



Curso de programación PHP

Jorge García Ochoa de Aspuru
-aka Bardok-

bardok@telefonica.net - shadow@bardok.net
<http://www.bardok.net>

Julio 2005 – e-ghost
ESIDE – Universidad de Deusto



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/2.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

© 2004/2005 – Jorge García (Bardok)

e-ghost – ESIDE – Universidad de Deusto



Índice

- Introducción
- HTML básico
- Configuración del servidor Apache
- Fundamentos de PHP
- PHP e información de usuario
- Bases de datos en PHP
 - Configuración del servidor de base de datos
 - PHP y MySQL



Introducción

- ¿Qué es PHP?
 - PHP: Hypertext Preprocessor
 - Lenguaje de scripting
 - Sintaxis similar a C, Java o Perl
 - Propósito general
 - Utilizado sobre todo para generación de páginas dinámicas HTML
 - Incrustado en páginas HTML
 - Open Source



Introducción

- Historia de PHP
 - PHP/FI: precursor de PHP (Rasmus Lerdorf, 1995)
 - Scripts en Perl para registrar accesos a su web
 - Se empezó a extender, hasta tener acceso a bases de datos y generación de páginas web dinámicas
 - En 1997 se liberó la versión PHP/FI 2.0
 - PHP 3: la primera versión similar al PHP actual (1998)
 - Proyecto universitario



Introducción

- PHP 4: se reescribe el núcleo (llamado Zend Engine)
 - Más modular
 - Más rendimiento
 - Más funcionalidad
 - Versión actualmente más utilizada de PHP
 - El curso se centra en ésta versión
- PHP 5:
 - Zend Engine 2.0
 - Gestión de clases mejorada
 - Muchas utilidades y mejoras en el API
 - Gestión de Bases de Datos mejorada



Introducción

- ¿Cómo funciona?
 - Nosotros vemos páginas web en un navegador:
 - Mozilla, Konqueror, Opera, Nautilus...
 - Un navegador interpreta HTML
 - (y hojas de estilo, Javascript, etc, pero eso está fuera del alcance de este curso ;-)
 - Un navegador NO sabe interpretar PHP
 - Entonces, ¿qué ocurre con el PHP?



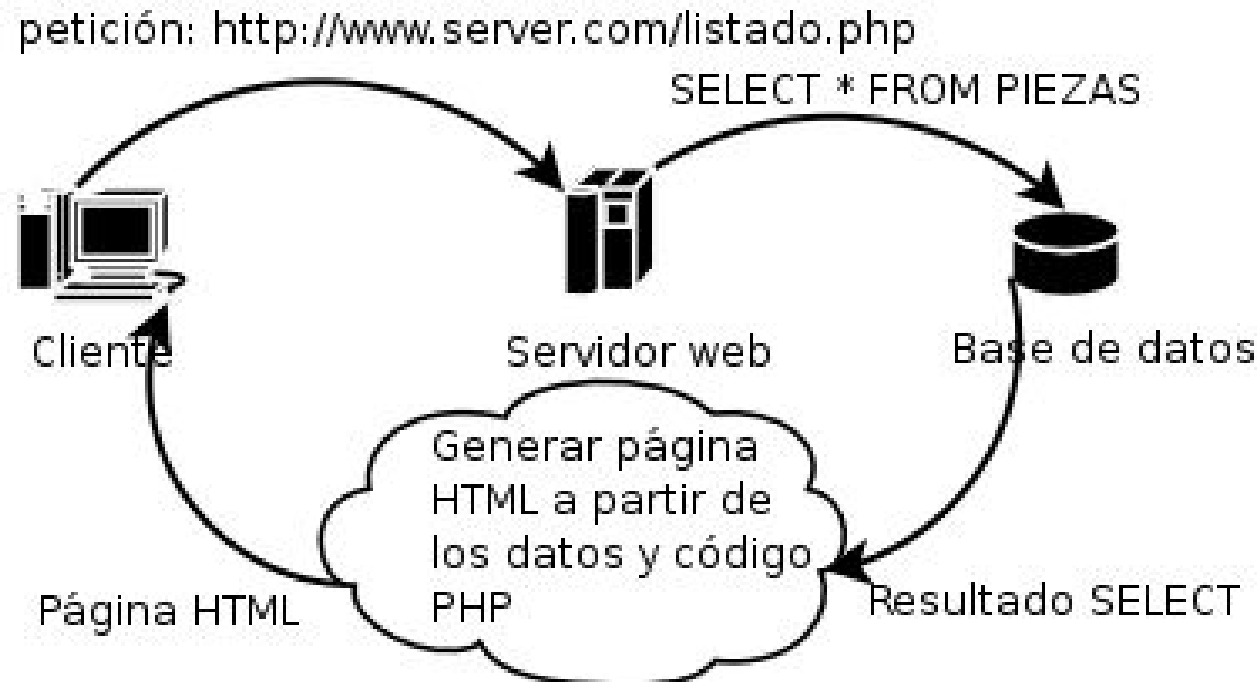
Introducción

- El navegador puede:
 - Mostrar una página alojada en el ordenador en el que él está instalado
 - Éste es el ordenador “cliente”
 - La página se muestra tal y como está guardada, no hay proceso posible: tiene que ser HTML
 - Pedir una página a un servidor
 - El servidor puede procesar la página antes de enviarla al navegador



Introducción

- Ejemplo: petición de una página PHP a un servidor web



HTML básico

- HTML es un lenguaje de etiquetas utilizado para crear páginas web
 - En nuestro caso utilizaremos XHTML
 - Sintaxis XML
 - Etiqueta con inicio y fin:
 - `<etiqueta>Contenido</etiqueta>`
 - Etiqueta sin contenido:
 - `<etiqueta />`
 - Las etiquetas de inicio (o las etiquetas sin contenido) pueden tener parámetros:
 - `<etiqueta param1="valor1" para2="valor2">`
 - Toda etiqueta de inicio debe tener su fin
 - Pueden anidarse



HTML básico

- Una página web básica:

```
<html>  
  <head>  
    <title>Título en la barra del navegador</title>  
  </head>  
  <body>  
    Contenido de la página  
  </body>  
</html>
```

- Etiqueta “html”: raíz del documento
- Sección “head”: información de la página
- Sección “body”: contenido de la página



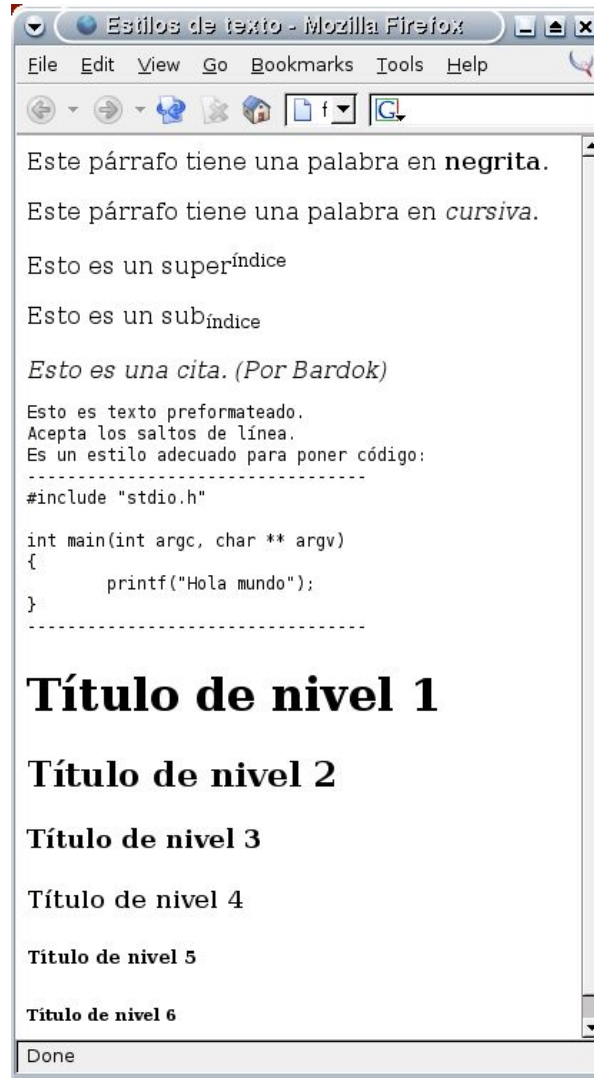
HTML básico

- Formateando textos...
 - Si insertamos un texto dentro de una sección, tendrá diferentes formatos
 - `<p>texto</p>`: párrafo de texto
 - `<i>texto</i>`: cursiva
 - `texto`: negrita
 - `<h[1..6]>texto</h[1..6]>`: estilos de título
 - `^{texto}`: superíndice
 - `_{texto}`: subíndice
 - `<pre>texto</pre>`: texto preformateado
 - Más estilos:
http://www.w3schools.com/html/html_formatting.asp
 - Ejercicio: `estilos.html`



HTML básico

- Estilos.html



HTML básico

- Imágenes
 - Etiqueta ``
 - El atributo “src” indica dónde se encuentra la imagen
 - Importante: utilizar paths relativos
 - Los atributos “width” y “height” permiten poner altura y anchura a la imagen
 - Ejemplo: imagen.html



HTML básico

- imagen.html
(para centrar la imagen, etiqueta <center>)



HTML básico

- Enlaces

- Nos permiten movernos entre páginas
- Su estructura es:

`texto, imagen, etc.`

- El destino debería tener el path:

