

GNU/Linux, software libre para la comunidad universitaria

Creación de páginas web dinámicas con PHP/MySQL

Copyright (C) 2008 David Vaquero Santiago pepesan_mail@yahoo.es, Copiar Pegar Digital S.L.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

COLABORADORES

	<i>TÍTULO :</i> GNU/Linux, software libre para la comunidad universitaria		
<i>ACCIÓN</i>	<i>NOMBRE</i>	<i>FECHA</i>	<i>FIRMA</i>
ESCRITO POR	David Vaquero Santiago	19 de octubre de 2009	

HISTORIAL DE REVISIONES

NÚMERO	FECHA	MODIFICACIONES	NOMBRE
1.0	11-04-2007		David Vaquero Santiago

Índice general

1. Introducción.	1
2. Principios de la Web	2
2.1. Principios de la programación web.	2
2.2. La página principal	2
2.3. El manejo de una tabla sencilla.	6
2.4. Añadir un nuevo registro.	8
2.5. El fichero ver.php del directorio	9
2.6. El fichero editar.php	10
2.7. El fichero borrar.php	12
2.8. El fichero buscar.php	13
2.9. Conclusiones	14
3. Otros ejemplos	15
4. Inclusión de relaciones	23
4.1. Directorio de Alumnos	23
4.2. Conclusiones	38
A. GNU Free Documentation License	39

Capítulo 1

Introducción.

Bienvenidos a esta documentación sobre la creación de aplicaciones Php/MySQL. Espero que os sea grata y que os ayude a programar aplicaciones web mejor en los ordenadores con GNU/Linux.

Capítulo 2

Principios de la Web

2.1. Principios de la programación web.

Siempre que nos vayamos a plantear el desarrollo de una aplicación web, debemos de disponer de una serie de conocimientos teórico- prácticos base, que son:

- Saber programar en HTML.
- Saber programar en CSS.
- Saber programar en PHP.
- Saber consultar BBDD con SQL.

Como estos conocimientos pueden echar para atrás en primera instancia, se pretende, mediante ejemplos fomentar el aprendizaje del desarrollo de este tipo de aplicaciones. Lo único que necesita es tener instalado y configurada una plataforma LAMP. Debe tomarse en consideración que los ejemplos aquí indicados se han colocado únicamente con fines educativos y que por lo tanto no deben tomarse como ejemplo de codificación de calidad en PHP.

2.2. La página principal

Como primer reto tenemos el de construir una página principal. Para este paso necesitamos más cosas de las que pensamos en un primer momento. Los pasos a dar son los siguientes:

- Saber el directorio raíz del servidor web. Por defecto en OpenSuse 10.2 es /srv/www/htdocs
- Crear un directorio para almacenar los ficheros de la aplicación. Para ello en Openoffice 10.2 debemos ejecutar como root el comando:

```
[root@Glisa-host htdocs]#  
mkdir /srv/www/htdocs/ejercicios
```

- Modificar los permisos del directorio para permitir la escritura de todos los usuarios (no es muy seguro pero sí funcional). Para ello ejecutaremos:

```
[root@Glisa-host htdocs]#  
chmod 777/srv/www/htdocs/ejercicios
```

Una vez cambiados los permisos ya puede utilizar su editor favorito para realizar todas la operaciones necesarias con ficheros y carpetas. A Partir de este momento siempre que se diga crear un fichero o carpeta será respecto al directorio: /srv/www/htdocs

- Creamos un fichero php principal para el sitio web, que se llame index.php , y copiamos el siguiente contenido:

```
<html>
  <head>
    <TITLE>Título Provisional</TITLE>
  </head>
  <body>
    <p>Mi Primer HTML</p>
  </body>
</html>
```

Mediante este ejemplo puede ver cómo se estructura un fichero html, no se ha metido nada de código php para no complicar más las cosas.

- Crear un fichero css y enlazarlo con el fichero index.php . El fichero css va a ayudar a que cambiar el aspecto de la aplicación de una manera automatizada para todos los ficheros .php . El contenido del CSS será similar al siguiente:

```
/* Definición de estilos en un archivo aparte */
/* Estilo para el documento */
BODY {font-family:Verdana,sans-serif;font-size:20pt;margin-left:0.25in; margin-right ←
:0.25in}
/* Estilo para la cabecera de nivel 2 */
P.cabecera {font-family:Verdana,sans-serif;font-size:15pt;color:red}
/* Estilos para otras etiquetas */
TD.titulo {font-family:Verdana,sans-serif;font-size:20;color:red}
TD.linea {font-family:Verdana,sans-serif;font-size:18;color:black}
:link { color: rgb(0, 0, 153) } /* for unvisited links */
:visited { color: rgb(153, 0, 153) } /* for visited links */
a:active { color: rgb(255, 0,102) } /* when link is clicked */
a:hover { color: rgb(0, 96, 255) } /* when mouse is over link*/
```

Mediante este ejemplo puede ver cómo se aplican modificaciones sobre atributos de tags HTML mediante la definición de una clase CSS. Para Enlazar el fichero en el fichero .php se haría las siguientes modificaciones en el index.php:

```
<html>
  <head>
    <TITLE>Título Provisional</TITLE>
    <link REL="stylesheet" TYPE="text/css"
      HREF="estilo.css" />
  </head>
  <body>
    <p CLASS="cabecera">Mi Primer HTML</p>
  </body>
</html>
```

Como puede ver se ha añadido un nuevo tag llamado LINK dentro de la cabecera del fichero .php que indica donde se encuentra el fichero .css , si se da cuenta, el fichero no tiene ningún tipo de ruta definida, sólo se identifica el nombre del fichero, ya que, dicho fichero se encuentra en el mismo directorio que el fichero index.php. También se puede ver como en el tag <p />se ha añadido una nueva propiedad denominada Class al que le hemos puesto el estilo definido como "cabecera" para el tag <p />en el fichero estilo.css

- Una vez definida la página principal y aplicado el estilo, necesita probar que funciona correctamente. Para lo cual abra su navegador favorito e introduzca la siguiente URL: <http://localhost/ejercicios/index.php> . Debería ver la Página definida en el fichero index.php. A partir de este momento siempre que se vaya a visualizar una página, asuma que la dirección URL siempre empezará por <http://localhost/ejemplos/>
- Una vez comprobado que la página está correctamente definida y verificada, resta pensar qué datos son los que necesitamos manejar. Este tipo de decisiones son las que se suelen realizar en la fase de análisis de un proyecto y dan normalmente como resultado un diagrama de entidad relación, que una vez pasado a una base de datos nos resulta un fichero .sql o fichero de datos de una base de datos. A continuación se muestra el fichero sql resultante del análisis de una aplicación de gestión de matriculaciones:

```
-- phpMyAdmin SQL Dump --
version 2.6.4-pl4 -- http://www.phpmyadmin.net -- --
Servidor: localhost -- Tiempo de generación:
26-01-2006 a las 17:18:36 -- Versión del
servidor: 5.0.18 -- Versión de PHP:
5.1.2-1ubuntu1 -- -- Base de datos: 'cfie' -- --
-----

-- -- Estructura de tabla para la tabla 'alumnos' --
DROP TABLE IF EXISTS 'alumnos'; CREATE TABLE 'alumnos'
( 'id' int(10) unsigned NOT NULL auto_increment,
'nombre' varchar(100) NOT NULL, 'dni' varchar(10) NOT
NULL, 'id_tipo_de_alumno' int(11) unsigned NOT NULL,
PRIMARY KEY ('id') ) ENGINE=MyISAM DEFAULT
CHARSET=latin1 AUTO_INCREMENT=6 ; -- -- Volcar la base
de datos para la tabla 'alumnos' -- INSERT INTO
'alumnos' ('id', 'nombre', 'dni', 'id_tipo_de_alumno')
VALUES (5, 'David Vaquero', '7077052A', 6); --
-----

-- -- Estructura de tabla para la tabla 'asignaturas'
-- DROP TABLE IF EXISTS 'asignaturas'; CREATE TABLE
'asignaturas' ( 'id' int(10) unsigned NOT NULL
auto_increment, 'nombre' varchar(100) NOT NULL, PRIMARY
KEY ('id') ) ENGINE=MyISAM DEFAULT CHARSET=latin1
AUTO_INCREMENT=4 ; -- -- Volcar la base de datos para
la tabla 'asignaturas' -- INSERT INTO 'asignaturas'
('id', 'nombre') VALUES (3, 'Algoritmia'); INSERT INTO
'asignaturas' ('id', 'nombre') VALUES (2, 'POO'); --
-----

-- -- Estructura de tabla para la tabla 'matriculas' --
DROP TABLE IF EXISTS 'matriculas'; CREATE TABLE
'matriculas' ( 'id_mat' int(10) unsigned NOT NULL
auto_increment, 'id_alum' int(10) unsigned NOT NULL,
'id_asig' int(10) unsigned NOT NULL, PRIMARY KEY
('id_mat') ) ENGINE=MyISAM DEFAULT CHARSET=latin1
AUTO_INCREMENT=23 ; -- -- Volcar la base de datos para
la tabla 'matriculas' -- INSERT INTO 'matriculas'
('id_mat', 'id_alum', 'id_asig') VALUES (10, 1, 3);
INSERT INTO 'matriculas' ('id_mat', 'id_alum',
'id_asig') VALUES (18, 1, 2); INSERT INTO 'matriculas'
('id_mat', 'id_alum', 'id_asig') VALUES (16, 1, 2);
INSERT INTO 'matriculas' ('id_mat', 'id_alum',
'id_asig') VALUES (17, 1, 3); INSERT INTO 'matriculas'
('id_mat', 'id_alum', 'id_asig') VALUES (22, 5, 2);
INSERT INTO 'matriculas' ('id_mat', 'id_alum',
'id_asig') VALUES (21, 5, 3); --
-----

-- -- Estructura de tabla para la tabla
'tipos_de_alumno' -- DROP TABLE IF EXISTS
'tipos_de_alumno'; CREATE TABLE 'tipos_de_alumno' (
'id_tipo_de_alumno' int(11) unsigned NOT NULL
auto_increment, 'nombre' varchar(50) NOT NULL, PRIMARY
KEY ('id_tipo_de_alumno') ) ENGINE=MyISAM DEFAULT
CHARSET=latin1 AUTO_INCREMENT=7 ; -- -- Volcar la base
de datos para la tabla 'tipos_de_alumno' -- INSERT INTO
'tipos_de_alumno' ('id_tipo_de_alumno', 'nombre')
VALUES (6, 'descarriado'); INSERT INTO
'tipos_de_alumno' ('id_tipo_de_alumno', 'nombre')
VALUES (5, 'trasto'); INSERT INTO 'tipos_de_alumno'
('id_tipo_de_alumno', 'nombre') VALUES (4,
'aplicado');
```


Como puede verse se han definido varias tablas y algunos datos ya introducidos para facilitar el aprendizaje.

