Introducción al programa EasyPHP.

EasyPHP es un paquete fácil de instalar y configurar cuya misión es instalar Apache, MySQL, Php y PhpMyAdmin en una máquina que disponga del S.O. Windows 9x, 2k y XP. De esta forma podremos trabajar en nuestro ordenador con aplicaciones Web construidas con código php para nuestra familiarización con ellas.

Apache es un servidor Web cuya misión es la de recibir y responder peticiones de páginas Webs.

PHP es un lenguaje de programación interpretado para la construcción de páginas Web del lado del servidor.

MySQL es un sistema de bases de datos de uso bajo Linux.

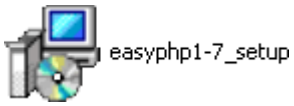
PHPMyAdmin es un sistema mediante el cual, podemos desde un navegador mantener una base de datos MySQL de manera fácil e intuitiva.

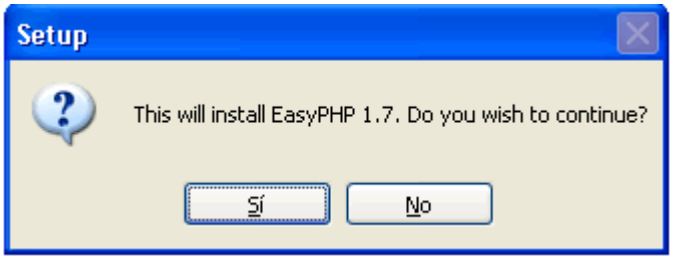
Instalación del programa EasyPHP.

Desde la página oficial, en la sección de descargas te puedes bajar la última versión del archivo instalador.

Además te conviene bajar el archivo de "Páginas locales en español".

Una ves que hayas bajado el instalador, procederemos a dar un doble click en el paquete instalador:



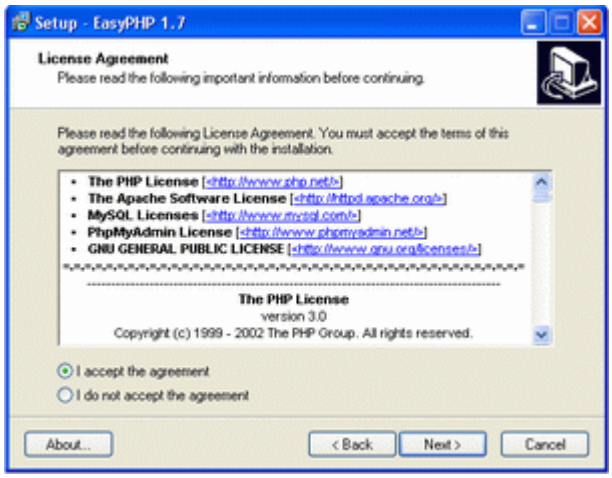


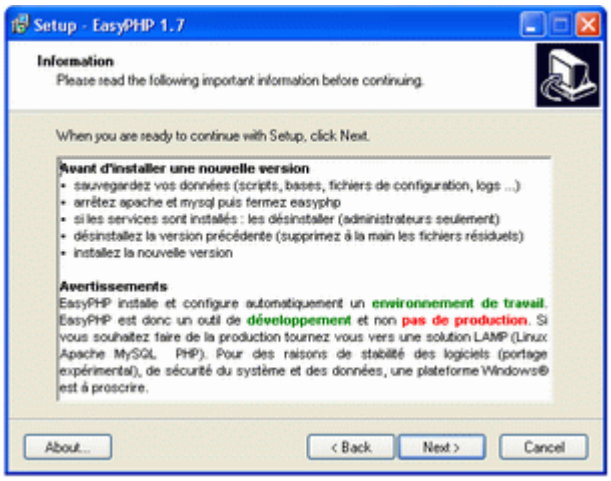
Ahora aparecerá una ventanita preguntandonos si deseamos instalar el EasyPHP 1.7; pulsamos sobre Sí.

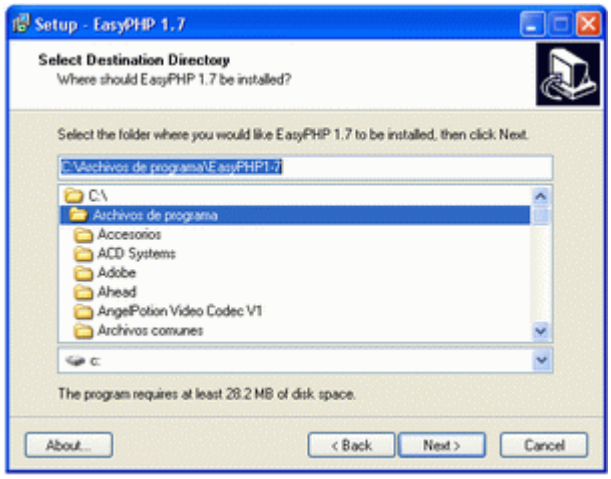
Luego procederemos a seguir contestando el asistente de instalación:

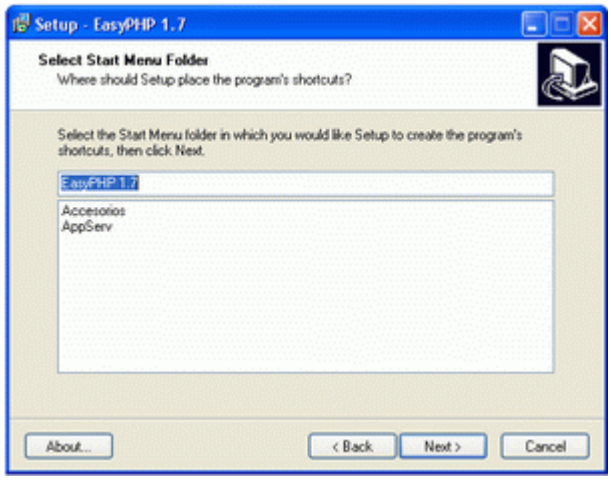
Después de leer la Bienvenida, le daremos un click en Next.

A continuación tendremos que aceptar los términos de la licencia "I accept ..." y pulsar sobre Next.

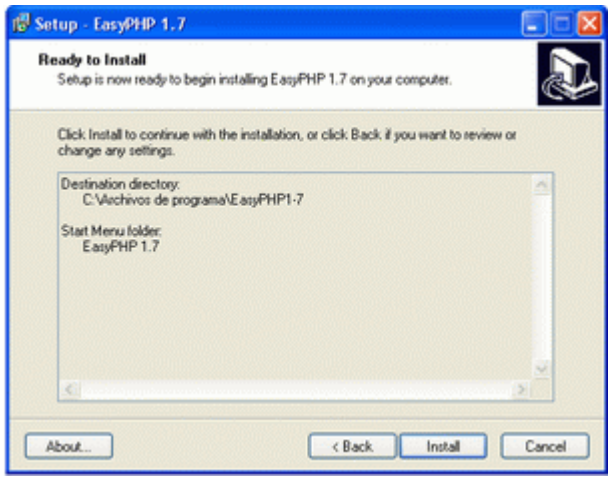




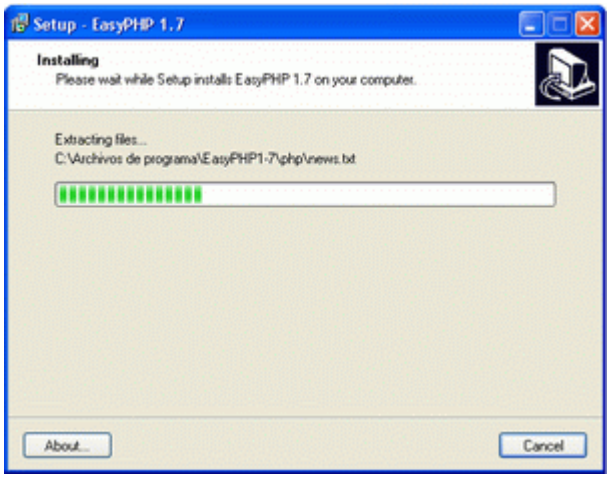




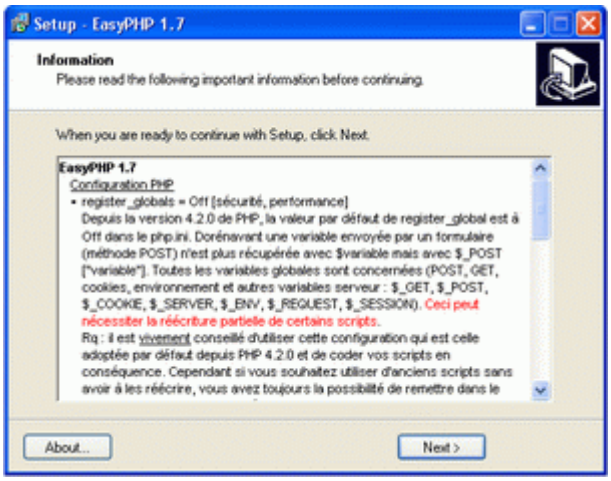
Nos pide seleccionar el menú donde querramos que instale el acceso al programa; podemos dejarlo tal y como está y pulsamos de nuevo sobre Next.



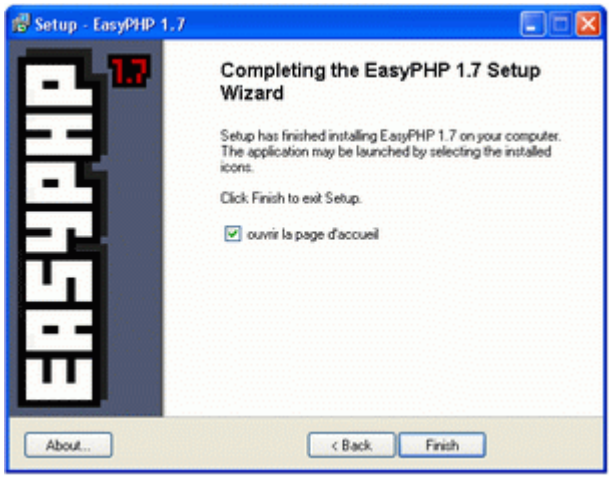
Después de haber preparado los archivos de instalación, el asistente realiza el proceso de instalación en nuestra máquina.



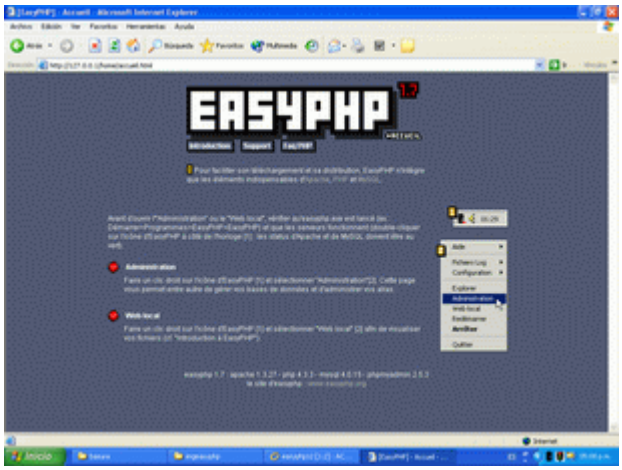
En estos momentos está extraendo los archivos que estaba en el instalador hacia la nueva carpeta creada.



En pantalla nos muestra el cómo configurar el paquete después de su completa instalación. Demosle un click en Next.



Por fín hemos terminado la instalación de EasyPHP 1.7 en nuestro PC. Si deseas puedes dejar seleccionado la casilla, para luego dar Click en Finish. Si dejastes seleccionado la casilla, nos mostrará una Bienvenida en nuestro explorador:



Puedes ver una animación del proceso de instalación:

Instalación páginas locales en español.

Para la instalación de las páginas locales en español, lo único que debemos hacer es depositar el contenido de la carpeta comprimida en la carpeta **C:\Archivos de programa\EasyPHP1-7.** A continuación se nos pedirá confirmación para reemplazar los archivos existentes, a lo que responderemos **Sí**.