- Absoluto: se encuentra en otro servidor

`http://www.google.es`

- Relativo al dominio: se encuentra en otro directorio virtual, dentro del servidor

`/noticias/lunes.html`

- Relativo al directorio de trabajo: se encuentra en el mismo directorio que la página actual

`pagina.html`



HTML básico

- Ejemplo: enlaces.html



HTML básico

- Listas

- Permiten enumerar elementos

- Pueden ser ordenadas (con la etiqueta ``) o desordenadas (``)
 - Los elementos van delimitados por la etiqueta ``
 - Ejemplo

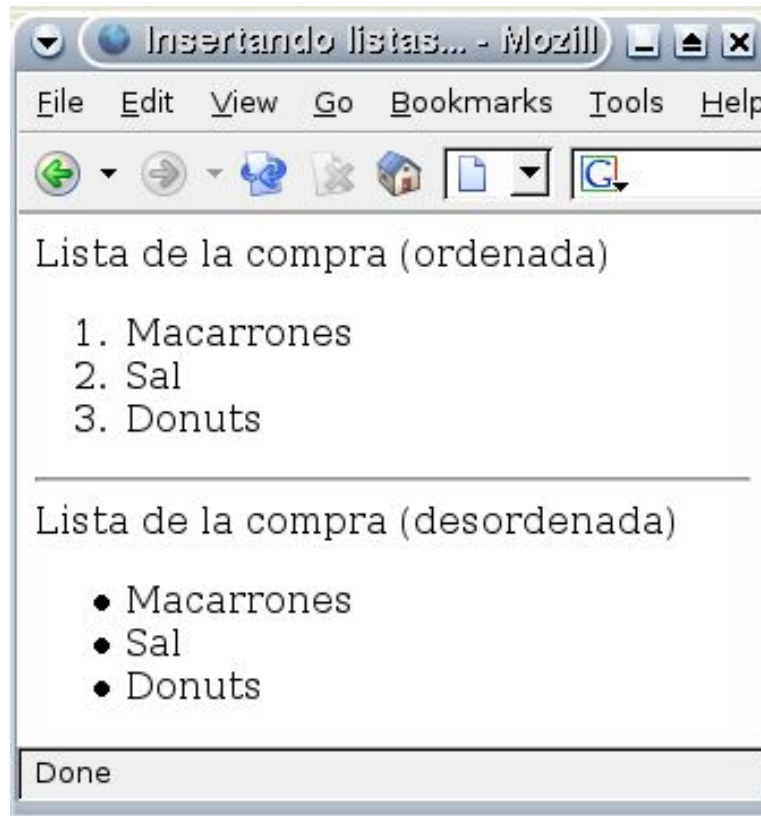
Lista de la compra:

```
<ul>
  <li>Macarrones</li>
  <li>Sal</li>
  <li>Donuts</li>
</ul>
```



HTML básico

- Ejemplo: listas.html



HTML básico

- Tablas
 - Permiten organizar elementos
 - Delimitadas por `<table>`
 - Se definen por filas (`<tr>`) y columnas (`<td>`)
 - Una tabla tiene filas, y cada fila, columnas
 - Todas las filas tienen el mismo número de celdas
 - Podemos hacer que una celda de una fila ocupe el espacio de varias celdas...
 - ... o que una celda esté en varias filas



HTML básico

- Una tabla sencilla

```
<table>
  <tr>
    <td>Celda 1</td>
    <td>Celda 2</td>
    <td>Celda 3</td>
  </tr>
  <tr>
    <td>Celda 4</td>
    <td>Celda 5</td>
  </tr>
  ...
</table>
```

Celda 1	Celda 2	Celda 3
Celda 4	Celda 5	Celda 6
Celda 7	Celda 8	Celda 9



HTML básico

- Juntando dos celdas (misma fila)

```
<table>
  <tr>
    <td>Celda 1</td>
    <td>Celda 2</td>
    <td>Celda 3</td>
  </tr>
  <tr>
    <td>Celda 4</td>
    <td colspan="2">Celda 5</td>
  </tr>
  ...
</table>
```

Celda 1	Celda 2	Celda 3
Celda 4	Celda 5	
Celda 7	Celda 8	Celda 9



HTML básico

- Juntando dos celdas (misma columna)

```
<table>
  <tr>
    <td>Celda 1 </td>
    <td rowspan="2">Celda 2 </td>
    <td>Celda 3 </td>
  </tr>
  <tr>
    <td>Celda 4 </td>
    <td>Celda 5 </td>
  </tr>
  ...
</table>
```

Celda 1	Celda 2	Celda 3
Celda 4	Celda 2	Celda 5
Celda 7	Celda 8	Celda 9



HTML básico

- Ejemplo: tablas.html

los bordes están puestos con el atributo:

```
<table border="1">
```



HTML básico

- Formularios

- Permiten introducir información

- Estructura:

- ```
<form method="modo de envío" action="página de recepción">
```

- controles del formulario:

- Cajas de texto...

- Cajas para password...

- Memos...

- Checkboxes...

- Listas desplegadas...

- ```
</form>
```

- Dentro de un formulario puede haber código HTML normal



HTML básico

- Página de recepción:

- Indica el elemento que va a procesar la información recibida
 - Lo normal, como veremos, es que sea una página PHP, aunque puede ser cualquier cosa
 - Ej: `<form action="procesar.php" method=...>`

- Modo envío:

- Permite especificar cómo se envía la información del formulario al servidor. Hay dos modos:
 - “get”: la información se envía en la URL
 - Ej: `procesar.php?login=bardok&pass=12345`
 - “post”: la información se envía oculta
 - Útil para enviar, por ejemplo, passwords



HTML básico

- Controles para el formulario

- Atributo “name”

- La información se enviará con ese nombre al servidor
- Es como si fuese el nombre de una variable

- Atributo “value”

- Opcional
- Indica un valor por defecto

- La mayoría se definen a través de la etiqueta “input”

- `<input type="tipo del control" name="nombre" [value="valor por defecto"] />`
- El “tipo de control” indica qué tipo de control estamos insertando



HTML básico

- Cuadro de texto
 - Recuadro para introducir una línea de texto
 - Ej: `<input type="text" name="login" />`
- Cuadro para introducir passwords
 - Recuadro que permite introducir una línea de texto oculto
 - Ej: `<input type="password" name="pass" />`
- Checkbox:
 - Permite seleccionar una opción
 - Ej: `<input type="checkbox" name="recordar" />`
- Botón de envío:
 - Permite enviar los datos del formulario al servidor
 - Ej: `<input type="submit" name="envio" value="Enviar" />`



HTML básico

- Botón de borrado:
 - Permite borrar el contenido de los controles, y volver a poner sus valores por defecto
 - `<input type="reset" name="borrado" value="Borrar" />`
- Ejemplo de los controles vistos hasta ahora:
 - formulario_sencillo.html



Formulario sencillo

File Edit View Go Bookmarks Tools

Login:

Password:

Recordar datos



HTML básico

- Listas desplegables

- Se definen con la etiqueta “select”
- También pueden tener un nombre (name)
- Las opciones se definen con la etiqueta “option”

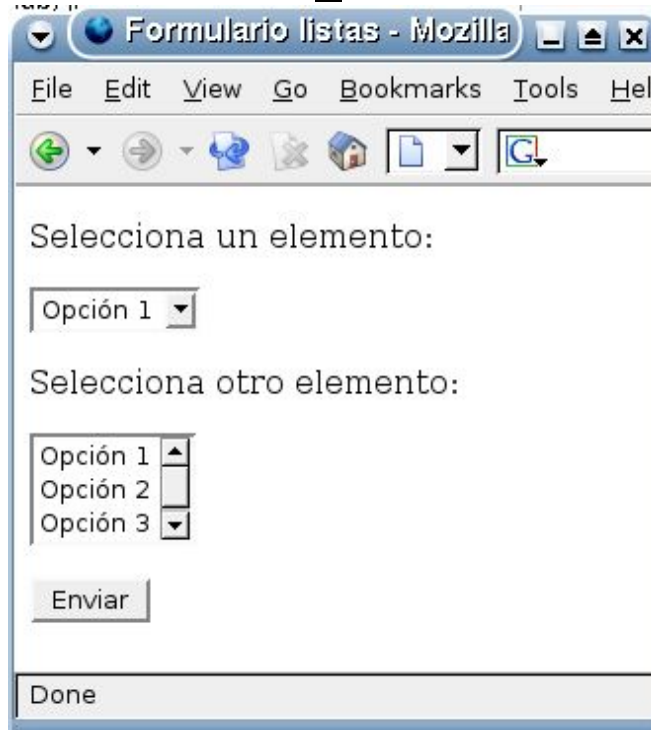
```
<select name="lista">  
  <option value="1">Opción 1</option>  
  <option value="2">Opción 2</option>  
  <option value="3">Opción 3</option>  
</select>
```

- El valor enviado será el “value” de la opción seleccionada
- Un control “select” puede tener un atributo “size” que indica el número de opciones visibles a la vez



HTML básico

- Ejemplo: formulario_listas.html



- Campos ocultos:
 - Permiten guardar valores sin que se muestren por pantalla

```
<input type="hidden" name="selected" value="3" />
```





Configuración del servidor Apache

- Vamos a utilizar el servidor Apache
 - <http://www.apache.org>
 - Servidor HTTP Open Source
 - En principio, soporta el protocolo HTTP
 - Se pide un fichero, el servidor lo devuelve
 - Extensible a través de módulos, como el módulo PHP
 - Si está cargado y configurado, en caso de que el fichero pedido sea una página PHP, se procesa y lo que se envía es el resultado