- Ahora lo que debe hacer es crear la base de datos y ejecutar el fichero .sql para definir las tablas y los datos de las tablas en dicha base de datos, esto lo puede hacer fácilmente mediante PhpMyAdmin. No se olvide de apuntar los datos de conexión a la base de datos: servidor base de datos, nombre de la base de datos, nombre de usuario y contraseña; los necesitará más tarde para crear el fichero de configuración config.php.

- Crear un directorio por cada entidad mediante el comando:

```
[root@Glisa-host ejercicios]#  
mkdir entidad
```

- Modificar los permisos de cada directorio creado:

```
[root@Glisa-host ejercicios]#  
chmod 777 entidad
```

- Una vez que están definidos los datos a manejar podemos finalizar el fichero index.php introduciendo los enlaces a los ficheros index.php que crearemos en cada nuevo directorio. A continuación verá el fichero index.php definitivo:

```
<html>  
  <head>  
    <TITLE>Título Provisional</TITLE>  
    <LINK REL="stylesheet" TYPE="text/css"  
      HREF="estilo.css" />  
  </head>  
  <body>  
    <A href="alumnos/index.php" name="alumnos">  
      Gestión de alumnos</A>  
    <br />  
    <A href="asignaturas/index.php"  
      name="asignaturas">Gestión de  
      asignaturas</A>  
    <BR />  
    <A href="tipos_de_alumno/index.php"  
      name="tipos_de_alumno">Gestión de  
      tipos de alumno</A>  
    <BR />  
  </body>  
</html>
```

Como puede ver se han añadido tres enlaces cada uno de ellos a un index.php que permite la gestión de cada una de las entidades del modelo de datos.

- Compruebe que se visualiza correctamente, actualice en su navegador web el index.php de la aplicación.
- A continuación definiremos nuestro primer fichero con código PHP, el config.php, lo crearemos en el directorio donde se encuentra el index.php de la página principal y aplicarle los permisos necesarios (siempre que cree in fichero recuerde aplicarle los permisos). El contenido del fichero debería ser similar al siguiente:

```
<?php  
//ponemos el nombre del servidor de mysql  
$server= 'localhost';  
//ponemos el nombre de la bbdd  
$database='ejemplo';  
//ponemos el nombre del usuario  
$user='root';  
//ponemos la contraseña del usuario  
$passwd='micontrasegna';  
?>
```

Substituya los valores entrecomillados, por los datos que ha anotado previamente. Como puede ver el fichero empieza con la identificación del lenguaje de script php y termina con la finalización de la parte PHP del fichero, en medio, disponemos de comentarios y asignaciones de variables, como puede ver todas las variables empiezan con el símbolo \$, la asignaciones en sí mismas son muy parecidas a otros lenguajes de programación. Y como puede observar toda sentencia se finaliza con el símbolo ;

Con esto dispone ya de los elementos para empezar construir su aplicación web. Ahora se procederá a explicar el funcionamiento de los ficheros .php que manejan una entidad.

2.3. El manejo de una tabla sencilla.

Vamos a empezar con una tabla sencilla para facilitar su comprensión, la de asignaturas. Empezaremos por crear el fichero index.php. Que sería así:

```
<?php
//incluimos el fichero config.php para coger los datos de la conexión
include_once('../config.php');
//cogemos las variables de la conexión para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexión
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");
//rellenamos la consulta, cambiar alumnos por el nombre de la tabla a listar
$sql="SELECT * FROM asignaturas";
//echo $sql;
//re realiza la consulta
$result=mysql_query($sql,$conexion);

?>
<html>
<head>
<TITLE>Listado de Asignaturas</TITLE>
<LINK REL="stylesheet" TYPE="text/css"
HREF="../estilo.css" />
</head>
<body>
<A href="../index.php" name="alumnos">Inicio</A>
<br />
<A href="../alumnos/index.php" name="alumnos">
Alumnos</A>
<br />
<A href="index.php" name="asignaturas">
Asignaturas</A>
<BR />
<A href="../tipos_de_alumno/index.php"
name="tipos_de_alumno">Tipo de alumno</A>
<br />
<p>Listado de Asignaturas.</p><?php
//aquí se coloca el enlace que permite añadir una asignatura
echo "<p><A HREF='nuevo.php'>Nuevo</A></p>";
//comienza el formulario que permite la búsqueda
echo "<form method='post' action='buscar.php'>";
//aquí coloca una caja de texto donde se escribe la búsqueda
echo "<input type='text' name='busqueda' size=20 maxlength =20>";
//aquí se coloca el botón a pulsar cuando se rellene la búsqueda
echo "<input type='submit' name='buscar' value='Buscar'></form><br>";
//se cogen los primeros resultados del listado
if($row= mysql_fetch_array($result)){
```

```
//si hay algun registro inicia la tabla que muestra los resultados del listado
echo "<TABLE BORDER='1'>";
//coloca la cabecera de la tabla, aqui se substituiria con los campos de la tabla
//colocando mas <TD CLASS=\"titulo\">campo</TD> dependiendo de los campos que haya
echo "<TR><TD CLASS=\"titulo\">Nombre</TD><TD colspan=3 CLASS=\"titulo\">Opciones</TD></TR>";
//empieza el bucle que coloca las líneas del listado
DO
{
//Coloca una línea en el listado
//si hay mas campos se ponen más <TD width=300 CLASS=\"linea\">".$row["nombre_campo"]."</TD>
//inicia la línea
echo "<TR>";
//pone el nombre del alumno
echo "<TD width=300 CLASS=\"linea\">".$row["nombre"]."</TD>";
//pone los enlaces de ver, editar y borrar asignaturas
echo "<TD CLASS=\"linea\"><A HREF=ver.php?id=".$row['id']. ">Ver</A></TD>";
echo "<TD width=50 CLASS=\"linea\"><A HREF=editar.php?id=".$row['id']. "> Editar</A></TD>";
echo "<TD width=50 CLASS=\"linea\"><A HREF=borrar.php?id=".$row['id']. "> Borrar</A></TD>";
//termina la fila
echo "</TR>";
}
//comprueba si hay mas registros que colocar
WHILE ($row=mysql_fetch_array($result));
//si no hay mas registros termina la tabla
echo "</TABLE><br>";
//coloca el enlace que permite volver a listar
echo "<A HREF=\"index.php\">Volver</A>";
}else{
//si no hay registros en la tabla informa de ello
echo "No hay ninguna asignatura en la BBDD<BR>";
//coloca el enlace que permite volver a listar
echo "<A HREF=\"index.php\">Volver</A>";

}
//termina el script
?>
</body>
</html>
```

Como puede verse en el fichero hay 4 partes bien diferenciadas, que se repetirán en todos los ficheros de index.php (de listado): una definidas como parte de un script PHP que consulta los datos a la tabla, otra donde hay código HTML de la cabecera del fichero HTML resultante, otra parte en PHP que permite generar la tabla con los datos y la última donde se cierran los tags HTML de la página web.

En la primera parte se incluye el fichero ../config.php que es donde centralizamos los datos de configuración para poder acceder a la base de datos. Después de realizar la conexión y seleccionar la BBDD se almacena en la variable \$sql la consulta a realizar. Para a continuación ejecutar la consulta en la base de datos y almacenar en la variable \$result los resultados de la consulta.

En la segunda parte se definen los tags HTML generales y el del menú de la Aplicación. El enlace a la página principal y al resto de directorios generados.

En la tercera parte, se pudo ver cómo generar el código HTML mediante los echos. en la primera sección se pudo ver cómo se genera el cuadro de búsqueda, en la segunda parte se puede ver cómo utilizar la sentencia condicional con sus dos casos: el primero en el que hay datos en el resultado y el segundo cuando no hay resultados a la consulta. En el primer caso se genera la cabecera de una tabla y luego se entra en un bucle para cada uno de los resultados(filas), se ponen celdas cada fila con cada uno

de los valores (en este caso un sólo campo registrable el nombre) y luego colcoa las operaciones típicas Ver, Editar y Borrar; cierra la fila y luego cierra la tabla. En el segundo caso simplemente saca un mensaje de que no hay asignaturas disponibles.

Merece una especial mención la manera de resolver las llamadas a cada una de las operaciones. Por el funcionamiento de una aplicación Web debe recordarse que una vez calculada la página se termina su ejecución e inicialmente no hay datos disponibles; por lo tanto en cada recarga de página el script debe ser ejecutado completamente hasta su finalización. Para facilitarle al lector su aprendizaje se van a generar ficheros .php para cada operación. La manera de abstraer los pasos de valores entre recargas de página es con los métodos GET y POST, en el caso de las operaciones del listado se pasan los datos mediante el método GET, es decir mediante la llamada al script, por ejemplo ver.php, seguido del símbolo ? seguido de una terna: nombre de parámetro, símbolo = y el valor; por ejemplo ?id=1 sería pasar el parámetro id con el valor 1. Después estos parámetros se podrán recuperar de una manera sencilla.

En la cuarta parte, simplemente se cierran los tag que se había abierto previamente para dar un código HTML Válido.

A continuación veremos los distintos tipos de enlaces que disponemos en la página de listado index.php del directorio.

2.4. Añadir un nuevo registro.

A continuación veremos cómo se estructura un fichero nuevo.php:

```
<?php
//incluimos el fichero config.php para coger los datos de la conexión
include_once('../config.php');
//cogemos las variables de la conexión para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexión
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");

?>
<html>
  <head>
    <TITLE>Nueva Asignatura</TITLE>
    <LINK REL="stylesheet" TYPE="text/css"
      HREF="../estilo.css" />
  </head>
  <body>
    <?php
      //comprueba si se envían datos para guardar
      //si no es así es que tiene que presentar el formulario
      if(!isset($_POST['nuevo']))
      {
        //coloca el principio del formulario para que vuelva al mismo script cuando ←
        se rellene
        echo "<form method=\"post\" action=\"nuevo.php\">";
        //aquí se deben poner tantas líneas echo como campos haya
        //cambiando el echo "Nombre : <input type=\"text\" name=\"nombre\" maxlength= ←
        =200 size=40><br>";
        // por echo "Nombre : <input type=\"text\" name=\"nombre_campo\" maxlength= ←
        longitud_campo size=tamaño_caja_texto><br>";
        //se coloca la caja de texto para el nombre
        echo "Nombre : <input type=\"text\" name=\"nombre\" maxlength=100 ←
        size=40><br>";
        ////se coloca el botón de guardar
        echo "<input type=\"submit\" name=\"nuevo\" value=\"Guardar\"></form ←
        ><br>";
      }else{
        //si se han metido los datos en el formulario
        //se deben colocar tantos isset($_POST['nombre_campo']) && como campos haya ←
        en la tabla
```

```
        if(isset($_POST['nombre'])){
//en esta linea se ponen los nombres de los campos separados por comas y en la parte de valores lo mismo pero con
//variables $_POST['nombre_campo']
        $sql="INSERT INTO `asignaturas` ( `id` , `nombre`)
        VALUES (NULL , '".$_POST['nombre']."' );";
        //echo $sql;
//se ejecuta la inserción de datos
        $result=mysql_query($sql,$conexion);
//se informa de que se añadido un nueva asignatura
        echo "<p>Asignatura Añadida</p>";
//se coloca el enlace que permite volver al listado
        echo "<p><A HREF='index.php'>Volver</A></p>";
        }else{
// si no nos han metido campos
        echo "<p>Vuelva a intentarlo</p>";
// se vuelve a colocar el formulario con los mismos campos que ahora, la diferencia es que se pone tantas veces como campos haya en
//los VALUE el $_POST['nombre_de_campo'] para que no haya que volver a teclearlo
        echo "<form method='post' action='nuevo.php' >";
// se coloca la caja del nombre
        echo "Nombre : <input type='text' name='nombre' value = '".$_POST['nombre']."' maxlength = 100 size=40><br>";
// se coloca el boton de guardar
        echo "<input type='submit' name='nuevo' value='Guardar' >";
        }
    }
    ?>
</body>
</html>
```

Como puede verse la estructura principal es muy parecida a la del fichero index.php del directorio. En la primera parte se hacen las inicializaciones de la BBDD, en la segunda se ponen los tags iniciales de HTML, en la tercera se hacen las comprobaciones, presentación del formulario e inserción del registro y en la cuarta parte los tags finales del HTML.