Utilización básica y configuración de EasyPHP.

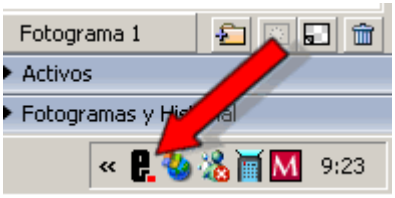
Una vez instalado el paquete tendremos que configurar algunos aspectos básicos como son:

- Actualización al idioma castellano.
- Definición de la carpeta base de nuestras páginas Web.
- Establecimiento de una contraseña para el administrador de MySQL.

Actualización al castellano.

Para iniciarlo seleccionamos **Inicio / Todos los programas / EasyPHP 1.7 / EasyPHP**

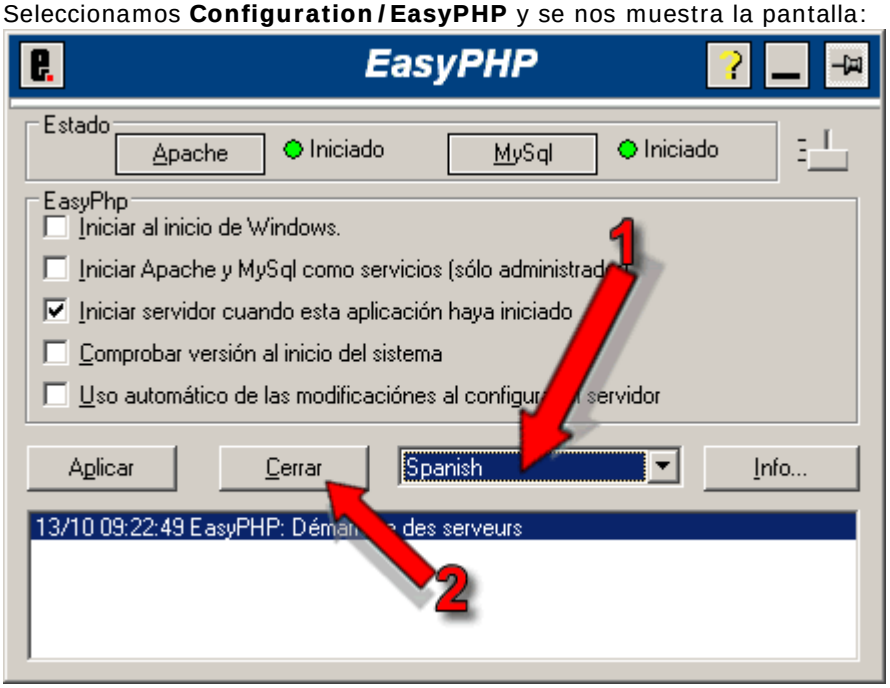
A continuación, en la barra de estado nos debe aparecer el logo del programa con un puntito rojo parpadeante:



Esto nos indica que el servidor Apache está activo y dispuesto a servir páginas Web.

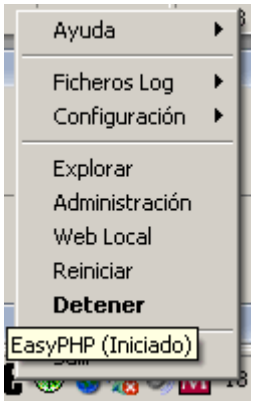
Pulsando sobre el botón derecho del ratón sobre el logo se nos muestran las posibles opciones que podemos realizar. De momento vamos a configurar el entorno al español:

- Pulsamos sobre el botón derecho en el logo.
- Seleccionamos **Configuration / EasyPHP** y se nos muestra la pantalla:



- Así, en la ventanita marcada seleccionaremos la opción **Spanish**, y a continuación pulsamos el botón **Cerrar.**

Ahora, si pulsamos sobre el botón derecho de nuevo, las opciones de uso ya se nos muestran en español:



Definición de la carpeta base de nuestras páginas Web.

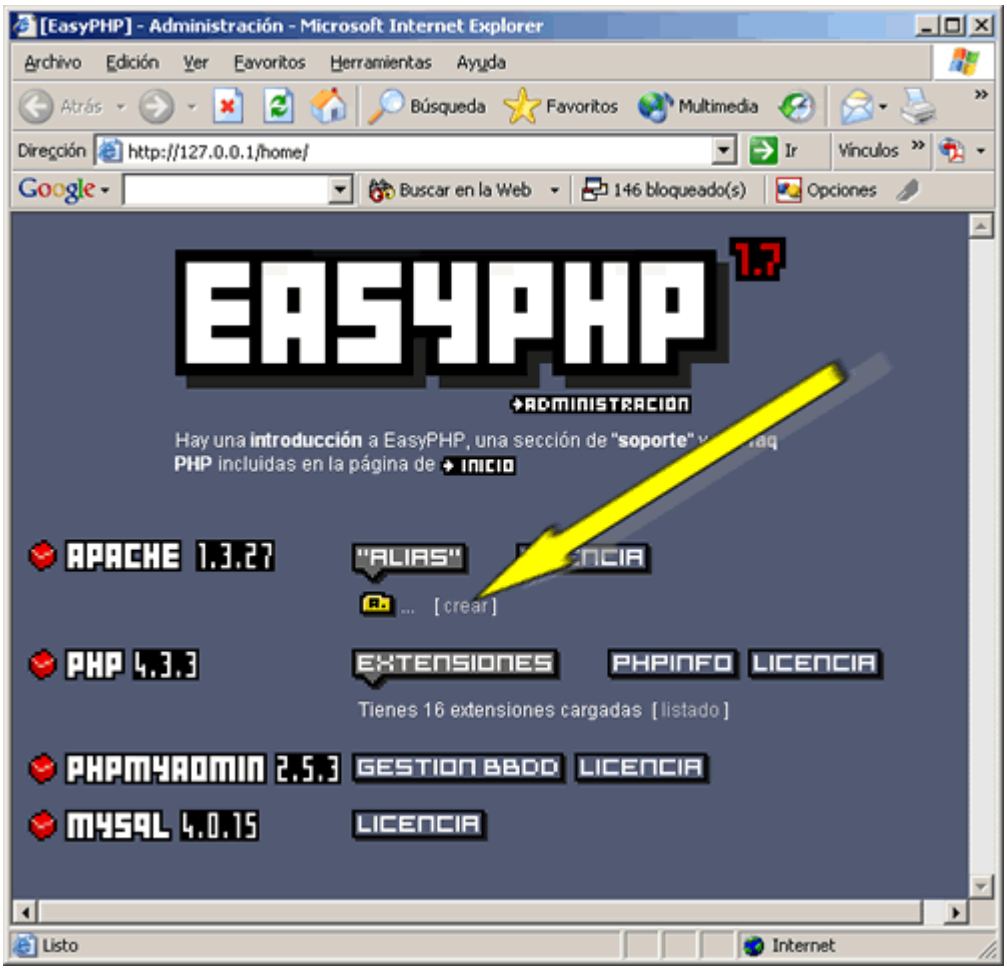
Por defecto, EasyPHP define una carpeta como la depositaria de todas las páginas que el servidor Apache debe servir.

Esta es la carpeta **C:\Archivos de Programa\EasyPHP1-7\www**

Luego todas las página que pensemos desarrollar, deberíamos situarlas en carpetas dentro de la mencionada.

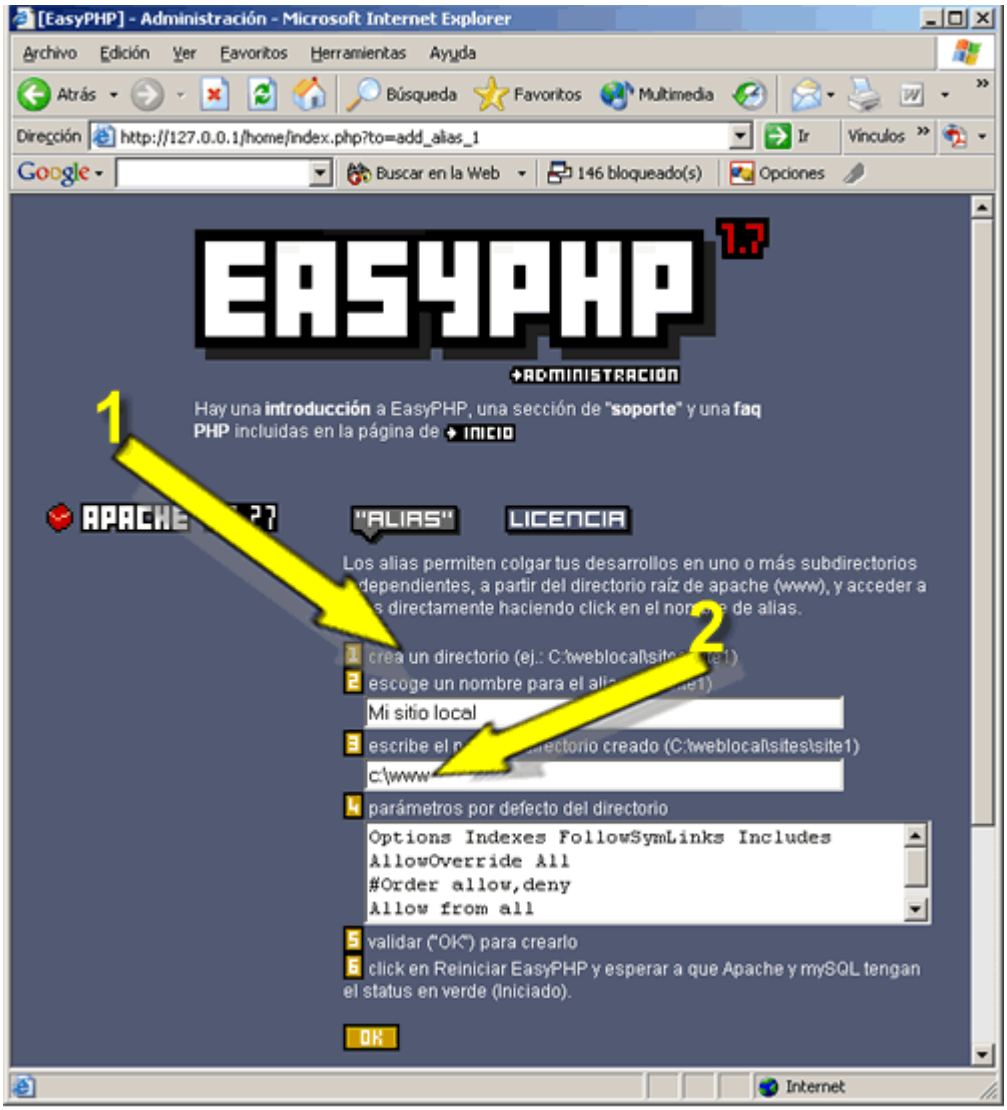
No obstante, si en nuestros discos duros ya disponemos de una carpeta donde se encuentras nuestras páginas web, podemos indicarselo a Apache para no tener que mudarlas a la carpeta www. Los alias en EasyPHP son otras carpetas donde se encuentras páginas:

Para ello seleccionamos la opción de **Administración**, tras lo que se nos muestra la página de configuración:



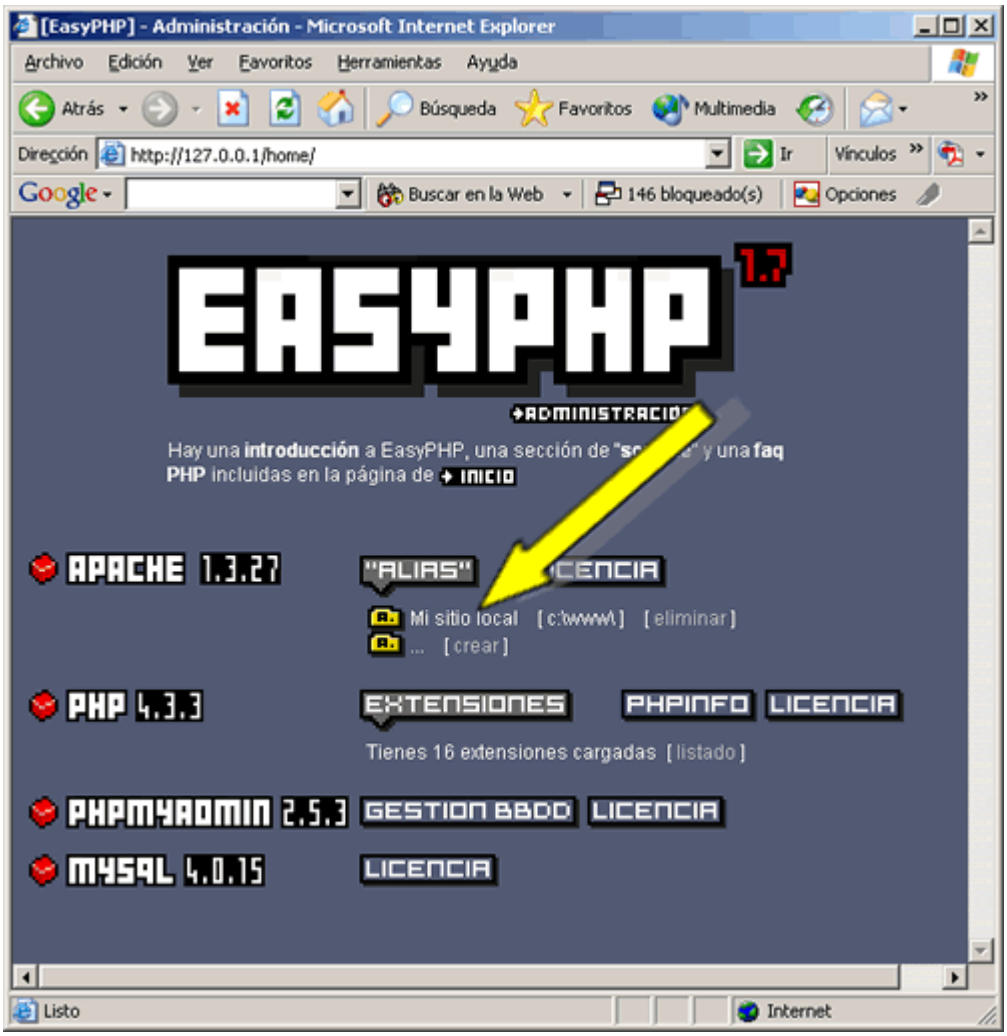
Pulsamos donde indica la flecha amarilla para crear un nuevo alias.

A continuación, en la posición 1 ponemos un nombre al nuevo alias y en la 2 escribimos la ruta de la carpeta:



Para finalizar bastará con pulsar sobre la tecla OK.

A partir de ese momento podremos visualizar nuestras páginas Web alojadas en el alias creado. Para ello accederemos de nuevo a la opción **Administración** de EasyPHP. Ahora se nos muestra en la ventana un acceso nuevo al alias creado.



Pulsando sobre él podrás visualizar su contenido.

Determinación de una contraseña para el administrador (root) de MySQL

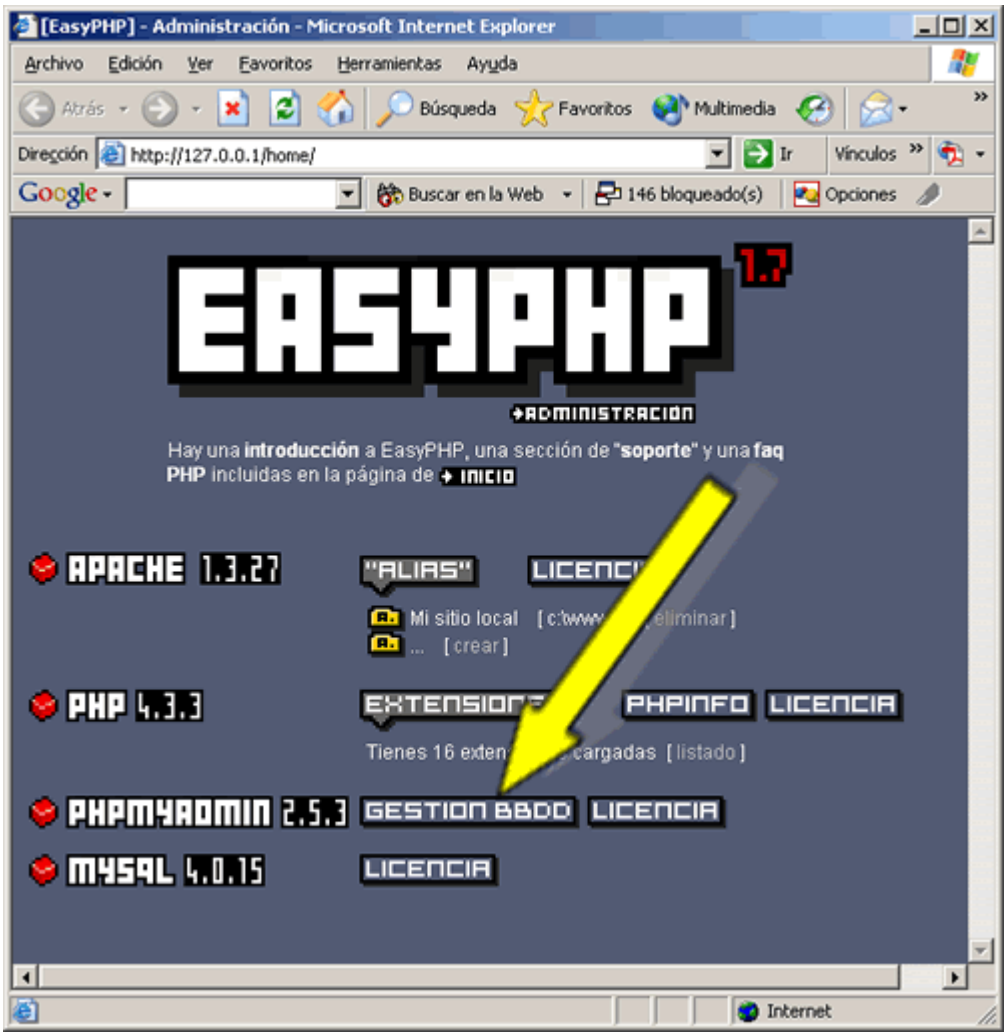
Recuerda que EasyPHP integra un sistema de gestión de bases de datos (MySQL) muy usado en Internet. Estos sistemas soportan los elementos habituales de las bases de datos: bases de datos formadas a su vez por tablas que se estructuran mediante campos y que guardan la información en los correspondientes registros.

No obstante, en los sistemas de gestión de bases de datos de cierta embergadura se definen también los usuarios que van a crear, modificar o consultar los datos existentes, habilitándose los permisos necesarios para ello.

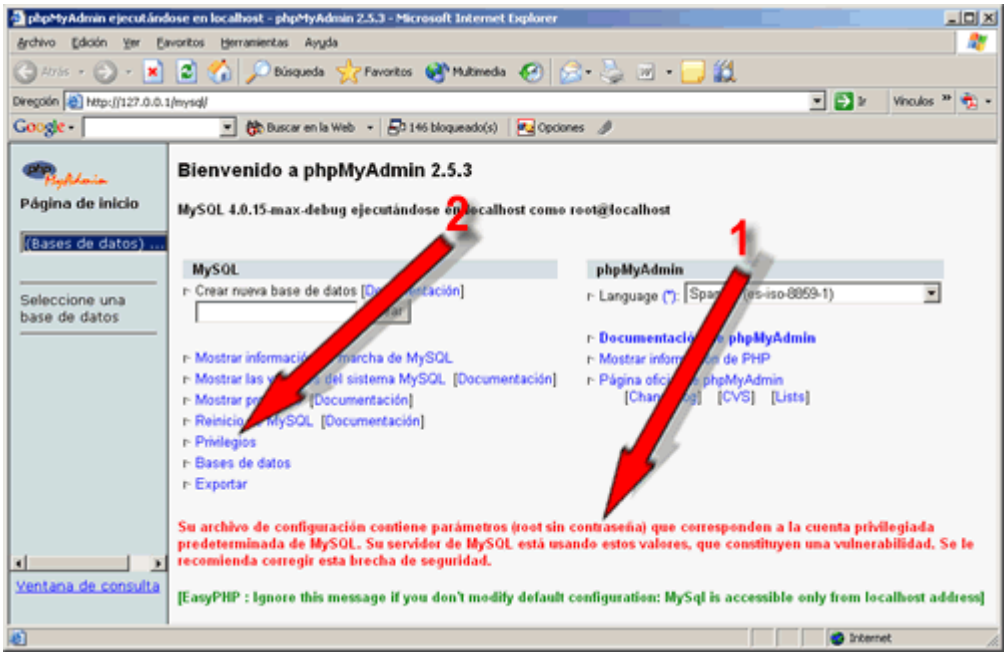
	Cuando EasyPHP instala el sistema MySQL de gestión de bases de datos, crea un único usuario llamado root que posee todos los privilegios sobre todas las bases de datos que se creen en el sistema. Luego parece razonable que antes de nada, establezcamos una contraseña para el usuario root ya que, de no hacerlo, nuestros datos estarían en peligro ante cualquier visita inesperada.

La forma más habitual que existe en la actualidad para mantener una base de datos del tipo MySQL es usar un gestor llamado **PHPMyAdmin** mediante el que, a través de cualquier navegador, podremos realizar todas las funciones necesarias de mantenimiento de nuestros datos.

Para ello accederemos a la opción de **Administración** de EasyPHP y seleccionamos **GESTIÓN BBDD** :



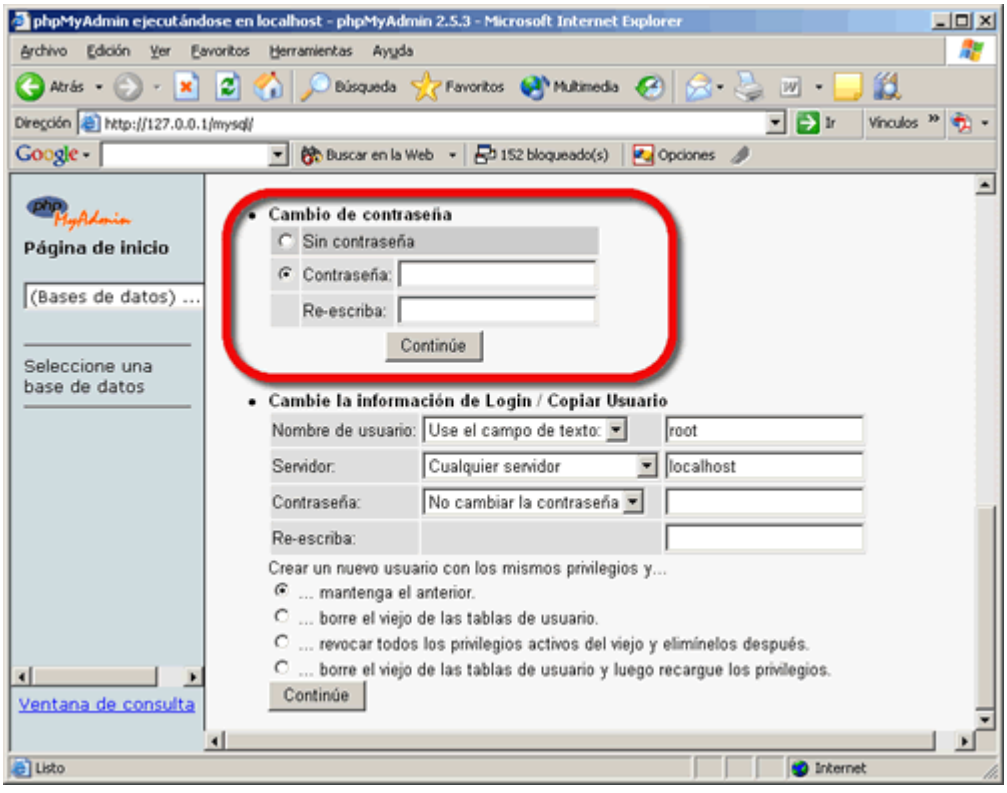
Esto nos dará entrada al PHPMyAdmin:



Observa que en la etiqueta 1 también te advierten de que el usuario root no dispone de contraseña.