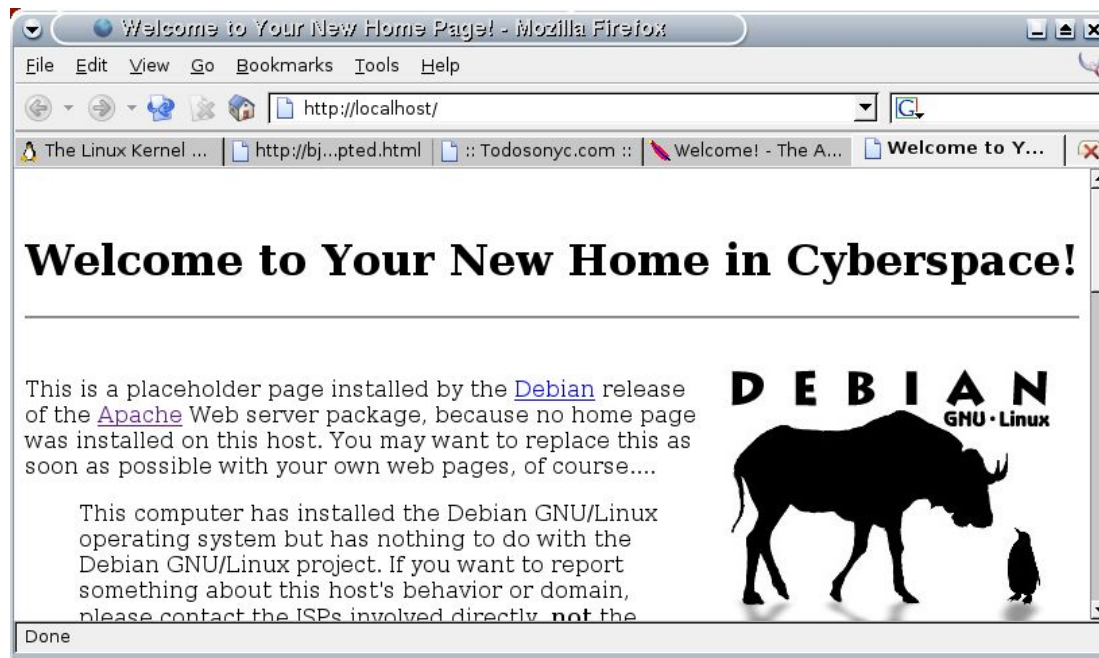
Configuración del servidor Apache

- El proceso descrito aquí es para configurar el servidor en una distribución Debian GNU/Linux
 - No debería variar sustancialmente para otras distribuciones
- Instalar los paquetes php4, apache, apache-common y apache utils
 - apt-get install php4 apache
 - Versiones (julio 2004):
 - Apache: 1.3.29
 - PHP: 4.3.4
 - Actualización de última hora: acaba de salir PHP 5



Configuración del servidor Apache

- Para comprobar que el servidor Apache está activo:
 - <http://localhost>
 - Aparecerá la siguiente página
 - Accedemos a través del servidor, no directamente





Configuración del servidor Apache

- Activamos el módulo PHP:
 - Editar el fichero “/etc/apache/httpd.conf”
 - Descomentar las líneas
 - AddType application/x-httpd-php .php
 - AddType application/x-httpd-php-source .phps
 - Reiniciar el servidor Apache:
 - `#/etc/init.d/apache restart`



Configuración del servidor Apache

- Finalmente, crearemos un directorio virtual en el servidor:
 - Accederemos a nuestras páginas en “http://localhost/cursophp”
 - Guardaremos nuestras páginas en “/home/usuario/cursophp”
- Para ello, editamos el fichero “/etc/apache/”
 - En la sección “Section 2” incluimos:

```
<IfModule mod_alias.c>
  Alias /cursophp/ /home/usuario/cursophp/
  <Directory /home/usuario/cursophp/>
    Options Indexes Includes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
  </Directory>
</IfModule>
```

- Reiniciamos el servidor



Configuración del servidor Apache

- Finalmente, comprobamos que funcionen las páginas PHP

- Creamos el fichero:

“/home/usuario/cursophp/holamundo.php”

```
<html>
```

```
  <body>
```

```
    <?php echo ("Hola mundo!"); ?>
```

```
  </body>
```

```
</html>
```

- Accedemos a él a través del servidor:

- <http://localhost/cursophp/holamundo.php>



Fundamentos de PHP

- Como se dijo en la introducción:
 - PHP se incrusta dentro del código HTML
 - PHP se procesa en el servidor
 - Sirve para crear páginas HTML que después se enviarán al cliente, y para realizar otras operaciones
 - Podemos generar una página según el contenido de un formulario...
 - ... o según el contenido de una base de datos...
 - ... o según la hora del servidor...
 - ...



Fundamentos de PHP

- ¿Cómo incrustamos PHP dentro del HTML?
 - Con la etiqueta `<?php [instrucciones] ?>`
 - El servidor procesa el código que hay entre “`<?php`” y “`?>`” (o simplemente “`<? ?>`”)
`<?php echo (“Hola mundo!”); ?>`
 - Con la etiqueta `<?=[expresión]?>`
 - El servidor evalúa la expresión, y escribe su resultado en la página web
`<?=”Hola mundo!” ?>`
 - No puede haber más código que la expresión que queremos evaluar y sacar por pantalla



Fundamentos de PHP

- ¿Cómo procesa el servidor el código?
 - Tenemos la página:

```
<html>
  <body>
    <?php echo ("Hola mundo!"); ?>
  </body>
</html>
```

- El servidor comienza a procesar la página
 - Si encuentra HTML, lo envía directamente al cliente
 - Si encuentra una etiqueta de PHP, procesa el código, y si el código escribe algo, esa “escritura” se incluye en el HTML que se envía al cliente



Fundamentos de PHP

- La página anterior enviaría:

```
<html>
  <body>
    Hola mundo!
  </body>
</html>
```

- El mismo resultado se conseguiría con:

```
<html>
  <body>
    <?="Hola mundo!" ?>
  </body>
</html>
```



Fundamentos de PHP

- Sintaxis

- La sintaxis es muy similar a la sintaxis de C o Java
 - Las instrucciones se escriben separadas por ';'.
 - Las estructuras de control (if, while, etc.) son muy similares
 - Las llamadas a función se realizan con paréntesis



Fundamentos de PHP

- Variables

- Las variables tienen dos diferencias con respecto a C:
 - Pueden cambiar de tipo fácilmente
 - Ej.: si en una variable hemos guardado un entero, podemos utilizarlo como un String, y viceversa
 - siempre que el String represente un entero
 - El nombre de la variable lleva '\$' delante
 - `$usuario = "pepe"`
- Es posible concatenar variables (strings) con el operador '.'

```
<?php
    $mensaje_1 = "Hola";
    $mensaje_2 = "mundo!";
    echo ($mensaje_1." ".$mensaje_2);
?>
```



Fundamentos de PHP

- Dentro de un string delimitado con comillas dobles se evalúan las variables

```
$nombre = "bardok";  
echo ("El valor de la variable nombre es $nombre.");
```
- Con la función "unset()" podemos quitar el valor de una variable

```
$var = "pepe";  
unset($var); // Ahora no tiene valor (NULL)
```
- La función "isset()" nos indica si una variable está inicializada

```
if (isset($var))  
{  
    echo ("Tiene valor");  
}
```



Fundamentos de PHP

- Operadores
 - Aritméticos

Operador	Símbolo	Ejemplo	Resultado
Suma	+	3 + 3	6
Resta	-	4 - 3	1
Multiplicación	*	4 * 3	12
División	/	6 / 2	3
Módulo (resto)	%	7 % 2	1
Incremento	++	\$x = 3; \$x++	4
Decremento	--	\$x = 3; \$x--	2

- Asignación

Operador	Símbolo	Equivalencia
Asignación	\$x = \$y	\$x = \$y
Asignación y suma	\$x += \$y	\$x = \$x + \$y
Asignación y resta	\$x -= \$y	\$x = \$x - \$y
Asignación y multiplicación	\$x *= \$y	\$x = \$x * \$y
Asignación y división	\$x /= \$y	\$x = \$x / \$y
Asignación y resto	\$x %= \$y	\$x = \$x % \$y





Fundamentos de PHP

- Comparación

Operador	Símbolo	Ejemplo	Resultado
Igual	==	3 == 3	cierto
Distinto	!=	4 != 3	cierto
Mayor	>	3 > 4	falso
Menor	<	3 < 4	cierto
Mayor o igual	>=	7 >= 2	cierto
Menor o igual	<=	3 <= 2	falso

- Lógicos

Operador	Símbolo	Ejemplo	Resultado
Y (and)	&&	cierto && falso	falso
O (or)		cierto falso	cierto
NO (not)	!	!cierto	falso



Fundamentos de PHP

- Condicionales
 - Condicional simple

```
if (<condición>
{...}
```
 - Condicional y alternativa

```
if (<condición>
{...}
else
{...}
```



Fundamentos de PHP

- Condicional múltiple

```
switch (<expresión>
{
    case <valor 1>: ...
        break;
    case <valor 2>: ...
        break;
    ...
    case <valor n>: ...
        break;
    default: ...
        break;
}
```