En la tercera parte se puede ver cómo se discrimina dependiendo de si el valor 'nuevo' de la variable POST está presente (si se está mandando el formulario, recordar el método POST explicado en el punto anterior, que se utiliza en el formulario de esta página, se expondrá más adelante). En el primer caso (no se ha enviado un formulario), se genera el formulario de alta de asignatura, indicando los campos necesarios, observase que el formulario se envía por el método POST, así se pueden pasar también los parámetros, al siguiente script que se ejecute. En el segundo caso el formulario ya se ha enviado, si se cumplen los requisitos para dar de alta un registro de asignatura se pasa a hacer la inserción, se informa de ello al usuario y se coloca el botón para volver; si no se cumplen los requisitos para dar de alta el registro, se informa de ello al usuario y se vuelve a presentar el formulario.

Por lo que se puede ver no hay muchas más diferencias con el resto de las partes del fichero.

2.5. El fichero ver.php del directorio

A continuación verá como construir el fichero de visualización de un registro:

```
<?php
//incluimos el fichero config.php para coger los datos de la conexión
include_once('../config.php');
//cogemos las variables de la conexión para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexión
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
```

```
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");
//rellenamos la consulta, cambiar alumnos por el nombre de la tabla a listar
$sql="Select * from asignaturas WHERE id=".$_GET['id'].>";";
//re realiza la consulta
$result=mysql_query($sql,$conexion);
//print $sql;
//se coge el registro del alumno
$row= mysql_fetch_array($result);
?>
<html>
  <head>
    <TITLE>Alumno:<?php echo $row['nombre']; ?></TITLE>
    <LINK REL="stylesheet" TYPE="text/css"
      HREF="../../../estilo.css" />
  </head>
  <body>
    <?php
      //en esta parte se muestran los campos del alumno
      //se debe añadir líneas tantas como campos tenga la tabla: echo "<p>Nombre de ↵
        campo:".$row["nombre_de_campo"]."</p>";
      //se muestra el código del alumno
      echo "<p>Código:".$row["id"]."</p>";
      //se muestra el nombre del alumno
      echo "<p>Nombre:".$row["nombre"]."</p>";
      //se coloca el enlace para que pueda editarse el alumno.
      echo "<p><a href=editar.php?id=".$_row['id'].">Editar</a></TD></p>";
      // se coloca el enlace para borrar el alumno
      echo "<p><a href=borrar.php?id=".$_row['id'].">Borrar</a></TD></p>";
      // se coloca el enlace para ir a listado de alumnos
      echo "<p><a href='index.php'>Atras</a></p>";
    ?>
  </body>
</html>
```

Como se puede observar en la primera parte del fichero se coge el parámetro por GET del id del registro y se realiza la consulta para coger el registro.

En la tercera parte simplemente se cogen los campos de la variable, se presentan al usuario y se pone el botón para volver al index.php

2.6. El fichero editar.php

A continuación veremos con se estructura el fichero editar.php:

```
<?php
//incluimos el fichero config.php para coger los datos de la conexi?n
include_once(' ../config.php');
//cogemos las variables de las coneci?n para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexi?n
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");
//rellenamos la consulta, cambiar alumnos por el nombre de la tabla a listar
$sql="Select * from asignaturas WHERE id=".$_GET['id'].>";";
//re realiza la consulta
$result=mysql_query($sql,$conexion);
//print $sql;
//se coge el registro del alumno
$row= mysql_fetch_array($result);
```

```
?>
<html>
<head>
  <TITLE>Editar Asignatura</TITLE>
  <LINK REL="stylesheet" TYPE="text/css"
    HREF="../../../estilo.css" />
</head>
<body>
  <?php
    //se mira si se est? enviando el formulario de edici?n del alumno
    if(!isset( $_POST['editar']))
    {
      //Si se est? enviando el formulario
      //echo "<p>Vuelva a intentarlo</p>";
      //se abre el formulario que envia los datos de las modificaciones de ese
      registro
        echo "<form method=\"post\" action=\"editar.php?id=".$_GET[' ←
          id'].\"\" >";
      // muestra el c?digo del alumno
        echo "Codigo:".$row['id'].<br>";
      // a partir de aqui se muestran los campos a rellenar en el formulario
      // si se substituye echo "Nombre : <input type=\"text\" name=\"nombre\" value ←
        =\"".$row['nombre'].\"\" maxlength =200 size=40><br>";
      // por echo "Nombre : <input type=\"text\" name=\"nombre_de_campo\" value ←
        =\"".$row['nombre'].\"\" maxlength =longitud_maxima_de_campo
      //size=tama?o_dte_la_caja><br>";
      // se muestra la caja del nombre del alumno
        echo "Nombre : <input type=\"text\" name=\"nombre\" value ←
          =\"".$row['nombre'].\"\" maxlength =100 size=40><br>";
      // se pone el bot?n de guardar los datos
        echo "<input type=\"submit\" name=\"editar\" value=\"Guardar ←
          \"></form><br>";
    }else{
      // si se env?a el formulario hay que coger los datos del formulario y ←
        meterlos en la BBDD
      // aqui se deben a?adir tantos isset($_POST['nombre_de_campo']) && como ←
        campos haya en la bbdd
        if(isset($_POST['nombre'])) {
      //se desactiva el contenido de la variable $sql
        unset($sql);
      // se rellena la consulta que inserta los datos
      //se ponen tantos 'nombre_de_campo' = '".$_POST['nombre_de_campo'].',
        $sql="UPDATE `asignaturas` SET `nombre` = '".$_POST['nombre ←
          '].' WHERE `id` =".$_GET['id'].\" LIMIT 1 ";
        //echo $sql;
      //se envia la inserci?n de datos en la BBDD
        $result=mysql_query($sql,$conexion);
      //Se informa por la p?gina web que el alumno ha sido modificado
        echo "<p>Asignatura Modificada</p>";
      //se coloca el bot?n que hace volver al listado de alumnos
        echo "<p><A HREF='index.php'>Volver</A></p>";
    }else{
      // se informa al usuario que no se han rellenado los campos
      echo "Vuelva a intentarlo.<br>";
      //se abre el formulario que envia los datos de las modificaciones de ←
        ese registro
        echo "<form method=\"post\" action=\"editar.php?id=".$_GET[' ←
          id'].\"\" >";
      // muestra el c?digo del alumno
        echo "Codigo:".$row['id'].<br>";
      // a partir de aqui se muestran los campos a rellenar en el formulario
      // si se substituye echo "Nombre : <input type=\"text\" name=\"nombre\" value ←
```

```
        =\"".$row['nombre'].\" \" maxlength =200 size=40><br>";
// por echo "Nombre : <input type=\"text\" name=\"nombre_de_campo\" value ←
        =\"".$row['nombre'].\" \" maxlength =longitud_maxima_de_campo
// size=tamaño_de_la_caja><br>";
// se muestra la caja del nombre del alumno
        echo "Nombre : <input type=\"text\" name=\"nombre\" value ←
        =\"".$row['nombre'].\" \" maxlength =200 size=40><br>";

// se pone el botón de guardar los datos
        echo "<input type=\"submit\" name=\"editar\" value=\"Guardar ←
        \"></form><br>";
    }
}
?>
</body>
</html>
```

En la primera parte se coge el parámetro id por GET y se cogen los datos de la tabla.

En la tercera parte es muy parecido al nuevo.php, exceptuando por que intenta colocar los datos del registro de la tabla en el formulario y hace un UPDATE en lugar de un INSERT en la consulta SQL. Los datos en sí mismos del formulario se cogen con POST

2.7. El fichero borrar.php

A continuación se puede ver la estructura del fichero borrar.php:

```
<?php
//incluimos el fichero config.php para coger los datos de la conexión
include_once(' ../config.php');
//cogemos las variables de las conexión para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexión
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");
//rellenamos la consulta, cambiar alumnos por el nombre de la tabla a listar
$sql="Select * from alumnos WHERE id=\"".$_GET['id'].\"";
//re realiza la consulta
$result=mysql_query($sql,$conexion);
//print $sql;
//se coge el registro del alumno
$row= mysql_fetch_array($result);
?>
<html>
<head>
<TITLE>Borrar Asignatura</TITLE>
<LINK REL="stylesheet" TYPE="text/css"
    HREF=" ../estilo.css" />
</head>
<body>
<?php
// se rellena la consulta que borra el alumno se substituye alumno por el ←
    nombre de la tabla
    $sql="DELETE from 'asignaturas' WHERE 'id' =\"".$_GET['id'].\" LIMIT 1 ;";
//echo $sql;
// se ejecuta la consulta se borrado
$result=mysql_query($sql,$conexion);
// se informa de que se ha eliminado un alumno
echo "<p>Asignatura eliminada</p>";
```



```
// se coloca el enlace que permite volver al listado
echo "<p><A HREF='index.php'>Volver</A></p>";

?>
</body>
</html>
```

En la primera parte cogemos los datos del registro que queremos borrar, por si en algún momento quisiéramos mostrarlo. En la tercera parte borramos el registro en sí mismo, informamos al usuario del cambio y colocamos el botón Volver.

