco de trabajo, ya que la administración de la red es tarea ardua y complicada. Nada mejor que implicar al profesorado para que gestione sus cursos. Así que a la “profe” MATILDE APÓSTROFE, jefa del Dpto de Lengua le vamos a dar de alta¹⁰ para que pueda crear y gestionar sus cursos. En la página de inicio pulsamos sobre **Usuarios**¹¹, obtendremos¹²



Pulsamos sobre **Nuevos Usuarios** e introducimos los datos, por último guardamos los cambios del formulario de introducción de datos y se nos mostrará la ventana de administración de usuarios.

Volvamos a la página de administración de usuarios y pulsemos sobre **Asignación de autores de cursos**¹³, desde ella y pulsando sobre la flecha (←) que hay delante del nombre hagamos que nuestra compañera Matilde pueda crear¹⁴ nuevos cursos y administrar los profesores de esos cursos



Hasta ahora Matilde puede crear cursos pero no asignarlos a una categoría, así que antes de terminar el trabajo como administrador vamos a crearle la categoría Lengua para que sus cursos se sitúen dentro de ella. Pulsamos¹⁵ sobre **Administración** y después sobre **Cursos**

¹⁰También se puede dar de alta ella sola y nosotros después, sólo tenemos que permitirle esta opción.

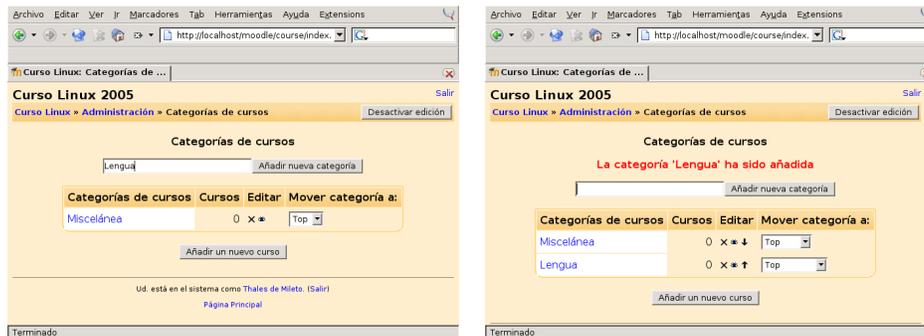
¹¹También se accede desde **Configuración**→**Administración**→**Usuarios**

¹²Como podemos observar el “manual” está incorporado.

¹³De esta forma podrá crear nuevos cursos y enseñar en ellos

¹⁴No pensemos que este proceso es “obligatorio”, el administrador es el root de la plataforma y se puede optar por que sea él solo el que tenga el control de toda ella.

¹⁵También se puede acceder aquí de otras formas, por ejemplo desde la página inicial de la aplicación.



Una vez creada la categoría Lengua podemos descansar un poco. Pero:



Dos notas a tener en cuenta antes de que pasemos el trabajo a Matilde:

- Para que podamos modificar la plataforma es necesario ver el botón **Desactivar edición**, es decir, la edición ha de estar activa (**Activar edición**).
- Es importante que cuando terminemos el trabajo pulsemos sobre **Salir** para finalizar la sesión.

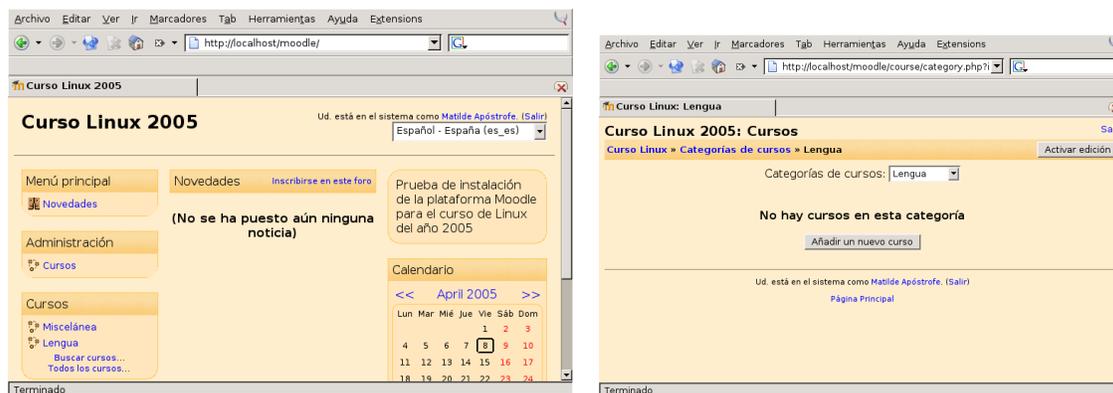
3.1.4. Nuestro primer curso



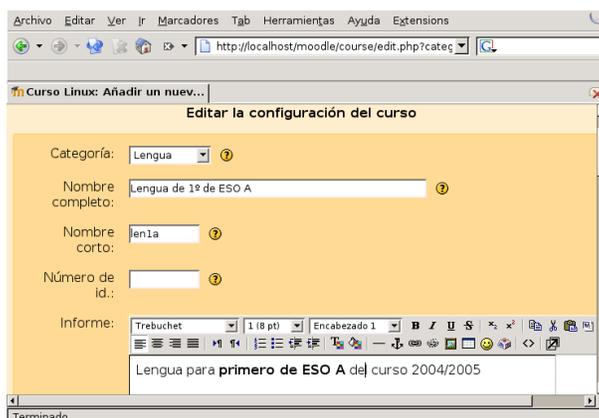
Además de la ayuda de contexto, en Moodle disponemos de una extensa documentación. En castellano podemos acceder a ella desde <http://moodle.org/course/view.php?id=11>. Merece la pena destacar el Manual de usuario y sobre todo el Manual del profesor. Para facilitar el acceso a ellos los hemos puesto también en la web del curso.

Como ya hemos dicho, el uso de Moodle es intuitivo (estáis trabajando con él en los cursos de formación). Pero para iniciarnos con él nada mejor que ver cómo configurar nuestro primer curso.

Le toca el turno a Matilde, va a crear un curso que integrará dentro de la categoría Lengua recién creada. Así que, se autentifica en la plataforma, pulsa sobre **Lengua** y crea su primer curso (Lengua de 1º de ESO A)

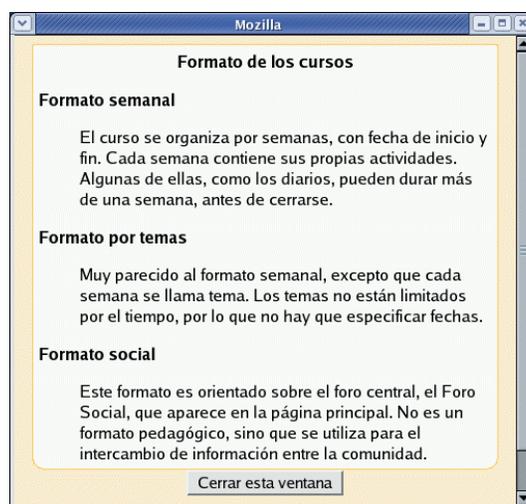


pulsa sobre **Añadir un nuevo Curso**, e introduce los datos adecuados en el formulario que se le presenta:

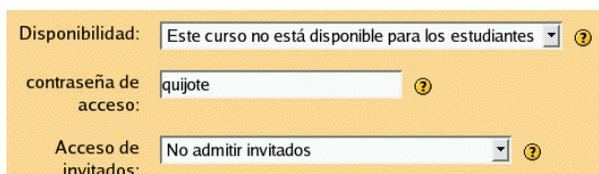


Además de lo reflejado en la captura, comentemos alguno de los campos “más” importantes:

Formato puede ser de tres tipos: semanal, temas y social. Como no sabe muy bien de qué va cada uno pulsa sobre  y se abre la ventana de ayuda:



está claro, el que mejor se ajusta a lo que ella pretende es el formato temas. Opta por él y sigue con el campo **Disponibilidad**, no le interesa que cualquier alumno pueda acceder a su curso, así que selecciona



y escribe la contraseña de acceso.