Fundamentos de PHP

- Ejemplo “if-else”:

```
if ($x < 4)
{
    echo (“<p>La variable es menor</p>”);
} else
{
    echo (“<p>La variable no es menor</p>”);
}
```

- Ejemplo “switch”:

```
switch ($variable)
{
    case 1: echo (“Número 1”);
            break;
    case 2: echo (“Número 2”);
            break;
    case 3: echo (“Número 3”);
            break;
    default: echo (“No está entre 1 y 3”); break;
}
```



Fundamentos de PHP

- Bucles

- while

- mientras se cumpla una condición, se ejecuta el código (puede no ejecutarse nunca)

- Ej.:

```
$a = 1;
while (a <= 10)
{
    echo ("<p>El número es ".$a."</p>");
    $a++;
}
```



Fundamentos de PHP

- do ... while

- Se ejecuta un código mientras se cumpla una condición (al menos se ejecuta una vez)

- Ej.:

```
$a = 0;
do
{
    $a++;
    echo ("<p>El número es".$a."</p>");
}
while ( a < 10);
```



Fundamentos de PHP

- for

- Ejecuta un código mientras se cumpla una condición
 - Puede no ejecutarse ninguna vez
 - Opcionalmente, se puede proporcionar:
 - Una expresión de inicialización, se ejecuta la primera vez
 - Una expresión de condición
 - Una expresión de paso, se ejecuta al final de cada pasada del bucle
 - Ej.:

```
for ($a = 1; $a <= 10; $a++)  
{  
    echo ("<p>El número es ".$a."</p>");  
}
```



Fundamentos de PHP

- foreach

- Se ejecuta un determinado código por cada uno de los elementos de una colección
- Por ejemplo, si tenemos un array con 3 elementos:

```
$arr = array("perro","gato","ornitorrinco");  
foreach ($arr as $elem)  
{  
    // En cada vuelta, elem guarda uno de los strings  
    echo ("

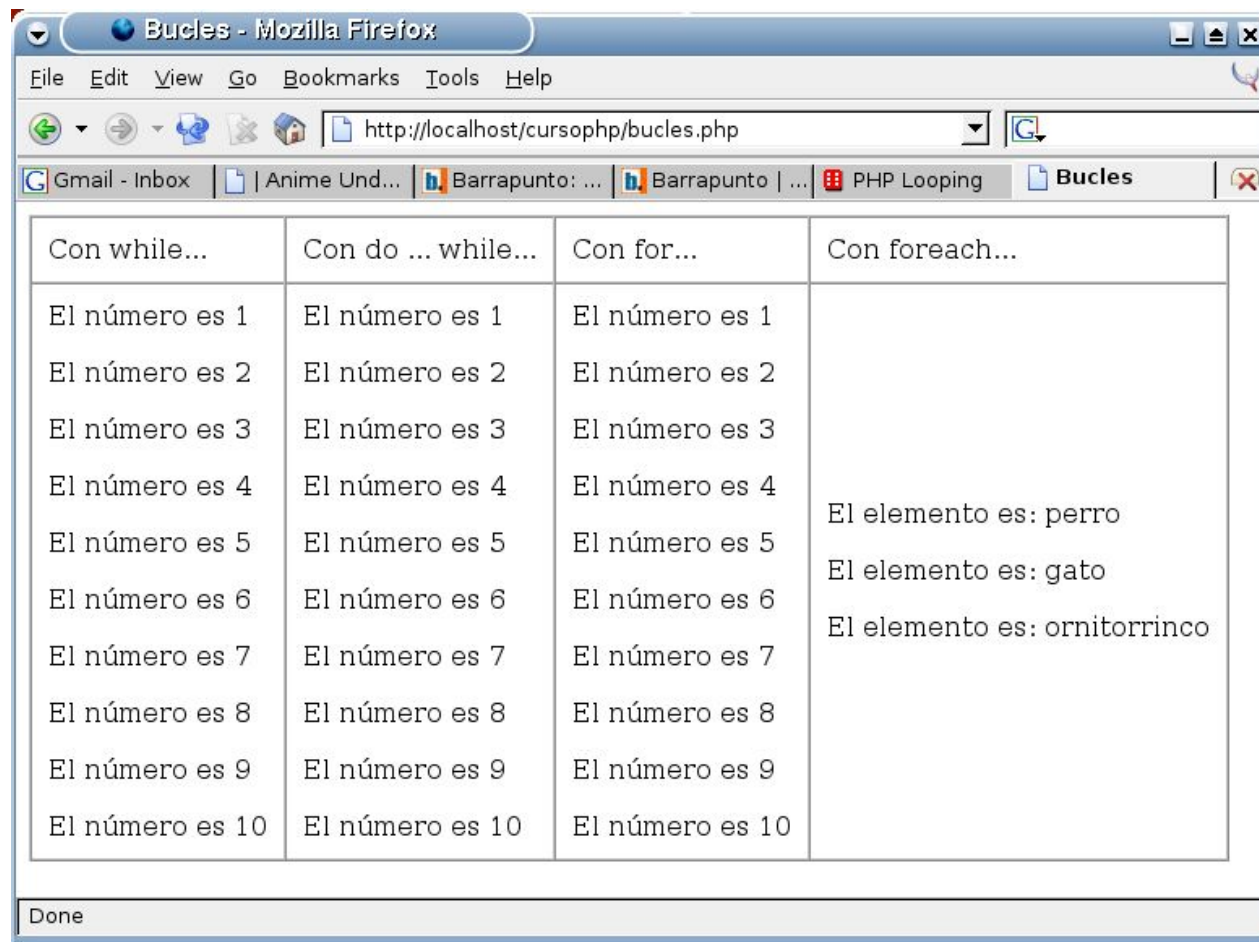
El elemento es: ".$elem."

");  
}
```



Fundamentos de PHP

- Ejemplo: bucles.php



Con while...	Con do ... while...	Con for...	Con foreach...
El número es 1	El número es 1	El número es 1	
El número es 2	El número es 2	El número es 2	
El número es 3	El número es 3	El número es 3	
El número es 4	El número es 4	El número es 4	El elemento es: perro
El número es 5	El número es 5	El número es 5	El elemento es: gato
El número es 6	El número es 6	El número es 6	El elemento es: ornitorrinco
El número es 7	El número es 7	El número es 7	
El número es 8	El número es 8	El número es 8	
El número es 9	El número es 9	El número es 9	
El número es 10	El número es 10	El número es 10	



Fundamentos de PHP

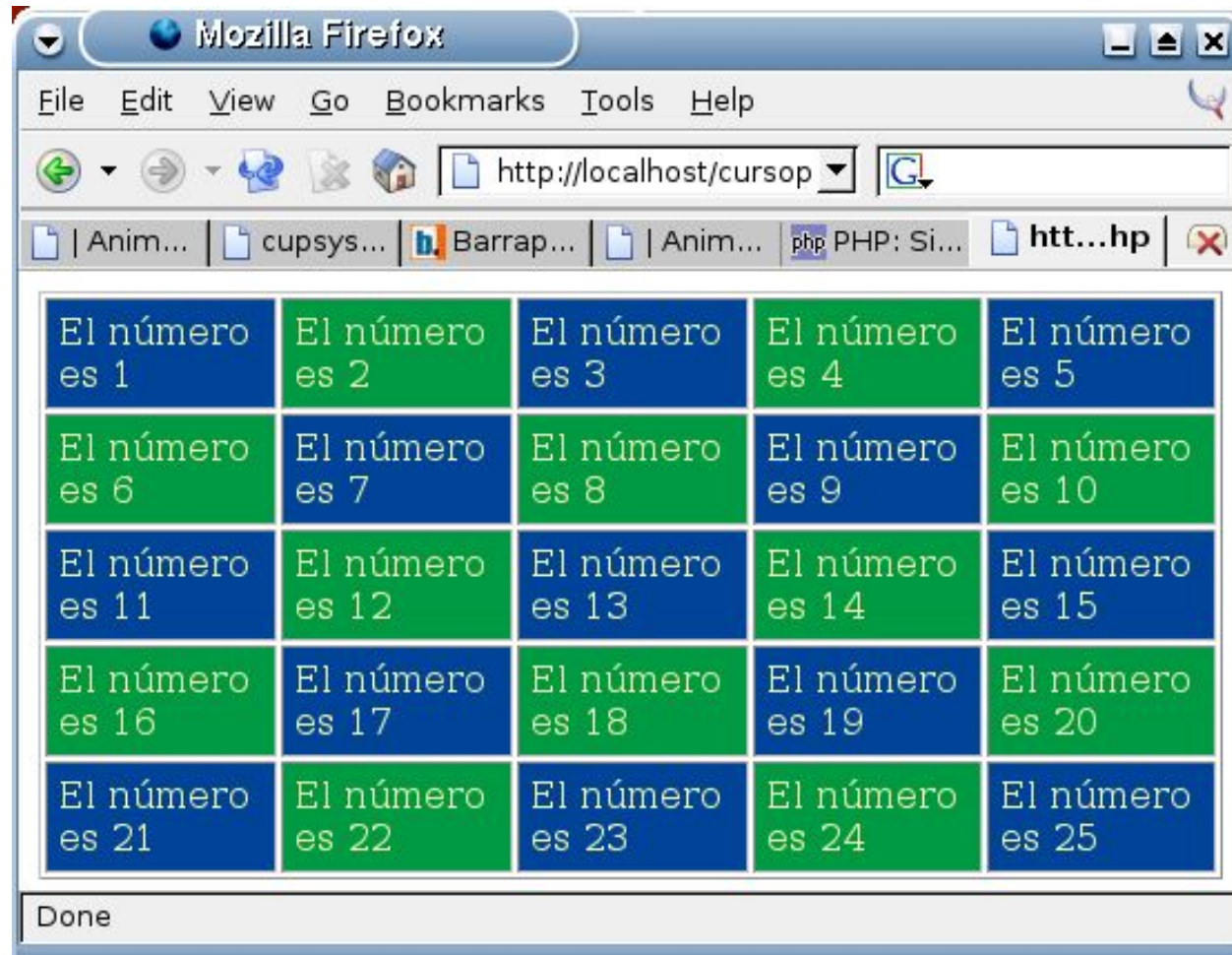
- Métodos avanzados de escape
 - PHP permite construcciones como:

```
<?php
    if ($expresion)
    {
        ?>
        <p>La expresión es cierta</p>
        <?php
    }
    else
    {
        ?>
        <p>La expresión es falsa</p>
        <?php
    }
?>
```