2.8. El fichero buscar.php

A continuación se muestra la estructura del fichero buscar.php:

```
<?php
//incluimos el fichero config.php para coger los datos de la conexión
include_once('../config.php');
//cogemos las variables de las conexión para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexión
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");

?>
<html>
<head>
<TITLE>Búsqueda de Asignaturas</TITLE>
<LINK REL="stylesheet" TYPE="text/css"
    HREF="../estilo.css" />
</head>
<body>
<p>Búsqueda de Asignaturas.</p><?php
// coloca el enlace que permite añadir un alumno
echo "<p><A HREF='nuevo.php'>Nuevo</A></p>";
// se inicia el formulario de la búsqueda
echo "<form method=\"post\" action=\"buscar.php\">";
// se indica que el campo de búsqueda tiene 20 caracteres de longitud
echo "<input type=\"text\" name=\"busqueda\" size=20 maxlength =100 value ←
    =\"".$_POST['busqueda']."\">";
// se coloca el boton que se parantara para realizar la busqueda
echo "<input type=\"submit\" name=\"buscar\" value=\"Buscar\"></form><br>";
// se rellena la busqueda por todos los campos de la tabla, se debe ←
    substituir alumnos por el nombre de la tabla y poner tantos
// nombre_de_campo LIKE ' %'".$_POST['busqueda']."' OR como campos haya.
$sql="SELECT * FROM asignaturas where nombre LIKE ' %'".$_POST['busqueda ←
    '']." %'";
// se realiza la busqueda
$result=mysql_query($sql,$conexion);
//Se comprueba si hay resultados en la busqueda
if($row= mysql_fetch_array($result)){
// si hay resultados crea la tabla
echo "<TABLE BORDER='1'>";
//pone los campos de la tabla tiene que haber tantos <TD CLASS=\"titulo\"> ←
    Nombre_de_campo</TD> como campos haya
echo "<TR><TD CLASS=\"titulo\">Nombre</TD><TD colspan=3 CLASS=\"titulo\"> ←
    Opciones</TD></TR>";
//coloca las líneas de la tabla
DO
{
```

```
//aqui se ponen tantas líneas <TD width=300 CLASS=\"linea\">\".$row[\" ←  
    nombre_de_campo\"].\"</TD> como campos haya en la tabla  
// se coloca el inicio de la fila  
echo "<TR>";  
//se coloca el campos del nombre  
echo "<TD width=300 CLASS=\"linea\">\".$row[\"nombre\"].\"</TD>";  
//se colocan los enlaces de ver editar y borrar alumno  
echo "<TD CLASS=\"linea\"><A HREF=ver.php?id=\".$row['id'].\">Ver</A></TD>";  
echo "<TD width=50 CLASS=\"linea\"><A HREF=editar.php?id=\".$row['id'].\"> ←  
    Editar</A></TD>";  
echo "<TD width=50 CLASS=\"linea\"><A HREF=borrar.php?id=\".$row['id'].\"> ←  
    Borrar</A></TD>";  
//se termina la fila  
echo "</TR>";  
}  
//Comprueba si se ha terminado de poner líneas  
WHILE ($row=mysql_fetch_array($result));  
//como se ha terminado de poner líneas se acaba la tabla  
echo "</TABLE><br>";  
//coloca el enlace para volver al listado  
echo "<A HREF=\"index.php\">Volver</A>";  
}else{  
    //si no hay registros que coincidan con la búsqueda se le indica al usuario  
    echo "La búsqueda no coincide con ningún registro de la BBDD<br>";  
    //se coloca el enlace del volver al listado  
    echo "<A HREF=\"index.php\">Volver</A>";  
}  
?>  
</body>  
</html>
```

Por lo que puede ver el fichero es muy parecido al fichero index.php, se ha separado para entender mejor el cuadro de búsqueda. En la consulta a realizar se tiene en cuenta el cuadro de búsqueda enviado por POST.

2.9. Conclusiones

Como puede ver las modificaciones para la adaptación de estos scripts para entidades son laboriosas pero simples de realizar. Como ejercicio propuesto añada más campos a la tabla y modifique los ficheros para adaptarse a esos cambios.

Capítulo 3

Otros ejemplos

A continuación se listarán los ficheros de los otros directorios:

index.php tipos de alumno

```
<?php
//incluimos el fichero config.php para coger los datos de la conexión
include_once('../config.php');
//cogemos las variables de la conexión para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexión
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");
//rellenamos la consulta, cambiar alumnos por el nombre de la tabla a listar
$sql="SELECT * FROM tipos_de_alumno";
//echo $sql;
//se realiza la consulta
$result=mysql_query($sql,$conexion);

?>
<html>
  <head>
    <TITLE>Listado</TITLE>
    <LINK REL="stylesheet" TYPE="text/css"
      HREF="../estilo.css" />
  </head>
  <body>
    <A href="../index.php" name="alumnos">Inicio</A>
    <br />
    <A href="../alumnos/index.php" name="alumnos">
      Alumnos</A>
    <br />
    <A href="../asignaturas/index.php" name="asignaturas">
      Asignaturas</A>
    <br />
    <A href="index.php" name="tipos_de_alumno">Tipos de
      Alumno</A>
    <BR />
    <p>Listado de Tipos de Alumno.</p><?php
    //aquí se coloca el enlace que permite añadir una asignatura
    echo "<p><A HREF='nuevo.php'>Nuevo</A></p>";
    //comienza el formulario que permite la búsqueda
    echo "<form method='post' action='buscar.php'>";
    //aquí coloca una caja de texto donde se escribe la búsqueda
    echo "<input type='text' name='busqueda' size=20 maxlength =20>";
```

```
//aqui se coloca el boton a pulsar cuando se rellene la busqueda
echo "<input type=\"submit\" name=\"buscar\" value=\"Buscar\"></form><br />";
//se cogen los primeros resultados del listado
if($row= mysql_fetch_array($result)){
//si hay algun registro inicia la tabla que muestra los resultados del listado
echo "<TABLE BORDER='1'>";
//coloca la cabecera de la tabla, aqui se substituiria con los campos de la
tabla
//colocando mas <TD CLASS=\"titulo\">campo</TD> dependiendo de los campos que
haya
echo "<TR><TD CLASS=\"titulo\">Nombre</TD><TD colspan=3 CLASS=\"titulo\">
Opciones</TD></TR>";
//empieza el bucle que coloca las líneas del listado
DO
{
//Coloca una línea en el listado
//si hay mas campos se ponen más <TD width=300 CLASS=\"linea\">".$row["
nombre_campo"]."</TD>
//inicia la línea
echo "<TR>";
//pone el nombre del alumno
echo "<TD width=300 CLASS=\"linea\">".$row["nombre"]."</TD>";
//pone los enlaces de ver, editar y borrar asignaturas
echo "<TD CLASS=\"linea\"><A HREF=ver.php?id=".$row['id_tipo_de_alumno'].">Ver
</A></TD>";
echo "<TD width=50 CLASS=\"linea\"><A HREF=editar.php?id=".$row['
id_tipo_de_alumno'].">Editar</A></TD>";
echo "<TD width=50 CLASS=\"linea\"><A HREF=borrar.php?id=".$row['
id_tipo_de_alumno'].">Borrar</A></TD>";
//termina la fila
echo "</TR>";
}
//comprueba si hay mas registros que colocar
WHILE ($row=mysql_fetch_array($result));
//si no hay mas registros termina la tabla
echo "</TABLE><br>";
//coloca el enlace que permite volver a listar
echo "<A HREF=\"index.php\">Volver</A>";
}else{
//si no hay registros en la tabla informa de ello
echo "No hay ningún tipo de alumno en la BBDD<BR>";
//coloca el enlace que permite volver a listar
echo "<A HREF=\"index.php\">Volver</A>";
}
//termina el script
?>
</body>
</html>
```

ver.php de tipos de alumno

```
<?php
//incluimos el fichero config.php para coger los datos de la conexión
include_once('../config.php');
//cogemos las variables de las conexión para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexión
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");
//rellenamos la consulta, cambiar alumnos por el nombre de la tabla a listar
```

```
$sql="Select * from tipos_de_alumno WHERE id_tipo_de_alumno=".$_GET['id'].";";
//re realiza la consulta
$result=mysql_query($sql,$conexion);
print $sql;
//se coge el registro del alumno
$row= mysql_fetch_array($result);
?>
<html>
  <head>
    <TITLE>Alumno:<?php echo $row['nombre']; ?></TITLE>
    <LINK REL="stylesheet" TYPE="text/css"
      HREF="../estilo.css" />
  </head>
  <body>
    <?php
      //en esta parte se muestran los campos del alumno
      //se debe añadir líneas tantas como campos tenga la tabla: echo "<p>Nombre de
        campo:".$row["nombre_de_campo"]."</p>";
      //se muestra el código del alumno
      echo "<p>Código:".$row["id_tipo_de_alumno"]."</p>";
      //se muestra el nombre del alumno
      echo "<p>Nombre:".$row["nombre"]."</p>";
      //se coloca el enlace para que pueda editarse el alumno.
      echo "<p><A HREF=editar.php?id=".$_row['id_tipo_de_alumno']. ">Editar</A></TD></p>";
    >";
      // se coloca el enlace para borrar el alumno
      echo "<p><A HREF=borrar.php?id=".$_row['id_tipo_de_alumno']. ">Borrar</A></TD></p>";
    >";
      // se coloca el enlace para ir a listado de alumnos
      echo "<p><A HREF='index.php'>Atras</A></p>";
    ?>
  </body>
</html>
```

editar.php de tipos de alumno

```
<?php
//incluimos el fichero config.php para coger los datos de la conexi?n
include_once('../config.php');
//cogemos las variables de las coneci?n para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexi?n
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");
//rellenamos la consulta, cambiar alumnos por el nombre de la tabla a listar
$sql="Select * from tipos_de_alumno WHERE id_tipo_de_alumno=".$_GET['id'].";";
//re realiza la consulta
$result=mysql_query($sql,$conexion);
//print $sql;
//se coge el registro del alumno
$row= mysql_fetch_array($result);
?>
<html>
  <head>
    <TITLE>Editar Tipo de Alumno</TITLE>
    <LINK REL="stylesheet" TYPE="text/css"
      HREF="../estilo.css" />
  </head>
  <body>
    <?php
      //se mira si se est? enviando el formulario de edici?n del alumno
```

```
if(!isset( $_POST['editar']))
{
//Si se est? enviando el formulario
//echo "<p>Vuelva a intentarlo</p>";
//se abre el formulario que envia los datos de las modificaciones de ese ←
registro
        echo "<form method=\"post\" action=\"editar.php?id=".$_GET['id'] ←
        '\"]\" >";
// muestra el c?digo del alumno
        echo "Codigo:".$row['id_tipo_de_alumno'].<br>";
// a partir de aqui se muestran los campos a rellenar en el formulario
// si se substituye echo "Nombre : <input type=\"text\" name=\"nombre\" value ←
        =\"".$_row['nombre'].\"\" maxlength =200 size=40><br />";
// por echo "Nombre : <input type=\"text\" name=\"nombre_de_campo\" value=\"".$_ ←
        $row['nombre'].\"\" maxlength =longitud_maxima_de_campo
// size=tama?o_dte_la_caja><br />";
// se muestra la caja del nombre del alumno
        echo "Nombre : <input type=\"text\" name=\"nombre\" value=\"".$_ ←
        $row['nombre'].\"\" maxlength =100 size=40><br />";
// se pone el bot?n de guardar los datos
        echo "<input type=\"submit\" name=\"editar\" value=\"Guardar ←
        \"></form><br />";
}else{
// si se env?a el formulario hay que coger los datos del formulario y meterlos ←
en la BBDD
// aqui se deben a?adir tantos isset($_POST['nombre_de_campo']) && como campos ←
haya en la bbdd
        if(isset($_POST['nombre'])) {
//se desactiva el contenido de la variable $sql
                unset($sql);
// se rellena la consulta que inserta los datos
//se ponen tantos 'nombre_de_campo' = '".$_POST['nombre_de_campo'].\"',
                $sql="UPDATE `tipos_de_alumno` SET `nombre` = '".$_POST['nombre ←
                '].\"' WHERE `id_tipo_de_alumno` =".$_GET['id'].\" LIMIT 1 ";
                //echo $sql;
//se envia la inserci?n de datos en la BBDD
                $result=mysql_query($sql,$conexion);
//Se informa por la p?gina web que el alumno ha sido modificado
                echo "<p>Tipo de alumno Modificado</p>";
//se coloca el bot?n que hace volver al listado de alumnos
                echo "<p><A HREF='index.php'>Volver</A></p>";
        }else{
// se informa al usuario que no se han rellenado los campos
echo "Vuelva a intentarlo.<br>";
//se abre el formulario que envia los datos de las modificaciones de ←
ese registro
        echo "<form method=\"post\" action=\"editar.php?id=".$_GET['id'] ←
        '\"]\" >";
// muestra el c?digo del alumno
        echo "Codigo:".$row['id_tipo_de_alumno'].<br>";
// a partir de aqui se muestran los campos a rellenar en el formulario
// si se substituye echo "Nombre : <input type=\"text\" name=\"nombre\" value ←
        =\"".$_row['nombre'].\"\" maxlength =200 size=40><br />";
// por echo "Nombre : <input type=\"text\" name=\"nombre_de_campo\" value=\"".$_ ←
        $row['nombre'].\"\" maxlength =longitud_maxima_de_campo
// size=tama?o_dte_la_caja><br />";
// se muestra la caja del nombre del alumno
        echo "Nombre : <input type=\"text\" name=\"nombre\" value=\"".$_ ←
        $row['nombre'].\"\" maxlength =200 size=40><br />";
// se pone el bot?n de guardar los datos
        echo "<input type=\"submit\" name=\"editar\" value=\"Guardar ←
```

```
        \"></form><br />";
    }
}
?>
</body>
</html>
```