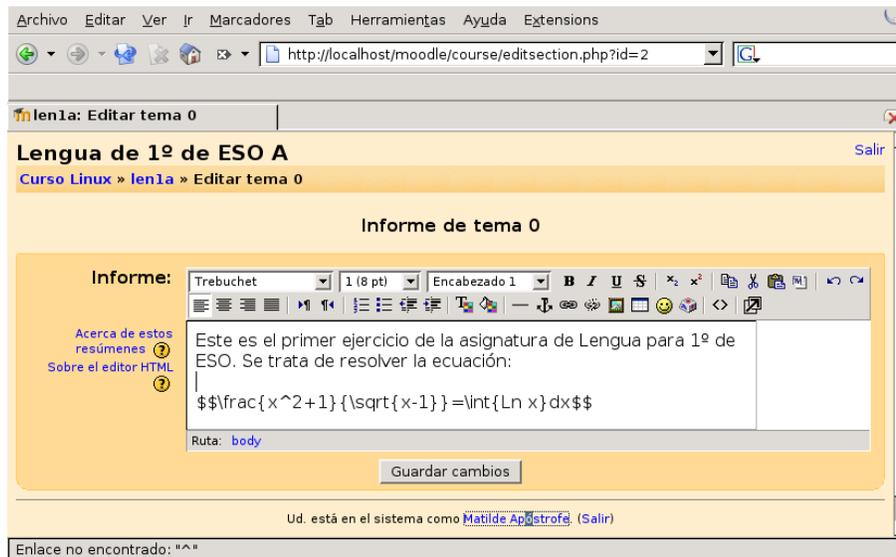
De esa forma, para que un alumno pueda después matricularse en el curso, además de darse de alta en Moodle, tendrá que conocer la palabra de paso. Es un sistema que permite que los alumnos se automatriculen en los cursos pero manteniendo nosotros el control sobre quién lo hace.



Además, opta por no admitir invitados y guarda los cambios. Entra entonces por primera vez en la ventana de su curso¹⁶ recién creado (¿os suena?) y activa la edición (**Activar edición**).



Como además, es aficionada a las matemáticas les va a gastar una broma a sus alumnos, y tras pulsar sobre el icono  que hay encima de **Foro de Noticias**, comienza a escribir su texto de bienvenida



¿Qué dirán sus alumnos y alumnas cuando vean el resultado?

¹⁶Si aparecen errores del tipo

"Fatal error: Allowed memory size of 8388608 bytes exhausted (tried to allocate 46080 bytes) in ...

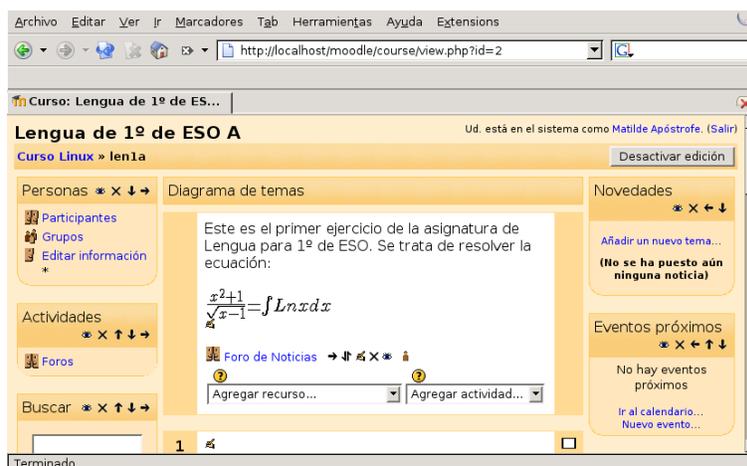
Fatal error: Allowed memory size of 8388608 bytes exhausted (tried to allocate 136 bytes) in Unknown on line 0"

lo podemos solucionar modificando la directiva del fichero de configuración de PHP

```
memory_limit = 8M
```

Un valor de 16 debe ser suficiente para Moodle.

Software libre y educación: redes, gestores de contenidos y seguridad



Para seguir trabajando y conocer qué puede seguir haciendo sólo tiene que echar mano de la **Ayuda** accesible desde esta misma página¹⁷ y sobre todo un documento ya comentado en la página 82: *El manual del profesor*.

3.1.5. Más configuración

En toda esta sección partiremos de la base de que somos el administrador de la plataforma.

Tamaño permitido de subida ficheros

Por defecto, la configuración de php no permite que subamos ficheros de más de 2MB. Si ese tamaño se nos queda pequeño tendremos que modificar nuestro sistema para que sí que lo permita. Supongamos que deseamos establecer un tamaño máximo de subida de ficheros de 16MB, si es así, los parámetros de configuración a cambiar son:

- Del fichero php.ini:

`post_max_size = 16M` con este valor establecemos el tamaño máximo de archivos enviados usando el método POST. Su valor condiciona el valor de la directiva siguiente así que debe ser siempre más grande que el asignado a `upload_max_filesize`.

`upload_max_filesize = 16M` del fichero de configuración de php (`php.ini`) ajustándola al valor deseado.

- Del fichero de configuración de Apache:

`LimitRequestBody 16000000` **No es obligatorio añadirla** ya que por defecto está a cero, es decir, sin límite. Sólo hemos de añadirla si deseamos controlar cuál es el tamaño máximo de fichero a servir por una solicitud HTTP. Este parámetro se puede usar en la configuración del servidor, de los hosts virtuales, a nivel de directorio y en los archivos `.htaccess`. Notar que su valor se da en Bytes¹⁸ y hemos redondeado a la baja. En el caso de que optemos por añadirla, un buen lugar para hacerlo es el fichero:

Debian `/etc/apache2/mods-available/php.conf`

Fedora `/etc/httpd/conf.d/php.conf`

Cuando terminemos de hacer los cambios, reiniciemos el servidor Apache, por ejemplo con:

¹⁷Un documento que os puede resultar interesante es <http://www.iesmurgi.org/~ljoya/moodle/> o en la versión en formato pdf disponible en http://www.iesmurgi.org/modules.php?name=Downloads&d_op=viewdownload&cid=3.

¹⁸El valor máximo es de 2147483647 bytes, es decir, 2GB.



```
#apache2ctl restart
```

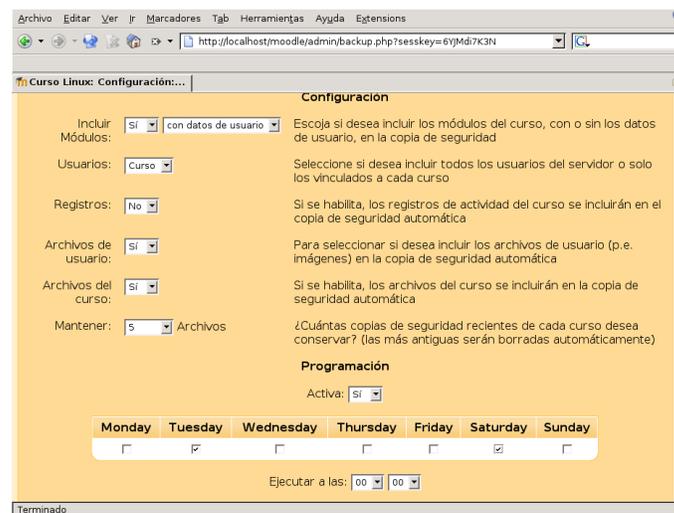
Además, tendremos que permitirlo desde Moodle: **Administración**→**Configuración**→ **Variables**



y después en la configuración de los cursos y módulos o actividades que podemos definir en ellos.