Fundamentos de PHP

- Ejemplo: tablafor.php



Fundamentos de PHP

- Matrices
 - Acceso a una posición
 - A través de “[...]”
`echo ($matriz[1]);`
 - El índice de una matriz puede ser cualquier cosa
`echo ($matriz[“gato”]);`
 - Creación de una matriz
 - A través de “array()”
`$animales = array(“Gato”, “Perro”, “Ornitorrinco”);`
 - En la posición 0 del array está el string “Gato”, en la 1 “Perro” y así sucesivamente



Fundamentos de PHP

- Podemos especificar el índice de cada uno de los elementos:

```
$matriz = array("felino" => "Gato", "canido" => "Perro",  
              "monotrema" => "Ornitorrinco");
```

- Si hacemos:

```
echo (array["monotrema"]);
```
- obtendremos la palabra "Ornitorrinco"

- Creación con corchetes

- Podemos crear elementos de una matriz sobre la marcha:

```
$matriz;  
$matriz[7] = "Texto de la posición 7";  
// si no especificamos un índice, se inserta en la siguiente posición  
$matriz[] = "Esto iría en la posición 8";
```



Fundamentos de PHP

- Modificación de elementos

- Podemos modificar un elemento de una matriz reasignando un valor
 - `$matriz = array(3 => "Oso", 5 => "Perro");`
 - `$matriz[3] = "Gato";`

- Eliminación de elementos

- Podemos eliminar elementos, o la matriz entera con "unset()"

```
$matriz = array(3 => "Oso", 5 => "Perro");  
unset($matriz[3]); // No hay nada en la posición 3  
unset($matriz); // No hay nada en la matriz
```



Fundamentos de PHP

- Recorrido de un array con “foreach”
 - La estructura foreach, además de ser utilizada como hemos visto en la sección de estructuras de control puede usarse para obtener también los índices del array:

```
foreach ($array as $clave => $elem)
{
    echo (“El elemento de la posición “.$clave.” es “.$elem);
}
```



Fundamentos de PHP

- Funciones
 - Podemos definir nuestras propias funciones

```
function <nombre_función> ($parm1, $parm2, ...)  
{  
    ...  
    return <resultado>;  
}
```

- Ej.:

```
<?php  
function saludar($nombre)  
{  
    return "Hola ".$nombre."!";  
}  
  
saludar("Bardok");
```

?>



Fundamentos de PHP

- Objetos
 - Definición de una clase

```
class nombre_clase [extends clase_padre]
{
    var $atributo;

    function nombre_clase() // Constructor
    {}

    function metodo()
    {}
}
```



Fundamentos de PHP

- Creación de objetos

- Si tenemos una clase “ClaseA” podremos crear objetos mediante el operador “new”

```
$objeto = new ClaseA;
```

- El puntero “\$this”

- Es una referencia al propio objeto. Se usa dentro de los métodos

```
class ClaseA
{
    var $num;

    function getNum()
    {
        return $this->num; // “num” no lleva “$”
    }
}
```



Fundamentos de PHP

- Constructores

- Un constructor es un método con el nombre de la clase. Se llama al crearse el objeto.

```
class Persona
{
    var $nombre;

    function Persona($nom)
    {
        $this->$nombre = $nom;
    }
}
```

```
$p = new Persona("Bardok");
```



Fundamentos de PHP

- Herencia

- PHP soporta herencia de clases

```
class ClaseA
{
    var $attrA;
    function getAttrA()
    { return $this->attrA; }
}
class ClaseB extends ClaseA
{
    var $attrB;
    function getAttrB()
    { return $this->attrB; }
}
$obj = new ClaseB;
$obj->getAttrA();
$obj->getAttrB();
```



Fundamentos de PHP

- Otros aspectos de interés:
 - Función “die(mensaje);”:
 - Para la ejecución del script, y envía un mensaje como salida a la página
 - Para errores críticos
 - Ej:

```
if ($error_critico)
{
    die("Error crítico. No se puede continuar.");
}
```



PHP e información de usuario

- Información desde formularios
 - La información de un formulario puede enviarse con el método “get” o el método “post”
 - Existiendo dos arrays en PHP llamados “\$_GET” y “\$_POST”
 - Se accede por el nombre del parámetro
 - Se obtiene el valor del parámetro



PHP e información de usuario

- Ejemplo: crear el siguiente fichero PHP (ejerparams.php):

```
<html>
  <body>
  <?
    foreach ($_GET as $nombre => $param)
    {
      ?>
      <p><?=$nombre.>: ".$param?></p>
      <?
    }
  ?>
  </body>
</html>
```

- Acceder a la página con la URL:
 - [http://localhost/cursophp/ejerparams.php?login=bar dok&pass=1234\\$email=mail@server.com](http://localhost/cursophp/ejerparams.php?login=bar dok&pass=1234$email=mail@server.com)



PHP e información de usuario

- Creamos la página “formparams.html” y enviamos los datos a “ejerparams.php”:

```
<html>
  <body>
    <form method="get" action="ejerparams.php">
      Login: <input type="text" name="login" /><br />
      Password: <input type="password" name="pass" /><br />
      Email: <input type="text" name="email" /><br />
      <input type="submit" value="Enviar" />
    </form>
  </body>
</html>
```

- Modificación: en lugar de utilizar el método “get” utilizar el “post”



PHP e información de usuario

- Para acceder a un dato en concreto dentro de los arrays “\$_POST” y “\$_GET”
 - La clave para acceder a un elemento es el nombre del parámetro
 - echo (“El nombre de usuario utilizado es ”.\$_GET[“login”]);
- Múltiples valores en un parámetro:
 - Un parámetro puede tener distintos valores a la vez:
 - El nombre del parámetro será “nombreparametro[]”:

```
<input type="checkbox" name="sel[]" value="v1"/>Valor 1  
<input type="checkbox" name="sel[]" value="v2"/>Valor 2
```

- El parámetro “nombreparametro” es un array con los elementos seleccionados.
- Si se seleccionan los dos checkboxes anteriores:

```
sel[0] => “v1”  
sel[1] => “v2”
```



PHP e información de usuario

- Variables de sesión
 - Podemos guardar información del usuario que se mantendrá mientras no se cierre el navegador
 - Se transmite de unas páginas a otras
 - El ejemplo más clásico: carrito de la compra
 - Otra utilidad: validación de usuarios
 - Todo lo que guardemos en el array “`$_SESSION`” quedará almacenado de este modo
 - Para poder usarlo: lo primero en la página, llamada a “`session_start();`”



PHP e información de usuario

- Ejemplo: página pública/privada
 - public_private.php

```
<?php
    session_start();
    if (isset($_GET["accion"]))
    {
        $accion = $_GET["accion"];
        if ($accion == "Login")
        {
            $_SESSION["login"] = $_GET["login"];
        }
        else if ($accion == "Desconectar")
        {
            unset($_SESSION["login"]);
        }
    }
    $registrado = isset($_SESSION["login"]);
    if ($registrado)
    {
        $login = $_SESSION["login"];
    }
?>
```





PHP e información de usuario

```
<html>
  <body>
    <?php
      if ($registrado)
      {
        ?>
          <p>Bienvenido, <b><?=$login?></b></p>
          <p><a href="private_login.php">Link a una página privada</a></p>
          <form method="get" action="public_private.php">
            <input type="submit" name="accion" value="Desconectar" />
          </form>
        <?php
          }else{
            ?>
              <p>Por favor, introduce tu nombre de usuario</p>
              <form method="get" action="public_private.php">
                Nombre de usuario: <input type="text" name="login" /><br />
                <input type="submit" name="accion" value="Login" />
              </form>
            <?php
              }
            ?>
          </body>
        </html>
```



PHP e información de usuario

- private_login.php

```
<?php
    session_start();
?>
<html>
    <body>
        <?php
            if (isset($_SESSION["login"]))
            {
                echo ("El nombre de usuario es ".$_SESSION["login"]);
            }
            else
            {
                echo("No hay nombre de usuario");
            }
        ?>
    </body>
</html>
```



Bases de datos en PHP

- Es posible acceder a bases de datos a través de PHP
 - Uso de módulos de bases de datos
 - Gran cantidad de bases de datos soportadas
 - En nuestro caso usaremos MySQL como servidor de bases de datos junto con PHPMyAdmin para administrar las mismas