Para asignarle una debes pulsar sobre la indicación 2.

Ahora debes Editar la información del usuario root para asignarle una contraseña:

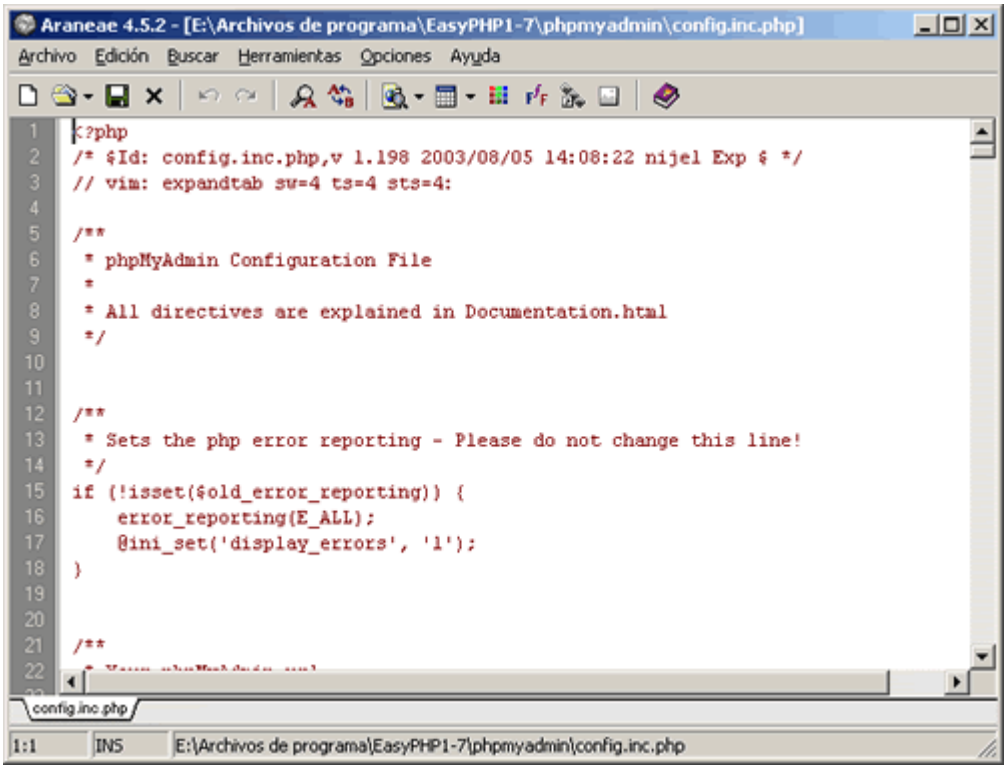


Después de pulsar sobre **Continúe** la contraseña quedará fijada en la Base de Datos MySQL.

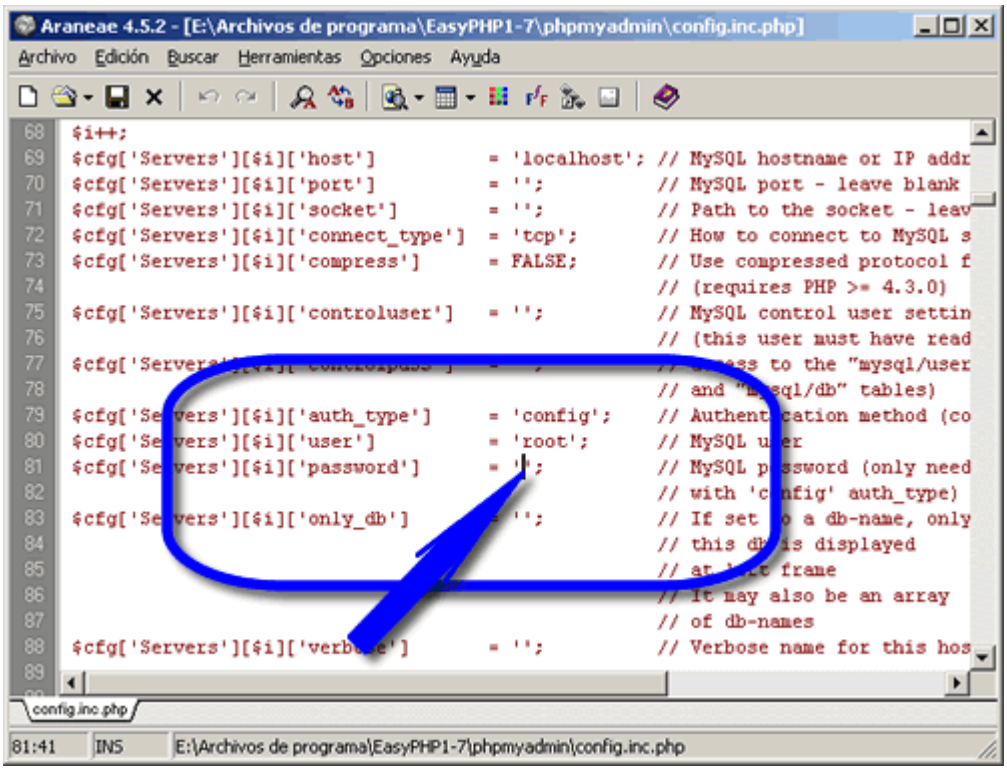
No obstante, esta información de nueva contraseña debe ser proporcionada al gestor PHPMyAdmin. Realmente la hemos cambiado en la base de datos MySQL, pero tendremos que comunicar esta nueva situación al gestor PHPMyAdmin para que, en la próxima sesión, pueda comunicarse con las bases de datos.

Para ello, mediante Areneae o cualquier otro editor de páginas tendremos que abrir el archivo:

c:\Archivos de Programa\EasyPHP1-7\phpmyadmin\config.inc.php



A lo largo del documento, encontrarás **tres** lugares donde aparece el texto:



Pues simplemente en los tres lugares donde se recoge la información del usuario root, donde se señala con la flecha tendrás que escribir entre las comillas la nueva contraseña que pusiste anteriormente.

Una vez hecho, guardas el documento, reinicias Apache y ya tienes todo protegido.

Ejemplo de desarrollo.

- Introducción
- PHP
- MySQ
- PHP con MySQL

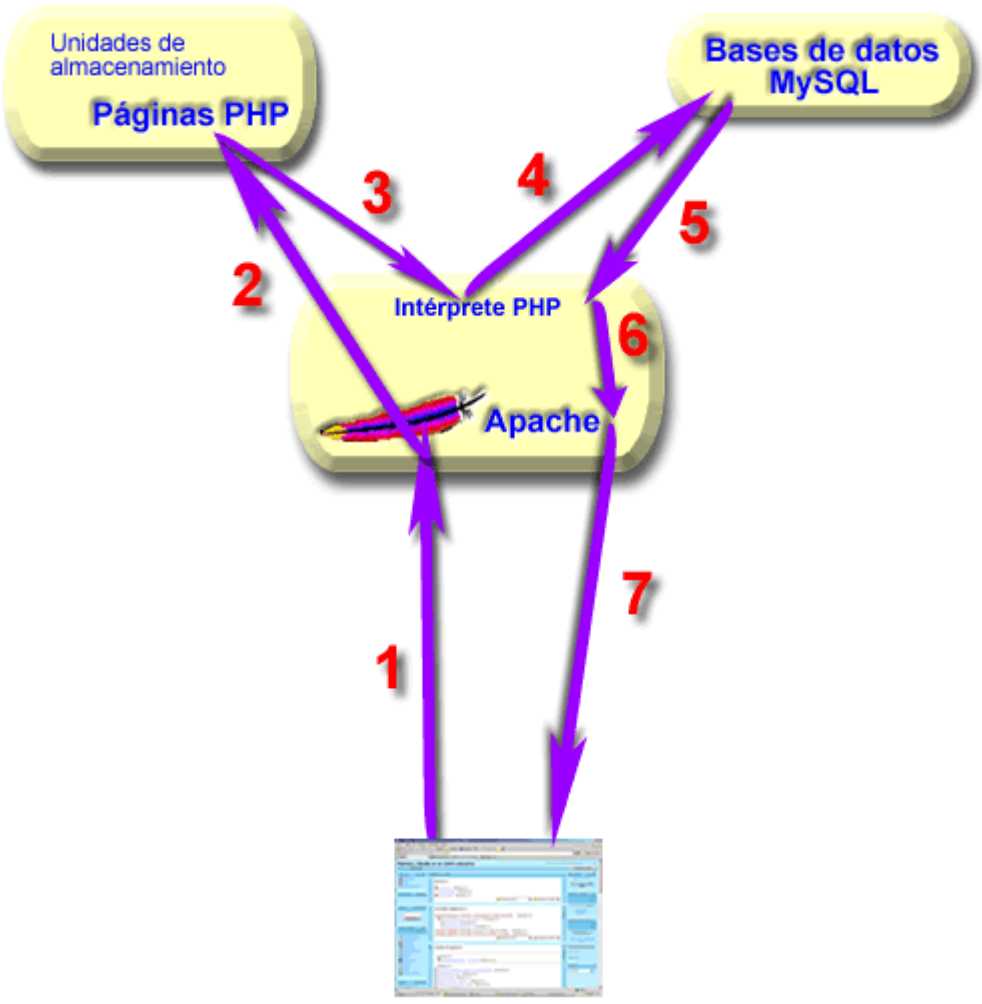
Introducción.

Ya sabemos que PHPNuke es un sistema de gestión de portales en Internet.

Moodle es un sistema de gestión de cursos y grupos en Internet.

Y ambos tienen en común, que están desarrollados en PHP y se apoyan en sistemas de gestión de bases de datos.

Por supuesto que nuestro objetivo no es aprender ni PHP ni gestión remota de bases de datos, pero no vendría mal desarrollar un pequeño ejemplo con el objetivo de comprender un poco mejor qué papel juegan cada uno de los actores en esta película (Apache, PHP MySQL, HTML, etc)



1. Nuestro cliente (navegador) realiza una petición de visualización de una página Web al servidor Apache.
2. El servidor Apache busca en sus unidades de almacenamiento la página solicitada. Si es una página HTML normal, la devuelve directamente al navegador que es el encargado de interpretarla y mostrarla correctamente.
3. Si es una página que contienen código PHP, es llevada a un interprete del código PHP que se encargará de "traducir" el mismo a código HTML que el navegador sea capaz de visualizar.
4. En muchos casos, el código PHP fuerza a que parte de los datos de la página Web sean incorporados desde una base de datos. Por ejemplo, podemos encontrar una instrución en PHP que indique al intérprete la necesidad de incorporar a la página todos los alumnos del grupo 1ESO-A. En estos casos, esta petición es realizada al gestor de bases de datos (MySQL, PostgreSQL, etc).
5. A continuación el gestor de BD devuelve los datos pedidos al intérprete de código PHP que los "incrusta" en el lugar correspondiente de la página.
6. Una vez confeccionada la página es devuelta a Apache.
7. Por último, Apache envia la página construida al cliente que le realizó la petición.

PHP.

Creando mi primera página PHP

Una vez instalado el programa EasyPHP, se creó una carpeta en la ruta **C:\Archivos de programa\EasyPHP1-7\www** donde se almacenarían todas las páginas que el servidor debe gestionar.

Del mismo modo comenté la posibilidad de crear "Alias", que eran otras carpetas que el servidor Apache podía gestionar.