Copias de seguridad

Un aspecto fundamental que no debemos descuidar como administradores de cualquier sistema es el mantener una buena política de copias de seguridad. En este sentido Moodle es una maravilla ya que no tenemos que hacer “casi nada” para conseguirlo. Pulsamos sobre **Configuración**→**Copias de Seguridad**



y programamos la política de copias. Es buena idea (si nuestro disco lo permite) mantener varios ficheros y programar las copias para que se realicen en un par de días y a una hora en que sepamos que el servidor está con poca carga. En cada curso se creará una carpeta accesible desde **Administración**→**Archivos**→**Backupdata**



Añadir cursos

Desde cualquier curso podemos crear copias de seguridad que tienen una funcionalidad añadida: podemos restaurarlas en cualquier otro moodle. Para crear una copia de seguridad para un curso en concreto pulsaremos sobre **Administración**→**Copia de seguridad**



Su uso no presenta ninguna dificultad.

- ➔ **Restaurar cursos de un Moodle a otro.** En esta práctica vamos a guiar la forma de restaurar un par de cursos en castellano que tenemos disponibles en la Web de Moodle <http://moodle.org/course/view.php?id=11>. Se trata de los cursos:

Recurso Moodle Para Profesores <http://moodle.org/mod/resource/view.php?id=2210>

Moodle Para Alumnos <http://moodle.org/mod/resource/view.php?id=2209>

Características de Moodle <http://moodle.org/mod/resource/view.php?id=2574>

Guiaremos el proceso sólo para el primero dejando los otros dos como ejercicio.

Bajaremos el fichero a nuestro equipo¹⁹. Después, desde la página principal de Moodle pulsaremos sobre **Restaurar**



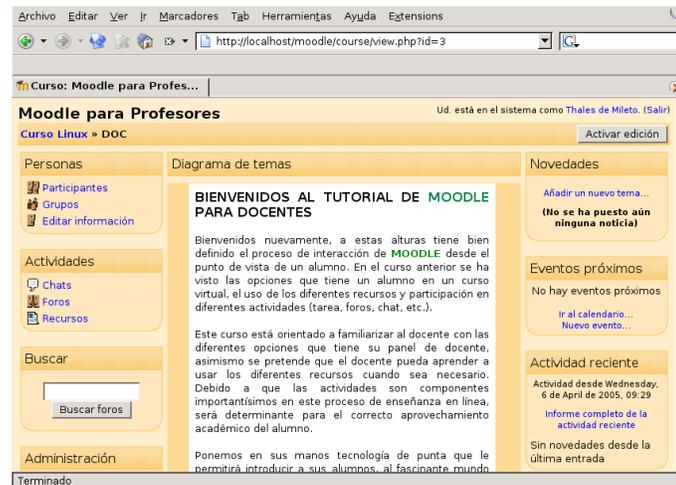
para subir el fichero a la plataforma. Una vez en ella, sólo hemos de marcarlo y pulsar sobre **restaurar**.

¹⁹Si intentamos subir a Moodle ficheros de un tamaño mayor del que permite la configuración del sistema (por defecto 2MB) no nos lo permitirá. Aunque aumentemos ese tamaño, la única solución para ficheros muy grandes es ponerlos en el sitio adecuado a mano. Por ejemplo con:

```
#cp fichero.zip /var/www/moodledata/1/
```



El proceso no presenta mayor dificultad y sólo hay que fijarse bien en las distintas opciones que se nos ofertan. Cuando acabemos tendremos creado un nuevo curso (si hemos elegido esa opción) similar a



Añadir módulos

Si bien las posibilidades de la instalación por defecto de Moodle son muy amplias, existen una serie de añadidos que nos puede interesar tener en nuestro sitio. Podemos ver cuáles hay disponibles en <http://moodle.org/download/modules/>. De los que hay disponibles, la mayoría están instalados pero hay otros que son interesantes y no lo están, por lo que tendremos que instalarlos nosotros. El proceso para todos ellos es parecido así que vamos a instalar sólo uno, se trata de:

Questionnaire que nos va a permitir poder realizar encuestas desde Moodle.

Para instalarlo, lo bajamos a nuestro ordenador y descomprimos el fichero en el lugar adecuado

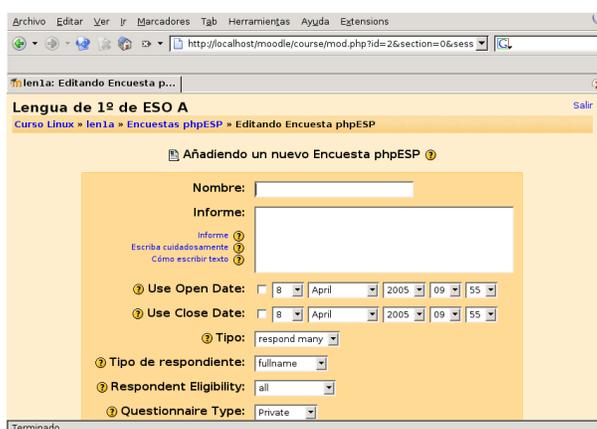
```
# cp questionnaire.zip /var/www/html/moodle/mod/
# cd /var/www/html/moodle/mod/
# unzip questionnaire.zip
```

Ya está, así de fácil. Cuando como administrador entremos en la configuración de los módulos²⁰ (en **Administración**) él sólo se encargará de todo: actualizar la base de datos, crear las tablas, etc.

²⁰Desde aquí, a su vez, podremos borrar los módulos que no nos interesen.



Añadir en un curso un cuestionario es fácil. Con la edición activa marcaremos sobre **Agregar Actividad...→Encuesta phpESP**



Actualizaciones de Moodle

Las versiones de Moodle se suceden casi a diario y a veces es necesario actualizar el sistema porque hay bugs que se han corregido para la versión con la que trabajamos o porque se han añadido funcionalidades interesantes que nuestra versión no soporta. Afortunadamente, este tema también está bien resuelto. Para no alargar la entrega tenéis la forma de hacerlo (en castellano) explicada en <http://localhost/moodle/doc/?file=upgrade.html>.

Se puede resumir en:

- Crear una copia de seguridad de la base de datos

```
#mysqldump -u root -p moodle > moodle-backup-10-04-2005.sql
```

- Movemos el moodle viejo a otro directorio por si acaso:

```
#mv moodle moodle.backup
```

- Descomprimos la última versión en el lugar adecuado

```
#tar xvzf moodle-ultimo.tgz
```

- Copiamos el fichero de configuración de la instalación “vieja” a donde debe de estar

```
#cp moodle.backup/config.php moodle
```

Si hemos hecho cambios en el Moodle que teníamos instalado tendremos que pasarlos al Moodle nuevo: por ejemplo módulos añadidos, temas instalados, etc.



- Por último sólo hemos de ejecutar (él sólo se encarga del resto):

`http://www.midominio.org/moodle/admin`



3.2. PHP-Nuke

¿Qué es PHP-Nuke?

PHP-Nuke es un sistema automatizado de noticias especialmente diseñado para ser usado en Intranets e Internet. El Administrador tiene el control total de su sitio Web, sus usuarios registrados, y tendrá a la mano un conjunto de herramientas poderosas para mantener una página web activa y 100 % interactiva usando bases de datos.

Su autor es Francisco Burzi, que es el que mantiene el código y realiza todas las modificaciones que lleva el paquete original. Podemos encontrar su trabajo en <http://www.phpnuke.org>. Los requisitos para usar PHP-Nuke, que veremos posteriormente, no forman parte del sistema y han de instalarse independientemente. (*Manual de referencia rápida para PHP-Nuke*, CARLOS PÉREZ PÉREZ <http://www.forodecanarias.org/doc/nuke/html/node4.html>)

3.2.1. Introducción

Los portales web son sitios web pensados para manejar una gran cantidad de información y permitir el mantenimiento de páginas web actualizadas y dinámicas usando bases de datos.

En http://www.aferve.com/index.php?module=bkbCompare&func=narrow_selection&id=1 tenéis un estudio comparativo de los distintos portales y sus características.

Hemos optado por PHP-Nuke porque es un clásico, su uso está bien documentado (lo que os permitirá seguir profundizando en su conocimiento) y es fácil de instalar. Además, una vez que se aprenda a manejar uno, el uso de cualquier otro portal es similar y no presentará ningún problema.