borrar.php de tipos de alumno

```
<?php
//incluimos el fichero config.php para coger los datos de la conexión
include_once('../config.php');
//cogemos las variables de las conexión para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexión
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");
//rellenamos la consulta, cambiar alumnos por el nombre de la tabla a listar
$sql="Select * from alumnos WHERE id=".$_GET['id'].>";";
//re realiza la consulta
$result=mysql_query($sql,$conexion);
//print $sql;
//se coge el registro del alumno
$row= mysql_fetch_array($result);
?>
<html>
<head>
<TITLE>Borrar Asignatura</TITLE>
<LINK REL="stylesheet" TYPE="text/css"
    HREF="../estilo.css" />
</head>
<body>
<?php
    // se rellena la consulta que borra el alumno se substituye alumno por el ←
    nombre de la tabla
    $sql="DELETE from `tipos_de_alumno` WHERE `id_tipo_de_alumno` =".$_GET['id']. " ←
        LIMIT 1 ;";
    //echo $sql;
    // se ejecuta la consulta se borrado
    $result=mysql_query($sql,$conexion);
    // se informa de que se ha eliminado un alumno
    echo "<p>Asignatura eliminada</p>";
    // se coloca el enlace que permite volver al listado
    echo "<p><A HREF='index.php'>Volver</A></p>";

    ?>
</body>
</html>
```

buscar.php de tipos de alumno

```
<?php
//incluimos el fichero config.php para coger los datos de la conexión
include_once('../config.php');
//cogemos las variables de las conexión para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexión
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");
```

```
?>
<html>
  <head>
    <TITLE>Búsqueda de Asignaturas</TITLE>
    <LINK REL="stylesheet" TYPE="text/css"
      HREF="../estilo.css" />
  </head>
  <body>
    <p>Búsqueda de Asignaturas.</p><?php
    // coloca el enlace que permite añadir un alumno
    echo "<p><A HREF='nuevo.php'>Nuevo</A></p>";
    // se inicia el formulario de la búsqueda
    echo "<form method='post' action='buscar.php'>";
    // se indica que el campo de busqueda tiene 20 caracteres de longitud
    echo "<input type='text' name='busqueda' size=20 maxlength=100 value='". ←
      $_POST['busqueda']."'>";
    // se coloca el boton que se paretara para realizar la busqueda
    echo "<input type='submit' name='buscar' value='Buscar'></form><br>";
    // se rellena la busqueda por todos los campos de la tabla, se debe substituir ←
      alumnos por el nombre de la tabla y poner
    // tantos nombre_de_campo LIKE '%"$_POST['busqueda']."' OR como campos haya.
    $sql="SELECT * FROM asignaturas where nombre LIKE '%"$_POST['busqueda']."'";
    // se realiza la busqueda
    $result=mysql_query($sql,$conexion);
    //Se comprueba si hay resultados en la busqueda
    if($row= mysql_fetch_array($result)){
    // si hay resultados crea la tabla
    echo "<TABLE BORDER='1'>";
    //pone los campos de la tabla tiene que haber tantos <TD CLASS="titulo"> ←
      Nombre_de_campo</TD> como campos haya
    echo "<TR><TD CLASS='titulo'>Nombre</TD><TD colspan=3 CLASS='titulo'> ←
      Opciones</TD></TR>";
    //coloca las líneas de la tabla
    DO
    {
    //aquí se ponen tantas líneas <TD width=300 CLASS="linea">".$row[" ←
      nombre_de_campo"]."</TD> como campos haya en la tabla
    // se coloca el inicio de la fila
    echo "<TR>";
    //se coloca el campos del nombre
    echo "<TD width=300 CLASS='linea'>".$row["nombre"]."</TD>";
    //se colocan los enlaces de ver editar y borrar alumno
    echo "<TD CLASS='linea'><A HREF=ver.php?id=".$row['id']. ">Ver</A></TD>";
    echo "<TD width=50 CLASS='linea'><A HREF=editar.php?id=".$row['id']. ">Editar ←
      </A></TD>";
    echo "<TD width=50 CLASS='linea'><A HREF=borrar.php?id=".$row['id']. ">Borrar ←
      </A></TD>";
    //se termina la fila
    echo "</TR>";
    }
    //Comprueba si se ha terminado de poner líneas
    WHILE ($row=mysql_fetch_array($result));
    //como se ha terminado de poner líneas se acaba la tabla
    echo "</TABLE><br>";
    //coloca el enlace para volver al listado
    echo "<A HREF='index.php'>Volver</A>";
    }else{
    //si no hay registros que coincidan con la busqueda se le indica al usuario
    echo "La búsqueda no coincide con ningún registro de la BBDD<BR>";
    //se coloca el enlace del volver al listado
    echo "<A HREF='index.php'>Volver</A>";
  }
}
```



```
}  
?>  
</body>  
</html>
```

nuevo.php de tipos de alumno

```
<?php  
//incluimos el fichero config.php para coger los datos de la conexión  
include_once('../config.php');  
//cogemos las variables de las conexión para que se puedan utilizar en el script  
global $server, $database, $user, $passwd;  
//abrimos la conexión  
$conexion = mysql_connect($server,$user,$passwd);  
//elegimos la BBDD  
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");  
  
?>  
<html>  
  <head>  
    <TITLE>Búsqueda de Asignaturas</TITLE>  
    <LINK REL="stylesheet" TYPE="text/css"  
      HREF="../estilo.css" />  
  </head>  
  <body>  
    <p>Búsqueda de Asignaturas.</p><?php  
    // coloca el enlace que permite añadir un alumno  
    echo "<p><A HREF='nuevo.php'>Nuevo</A></p>";  
    // se inicia el formulario de la búsqueda  
    echo "<form method='post' action='buscar.php'>";  
    // se indica que el campo de búsqueda tiene 20 caracteres de longitud  
    echo "<input type='text' name='busqueda' size=20 maxlength=100 value='". ←  
      $_POST['busqueda']."'>";  
    // se coloca el boton que se paretara para realizar la busqueda  
    echo "<input type='submit' name='buscar' value='Buscar'></form><br>";  
    // se rellena la busqueda por todos los campos de la tabla, se debe substituir ←  
      alumnos por el nombre de la tabla y poner  
    // tantos nombre_de_campo LIKE ' %'$_POST['busqueda']."' OR como campos haya.  
    $sql="SELECT * FROM asignaturas where nombre LIKE ' %'$_POST['busqueda']."'";  
    // se realiza la busqueda  
    $result=mysql_query($sql,$conexion);  
    //Se comprueba si hay resultados en la busqueda  
    if($row= mysql_fetch_array($result)){  
      // si hay resultados crea la tabla  
      echo "<TABLE BORDER='1'>";  
      //pone los campos de la tabla tiene que haber tantos <TD CLASS="titulo"> ←  
        Nombre_de_campo</TD> como campos haya  
      echo "<TR><TD CLASS='titulo'>Nombre</TD><TD colspan=3 CLASS='titulo'> ←  
        Opciones</TD></TR>";  
      //coloca las líneas de la tabla  
      DO  
      {  
        //aquí se ponen tantas líneas <TD width=300 CLASS="linea">$.row[" ←  
          nombre_de_campo"]."</TD> como campos haya en la tabla  
        // se coloca el inicio de la fila  
        echo "<TR>";  
        //se coloca el campos del nombre  
        echo "<TD width=300 CLASS='linea'>$.row['nombre']."</TD>";  
        //se colocan los enlaces de ver editar y borrar alumno  
        echo "<TD CLASS='linea'><A HREF=ver.php?id=$.row['id']>Ver</A></TD>";  
        echo "<TD width=50 CLASS='linea'><A HREF=editar.php?id=$.row['id']>Editar ←  
          </A></TD>";
```

```
echo "<TD width=50 CLASS=\"linea\"><A HREF=borrar.php?id=\".$row['id'].\">Borrar ←  
    </A></TD>";  
//se termina la fila  
echo "</TR>";  
}  
//Comprueba si se ha terminado de poner líneas  
WHILE ($row=mysql_fetch_array($result));  
//como se ha terminado de poner líneas se acaba la tabla  
echo "</TABLE><br>";  
//coloca el enlace para volver al listado  
echo "<A HREF=\"index.php\">Volver</A>";  
}else{  
//si no hay registros que coincidan con la búsqueda se le indica al usuario  
echo "La búsqueda no coincide con ningún registro de la BBDD<BR>";  
//se coloca el enlace del volver al listado  
echo "<A HREF=\"index.php\">Volver</A>";  
  
}  
?>  
</body>  
</html>
```