Nuestro primer ejercicio será la creación de una página PHP y su posterior visualización.

En primer lugar crearemos una carpeta llamada **pruebas** dentro de la carpeta **C:\Archivos de programa\EasyPHP1-7\www**

Ahora debes usar cualquier editor de páginas Web que tengas, o incluso un editor de texto. Si no tienes ninguno, en el CD del curso llevas un editor de páginas Web de licencia libre: **ARANEAE**

En un archivo nuevo en blanco incluirás el siguiente código:

```
<HTML>
<HEAD>
<TITLE>Mi página PHP</TITLE>
</HEAD>
<BODY>

<?
echo 'HOLA MUNDO';
?>

</BODY>
</HTML>
```

Repito que no es objeto de este curso aprender ni HTML ni PHP pero con algún pequeño ejemplo podremos comprender cómo funciona:

- **<HTML>** Indica al navegador que el código que se encuentra a continuación es HTML.
- **<HEAD>** Sirve para concretar algunas características generales de la página:
- **<TITLE>** Es la cápsula encargada de recoger el texto que se mostrará en la parte superior de la ventana del navegador.
- **<?** Indica que el código que viene a continuación es PHP.

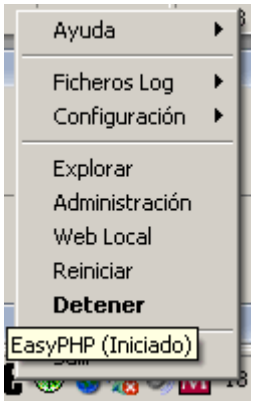
- **echo** es una instrucción PHP cuya misión es simplemente mostrar en el navegador el texto que aparece entrecomillado.
- **?>** es para indicar que se acabó el código PHP.

Ahora tan solo te queda guardar dentro de la carpeta **C:\Archivos de programa\EasyPHP1-7\www\pruebas** el archivo que has creado, con el nombre **hola.php**

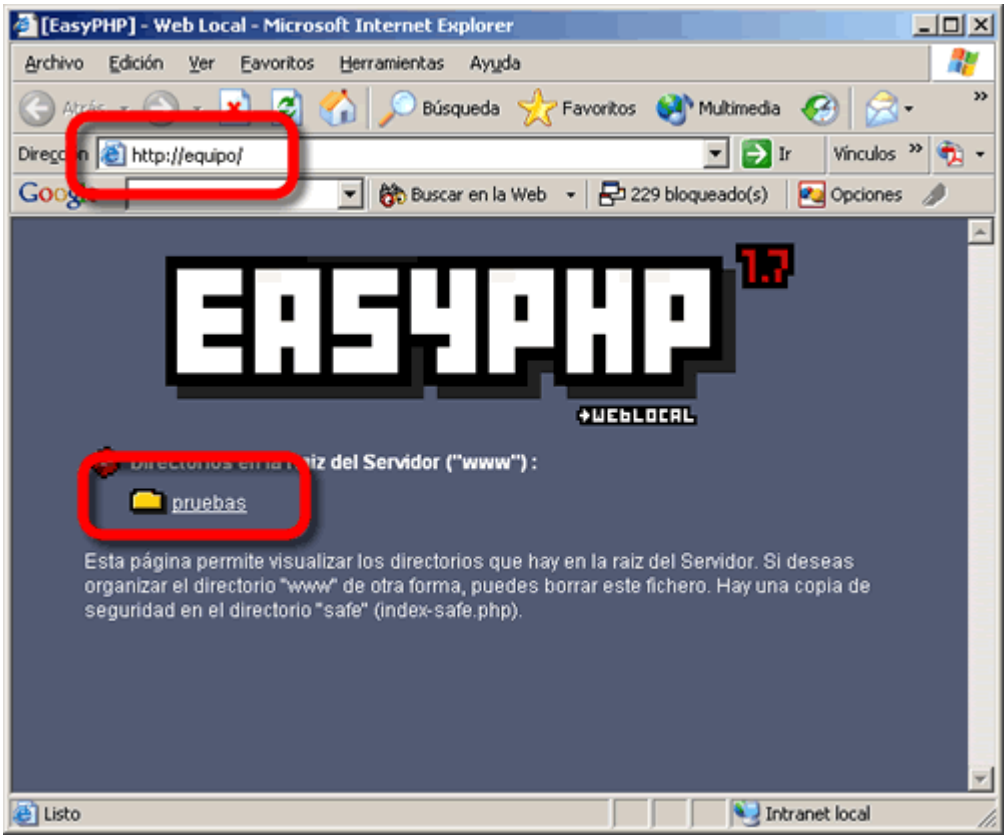
Comprobando que funciona

Para ello, lo primero que debemos hacer es arrancar EasyPHP, **Inicio/Todos los Programas/EasyPHP 1.7/EasyPHP**

Ahora debes comprobar que el servidor Apache está iniciado:



Ahora sobre el icono del programa, pulsando con el botón derecho del ratón seleccionarás **Web Local**, lo que debe mostrarnos la página de inicio de la carpeta www:



Observa que en el navegador aparece el nombre que Apache ha asignado a nuestro equipo: **http://equipo/**

Por defecto, EasyPHP asigna a nuestra máquina local una IP de funcionamiento que siempre es **127.0.0.1**

Luego, también podremos acceder a nuestro sitio local si, en la barra de direcciones del navegador escribimos **http://127.0.0.1/**

Cuando instalaste EasyPHP, el programa instaló en la carpeta www un archivo llamado

index.php

que es el que realmente estás visualizando. Luego si en la barra de direcciones de tu equipo, en vez de escribir

http://equipo/

escribes

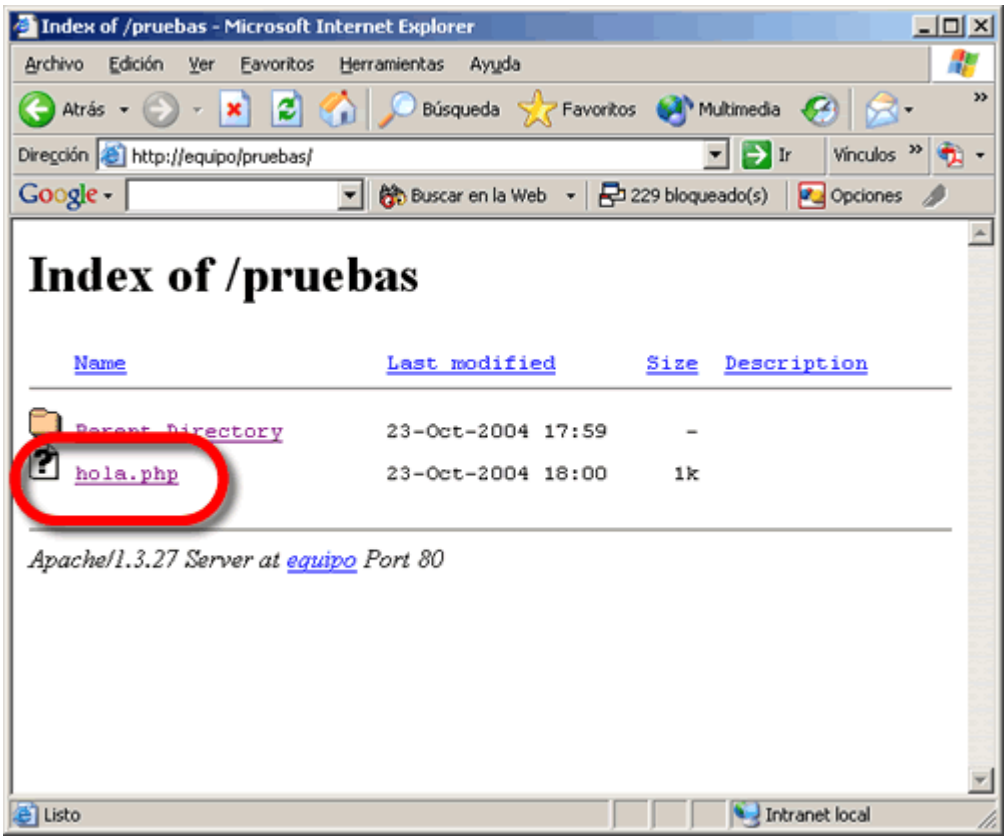
http://equipo/index.php

comprobarás que el efecto es el mismo. Esto ocurre porque cuando le pedimos a Apache que muestre el contenido de una carpeta en vez de un archivo, lo primero que hace es buscar si en su interior existe algún archivo con nombre

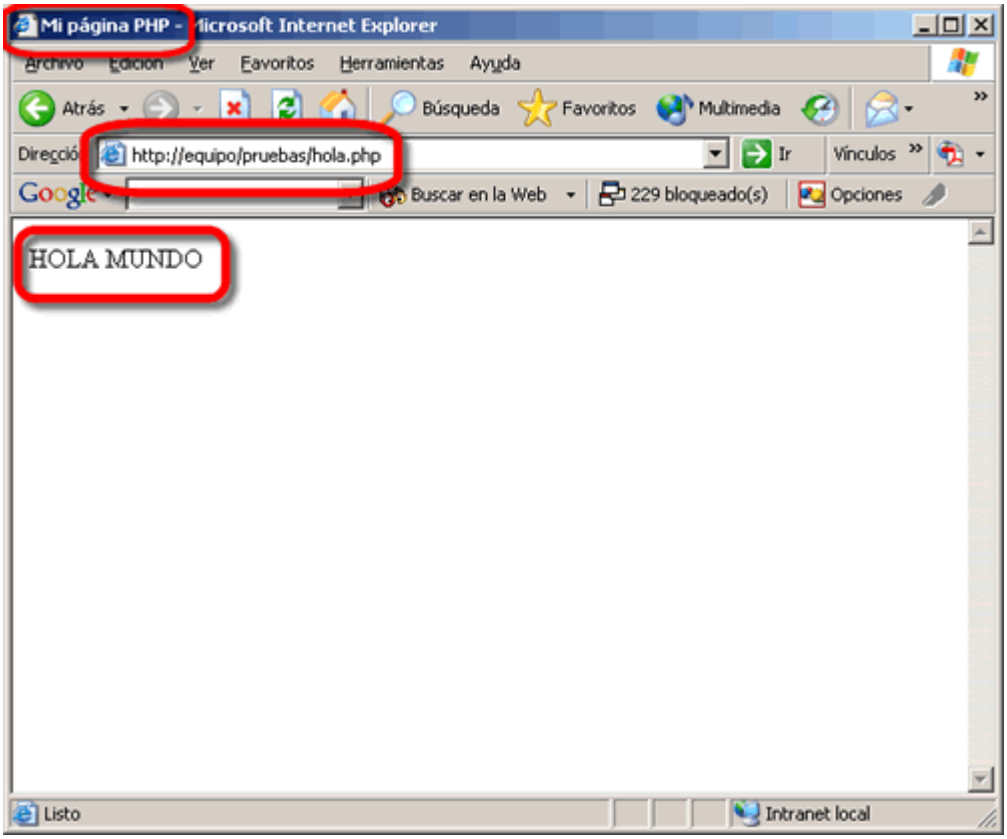
index.htm, index.html o index.php fundamentalmente.

Si lo encuentra, lo envia al navegador y si no lo encuentra puede mostrar o no el contenido completo de la carpeta en función de la configuración que tenga.

Observa que la página index.php está creada de tal manera que te muestra el contenido de la carpeta raiz www, y por tanto ha mostrado la carpeta pruebas. Si ahoras haces clic sobre el enlace a la carpeta, Apache buscará algún archivo llamado index, y como de momento no existe ninguno, lo que hace es mostrar el contenido de la carpeta prueba (ya que por defecto el programa configura así el servidor):



observa que aparece el archivo hola.php que creamos anteriormente; si pulsamos sobre él se nos mostrará la página PHP:



Ahora no estaría mal que

probaras que, con Apache desactivado, si intentas abrir desde el navegador la página hola.php no podrás visualizarla.