3.2.2. Instalación de PHP-Nuke



Con objeto de homogeneizar el proceso de instalación en ambas distribuciones partimos de la configuración del raíz del servidor Web de Apache, en Debian es similar a la de Fedora (véase 3.1.2 en la página 74) .

Vamos a ver la potencia de instalar MySQL (servidor de base de datos) y, junto con PHP y el software PHPNuke (<http://phpnuke.org/>), montar una web altamente configurable.

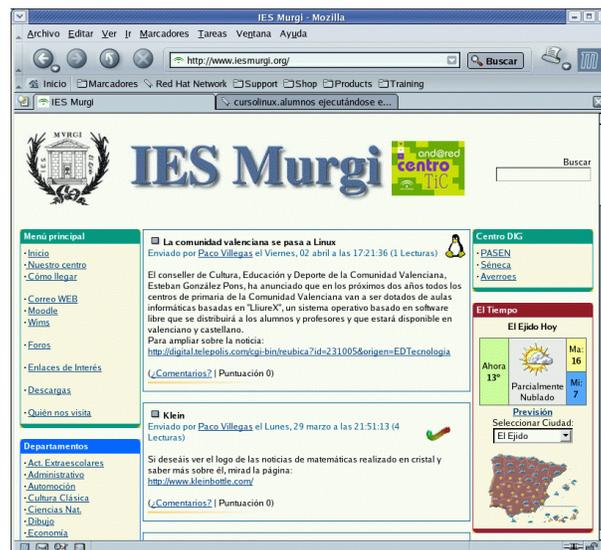


Figura 3.3: Web del IES Murgi

Para eso necesitamos:

1. Tener instalado Apache
2. Tener instalado PHP
3. Tener instalada la base de datos MySQL
4. Instalar el módulo que permite a PHP disponer de soporte de base de datos MySQL
5. Descomprimir el PHPNuke²¹ en `/var/www` (esto es opcional, pero así se queda ya puesto en su sitio):

```
# cp PHP-Nuke-7.5.zip /var/www
# cd /var/www
# unzip PHP-Nuke-7.5.zip
```

6. Una vez descomprimido y tras situarnos en el directorio `/var/www/sql` ejecutemos²²:

```
# mysqladmin -u root -p create nuke
```

para crear la base de datos `nuke`, y

```
# mysql -u root -p nuke < nuke.sql
```

para crear las tablas de esta base de datos según se establece en el fichero `nuke.sql`.

7. Ajustar la contraseña²³ de la base de datos en el fichero `/var/www/html/config.php`

```
$dbname = "root";
$dbpass = "contraseña";
```



Podemos tener varios Nukes instalados en nuestra máquina. Para eso sólo debemos descomprimirlos en directorios diferentes y crear bases de datos diferentes para cada uno de ellos, por ejemplo para el segundo nuke

```
# mysqladmin -u root -p create nuke2
# mysql -u root -p nuke2 < nuke.sql
```

y ajustar la variable adecuada a la base de datos para ese phpNuke, en concreto en el fichero `config.php`

```
$dbname = "nuke2";
```

8. Comprobar que todo está bien, apuntando con nuestro navegador a²⁴ `http://localhost/`

²¹Lo podemos bajar de `http://www.phpnuke.org/modules.php?op=modload&name=Downloads&d_op=viewdownload&cid=1`.

²²Véase el fichero `/var/www/html/INSTALL` para ampliar sobre los detalles de la instalación.

Podemos usar phpMyAdmin para hacerlo, pero en general es más rápido desde el modo comando.

²³Lo deseable es que sea otro usuario el que pueda conectar con la base de datos. para saber cómo se hace con phpMyAdmin debéis consultar en la página 76 cuando se explica para Moodle.

²⁴

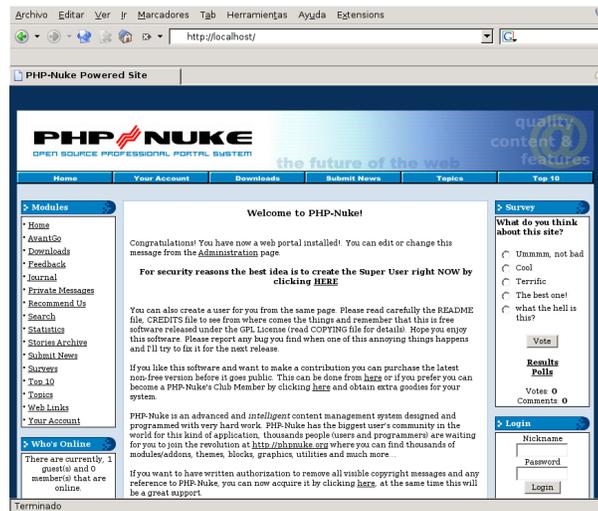
- En Debian es mejor ajustar el `DocumentRoot` modificando el fichero `/etc/apache2/sites-available/default` y ajustarlo a

```
DocumentRoot /var/www/html
```

Una vez cambiado:

```
# apache2ctl restart
```

- Se consigue administrar el portal apuntando a: `http://localhost/admin.php`



¡Bienvenidos a PHP-Nuke!

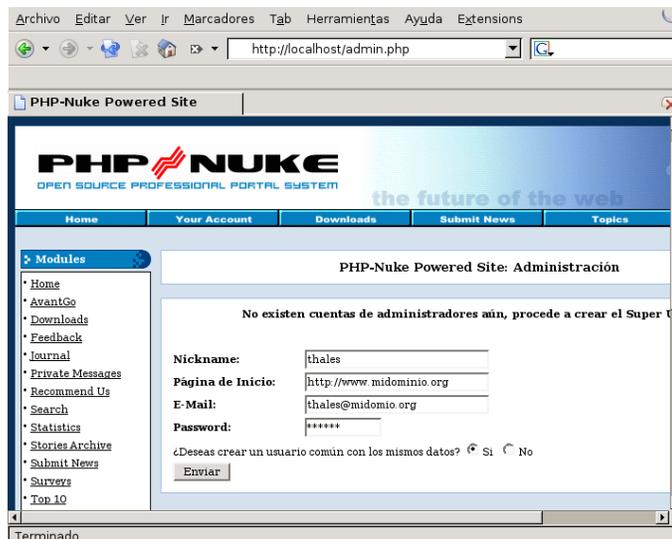
¡Enhorabuena! Ya tiene instalado un portal web! Puede editar o cambiar este mensaje desde la página "Administration".

Por razones de seguridad lo mejor es crear AHORA MISMO el Super-usuario pinchando AQUÍ.

En primer lugar pongamos el portal en castellano para eso optemos porque el interfaz se muestre así, seleccionando el idioma español del bloque de la derecha



Una vez pasado este trámite, vamos crear el super-usuario (para nosotros el socorrido THALES) pinchando allí²⁵:



²⁵Después, una vez autenticados como super-usuario, accederemos a la web de administración.

 Al crear la cuenta *GOD* (se puede poner el nombre que se quiera) no podemos usar ni espacios ni caracteres “extraños”. Si los usamos no podremos acceder al portal. En este caso, una solución rápida (ya que estamos empezando) puede ser:

```
# mysqladmin -u root -p drop nuke
Enter password:
Dropping the database is potentially a very bad thing to do.
Any data stored in the database will be destroyed.
```

Do you really want to drop the 'nuke' database [y/N]

y, tras confirmar, volver a realizar de nuevo el paso 6.

9. Enviemos el formulario y autentifiquémonos en el portal como administrador²⁶. Accederemos a la we de administración:



Figura 3.4: phpNuke admin

Nos queda configurar el portal, ponerlo por defecto en castellano, crear usuarios del sistema, etc. Pero eso lo dejamos para el punto siguiente, salgamos de él pulsando sobre **Salir** .



- Antes de meternos en la configuración, en `/var/www/html/docs` se nos ha instalado la documentación que acompaña al portal. Más de una duda seguro que se puede resolver desde aquí.

²⁶Es conveniente guardar a buen recaudo los datos del administrador y la contraseña de acceso.