Capítulo 4

Inclusión de relaciones

4.1. Directorio de Alumnos

index.php de alumnos

```
<?php
//incluimos el fichero config.php para coger los datos de la conexión
include_once('../config.php');
//cogemos las variables de las conexión para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexión
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");

?>
<html>
  <head>
    <TITLE>Búsqueda de Asignaturas</TITLE>
    <LINK REL="stylesheet" TYPE="text/css"
      HREF="../estilo.css" />
  </head>
  <body>
    <p>Búsqueda de Asignaturas.</p><?php
    // coloca el enlace que permite añadir un alumno
    echo "<p><A HREF='nuevo.php'>Nuevo</A></p>";
    // se inicia el formulario de la búsqueda
    echo "<form method='post' action='buscar.php'>";
    // se indica que el campo de busqueda tiene 20 caracteres de longitud
    echo "<input type='text' name='busqueda' size=20 maxlength=100 value =\
      \"$_POST['busqueda']\">";
    // se coloca el boton que se paretara para realizar la busqueda
    echo "<input type='submit' name='buscar' value='Buscar'></form><br>";
    // se rellena la busqueda por todos los campos de la tabla, se debe ←
    substituir alumnos por el nombre de la tabla y
    // poner tantos nombre_de_campo LIKE ' %'$_POST['busqueda']%' OR como ←
    campos haya.
    $sql="SELECT * FROM asignaturas where nombre LIKE ' %'$_POST['busqueda' ←
      ']'%'";
    // se realiza la busqueda
    $result=mysql_query($sql,$conexion);
    //Se comprueba si hay resultados en la busqueda
    if($row= mysql_fetch_array($result)){
    // si hay resultados crea la tabla
```

```
echo "<TABLE BORDER='1'>";
//pone los campos de la tabla tiene que haber tantos <TD CLASS=\"titulo\"> ←
Nombre_de_campo</TD> como campos haya
echo "<TR><TD CLASS=\"titulo\">Nombre</TD><TD colspan=3 CLASS=\"titulo\"> ←
Opciones</TD></TR>";
//coloca las líneas de la tabla
DO
{
//aqui se ponen tantas líneas <TD width=300 CLASS=\"linea\">".$row[" ←
nombre_de_campo"]."</TD> como campos haya en la tabla
// se coloca el inicio de la fila
echo "<TR>";
//se coloca el campos del nombre
echo "<TD width=300 CLASS=\"linea\">".$row["nombre"]."</TD>";
//se colocan los enlaces de ver editar y borrar alumno
echo "<TD CLASS=\"linea\"><A HREF=ver.php?id=".$row['id']. ">Ver</A></TD>";
echo "<TD width=50 CLASS=\"linea\"><A HREF=editar.php?id=".$row['id']. "> ←
Editar</A></TD>";
echo "<TD width=50 CLASS=\"linea\"><A HREF=borrar.php?id=".$row['id']. "> ←
Borrar</A></TD>";
//se termina la fila
echo "</TR>";
}
//Comprueba si se ha terminado de poner líneas
WHILE ($row=mysql_fetch_array($result));
//como se ha terminado de poner líneas se acaba la tabla
echo "</TABLE><br>";
//coloca el enlace para volver al listado
echo "<A HREF=\"index.php\">Volver</A>";
}else{
//si no hay registros que coincidan con la busqueda se le indica al usuario
echo "La búsqueda no coincide con ningún registro de la BBDD<BR>";
//se coloca el enlace del volver al listado
echo "<A HREF=\"index.php\">Volver</A>";

}
?>
</body>
</html>
```

Nótese como en la tercera parte del fichero hay una nueva consulta en busca de los nombres de los tipos de alumno para poder encontrar el nombre de la que está relacionada con cada alumno. Por lo demás es como cualquier otro index.php de directorio.

ver.php de alumnos

```
<?php
//incluimos el fichero config.php para coger los datos de la conexión
include_once('../config.php');
//cogemos las variables de las conexión para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexión
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");
//rellenamos la consulta, cambiar alumnos por el nombre de la tabla a listar
$sql="Select * from alumnos WHERE id=".$_GET['id']. " ";
//re realiza la consulta
$result=mysql_query($sql,$conexion);
//print $sql;
//se coge el registro del alumno
$row= mysql_fetch_array($result);
?>
```

```
<html>
<head>
  <TITLE>Alumno:<?php echo $row['nombre']; ?></TITLE>
  <LINK REL="stylesheet" TYPE="text/css"
    HREF="../../../estilo.css" />
</head>
<body>
  <?php
    //en esta parte se muestran los campos del alumno
    //se debe añadir líneas tantas como campos tenga la tabla: echo "<p>Nombre de ↵
      campo:".$row["nombre_de_campo"]."</p>";
    //se muestra el código del alumno
    echo "<p>Código:".$row["id"]."</p>";
    //se muestra el nombre del alumno
    echo "<p>Nombre:".$row["nombre"]."</p>";
    //se muestra el dni del alumno
    echo "<p>DNI:".$row["dni"]."</p>";
    $sql2="Select * from tipos_de_alumno where id_tipo_de_alumno=".$row[' ↵
      id_tipo_de_alumno'].";";
    //echo $sql2;
    //re realiza la consulta para meter en $result2 los tipos de alumnos
    $result2=mysql_query($sql2,$conexion);
    //coloca en $row2 el primer tipo de alumno
    $row2= mysql_fetch_array($result2);
    echo "<p>Tipo de Alumno:".$row2["nombre"]."</p>";
    //se coloca el enlace para que pueda editarse el alumno.
    echo "<p><A HREF=editar.php?id=".$row['id']. ">Editar</A></TD></p>";
    // se coloca el enlace para borrar el alumno
    echo "<p><A HREF=borrar.php?id=".$row['id']. ">Borrar</A></TD></p>";
    // se coloca el enlace para ver la matriculas de alumno
    echo "<p><A HREF=ver_matriculas.php?id=".$row['id']. ">Ver Matriculas</A></TD ↵
      ></p>";
    // se coloca el enlace para ir a listado de alumnos
    echo "<p><A HREF='index.php'>Atras</A></p>";
    ?>
  </body>
</html>
```

Aquí al final de la tercera parte se ve como se muestra un enlace a un nuevo fichero que permite la visualización de las matrículas de dicho alumno.

editar.php de alumnos

```
<?php
//incluimos el fichero config.php para coger los datos de la conexi?n
include_once('../config.php');
//cogemos las variables de las conexi?n para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexi?n
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");
//rellenamos la consulta, cambiar alumnos por el nombre de la tabla a listar
$sql="Select * from alumnos WHERE id=".$_GET['id'].";";
//re realiza la consulta
$result=mysql_query($sql,$conexion);
//print $sql;
//se coge el registro del alumno
$row= mysql_fetch_array($result);
?>
<html>
  <head>
```

```
<TITLE>Nuevo Alumno</TITLE>
<LINK REL="stylesheet" TYPE="text/css"
HREF="../../../estilo.css" />
</head>
<body>
<?php
//se mira si se est? enviando el formulario de edici?n del alumno
if(!isset( $_POST['editar']))
{
//Si se est? enviando el formulario
//echo "<p>Vuelva a intentarlo</p>";
//se abre el formulario que envia los datos de las modificaciones de ese ←
registro
        echo "<form method=\"post\" action=\"editar.php?id=\".$_GET[' ←
        id'].\"\" >";
// muestra el c?digo del alumno
        echo "Codigo:".$row['id'].<br>";
// a partir de aqui se muestran los campos a rellenar en el formulario
// si se substituye echo "Nombre : <input type=\"text\" name=\"nombre\" value ←
        =\"\".$row['nombre'].\"\" maxlength =200 size=40><br>";
// por echo "Nombre : <input type=\"text\" name=\"nombre_de_campo\" value ←
        =\"\".$row['nombre'].\"\" maxlength =longitud_maxima_de_campo
// size=tama?o_dte_la_caja><br>";
// se muestra la caja del nombre del alumno
        echo "Nombre : <input type=\"text\" name=\"nombre\" value ←
        =\"\".$row['nombre'].\"\" maxlength =200 size=40><br>";
// se muestra la caja del dni
        echo "DNI : <input type=\"text\" name=\"dni\" value=\"\".$row ←
        ['dni'].\"\" maxlength =10 size=10><br>";
//inicia la combo de los tipos
echo "Tipo de Alumno:";
echo "<select name=\"tipo_de_alumno\">";
//rellena la sql para buscar las asignaturas
$sql2="Select * from tipos_de_alumno;";
//re realiza la consulta para meter en $result2 las asignaturas
$result2=mysql_query($sql2,$conexion);
//coloca en $row2 la primera asignatura
$row2= mysql_fetch_array($result2);
DO
{
//coloca la asignatura en el listado
echo "<option value=\"\".$row2['id_tipo_de_alumno'].\"\"";
if($row2['id_tipo_de_alumno']==$row['id_tipo_de_alumno'])
        echo "selected";
echo ">\".$row2['nombre'].</option>";
}
//comprueba si hay m?s tipos de alumno
while ($row2= mysql_fetch_array($result2));
//si no hay mas tipos de alumno cierra el listado
echo "</select> <br><br>";
// se pone el bot?n de guardar los datos
        echo "<input type=\"submit\" name=\"editar\" value=\"Guardar ←
        \"></form><br>";
}else{
// si se env?a el formulario hay que coger los datos del formulario y ←
meterlos en la BBDD
// aqui se deben a?adir tantos isset($_POST['nombre_de_campo']) && como ←
campos haya en la bbdd
        if(isset($_POST['nombre']) && isset($_POST['dni']) && isset($_POST[' ←
        tipo_de_alumno'])){
//se desactiva el contenido de la variable $sql
        unset($sql);
```

```
// se rellena la consulta que inserta los datos
//se ponen tantos 'nombre_de_campo' = '".$_POST['nombre_de_campo'].",',

        $sql="UPDATE `alumnos` SET `nombre` = '".$_POST['nombre' ↵
            ']."', `dni` = '".$_POST['dni']."', `id_tipo_de_alumno` = ↵
            '".
        $_POST['tipo_de_alumno']."' WHERE `id` = '".$_GET['id'].'" LIMIT ↵
            1 ";
        //echo $sql;
        //print_r ($_POST);
//se envia la inserci?n de datos en la BBDD
        $result=mysql_query($sql,$conexion);
//Se informa por la p?gina web que el alumno ha sido modificado
        echo "<p>Alumno Modificado</p>";
//se coloca el bot?n que hace volver al listado de alumnos
        echo "<p><A HREF='index.php'>Volver</A></p>";
    }else{
        // se informa al usuario que no se han rellenado los campos
print_r($_POST);
        echo "Vuelva a intentarlo.<br>";
        //se abre el formulario que envia los datos de las modificaciones de ↵
        ese registro
        echo "<form method=\"post\" action=\"editar.php?id=\".$_GET[' ↵
            id'].\"\" >";
// muestra el c?digo del alumno
        echo "Codigo:".$row['id'].<br>";
// a partir de aqui se muestran los campos a rellenar en el formulario
// si se substituye echo "Nombre : <input type=\"text\" name=\"nombre\" value ↵
        =\"".$_row['nombre'].\"\" maxlength =200 size=40><br>";
// por echo "Nombre : <input type=\"text\" name=\"nombre_de_campo\" value ↵
        =\"".$_row['nombre'].\"\" maxlength =longitud_maxima_de_campo
// size=tama?o_dte_la_caja><br>";
// se muestra la caja del nombre del alumno
        echo "Nombre : <input type=\"text\" name=\"nombre\" value ↵
            =\"".$_row['nombre'].\"\" maxlength =200 size=40><br>";
// se muestra la caja del dni
        echo "DNI : <input type=\"text\" name=\"dni\" value=\"".$_row ↵
            ['dni'].\"\" maxlength =10 size=10><br>";
// se pone el bot?n de guardar los datos
        echo "<input type=\"submit\" name=\"editar\" value=\"Guardar ↵
            \"></form><br>";
    }
}
?>
</body>
</html>
```