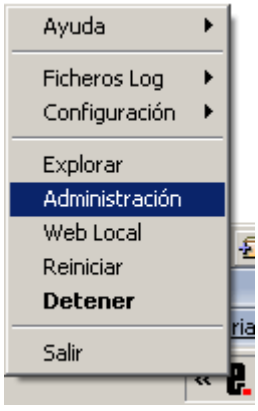
Con esta comprobación es cuando realmente demostramos que el navegador por si solo no es capaz de "entender" páginas en PHP, sino que necesita del servidor Apache acompañado del correspondiente interprete de PHP para poder mostrar su contenido final.

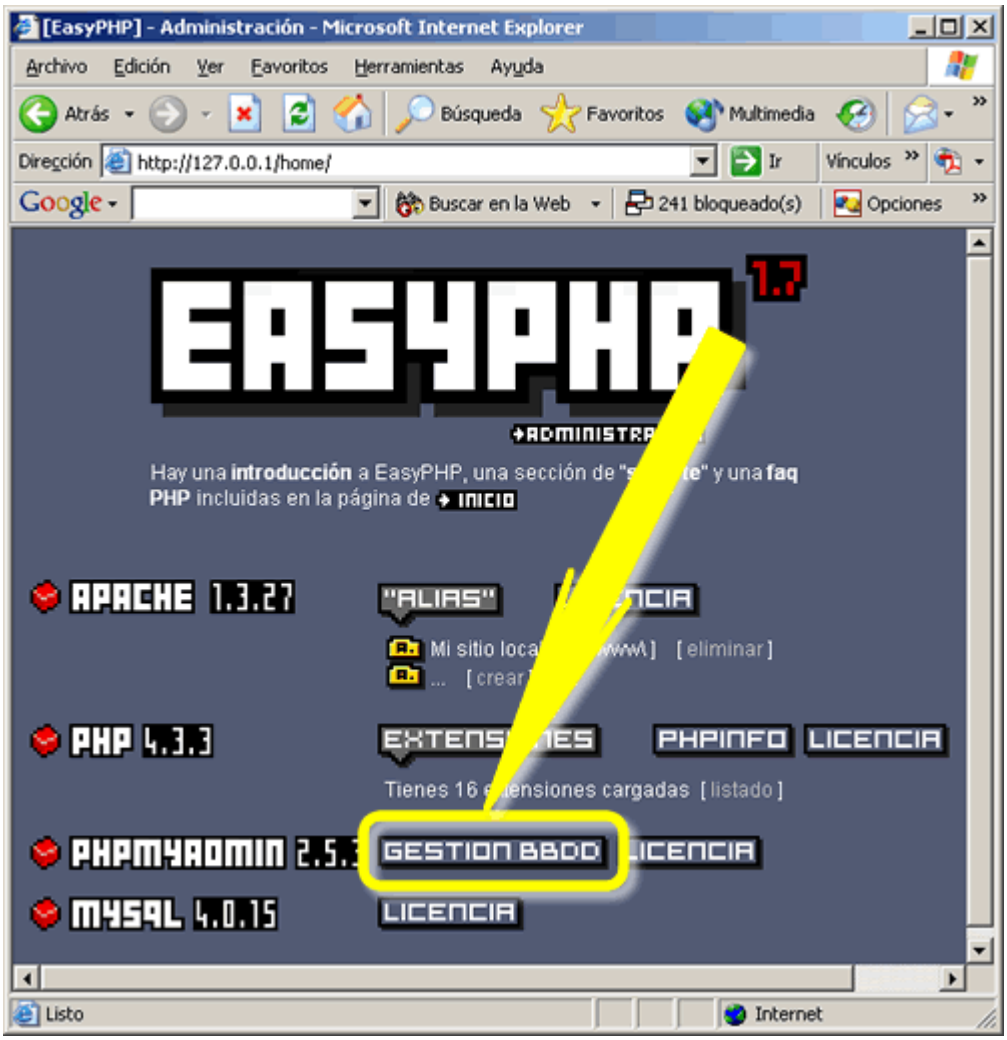
MySQL.

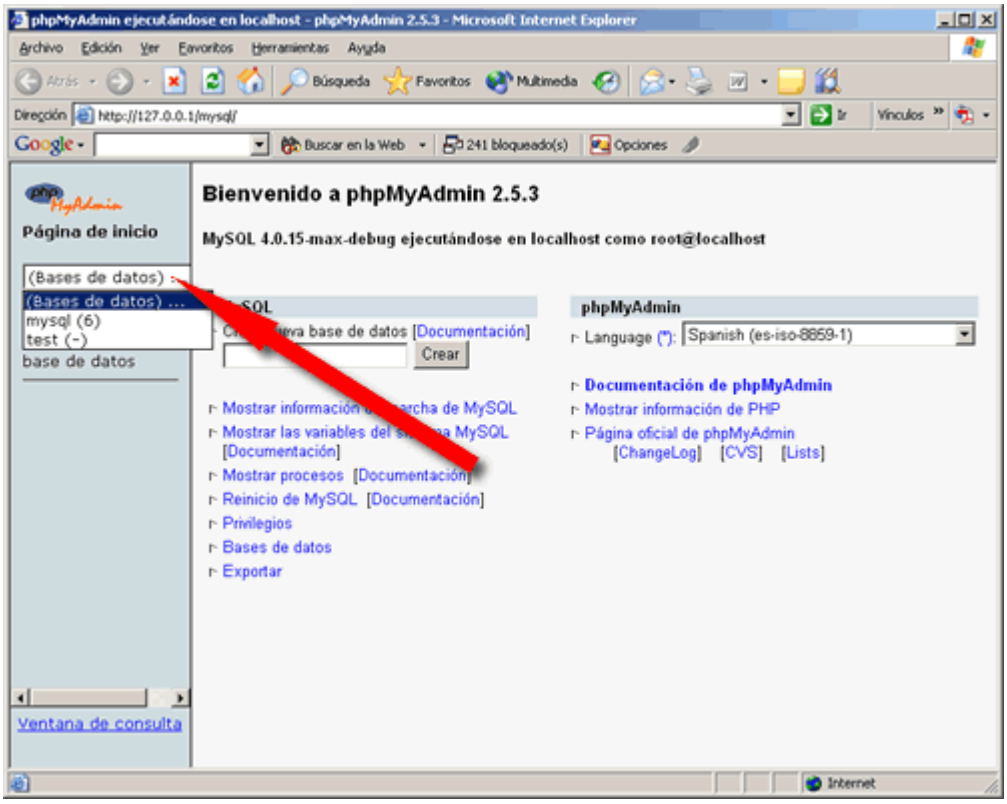
MySQl es el gestor de bases de datos incluidas en el programa EasyPHP. Paga gestionar de manera más fácil los datos se usa un gestor llamado PHPMyAdmin, que nos permite mantener nuestras bases de datos mediante formularios que aparecen en nuestro navegador.

Ahora, mediante un ejemplo, veamos cual es la mecánica general de funcionamiento de estos gestores de información. Tampoco es objetivo de este curso el manejo de bases de datos, pero creo necesario el desarrollo de este ejemplo para un mejor aprovechamiento de estas útiles herramientas.

Para acceder, con el EasyPHP arrancado, seleccionamos **Administración**.



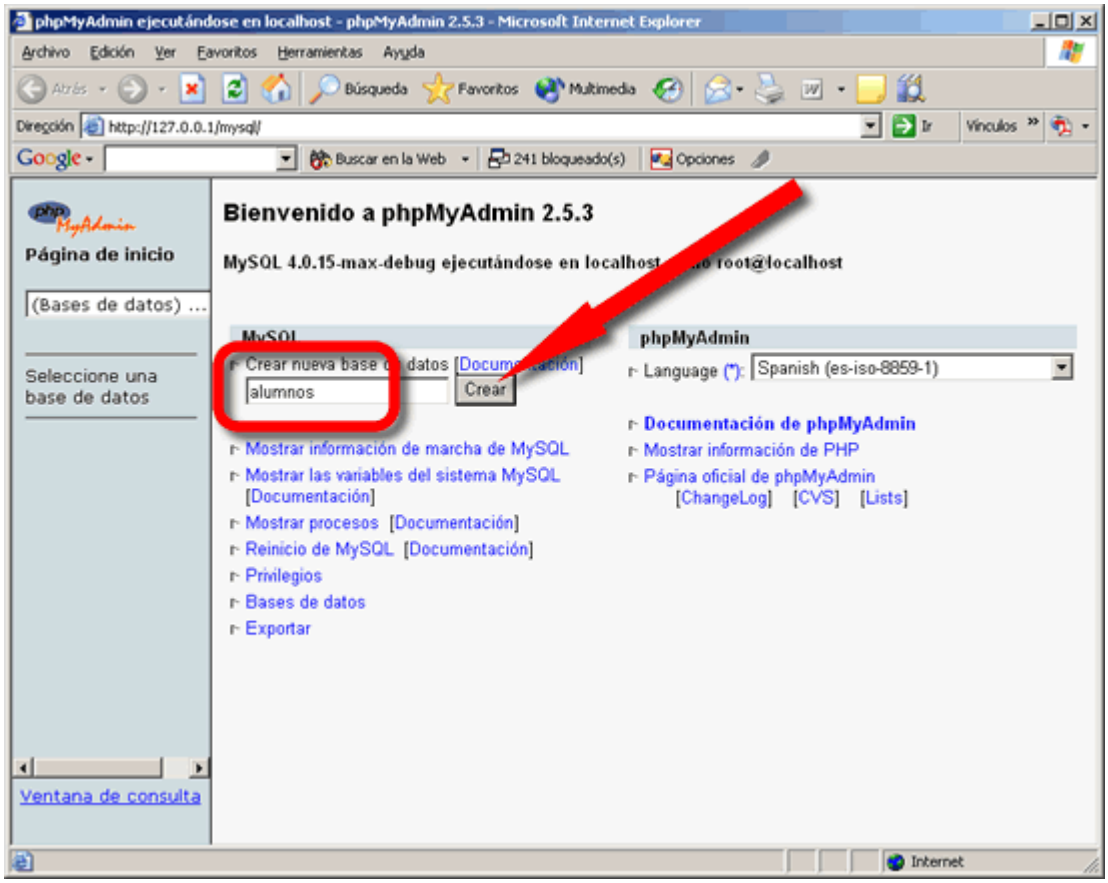




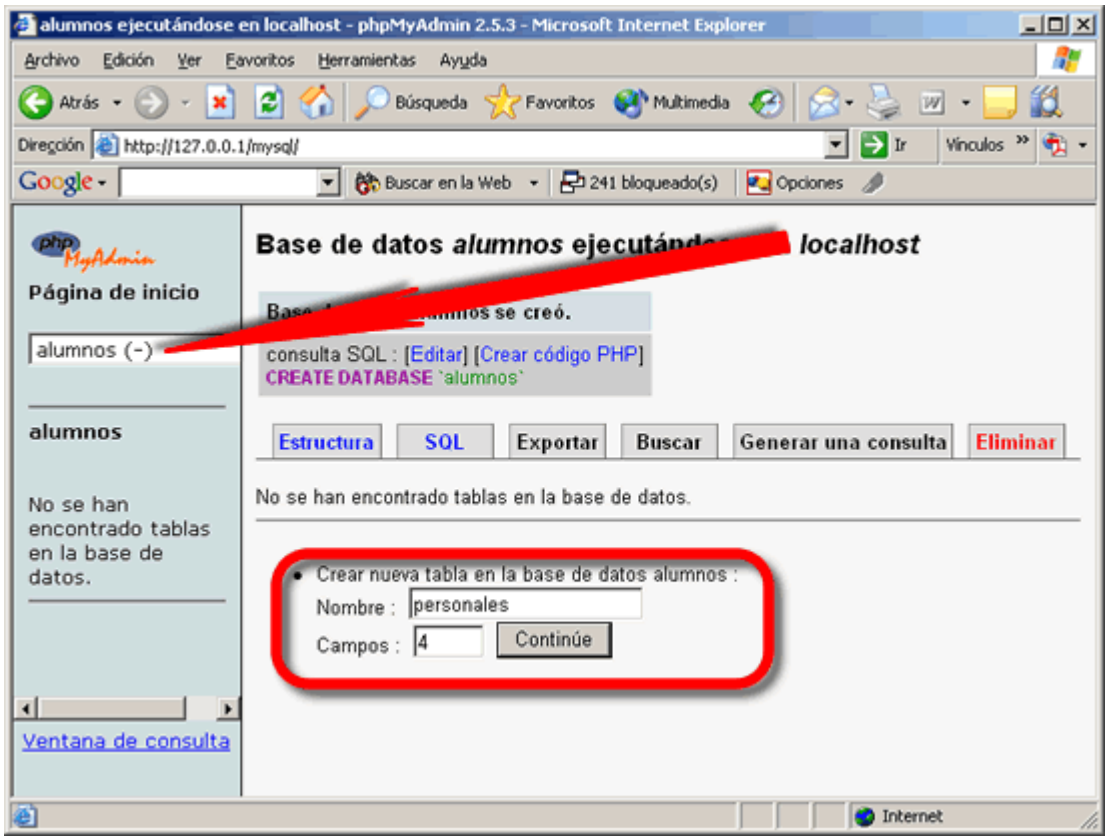
Pulsando sobre **(Bases de datos)** se os despliega una ventanita que muestra las bases de datos con la

que contamos: mysql con 6 tablas que es donde propiedad del sistema; además tienes otra test sin ninguna tabla para pruebas.