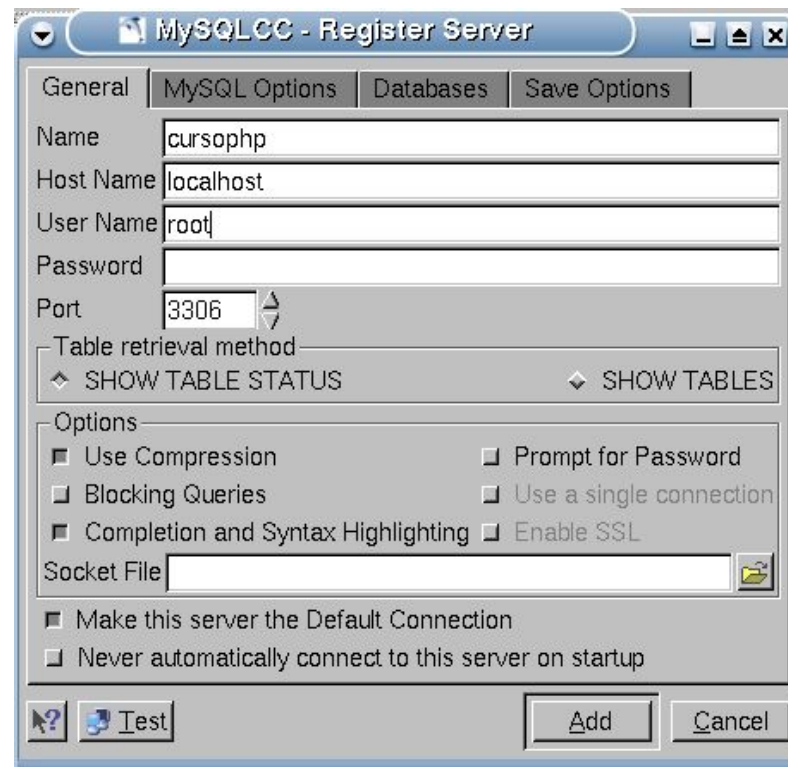
Configuración del servidor de base de datos

- Paquetes a instalar
 - En el caso de utilizar Debian, los paquetes a instalar son:
 - mysql-client, mysql-server, php4-mysql, phpmyadmin, mysqlcc, mysqlcc-i18n
 - mysql-client y mysql-server instalarán tanto el cliente como el servidor de base de datos
 - En caso de que la base de datos tenga un ordenador servidor propio, habría que instalar el servidor en él, y el cliente en el servidor web
 - php4-mysql instala el módulo que permite a PHP interactuar con MySQL
 - mysqlcc es un frontend de configuración de bases de datos, y phpmyadmin permite administración web



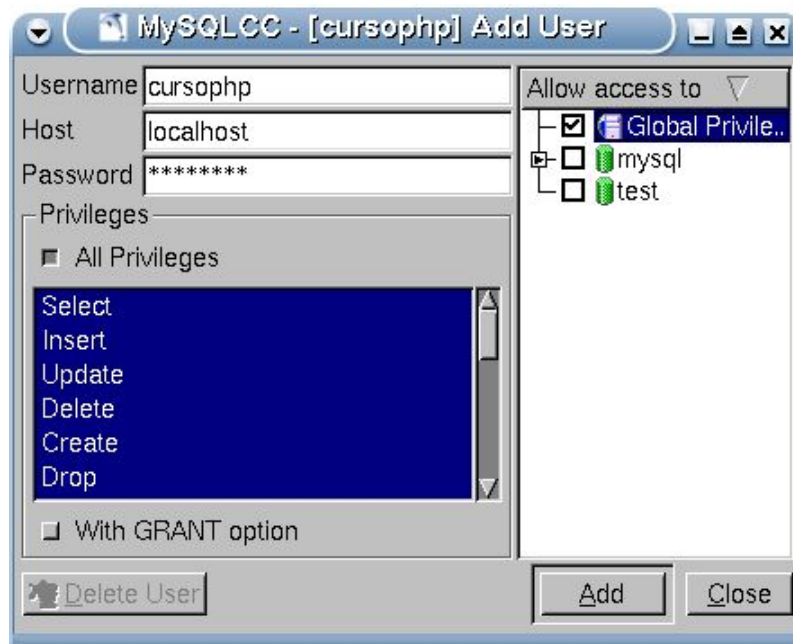
Configuración del servidor de base de datos

- Creación de una base de datos y un usuario administrador
 - Ejecutar mysqlcc y crear una conexión:



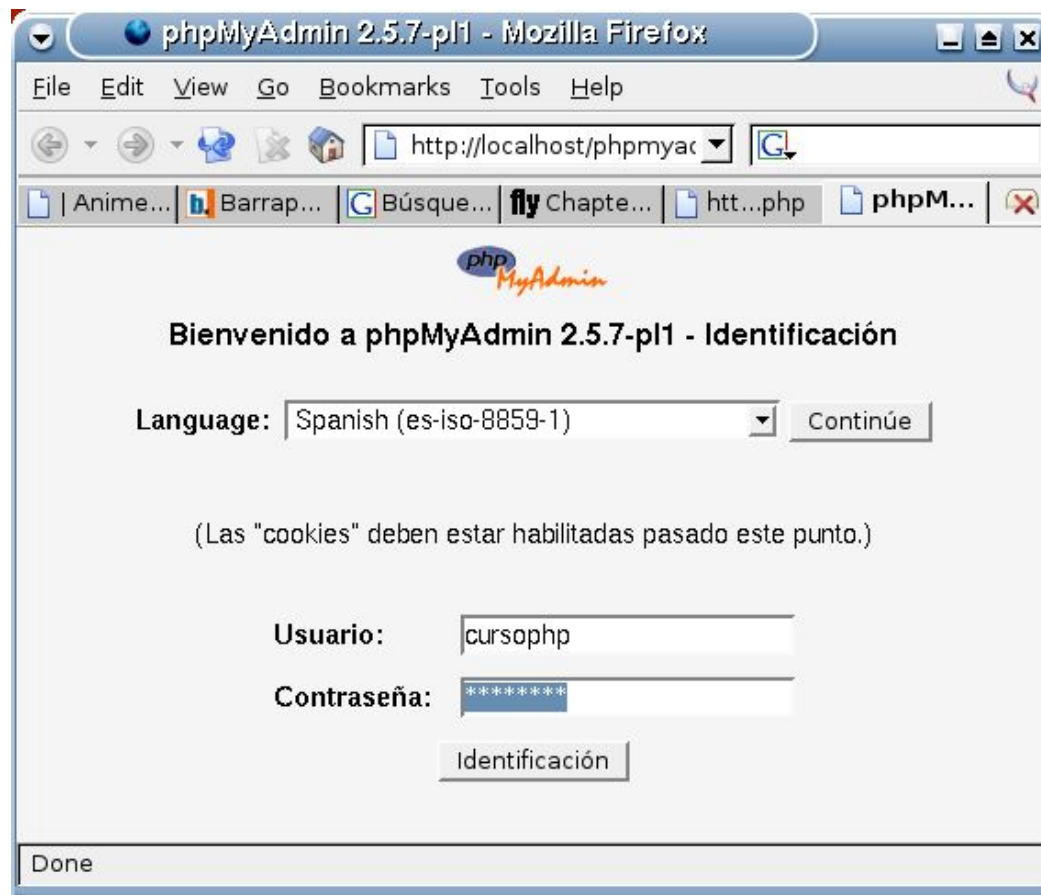
Configuración del servidor de base de datos

- Crear un usuario llamado “cursophp” con clave “cursophp” y todos los privilegios globales en localhost
 - NOTA: esto es terriblemente inseguro, pero útil para este curso



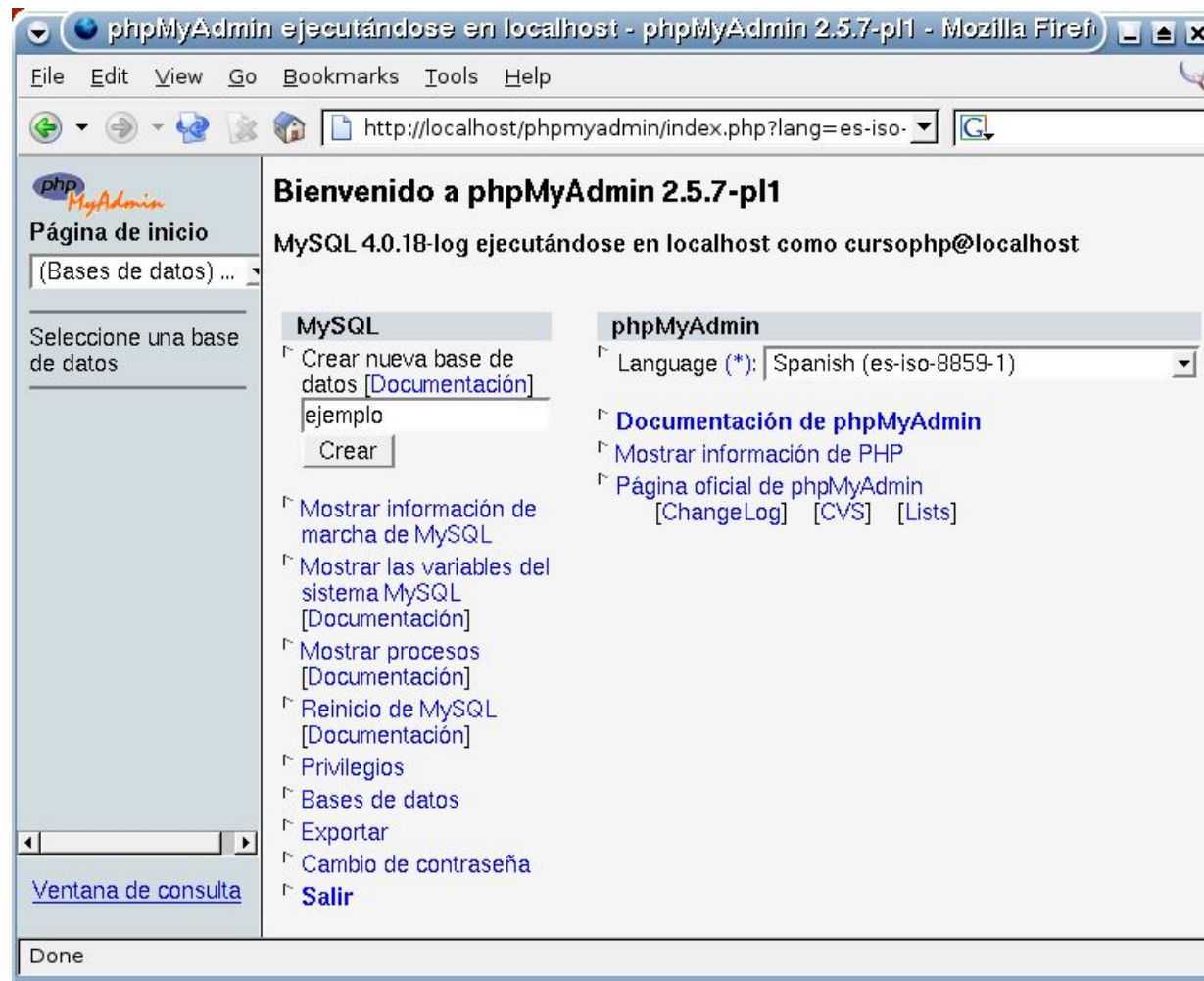
Configuración del servidor de base de datos