- Si lo que desamos es actualizar nuestro Nuke a una versión posterior podéis consultar en: <http://www.phpnuke-hispano.com/modules.php?name=News&file=article&sid=77>

3.2.3. Configuración básica del portal.

 Existen varios documentos bastante buenos para ampliar sobre este tema, por ejemplo:

- <http://www.conocimientosweb.net/portal/modules/Manual/index.htm>
- <http://www.forodecanarias.org/doc/nuke/html/>

y en italiano:

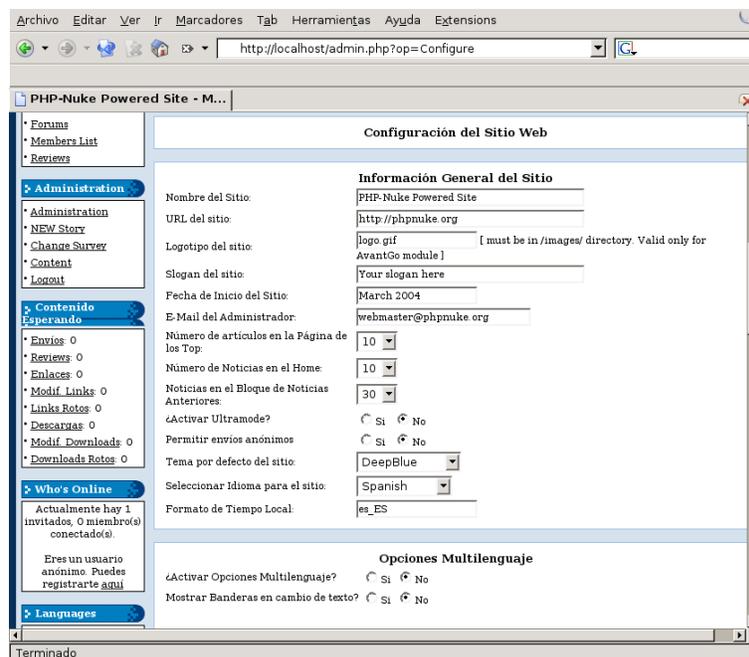
- <http://www.claudioerba.com/>

Veamos algunos aspectos básicos de configuración. Desde el navegador web escribamos:

<http://localhost/admin.php>

y tras autenticarnos accederemos de nuevo a la web de administración (figura 3.4 en la página anterior).

Después, si pulsamos sobre **Preferencias**  nos llevará a una página desde la que podremos configurar los aspectos generales del portal²⁷. Si nos desplazamos hacia abajo en la ventana del navegador, podremos modificar el valor de la lista desplegable **Seleccionar el Idioma para el sitio** (*Spanish*), además cambiaremos el **Formato de Tiempo Local** (*Locale Time Format*) a *es_ES*



Está claro que antes de salir deberíamos adecuar algunos de los campos de esta sección y adaptarlos a nuestro sitio.

De esta ventana comentaremos únicamente que podemos optar por distintos temas por defecto para nuestro phpNuke. Sólo hay que pulsar sobre **Tema por defecto del sitio**, ir seleccionando los que nos van apareciendo en la lista desplegable y guardar los cambios.

²⁷Una descripción más completa de las distintas opciones se puede consultar en: <http://www.forodecanarias.org/doc/nuke/html/node36.html>

Analicemos algunas de las posibilidades de la página de administración. Seguiremos el orden “lógico” inicial de configuración y no el que aparece en la Web.

Mensajes (Administración)

Desde este apartado podemos modificar el mensaje de bienvenida. Si accedemos a él podemos (tendremos que desplazarnos hacia abajo en la página Web)

1. Desactivar el mensaje de bienvenida por defecto. Para eso pulsamos sobre **Editar** el mensaje inicial

Lista de Mensajes					
ID	Título	Lenguaje	Visible para	Activo	Funciones
1	Welcome to PHP-Nuke!	Todo	Todo el mundo	Si	(Editar - Borrar)

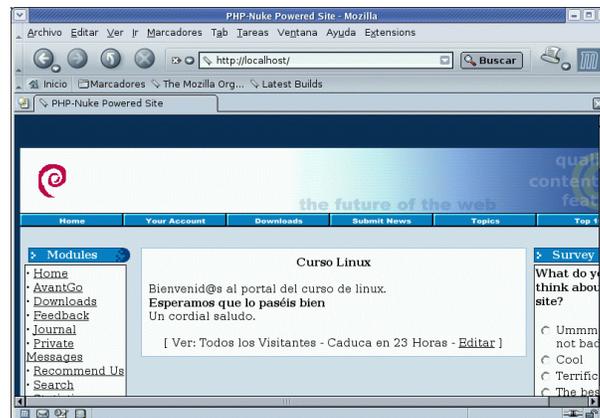
y después sobre **Desactivar mensaje** Active? Si No y [**Guardar Cambios**]

2. Añadir nuestro primer mensaje, para eso escribiremos

Título:	Curso Linux
Contenido:	Bienvenid@s al portal del curso de linux. Esperamos que lo paséis bien Un cordial saludo!

 Si queremos remarcar el texto hay que usar las etiquetas del html.

El resto de opciones no presentan problema. Optamos por dejar las opciones por defecto y pulsamos sobre [**Agregar Mensaje**].



Topics (Módulos)

Es desde donde definiremos los temas en los que se clasificarán nuestras noticias. Por defecto sólo viene un tema predeterminado y, como es obvio, aquí tendremos que dedicar un tiempo antes de que nuestra web esté operativa. Vamos a añadir un tema sobre las cuestiones relacionadas con las matemáticas.

1. Lo primero es buscar un gráfico que se adecúe a nuestro propósito, por ejemplo, el fichero²⁸

²⁸Tendrán que ser gráficos apropiados: png, gif o jpg y de reducido tamaño.

/usr/share/pixmaps/gnome-gnotravex.png (.

2. Situémoslo en su sitio:

```
# cp /usr/share/pixmaps/gnome-gnotravex.png
   /var/www/html/images/topics/gnomegnotravex.png
```



Notar que hemos sido cuidadosos y en el nombre de fichero no hemos permitido caracteres extraños (el guión), si no lo hacemos así, phpNuke no podrá trabajar con ese gráfico.

3. Añadamos el tema.

Agregar un Tópico

Nombre del Tópico:
(sólo un nombre sin espacios - máx: 20 caracteres)
(por ejemplo: juegosyhobbies)

Texto del Tópico:
(el texto completo o descripción del Tópico - máx: 40 caracteres)
(por ejemplo: Juegos y Hobbies)

Imagen del Tópico:



News (Módulos)

Una vez definidos los temas, desde aquí, podemos añadir las noticias que se mostrarán en nuestro portal. Su uso no es complicado y sólo hay que tener en cuenta que:

- Hay que seleccionar obligatoriamente un tema principal desde la lista desplegable **Tópico**. Es opcional marcar las casillas de **Tópicos asociados**.
- Para remarcar el texto es necesario usar las etiquetas de html.
- El **Texto de la Noticia** es lo que se verá en la página principal. Al **Texto extendido** hay que acceder.

¿Quieres programar esta historia? Si No

Ahora es: March 26, 2004 @ 01:50:04

Día: Mes: Año:

Hora: : : 00

- Podemos programar la fecha en que se mostrará la noticia en la página principal (por ejemplo el 31/12/2004 a las 0 horas para felicitar del año nuevo) y añadir una encuesta a cada noticia que pongamos.
- Si pulsamos sobre **Aceptar** estando activa la opción de **Vista Previa** se nos mostrará cómo quedará la noticia, pero no se envía. Para enviarla hay que optar por **Enviar Noticia** y **Aceptar**.

- Desde la ventana principal de administración siempre podremos editar o borrar las noticias que nosotros hayamos enviado.

Analicemos ya con cierto orden el resto de opciones

Administración



Respaldo

Es para crear una copia de seguridad de la base de datos de nuestro PHP-Nuke.