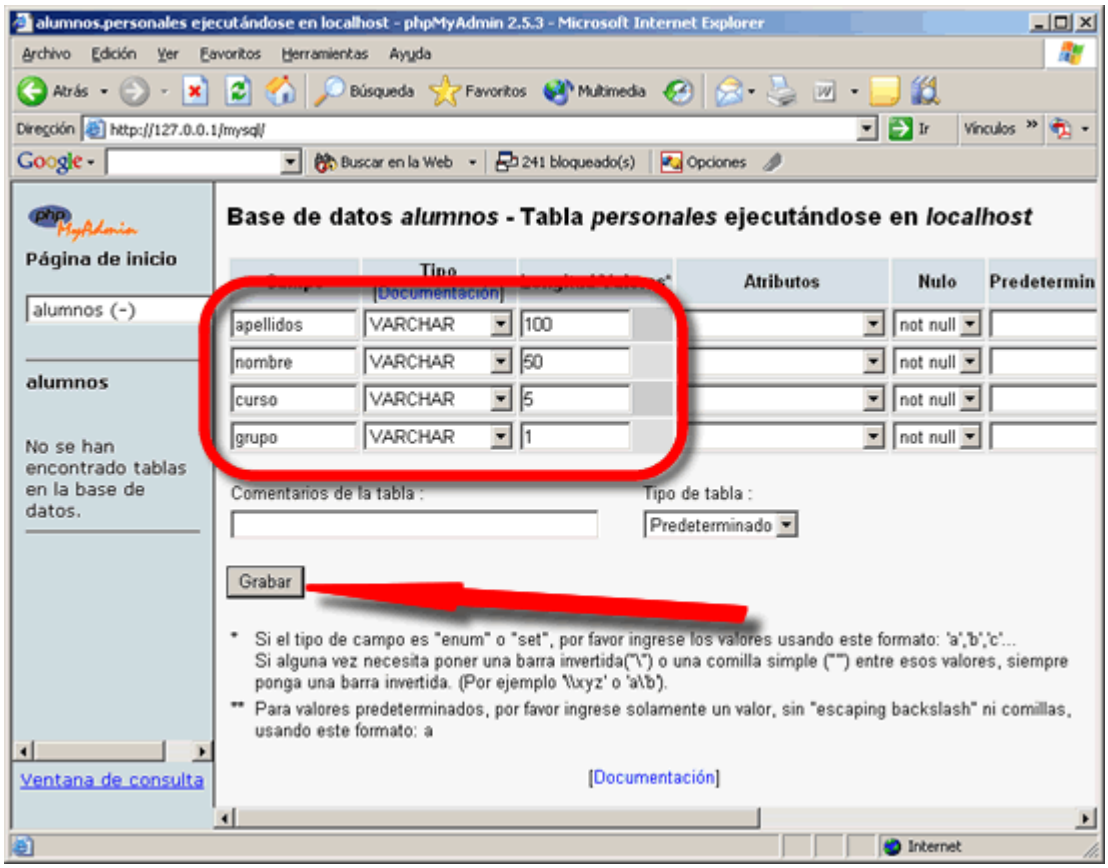
Ahora podemos crear una propia llamada **alumnos**:



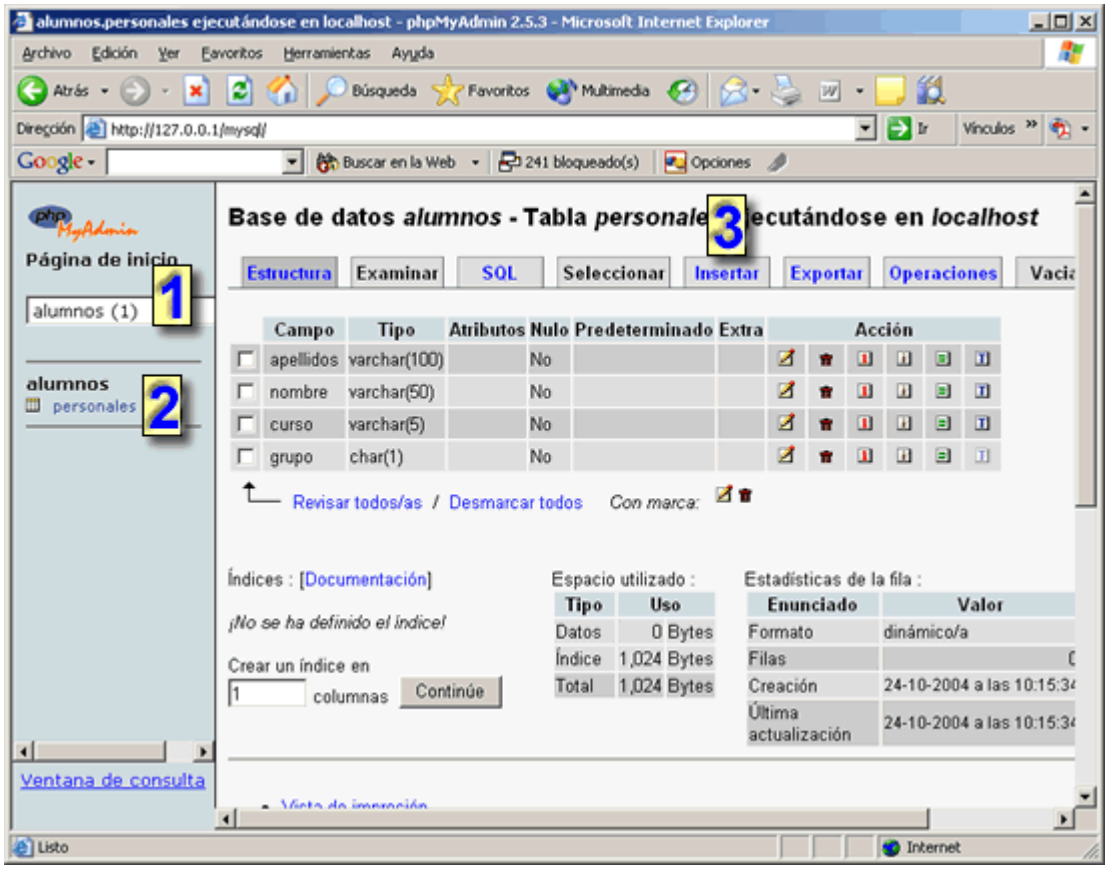
Ahora en la ventana de **(Bases de datos)...** ya debe aparecer la nuestra:



Creamos ahora así, una nueva tabla llamada **personales** con cuatro campos (apellidos, nombre, curso, grupo) a modo de ejemplo. A continuación nos pedirá información adicional como es el nombre de los campos, tipos, etc. Y como no es objetivo nuestro, tan solo nos limitaremos a dar de alta los datos correspondientes:



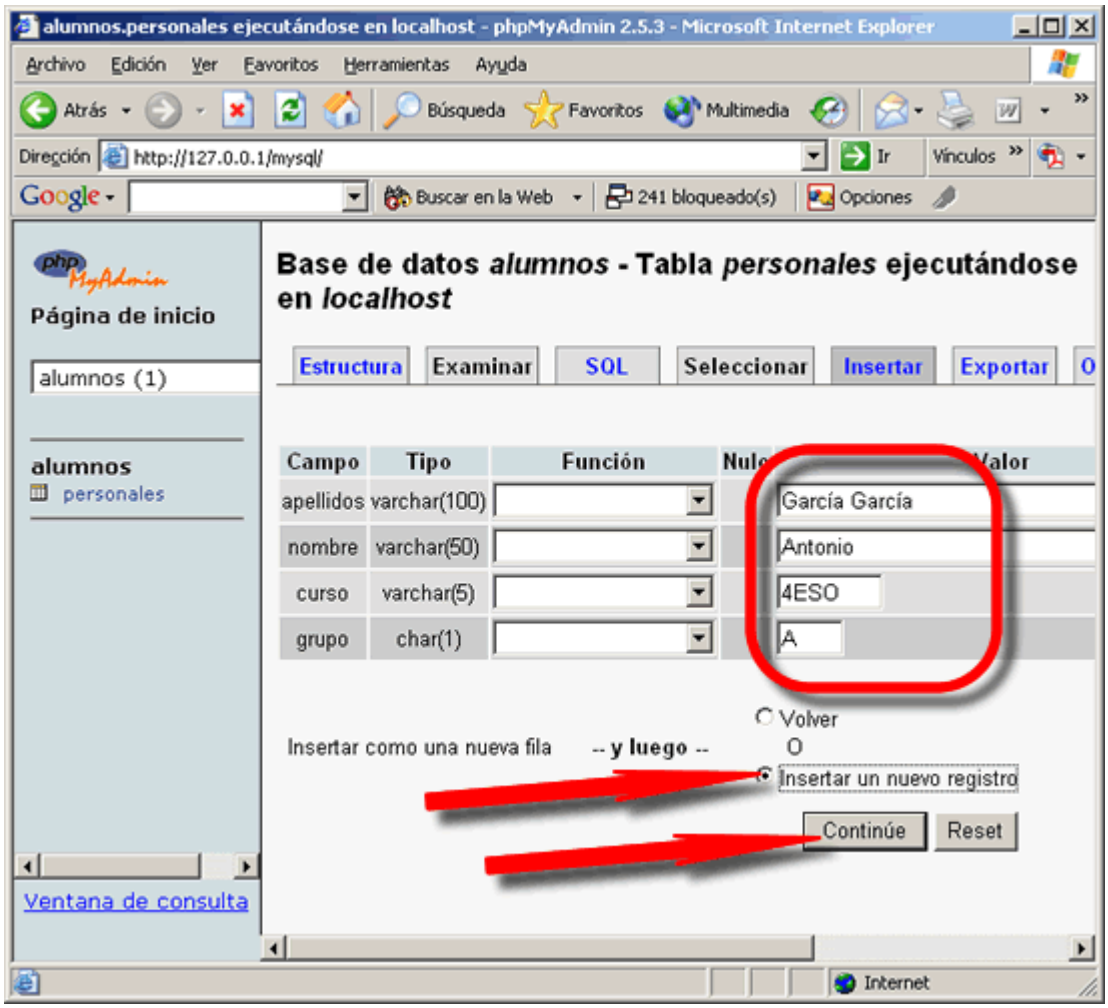
Una vez creada nuestra nueva tabla, la base de datos queda:



En la posición 1 puedes observar que la etiqueta de la BD alumnos ya aparece como **alumnos (1)** indicando que posee una tabla.