Cabe destacar la segunda b?sq?da en la tabla de tipos de alumno para ocnstuir el tag select (combobox)

borrar.php de alumnos

```
<?php
//incluimos el fichero config.php para coger los datos de la conexi?n
include_once('../config.php');
//cogemos las variables de las conexi?n para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexi?n
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");
//rellenamos la consulta, cambiar alumnos por el nombre de la tabla a listar
$sql="Select * from alumnos WHERE id='".$_GET['id'].\"";
```

```
//re realiza la consulta
$result=mysql_query($sql,$conexion);
//print $sql;
//se coge el registro del alumno
$row= mysql_fetch_array($result);
?>
<html>
<head>
<TITLE>Borrar Alumno</TITLE>
<LINK REL="stylesheet" TYPE="text/css"
HREF="../estilo.css" />
</head>
<body>
<?php
// se rellena la consulta que borra el alumno se substituye alumno por el ←
nombre de la tabla
$sql="DELETE from 'alumnos' WHERE 'id' ='".$_GET['id']."' LIMIT 1 ;";
//echo $sql;
// se ejecuta la consulta se borrado
$result=mysql_query($sql,$conexion);
// se informa de que se ha eliminado un alumno
echo "<p>Alumno eliminado</p>";
// se coloca el enlace que permite volver al listado
echo "<p><A HREF='index.php'>Volver</A></p>";

?>
</body>
</html>
```

buscar.php de alumnos

```
<?php
//incluimos el fichero config.php para coger los datos de la conexión
include_once('../config.php');
//cogemos las variables de las conexión para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexión
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");

?>
<html>
<head>
<TITLE>Búsqueda de Alumnos</TITLE>
<LINK REL="stylesheet" TYPE="text/css"
HREF="../estilo.css" />
</head>
<body>
<p>Búsqueda de Alumnos.</p><?php
// coloca el enlace que permite añadir un alumno
echo "<p><A HREF='nuevo.php'>Nuevo</A></p>";
// se inicia el formulario de la búsqueda
echo "<form method='post' action='buscar.php'>";
// se indica que el campo de búsqueda tiene 20 caracteres de longitud
echo "<input type='text' name='busqueda' size=20 maxlength =20 value='\". ←
$_POST['busqueda'].\">";
// se coloca el boton que se paretara para realizar la busqueda
echo "<input type='submit' name='buscar' value='\"Buscar\"></form><br>";
// se rellena la busqueda por todos los campos de la tabla, se debe ←
substituir alumnos por el nombre de la tabla y poner tantos ←
nombre_de_campo LIKE ' %'$_POST['busqueda']%' OR como campos haya.
```



```
$sql="SELECT * FROM alumnos where nombre LIKE ' %'$_POST['busqueda'].%' OR  
dni LIKE ' %'$_POST['busqueda'].%'";  
// se realiza la busqueda  
$result=mysql_query($sql,$conexion);  
//Se comprueba si hay resultados en la busqueda  
if($row= mysql_fetch_array($result)){  
// si hay resultados crea la tabla  
echo "<TABLE BORDER='1'>";  
//pone los campos de la tabla tiene que haber tantos <TD CLASS=\"titulo\"> ←  
Nombre_de_campo</TD> como campos haya  
echo "<TR><TD CLASS=\"titulo\">Nombre</TD><TD CLASS=\"titulo\">DNI</TD><TD ←  
colspan=3 CLASS=\"titulo\">Opciones</TD></TR>";  
//coloca las líneas de la tabla  
DO  
{  
//aquí se ponen tantas líneas <TD width=300 CLASS=\"linea\">".$row[" ←  
nombre_de_campo"]."</TD> como campos haya en la tabla  
// se coloca el inicio de la fila  
echo "<TR>";  
//se coloca el campos del nombre  
echo "<TD width=300 CLASS=\"linea\">".$row["nombre"]."</TD>";  
//se coloca el campo del dni  
echo "<TD width=160 CLASS=\"linea\">".$row["dni"]."</TD>";  
//se colocan los enlaces de ver editar y borrar alumno  
echo "<TD CLASS=\"linea\"><A HREF=ver.php?id=".$row['id'].>Ver</A></TD>";  
echo "<TD width=50 CLASS=\"linea\"><A HREF=editar.php?id=".$row['id'].> ←  
Editar</A></TD>";  
echo "<TD width=50 CLASS=\"linea\"><A HREF=borrar.php?id=".$row['id'].> ←  
Borrar</A></TD>";  
//se termina la fila  
echo "</TR>";  
}  
//Comprueba si se ha terminado de poner líneas  
WHILE ($row=mysql_fetch_array($result));  
//como se ha terminado de poner líneas se acaba la tabla  
echo "</TABLE><br>";  
//coloca el enlace para volver al listado  
echo "<A HREF=\"index.php\">Volver</A>";  
}else{  
//si no hay registros que coincidan con la busqueda se le indica al usuario  
echo "La búsqueda no coincide con ningún registro de la BBDD<br>";  
//se coloca el enlace del volver al listado  
echo "<A HREF=\"index.php\">Volver</A>";  
  
}  
?>  
</body>  
</html>
```

nuevo.php de alumnos

```
<?php  
//incluimos el fichero config.php para coger los datos de la conexión  
include_once('../config.php');  
//cogemos las variables de las conexión para que se puedan utilizar en el script  
global $server, $database, $user, $passwd;  
//abrimos la conexión  
$conexion = mysql_connect($server,$user,$passwd);  
//elegimos la BBDD  
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");  
  
?>
```

```
<html>
<head>
  <TITLE>Nuevo Alumno</TITLE>
  <LINK REL="stylesheet" TYPE="text/css"
    HREF="../../../estilo.css" />
</head>
<body>
  <?php
    //comprueba si se entas enviando datos para guardar
    //si no es asi es que tiene que presentar el formulario
    if(!isset( $_POST['nuevo']))
    {
      //coloca el principio del formulario para que vuelva al mismo script cuando ←
      se rellene
      echo "<form method=\"post\" action=\"nuevo.php\">";
      //aqui se deben poner tantas líneas echo como campos haya
      //cambiando el echo "Nombre : <input type=\"text\" name=\"nombre\" maxlength ←
      =200 size=40><br>";
      // por echo "Nombre : <input type=\"text\" name=\"nombre_campo\" maxlength= ←
      longitud_campo size=tamaño_caja_texto><br>";
      // se muestra la caja del nombre del alumno
      echo "Nombre : <input type=\"text\" name=\"nombre\" maxlength ←
      =100 size=40><br>";
      // se muestra la caja del dni del alumno
      echo "DNI : <input type=\"text\" name=\"dni\" maxlength =15 ←
      size=9><br>";
      //inicia la combo de los tipos
      echo "Tipo de Alumno:";
      echo "<select name=\"tipo_de_alumno\">";
      //rellena la sql para buscar las asignaturas
      $sql2="Select * from tipos_de_alumno;";
      //re realiza la consulta para meter en $result2 las asignaturas
      $result2=mysql_query($sql2,$conexion);
      //coloca en $row2 la primera asignatura
      $row2= mysql_fetch_array($result2);
      DO
      {
        //coloca la asignatura en el listado
        echo "<option value=\"".$row2['id_tipo_de_alumno']."\">".$row2['nombre'].</ ←
        option>";
      }
      //comprueba si hay más tipos de alumno
      while ($row2= mysql_fetch_array($result2));
      //si no hay mas tipos de alumno cierra el listado
      echo "</select> <br><br>";
      ////se coloca el botón de guardar
      echo "<input type=\"submit\" name=\"nuevo\" value=\"Guardar\"></form ←
      ><br>";
    }else{
      //si se han metido los datos en el formulario
      //se deben colocar tantos isset($_POST['nombre_campo']) && como campos haya ←
      en la tabla
      if(isset($_POST['nombre']) && isset($_POST['dni']) && isset($_POST[' ←
      tipo_de_alumno'])){
        //en esta linea se ponen los nombres de los campos separados por comas y en ←
        la parte de values lo mismo pero con variables $_POST['nombre_campo']
        //se reinician las variables de la busqueda
        unset($result2);
        unset($row2);

        //rellena la sql para buscar las asignaturas
        $sql2="Select * from tipos_de_alumno;";
```

```
//echo $sql2;
//re realiza la consulta para meter en $result2 las asignaturas
$result2=mysql_query($sql2,$conexion);
//echo $result2;
//coloca en $row2 la primera asignatura
$row2= mysql_fetch_array($result2);
//busca que id tiene la asignatura
DO
{
//si el nombre coincide con el elegido en el formulario coge el nombre y el ←
id de la asignatura
if($row2['id_tipo_de_alumno']==$_POST['tipo_de_alumno']){
//echo "hemos entrao!!";
$nombre_asig=$row2['nombre'];
//echo $nombre_asig;
$id_tipo_de_alumno=$row2['id_tipo_de_alumno'];
//echo $id_asig;
break;
}
}
//comprueba si hay más asignaturas
while ($row2= mysql_fetch_array($result2));
$sql="INSERT INTO `alumnos` ( `id` , `nombre` , `dni` , `id_tipo_de_alumno` )
VALUES (NULL , '". $_POST['nombre']."' , '". $_POST['dni']."' , ←
'".$_id_tipo_de_alumno."');"
//echo $sql;
//se ejecuta la inserción de datos
$result=mysql_query($sql,$conexion);
//se informa de que se añadido un nuevo alumno
echo "<p>Alumno Añadido</p>";
//se coloca el enlace que permite volver al listado
echo "<p><A HREF='index.php'>Volver</A></p>";
}
}
// si no nos han metido campos
echo "<p>Vuelva a intentarlo</p>";
// se vuelve a colocar el formulario con los mismos campos que ahora, la ←
diferencia es que se pone tantas veces como campos haya en los VALUE el ←
$_POST['nombre_de_campo'] para que no haya que volver a teclearlo

echo "<form method=\"post\" action=\"nuevo.php?id=\".$_GET['id ←
'"]."\">";
//aquí se deben poner tantas líneas echo como campos haya
//cambiando el echo "Nombre : <input type=\"text\" name=\"nombre\" maxlength ←
=200 size=40><br>";
// por echo "Nombre : <input type=\"text\" name=\"nombre_campo\" maxlength= ←
longitud_campo size=tamaño_caja_texto><br>";
// se muestra la caja del nombre del alumno
echo "Nombre : <input type=\"text\" name=\"nombre\" maxlength ←
=100 size=40><br>";
// se muestra la caja del dni del alumno
echo "DNI : <input type=\"text\" name=\"nombre\" maxlength ←
=15 size=9><br>";
//inicia la combo de los tipos
echo "<select name=\"tipo_de_alumno\">";
//rellena la sql para buscar las asignaturas
$sql2="Select * from tipos_de_alumno;";
//re realiza la consulta para meter en $result2 las asignaturas
$result2=mysql_query($sql2,$conexion);
//coloca en $row2 la primera asignatura
$row2= mysql_fetch_array($result2);
DO
{
```

```
//coloca la asignatura en el listado
echo "<option value=\"\".$row2['id_tipo_de_alumno'].\">\".$row2['nombre'].\"</option>";
}
//comprueba si hay más tipos de alumno
while ($row2= mysql_fetch_array($result2));
//si no hay mas tipos de alumno cierra el listado
echo "</select> <br><br>";
////se coloca el botón de guardar
echo "<input type=\"submit\" name=\"nuevo\" value=\"Guardar\"></form>";
}
}
?>
</body>
</html>
```