Banner

Por si deseamos añadir carteles publicitarios a nuestro sitio y gestionar la “cartera” de clientes.



Bloques

Es uno de los elementos fundamentales de nuestro portal.

Administración de Bloques						
Título	Posición	Peso	Tipo	Estado	Visible para	Funciones
Modules	Izquierda	1	ARCHIVO	Activo	Todo el mundo	[Editar Desactivar Borrar Ver]
Administration	Izquierda	2	SISTEMA	Activo	Sólo Administradores	[Editar Desactivar Borrar Ver]
Who's Online	Izquierda	3	ARCHIVO	Activo	Todo el mundo	[Editar Desactivar Borrar Ver]
Search	Izquierda	4	ARCHIVO	Inactivo	Todo el mundo	[Editar Activar Borrar Ver]
Languages	Izquierda	5	ARCHIVO	Activo	Todo el mundo	[Editar Desactivar Borrar Ver]
Random Headlines	Izquierda	6	ARCHIVO	Inactivo	Todo el mundo	[Editar Activar Borrar Ver]

Desde aquí podemos administrar los bloques de nuestro portal. Las dos columnas “importantes” son:

Peso Nos permite subir o bajar un bloque con sólo pulsar sobre las flechas ↑↓

Funciones al pulsar sobre el enlace se ejecuta la acción asociada.

Editar para modificar el bloque

Activar/Desactivar se trata de un “interruptor” que nos permite que un bloque se muestre o no en la página principal.

Borrar pues eso.

Ver para acceder a una vista previa de él.

➔ **Añadamos un bloque:** en esta práctica vamos a añadir un bloque en el que se mostrarán las noticias de la web de *Barrapunto*. Para poder hacerlo necesitamos que el sitio de donde las vamos a “coger” tenga un fichero dentro de su servidor, del tipo RSS/RDF. La URL es:

<http://backends.barrapunto.com/barrapunto.rss>

Agregar Bloque

Título:

Archivo RSS/RDF del sitio: [Setup]
(Selecciona "No Definido" y escribe el URL o selecciona simplemente un sitio en la lista para desplegar sus noticias)

Nombre de Archivo: (Selecciona un bloque a ser incluido. Todos los demás campos serán ignorados)

Si guardamos los cambios (**Crear Bloque**) y nos situamos sobre la página principal tendremos un bloque donde aparecerán las noticias de la página que hemos elegido antes, su contenido será similar al que se muestra en la figura lateral. ■



Editar Admins

Si trabajamos en un centro de enseñanza y hemos sido o somos los WebMaster sabemos que es más que deseable poder repartir el trabajo. Desde esta página podemos dar de alta a usuarios que van a poder administrar con nosotros el portal. Está en nuestras manos definir qué nivel de “trabajo” le vamos a permitir al autor:

Administración de Autores

Editar Admins

thales Todo Modificar Información Administrador God*

*(la cuenta GOD no puede borrarse)

Agregar Autor

Nombre: después (requerido, no podrá ser cambiado)

Nickname: Requerido

E-Mail: Requerido

URL:

Permisos:

Content Downloads Encyclopedia

FAQ News Reviews

Surveys Topics Web Links

Your Account

Super Usuario

ATENCIÓN: si seleccionas Super Usuario, tendrá acceso total

Password: Requerido

Agregar Autor

Para que el nuevo “administrador” ejecute sus funciones en nuestra máquina www.midominio.org tendrá que usar la URL: <http://www.midominio.org/admin.php>



Forums

phpNuke integra como módulo uno de los mejores paquetes para gestionar foros de la actualidad. Se trata de sistema de foros phpBB

Welcome to phpBB

Thank you for choosing phpBB as your forum solution. This screen will give you a quick overview of all the various statistics of your board. You can get back to this page by clicking on the [Admin Index](#) link in the left pane. To return to the index of your board, click the phpBB logo also in the left pane. The other links on the left hand side of this screen will allow you to control every aspect of your forum experience. Each screen will have instructions on how to use the tools.

Forum Statistics

Statistic	Value	Statistic	Value
Number of posts:		Posts per day:	
Number of topics:		Topics per day:	
Number of users:	1	Users per day:	0.00
Board started:	Sun Feb 17, 2002 11:10 am	Avatar directory size:	Not available
Database size:	155.84 KB	Gzip compression:	OFF

Who is Online

La web del programa es, <http://www.phpbb.com/>, desde la sección *downloads* podemos acceder

barrapunto.com

- Los países embargados podrán publicar en el IEEE
- Sanyo lanza los cedés de maíz
- ¿Cómo veis /?
- Patentes de software en Europa: de vuelta a las andadas
- Entrevista con el CEO de SCO en "El País"
- Integrando Mozilla y Firefox en KDE
- Los próximos pasos de Google
- Alianza entre Mozilla y Gnome?
- Vicepresidente de Sun renuncia debido a reconciliación con Microsoft
- Moodle: La mejor herramienta libre para crear cursos

[Leer más...](#)

a los módulos de lenguaje. Se trata de bajarnos los que nos van a permitir poner nuestros foros en castellano, es decir, los paquetes²⁹:

```
lang_spanish.tar.gz
subSilver_spanish.tar.gz
```

Descomprimos el fichero `lang_spanish.tar.gz` en `/var/www/html/modules/Forums/language`

```
#cp lang_spanish.tar.gz /var/www/html/modules/Forums/language
#cd /var/www/html/modules/Forums/language
#tar -xzf lang_spanish.tar.gz
```

y, tras entrar en el menú de administración del phpNuke, iniciamos la sección de foros. En el menú de la izquierda nos situamos sobre **General Admin Configuration** y en **Default Language** seleccionamos el idioma **español** (antes de descomprimir el fichero no aparece esa opción) y enviamos el formulario³⁰.

Estilo por defecto	subSilver
Ignorar el estilo del Usuario Se utilizará el estilo seleccionado por defecto sin importar la elección del usuario	<input checked="" type="radio"/> Sí <input type="radio"/> No
Lenguaje por Defecto	Spanish
Formato de la Fecha La sintaxis usada es idéntica a la función <code>date()</code> de PHP	D M d, Y g:i a

De esta forma conseguimos que el entorno esté en castellano.

Pero no está castellanizado “del todo”, el tema por defecto se denomina *subSilver* y, además, deseamos que los gráficos estén en consonancia con el idioma que acabamos de instalar. Nada más simple, pongamos el segundo fichero donde debe estar y desempaquetémoslo:

```
# cp subSilver_spanish.tar.gz /var/www/html/modules/Forums/templates/
# cd /var/www/html/modules/Forums/templates/
# tar -xzf subSilver_spanish.tar.gz
```

Está claro que ahora es más fácil modificar y adecuar la configuración de los foros: se deja como ejercicio :-).



Users Groups

Permite crear grupos y asignar usuarios a esos grupos, además se puede conseguir que los grupos puedan ver sólo ciertos módulos y bloques.



HTTP Referers

¿Quién enlaza nuestro sitio? El número (1000 por defecto) de referencias se controla desde la página de **Preferencias**

Opciones Variadas	
Activar HTTP Referers	<input checked="" type="radio"/> Sí <input type="radio"/> No
¿Cuántas Referers quieres como máximo?	1000
¿Activar Comentarios en las Encuestas?	<input checked="" type="radio"/> Sí <input type="radio"/> No
¿Activar Comentarios para las Noticias?	<input checked="" type="radio"/> Sí <input type="radio"/> No

²⁹Podemos optar por la versión en formato `.zip` o la `tar.gz`

³⁰Si no nos funciona bien, probemos creando el fichero:

```
#touch lang-spanish.php
```

y se debe eliminar el problema.



IP Ban

Para bloquear las IP de los “chicos malos”



Módulos

Se trata de uno de los elementos más importantes de la configuración. Los módulos son los que dotan (junto con los bloques) de contenido a nuestra Web.

En esta sección accederemos a una tabla en la que se nos informa de los distintos módulos, así como de sus características.