- Accedemos a PHPMyAdmin
 - <http://localhost/phpmyadmin>



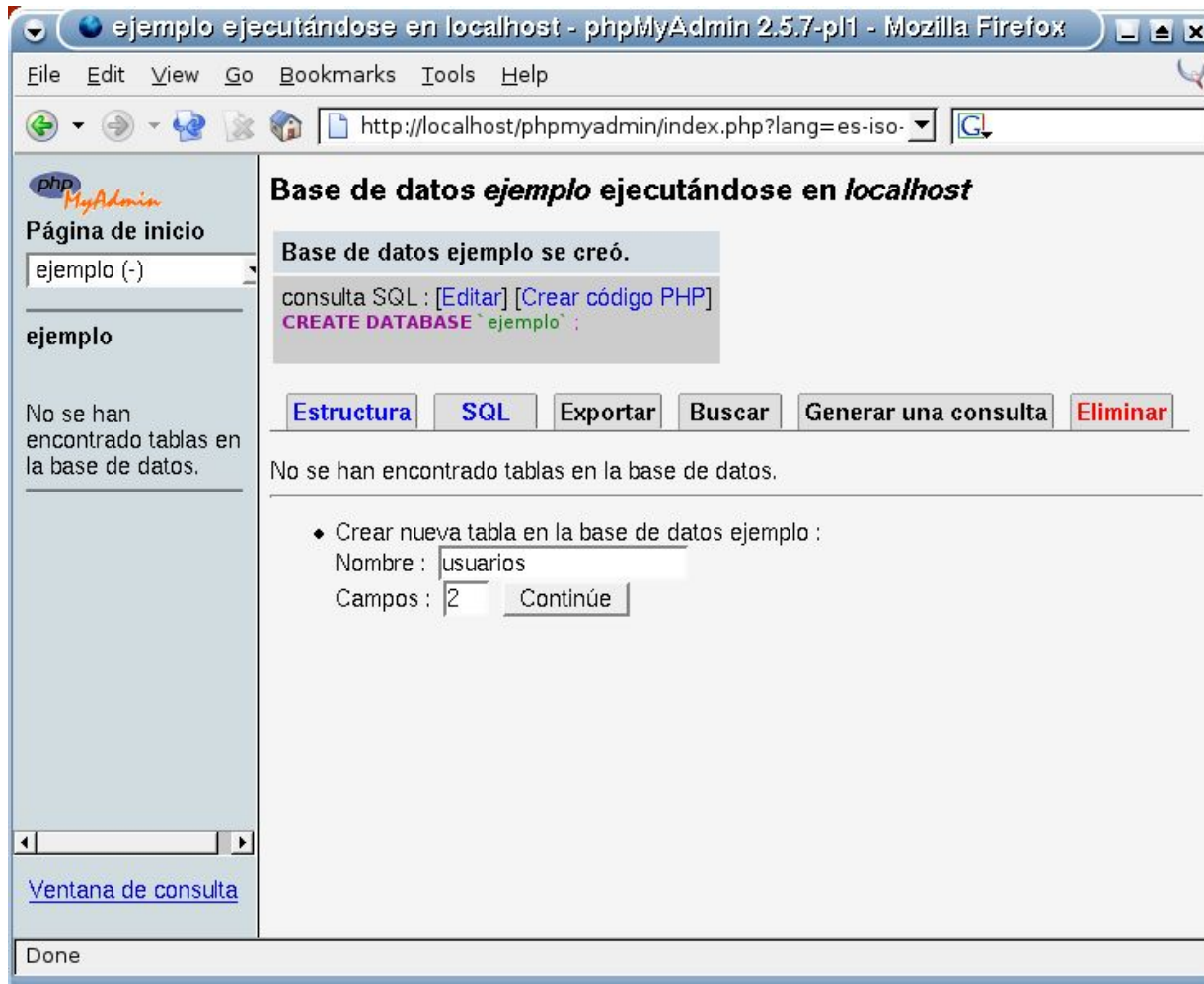
Configuración del servidor de base de datos

- Creamos una base de datos “ejemplo”



Configuración del servidor de base de datos

- Creación de una tabla



The screenshot shows the phpMyAdmin interface in a Mozilla Firefox browser window. The title bar reads "ejemplo ejecutándose en localhost - phpMyAdmin 2.5.7-pl1 - Mozilla Firefox". The address bar shows "http://localhost/phpmyadmin/index.php?lang=es-iso-". The main content area displays the "Base de datos ejemplo ejecutándose en localhost" page. A message states "Base de datos ejemplo se creó." with links for "consulta SQL : [Editar] [Crear código PHP]". Below this, the SQL command "CREATE DATABASE `ejemplo`;" is shown. A row of buttons includes "Estructura", "SQL", "Exportar", "Buscar", "Generar una consulta", and "Eliminar". The text "No se han encontrado tablas en la base de datos." is displayed. A section for creating a new table is active, with "Nombre : usuarios" and "Campos : 2" entered, and a "Continúe" button. The status bar at the bottom shows "Done".



Configuración del servidor de base de datos

- Estructura de una tabla



The screenshot shows the phpMyAdmin interface in a Mozilla Firefox browser window. The address bar shows the URL: `http://localhost/phpmyadmin/index.php?lang=es-iso-8859-1&server=1`. The main content area displays the structure of a table named 'usuarios' in a database named 'ejemplo'. The table structure is shown in a table format with columns: Campo, Tipo, Longitud/Valores*, Atributos, Nulo, Predeterminado**, Extra, and Primaria.

Campo	Tipo	Longitud/Valores*	Atributos	Nulo	Predeterminado**	Extra	Primaria
login	VARCHAR	20		not null			<input checked="" type="radio"/>
password	VARCHAR	20		not null			<input type="radio"/>

Comentarios de la tabla :
Tipo de tabla : Predeterminado

Grabar

* 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

[\[Documentación\]](#)



PHP y MySQL

- Una vez instalado el módulo de MySQL tenemos un API de funciones para interactuar con bases de datos MySQL
- Pasos a seguir:
 - Conexión con el servidor de Base de Datos
 - Selección de la base de datos
 - Trabajar con las tablas
 - En caso de selección, trabajar con los registros
 - Cerrar la conexión con la base de datos



PHP y MySQL

- Conexión con el servidor de base de datos
 - `$db = mysql_connect("host","usuario","password");`
 - La conexión a la base de datos devuelve un identificador de la conexión a la base de datos
 - Si ha habido error, el objeto devuelto será FALSE
- Selección de la base de datos
 - `mysql_select_db("nombre_de_base_de_datos", $db);`
 - Las operaciones posteriores se efectuarán sobre la base de datos seleccionada
 - Si ha habido error, devuelve FALSE



PHP y MySQL

- Trabajar con las tablas

- `mysql_query("sentencia_sql");`
- La sentencia SQL puede ser una inserción, borrado, actualización o selección
- Ej:
 - `mysql_query("INSERT INTO usuarios VALUES ("pepe","aaaaa"))");`
- En el caso de realizarse una selección, se devolverá un objeto con el identificador del resultado de la misma
 - `$res = mysql_query("SELECT login,password FROM usuarios");`



PHP y MySQL

- Una vez realizada la selección, podemos recorrer el resultado utilizando:
 - `mysql_fetch_row($res);`
 - Esta función devuelve FALSE si no hay más filas para procesar, y en caso contrario, devuelve la fila
 - Podemos acceder a los valores de esta manera:

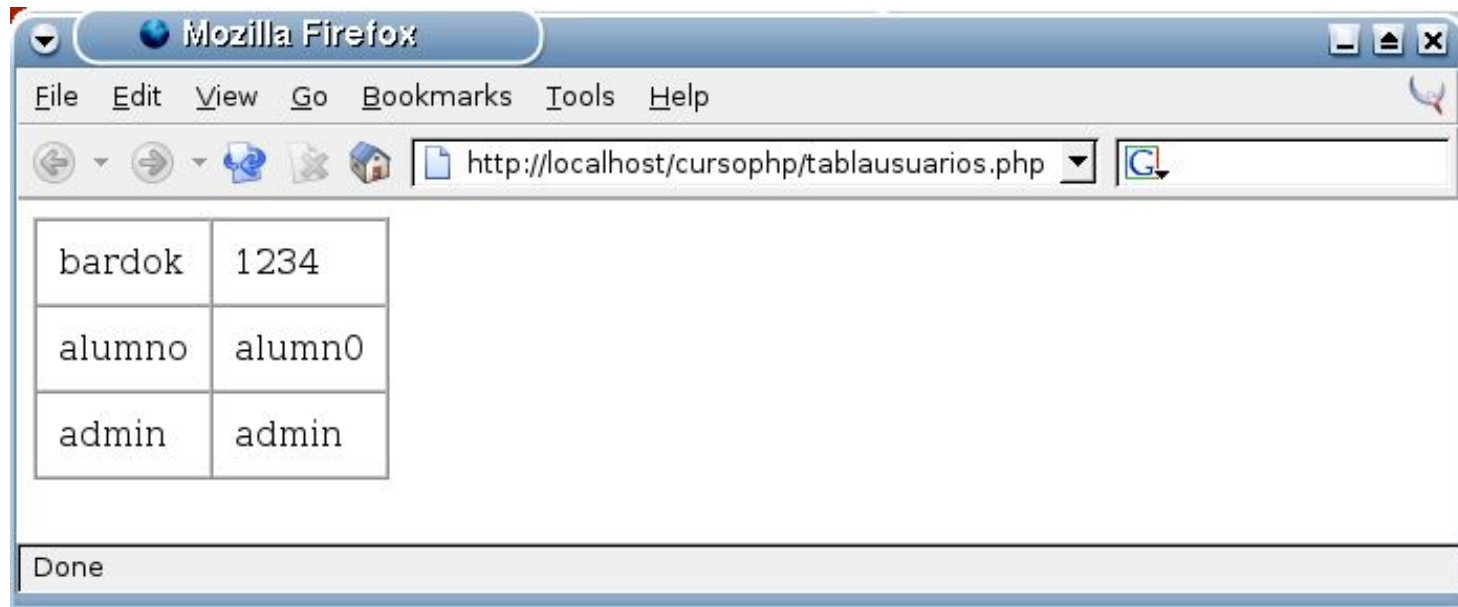
```
$res = mysql_query("SELECT login, password FROM usuarios");
while (list($l,$p) = mysql_fetch_row($res))
{
    echo("El login es ".$l." y el password ".$p."." );
}
```
- Cerrar la conexión con la base de datos
 - `mysql_close($db);`



PHP y MySQL

- Ejemplo:

- Introducir elementos en la tabla “usuarios” a través de PHPMyAdmin, y crear una página PHP que los muestre:



PHP y MySQL

- Ejercicio:
 - Dar de alta una tabla en la base de datos (con la estructura que se quiera)
 - Insertar elementos en la tabla
 - Crear una página que tenga
 - Un formulario que permita altas, bajas y modificaciones en la tabla
 - Un listado con los elementos de la tabla
 - Los elementos son enlaces que, pulsados, nos llevan a la misma página, pero con el elemento seleccionado (sus datos aparecen en el formulario)



PHP y MySQL

- Propuesta de tabla:



The screenshot shows the phpMyAdmin interface in Mozilla Firefox. The browser title is "ejemplo.grupos ejecutándose en localhost - phpMyAdmin 2.5.7-pl1 - Mozilla Firefox". The address bar shows "http://localhost/phpmyadmin/index.php?lang=es-iso-8859-1&server=1". The main content area displays the structure of a table named "ejemplo" in the "ejemplo" database. The table has the following fields:

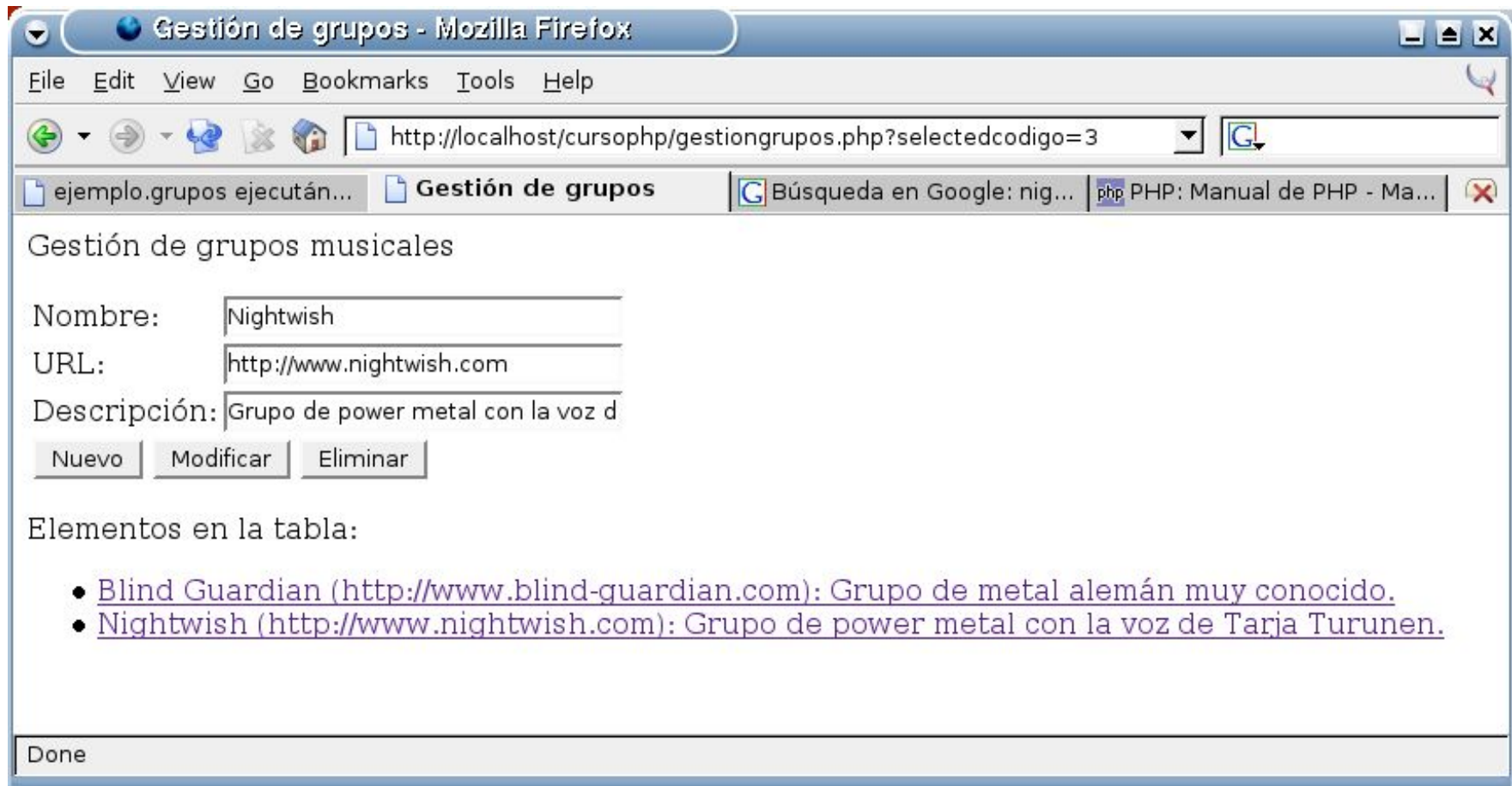
Campo	Tipo [Documentación]	Longitud/Valores*	Atributos	Nulo	Predeterminado**	Extra	Primaria	Índice
codigo	INT	5		not null		auto_increment	<input type="radio"/>	<input type="radio"/>
nombre	VARCHAR	20		not null			<input type="radio"/>	<input type="radio"/>
url	VARCHAR	30		not null			<input type="radio"/>	<input type="radio"/>
descripcion	LONGTEXT			not null			<input type="radio"/>	<input type="radio"/>

Below the table, there are fields for "Comentarios de la tabla:" and "Tipo de tabla:" (set to "Predeterminado").



PHP y MySQL

- Propuesta de interfaz (gestiongrupos.php):



PHP y MySQL

- Propuesta de estructura:
 - La página se puede dividir en varios “bloques”
 - Conexión a la base de datos
 - Gestión de los comandos
 - Todos los elementos de tipo “submit” tienen como nombre “accion” y como “value” la accion que realizan
 - Recuperación de los datos del grupo seleccionado (si lo hay)
 - Al seleccionar un grupo, se pasará como parámetro el código del mismo
 - Página web en sí
 - Formulario: si hay algún grupo seleccionado, sus valores se muestran en los componentes, se muestran los botones “modificar” y “eliminar” y se guarda el código seleccionado en un campo “hidden”
 - Listado: recorrido de la tabla. Cada elemento es un link a la propia página con el formato:
“gestiongrupos.php?selectedcodigo=<codigo>” (ej.:
gestiongrupos.php?selectedcodigo=3)
 - Desconexión de la base de datos