En la posición 2, se te mostrarán todas las tablas de la BD y pulsando sobre ellas podrás accede a las mismas para editar su estructura, añadir datos, modificarlos, etc.

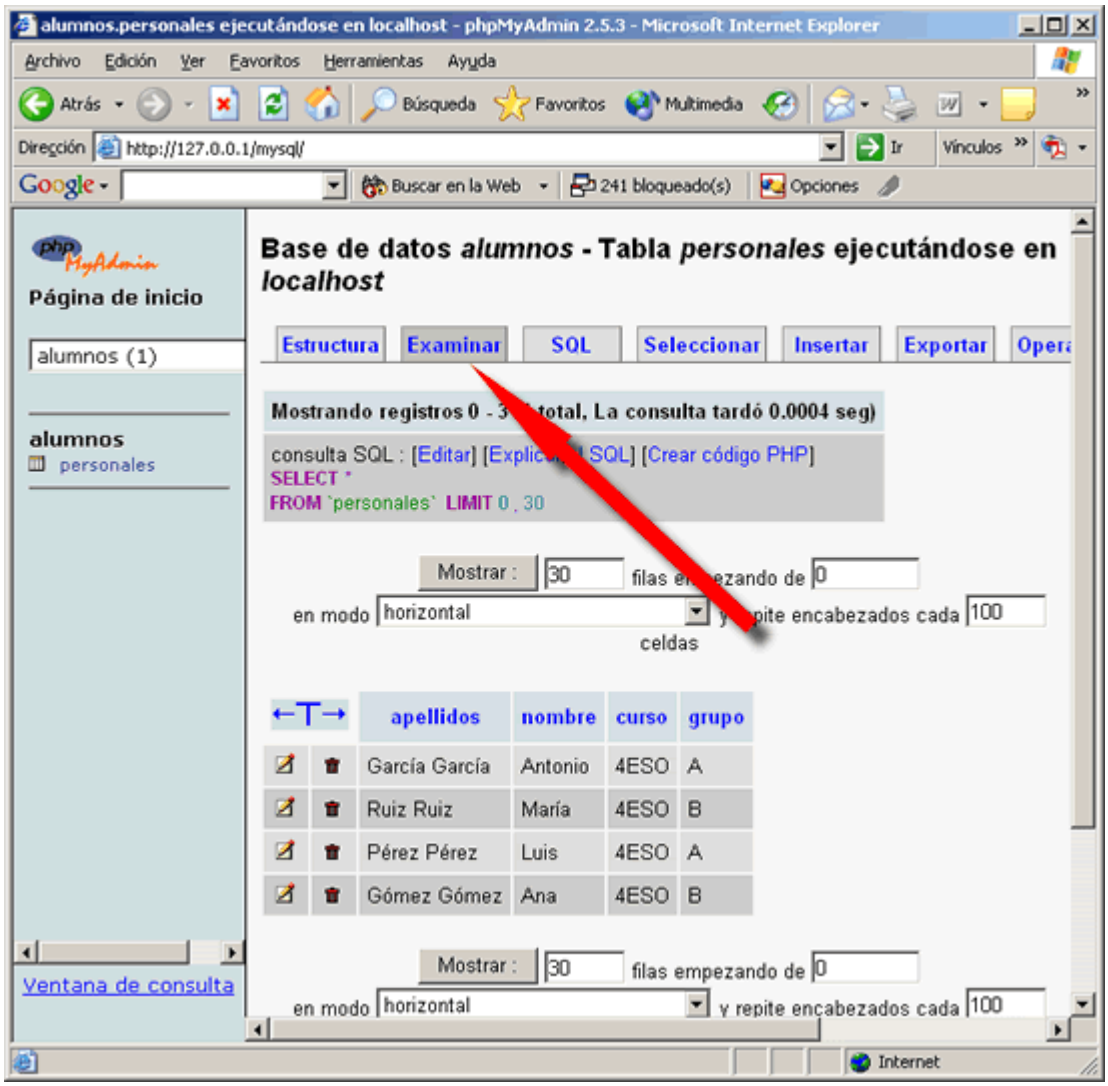
El número 3 te indica el lugar donde podrás insertar datos en tu tabla:



Mediante este procedimiento vamos a añadir 4 registros nuevos a nuestra tabla:

apellidos	nombre	curso	grupo
García García	Antonio	4ESO	A
Ruiz Ruiz	María	4ESO	B
Pérez Pérez	Luis	4ESO	A
Gómez Gómez	Ana	4ESO	B

Ahora desde la pestaña examinar podrás acceder a los datos introducidos:



PHP con MySQL.

Hasta el momento hemos conseguido visualizar una página PHP en nuestro equipo, a continuación hemos creado una base de datos propia con algunos datos de ejemplo.

Ahora vamos a desarrollar un pequeño ejemplo en el que visualicemos una página PHP que conecta con la base MySQL para solicitar los datos necesarios.

Para ello con el editor Araneae creamos una página nueva que guardaremos en la carpeta pruebas con el nombre de alumnos.php, y cuyo contenido sea:

Recuerda cambiar "tu_contraseña" por la que introduciste como root en MySQL.

```
<HTML>
<HEAD>
<TITLE>Otro ejemplo</TITLE>
</HEAD>
<BODY>

LISTADO DE ALUMNOS <BR>

<?
```

```
$sql="select * from personales";
mysql_connect("127.0.0.1","root","tu_contraseña");
$rs=mysql_db_query("alumnos",$sql);
while ($row = mysql_fetch_array($rs)){
echo $row["apellidos"].", ".$row["nombre"]." - ".$row["curso"]." -
".$row["grupo"]."<br>";
}
?>

</BODY>
</HTML>
```

Veamos un poco el significado del código:

```
<BODY>

LISTADO DE ALUMNOS <BR>
```

Aquí hemos escrito una cabecera para el listado;
 produce un salto de línea.

```
<?
$sql="select * from personales";
mysql_connect("127.0.0.1","root","tu_contraseña");
```

Con **<?** iniciamos el código PHP.

En **$sql** establecemos la consulta que queremos hacer a nuestra base de datos.

Mediante **mysql_connect("127.0.0.1","root","tu_contraseña");** establecemos la conexión con nuestra base de datos.

Recuerda cambiar "tu_contraseña" por la que introduciste como root en MySQL.

```
$rs=mysql_db_query("alumnos",$sql);
```

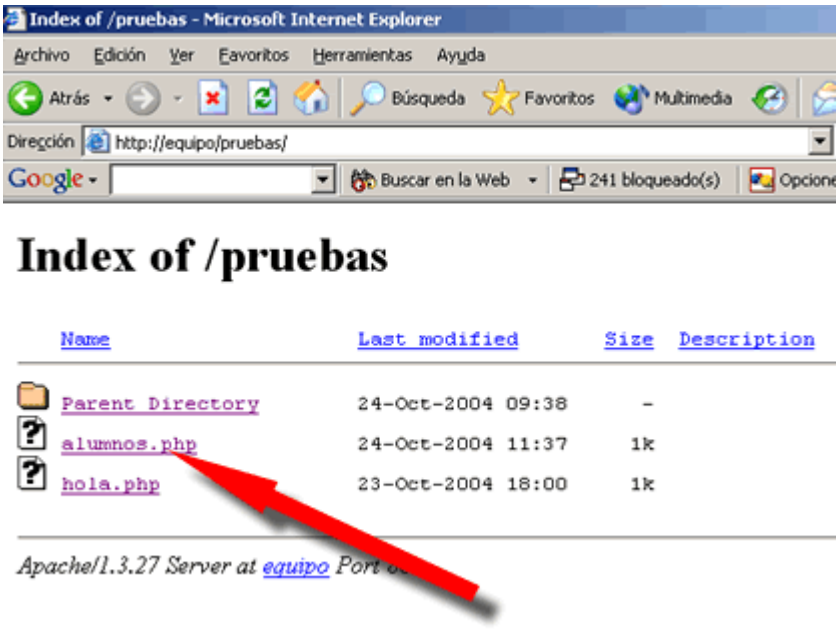
Ahora le indicamos a MySQL que queremos hacer en la BD alumnos la consulta $sql.

```
while ($row = mysql_fetch_array($rs)){
echo $row["apellidos"].", ".$row["nombre"]." - ".$row["curso"]." - ".$row["grupo"]."<br>";
}
```

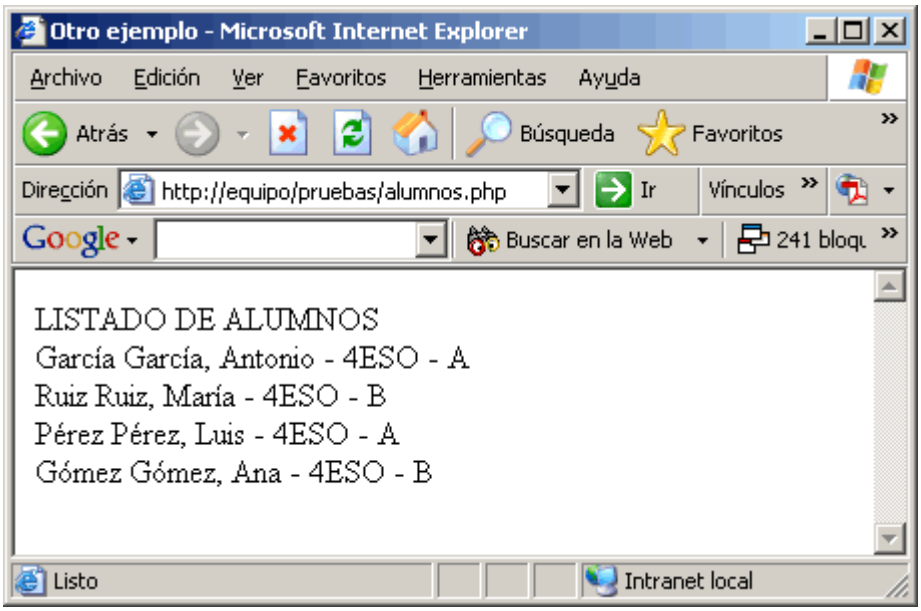
mostramos los datos.

Ahora comprobemos que funciona:

Accede a la Web local del EasyPHP, y entra en pruebas:



Y "picando" sobre alumnos.php tendríamos ... **Guauu!**



Copias de seguridad de MySQL

Una vez creada nuestra base de datos y enlazada desde PHP, podemos permitir a los visitantes de nuestro sitio que actualicen, añadan o incluso borren registros de la misma.

Por tanto suele ser habitual que aparezca la necesidad de realizar una copia de seguridad de nuestros datos en un archivo externo.

Copia de seguridad.

1. En nuestro caso usaremos el PHPMyAdmin. Para ello iniciamos **EasyPHP** y accedemos a la

opción **Administración**, y accedemos a **Gestión BBDD.**
2. Seleccionamos la Base de Datos **alumnos.**
3. Hacemos clic sobre la pestaña **Exportar.**
4. Las opciones marcadas por defecto son adecuadas; no obstante, para optimizar el tamaño de nuestros ficheros de copias de seguridad resulta bueno marcar la casilla **"comprimido con gzip",** de esta forma optimizamos el tamaño de nuestro archivo.
5. Pulsamos sobre **continue**.
6. Y se nos muestra la ventana para **Guardar** el archivo de copia de seguridad.

Restauración de datos.

Cuando en alguna ocasión necesites recuperar los datos de una copia de seguridad procederemos de la siguiente forma:

1. Vamos a crear una nueva base de datos llamada **prueba**, donde cargaremos los datos de la copia de seguridad anterior.
2. Para ello, desde PHPMyAdmin creamos la nueva base de datos llamada **prueba**.
3. A continuación, con la base de datos seleccionada pulsamos sobre la pestaña **SQL**.
4. Mediante **Examinar...** buscamos el archivo de copia de seguridad.
5. Pulsamos sobre **Continúe**, y ya está.