Muestra el estado actual de tus módulos/addons y permite cambiar su estado activando o desactivado. Nuevos módulos copiados en el directorio `/modules/` serán automáticamente agregados en estado `Inactivo` cuando recargues esta página.

Si deseas borrar un módulo, simplemente borra el directorio del mismo de `/modules/`, el sistema hará la actualización de forma automática para mostrar los cambios.

-- CUIDADO --

El título en negritas del módulo representa el módulo que tienes actualmente en el Homepage. No podrás desactivar este módulo o especificar restricciones mientras sea el módulo del Home! Si borras el directorio del módulo verás un error en el Homepage. También, este módulo ha sido reemplazado con un link a `Home` desde el bloque de módulos.

[·] implica un módulo cuyo nombre y enlace no son visibles en el Bloque de Módulos

Título	Título Propio	Estado	Visible para	Group	Funciones
Addon_Sample	Addon Sample	Inactivo	Sólo Administradores	Ninguno	[Editar Activar Al Home]
AvantGo	AvantGo	Activo	Todo el mundo	Ninguno	[Editar Desactivar Al Home]
Content	Content	Inactivo	Todo el mundo	Ninguno	[Editar Activar Al Home]
Downloads	Downloads	Activo	Todo el mundo	Ninguno	[Editar Desactivar Al Home]
Encyclopedia	Encyclopedia	Inactivo	Todo el mundo	Ninguno	[Editar Activar Al Home]
FAQ	FAQ	Inactivo	Todo el mundo	Ninguno	[Editar Activar Al Home]
Feedback	Feedback	Activo	Todo el mundo	Ninguno	[Editar Desactivar Al Home]

Título nombre del directorio que contiene al módulo.

Título propio nombre del módulo, este valor lo podemos castellanizar.

Estado si está activo o no

Visible para ¿qué usuarios pueden acceder a él?, ¿todos, sólo los registrados o sólo los administradores?

Funciones para editar, activar/desactivarlo o ponerlo como módulo por defecto en la página principal (*Al Home*)

No sólo podemos usar los que vienen por defecto. Añadir un módulo a phpNuke (en general) es bastante sencillo: sólo hay que bajar el módulo y descomprimirlo en el lugar adecuado³¹ (`/var/www/html/modules`). Después tenemos que recargar la página desde la que controlamos los módulos, y se añadirá a la tabla de características de los módulos. Como por defecto estará desactivado, tendremos que activarlo y castellanizarlo. Veamos un ejemplo de cómo añadir un módulo.

➔ **Coppermine:** este módulo³² permite para crear álbumes de fotos³³ dentro de portal, y además está traducido al español. Podemos bajarlo de <http://www.nukebazar.com/>, la única “pega” es que antes hay que registrarse.³⁴

³¹Puede que además tengamos que “retocar” la base de datos nuke.

³²La Web del programa es: <http://coppermine.sourceforge.net>

³³Existen en la Web múltiples utilidades de este tipo. Además de la comentada merece la pena probar:

gallery <http://gallery.menalto.com/>

My Photo Gallery <http://www.fuzzymonkey.org/cgi-bin/newfuzzy/software.cgi>

³⁴Está a vuestra disposición en la Web del curso

Una vez en nuestra máquina, lo descomprimos

```
$ mkdir coppermine
$ cp coppermine1,1d6.5.zip coppermine
$ cd coppermine
$ unzip coppermine1,1d6.5.zip
```

Al descomprimir se crean varios directorios³⁵: **block** y **modules**. Lo primero será actualizar la base de datos nuke con:

```
#mysql -u root -p nuke <coppermine.sql
```

Por último copiamos el contenido de los directorios anteriores en los lugares adecuados:

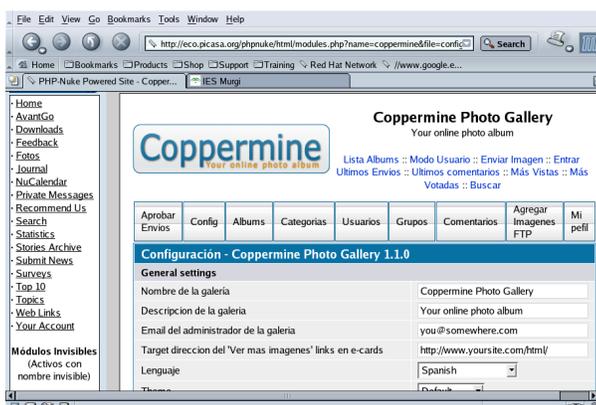
block en `/var/www/html/block`

modules en `/var/www/html/modules`

Ya está. Reiniciemos el bloque de módulos y activémoslo. Mejor si además le ponemos un nombre en castellano.

coppermine	Fotos	Activo	Todo el mundo	Ninguno	[Editar Desactivar Al Home]
------------	-------	--------	---------------	---------	-----------------------------------

Para administrarlo, pulsaremos sobre **Fotos** del bloque **Módulos (Modules)** de la página principal del portal



Al pulsar sobre **Config** podremos optar por cambiar el idioma y configurarlo a nuestro gusto.

Para no perder de vista lo que es importante (por ahora) aparcemos la configuración de este módulo hasta la 3.2.4 en la página 104. En ella se explica cómo crear una galería y la forma de introducir nuestras fotos. Así que continuemos con la configuración del portal. ■

@ Boletín

Para enviar un correo (boletín) a los usuarios que están suscritos o a todos los usuarios del portal.

⚙️ Optimize DB

Para optimizar la base de datos de nuestro portal.

🕒 Envios

Permite configurar desde **Preferencias** si deseamos que usuarios anónimos puedan poner artículos en nuestro portal o si, antes de que se publique, el administrador o usuarios han de darle el visto bueno.

³⁵Además del fichero que explica cómo se instala: `Installation-english.txt`

Moderación de los Comentarios

Tipo de Moderación:

Si establecemos algún tipo de moderación, aquí se almacenarán las noticias pendientes de publicar. Depende de nosotros decidir qué noticias publicamos y cuáles no.

Módulos



Contenido

Nos permite crear páginas de contenido diverso (“cajón de sastre”) clasificadas por temas. Si usamos la etiqueta `<!--pagebreak-->` podremos crear documentos (usando las etiquetas del HTML) de varias páginas.



Descargas (Downloads)

En primer lugar tendremos que crear las categorías de descargas, después añadiremos los programas y/o materiales que deseamos se puedan bajar desde nuestro portal. Una vez creada la categoría principal se nos “complicará” la página permitiéndonos añadir subcategorías y descargas clasificadas dentro de esas categorías. No presenta mayor dificultad.



Enciclopedia

Permite generar una enciclopedia basada en categorías. Se pone el nombre a la enciclopedia y después se agregan los términos. El trabajo es ir alimentando los términos.



FAQ

O su traducción PUF, es decir, que como estamos de trabajo fatal, seguro que diremos ¡puuf! cuando nos digan que hagamos la FAQ (*Frequently Asked Questions*). Es broma: se trata de que podemos crear una serie de preguntas de uso frecuente (PUF) con sus respuestas, que facilite a los usuarios el uso del portal.



Reviews

Abre una página dedicada al análisis de productos y servicios. Los usuarios pueden introducir comentarios amplios sobre un determinado producto, su nombre, e-mail, enlaces. Permite un análisis más completo e interactivo que las encuestas.



Secciones

Para crear secciones especiales, es el lugar adecuado para situar artículos que no entran dentro de la página principal.



Encuestas (Survey/Polls)

Administración de encuestas, su uso es sencillo e intuitivo.



Enlaces (Web Links)

Para gestionar los enlaces a páginas interesantes desde nuestro portal. Primero se crean las categorías (Ejemplo: enlaces de matemáticas) y después, dentro de cada categoría, se añaden los enlaces.



Usuarios

En PHP-Nuke podemos encontrar tres tipos de usuarios: superusuarios (administradores), todo el mundo (usuarios que visitan la página pero no se registran) y usuarios registrados (al registrarse se les envía un login y una contraseña que les permite interactuar con la página, por ej. enviando noticias, participando en foros, etc). Desde aquí podemos gestionar estos últimos.



Efemérides

Desde aquí tendremos la posibilidad de programar las efemérides que deseamos se muestren en la web principal para los días establecidos.

➔ Cambiemos de Logo (véase la figura 3.3 en la página 91)

Para poder poner un logo personalizado a nuestro portal lo hemos de hacer de forma manual. Existen dos formas de hacerlo:

1. Tendremos que garantizarnos que nuestro logo es un fichero en formato `gif` y de nombre `logo.gif`. Los temas de phpNuke se guardan en el directorio `html/themes`

```
# ls /var/www/html/themes/
3D-Fantasy ExtraLite Karate Odyssey SlashOcean
Anagram index.html Milo Sand_Journey Sunset
DeepBlue Kaput NukeNews Slash Traditional
```

En general, en cada uno de ellos existe un directorio de nombre `images`. En él se encuentra el fichero `logo.gif`. Sólo hemos de sobrescribirlo.

2. La segunda forma de hacerlo es:

- a) Crear nuestro logo, por ejemplo, `logomicentro.png` y ponerlo en el subdirectorio `images` de ese tema (por ejemplo en `/var/www/html/themes/Kaput/images`).

- b) Entrar en el directorio del tema en el que deseamos poner nuestro logo y modificar el fichero `theme.php`³⁶, buscar una línea de la forma:

```

```

y modificarla a nuestro gusto

```

```

■

3.2.4. Coppermine

Sigamos con este módulo que habíamos dejado a medias y vamos a iniciarnos sobre su uso³⁷. Vamos a crear un álbum de forma rápida en el que poner las fotos de la última Olimpiada Matemática Thales, esas fotos las hemos puesto en un directorio de nombre `olimpiada`. De nuevo os recordamos que no pretendemos tratar todas las posibilidades, sólo dar una primera idea de qué se puede hacer.



Antes de seguir, varias cuestiones a tener en cuenta si las fotos son de aproximadamente 512KB o más. Si es así, en la pestaña que permite configurar Coppermine (**Config**) accedemos a dos parámetros de configuración para los cuales es necesario aumentar³⁸ los valores:

- Máximo tamaño de los ficheros añadidos por los usuarios (KB), por defecto está a 1024, doblar su valor no es mala idea.
- Máxima anchura o altura de las imágenes/videos añadidos (pixels). Puede que no sea necesario modificar en exceso este parámetro, en cualquier caso aumentarlo un poco no debe generar ningún problema. Se trata de ajustarlo a las fotos de la máquina con que trabajamos.
- Añadir la ruta del programa `convert`, en general será de la forma `/usr/bin/`

Además, si al intentar añadir fotos a un álbum, el proceso no finaliza, se debe a que tenemos que aumentar la directiva `memory_limit` del archivo de configuración de PHP (`php.ini`) a 32 MB al menos³⁹. Si no tiene ese valor no podremos añadir a los álbumes fotos de calidad media.

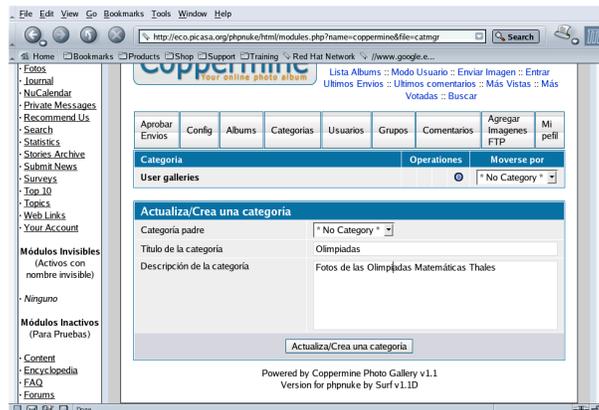
La primera labor a realizar consiste en crear una categoría en la que pondremos las fotos de las distintas Olimpiadas celebradas. Para eso pulsamos sobre la pestaña **Categorías** y le ponemos el nombre y descripción adecuados.

³⁶En los temas *3D-fantasy*, *NukeNews* y *Odyssey* se trata del fichero `header.html`

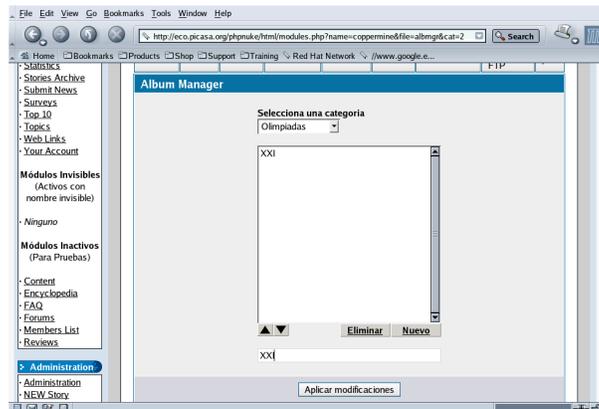
³⁷Un manual en Inglés sobre él: <http://coppermine.sourceforge.net/manual.php>

³⁸Si además, deseamos permitir que se puedan subir ficheros de más de 2MB véase 3.1.5 en la página 85

³⁹Habrà que reiniciar el servidor web para que los cambios tengan efecto.



Creemos ahora el álbum de fotos para esta convocatoria de Olimpiada, se trata de la XXI edición, así que pulsamos sobre **Albums** y, al pulsar sobre **Nuevo** añadimos el álbum XXI dentro de la categoría Olimpiadas.



Una vez que tenemos creado el álbum, vamos a añadir todas las fotos de una tajada (es posible hacerlo de una en una, tanto como administrador como permitiéndolo a los usuarios). Pondremos nuestras fotos en el lugar adecuado⁴⁰ para no tener que subirlas una a una, por ejemplo con

```
# mv olimpiada/ /var/www/html/modules/coppermine/albums/userpics/
```

Y nos garantizamos que el directorio y las fotos tengan los permisos adecuados, bien haciendo que sean del usuario Apache⁴¹

```
# cd /var/www/html/modules/coppermine/albums/userpics/
# chown www-data olimpiada/*
# chown www-data olimpiada
```

o bien relajándole al directorio los permisos al modo 777. Una vez que tenemos nuestras fotos listas pulsamos sobre **Agregar Imágenes**, nos situamos en el directorio `olimpiada` y marcamos las que deseamos añadir al álbum.

⁴⁰Este directorio se puede cambiar desde la pestaña **Config**

⁴¹En realidad es suficiente con que tenga permisos sobre el directorio. El que tenga permisos sobre las fotos nos sirve sólo en el que caso de que deseemos hacer cambios sobre ellas desde el propio programa.

En Fedora, el usuario por defecto del servidor Web no es `www-data`, es `apache`.



El resultado se puede comprobar en la captura que sigue.



Ya podemos ir añadiendo las de la semana cultural de Instituto, las del viaje de estudios, etc. Sacarle más partido a este magnífico módulo es cosa vuestra.

Bibliografía

- [1] *Creación de sitios web con PHP4*, F^o JAVIER GIL RUBIO y otros, Osborne McGraw-Hill
- [2] *Desarrollo Web con PHP y MySQL*, LUKE WELLING & LAURA THOMSON, Anaya Multimedia
- [3] *MySQL*, PAUL DUBOIS, Editorial Prentice-Hall
- [4] *MySQL*, IAN GILFILLAN, Anaya Multimedia
- [5] *Servidor Apache*. RICH BOWEN & KEN COAR. Prentice Hall
- [6] *Servidor Apache 2*. MOHAMMED J. KABIR. Anaya Multimedia.
- [7] <http://www.mysql-hispano.org/articles.php>
- [8] <http://www.conocimientosweb.net/portal/modules/Manual/general5.htm>
- [9] <http://html.conclase.net/w3c/html401-es/cover.html>
- [10] <http://www.rinconastur.com/php/>
- [11] <http://dev.mysql.com/doc/>
- [12] Manual de PHP <http://www.php.net/manual/es/>