Cabe destacar la segunda búsqueda en la tabla de tipos de alumno para construir el tag select (combobox)

matricular.php alumnos

```
<?php
//incluimos el fichero config.php para coger los datos de la conexión
include_once('../config.php');
//cogemos las variables de la conexión para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexión
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");
//rellenamos la consulta, cambiar alumnos por el nombre de la tabla a listar
$sql="Select * from alumnos WHERE id=\"$_GET['id']\".";
//print $sql;
//re realiza la consulta
$result=mysql_query($sql,$conexion);
//mete el alumno en la variable $row
$row= mysql_fetch_array($result);

//print $sql2;
?>
<html>
<head>
<TITLE>Matricular
Alumno:<?php echo $row['nombre']; ?></TITLE>
<LINK REL="stylesheet" TYPE="text/css"
HREF="../estilo.css" />
</head>
<body>
<?php
//comprueba si se están enviando datos para guardar
//si no es así es que tiene que presentar el formulario
if(!isset($_POST['nuevo']))
{
//coloca el principio del formulario para que vuelva al mismo script cuando se rellene
echo "<form method=\"post\" action=\"matricular.php?id=\"$_GET['id']\">";
//aquí se deben poner tantas líneas echo como campos haya
//cambiando el echo "Nombre : <input type=\"text\" name=\"nombre\" maxlength=
=200 size=40><br>";
// por echo "Nombre : <input type=\"text\" name=\"nombre_campo\" maxlength=
longitud_campo size=tamaño_caja_texto><br>";
```

```
//se muestra el código del alumno
echo "<p>Código: ".$row["id"]."</p>";
//se muestra el nombre del alumno
echo "<p>Nombre: ".$row["nombre"]."</p>";

//se coloca la caja oculta con el id del alumno
echo "<input type=\"hidden\" name=\"id_alum\" value=\"".$row['id'] ↵
    '].\">";
//inicia la combo de las asignaturas
echo "<select name=\"asignatura\">";
//rellena la sql para buscar las asignaturas
$sql2="Select * from asignaturas;";
//re realiza la consulta para meter en $result2 las asignaturas
$result2=mysql_query($sql2,$conexion);
//coloca en $row2 la primera asignatura
$row2= mysql_fetch_array($result2);
DO
{
    //coloca la asignatura en el listado
    echo "<option value=\"".$row2['id'].\">".$row2['nombre'].\"</option>";
}
//comprueba si hay más asignaturas
while ($row2= mysql_fetch_array($result2));
//si no hay mas asignatutas cierra el listado de asignaturas
echo "</select> <br><br>";
////se coloca el botón de guardar
echo "<input type=\"submit\" name=\"nuevo\" value=\"Guardar\"></form ↵
    ><br>";
}else{
    //si se han metido los datos en el formulario
    //se deben colocar tantos isset($_POST['nombre_campo']) && como campos haya ↵
        en la tabla
        if(isset($_POST['id_alum']) && isset($_POST['asignatura']))){

//se reinician las variables de la busqueda
unset($result2);
unset($row2);

//rellena la sql para buscar las asignaturas
$sql2="Select * from asignaturas;";
//echo $sql2;
//re realiza la consulta para meter en $result2 las asignaturas
$result2=mysql_query($sql2,$conexion);
//echo $result2;
//coloca en $row2 la primera asignatura
$row2= mysql_fetch_array($result2);
//busca que id tiene la asignatura
DO
{
    //si el nombre coincide con el elegido en el formulario coge el nombre y el ↵
        id de la asignatura
    if($row2['id']==$_POST['asignatura']){
        //echo "hemos entrao!!";
        $nombre_asig=$row2['nombre'];
        //echo $nombre_asig;
        $id_asig=$row2['id'];
        //echo $id_asig;
        break;
    }
}
//comprueba si hay más asignaturas
while ($row2= mysql_fetch_array($result2));
```

```

        $sql="INSERT INTO `matriculas` ( `id_mat`, `id_alum`, `id_asig` )
        VALUES (NULL , ".$_POST['id_alum'].", ".$_POST['id_asig'].")";
        //echo $sql;
//se ejecuta la inserción de datos
$result=mysql_query($sql,$conexion);
//se informa de que se añadido un nuevo alumno
echo "<p>Matrícula añadida del alumno ".$row['nombre'].", la asignatura ".$nombre_asig."</p>";
//se coloca el enlace que permite volver al listado
echo "<p><A HREF='ver_matriculas.php?id=".$_GET['id']."'> Volver</A></p>";
    }else{
// si no nos han metido campos
        echo "<p>Vuelva a intentarlo</p>";
// se vuelve a colocar el formulario con los mismos campos que ahora, la diferencia es que se pone tantas veces como campos haya en los VALUE el $_POST['nombre_de_campo'] para que no haya que volver a teclearlo
        echo "<form method='post' action='matricular.php'>";
//aquí se deben poner tantas líneas echo como campos haya
//cambiando el echo "Nombre : <input type='text' name='nombre' maxlength=200 size=40><br>";
// por echo "Nombre : <input type='text' name='nombre_campo' maxlength=longitud_campo size=tamaño_caja_texto><br>";
//se muestra el código del alumno
echo "<p>Código:".$row["id"]."</p>";
//se muestra el nombre del alumno
echo "<p>Nombre:".$row["nombre"]."</p>";

//se coloca la caja oculta con el id del alumno
        echo "id : <input type='hidden' name='id_alum' value='".$row['id'].><br>";
//inicia la combo de las signaturas
echo "<select name='asignatura'>";
//rellena la sql para buscar las asignaturas
$sql2="Select * from asignaturas";
//re realiza la consulta para meter en $result2 las asignaturas
$result2=mysql_query($sql2,$conexion);
//coloca en $row2 la primera asignatura
$row2= mysql_fetch_array($result2);
DO
{
//coloca la asignatura en el listado
echo "<option>".$row2['nombre'].</option>";
}
//comprueba si hay más asignaturas
while ($row2= mysql_fetch_array($result2));
//si no hay mas asignatutas cierra el listado de asignaturas
echo "</select> ";
////se coloca el botón de guardar
        echo "<input type='submit' name='nuevo' value='Guardar'></form><br>";
    }
}
?>
</body>
</html>

```

Este es un caso un poco peculiar, ya que dado un alumno se le intenta matricular en alguna de las asignaturas disponibles. Por lo tanto debe buscar todas las asignaturas y colocarlas en el select. Y luego hacer la insercción en la tabla de relaciones (matriculas).

ver_matriculas.php alumnos

```
<?php
//incluimos el fichero config.php para coger los datos de la conexión
include_once('../config.php');
//cogemos las variables de las conexión para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexión
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");
//rellenamos la consulta, cambiar alumnos por el nombre de la tabla a listar
$sql="Select * from alumnos WHERE id=".$_GET['id'].>";";
//print $sql;
//re realiza la consulta
$result=mysql_query($sql,$conexion);
//mete el alumno en la variable $row
$row= mysql_fetch_array($result);

//print $sql2;
?>
<html>
<head>
<TITLE>Matricular
Alumno:<?php echo $row['nombre']; ?></TITLE>
<LINK REL="stylesheet" TYPE="text/css"
HREF="../estilo.css" />
</head>
<body>
<?php
//comprueba si se entas enviando datos para guardar
//si no es asi es que tiene que presentar el formulario
if(!isset( $_POST['nuevo']))
{
//coloca el principio del formulario para que vuelva al mismo script cuando ←
se rellene
echo "<form method=\"post\" action=\"matricular.php?id=".$_GET['id ←
'].\">";
//aqui se deben poner tantas líneas echo como campos haya
//cambiando el echo "Nombre : <input type=\"text\" name=\"nombre\" maxlength ←
=200 size=40><br>";
// por echo "Nombre : <input type=\"text\" name=\"nombre_campo\" maxlength= ←
longitud_campo size=tamaño_caja_texto><br>";
//se muestra el código del alumno
echo "<p>Código:".$row["id"]."</p>";
//se muestra el nombre del alumno
echo "<p>Nombre:".$row["nombre"]."</p>";

//se coloca la caja oculta con el id del alumno
echo "<input type=\"hidden\" name=\"id_alum\" value=\"".$_row['id ←
'].\">";
//inicia la combo de las asignaturas
echo "<select name=\"asignatura\">";
//rellena la sql para buscar las asignaturas
$sql2="Select * from asignaturas;";
//re realiza la consulta para meter en $result2 las asignaturas
$result2=mysql_query($sql2,$conexion);
//coloca en $row2 la primera asignatura
$row2= mysql_fetch_array($result2);
DO
```

```
{
//coloca la asignatura en el listado
echo "<option value=\"".$row2['id']. "\">".$row2['nombre']. "</option>";
}
//comprueba si hay más asignaturas
while ($row2= mysql_fetch_array($result2));
//si no hay mas asignatutas cierra el listado de asignaturas
echo "</select> <br><br>";
////se coloca el botón de guardar
echo "<input type=\"submit\" name=\"nuevo\" value=\"Guardar\"></form <
><br>";
}else{
//si se han metido los datos en el formulario
//se deben colocar tantos isset($_POST['nombre_campo']) && como campos haya <
en la tabla
if(isset($_POST['id_alum']) && isset($_POST['asignatura']))){

//se reinician las variables de la busqueda
unset($result2);
unset($row2);

//rellena la sql para buscar las asignaturas
$sql2="Select * from asignaturas;";
//echo $sql2;
//re realiza la consulta para meter en $result2 las asignaturas
$result2=mysql_query($sql2,$conexion);
//echo $result2;
//coloca en $row2 la primera asignatura
$row2= mysql_fetch_array($result2);
//busca que id tiene la asignatura
DO
{
//si el nombre coincide con el elegido en el formulario coge el nombre y el <
id de la asignatura
if($row2['id']==$_POST['asignatura']){
//echo "hemos entrao!!";
$nombre_asig=$row2['nombre'];
//echo $nombre_asig;
$id_asig=$row2['id'];
//echo $id_asig;
break;
}
}
//comprueba si hay más asignaturas
while ($row2= mysql_fetch_array($result2));

$sql="INSERT INTO `matriculas` ( `id_mat` , `id_alum` , ` <
id_asig` )
VALUES (NULL , ".$_POST['id_alum']. " , ".$_POST['asignatura <
'].")";
//echo $sql;
//se ejecuta la inserción de datos
$result=mysql_query($sql,$conexion);
//se informa de que se añadido un nuevo alumno
echo "<p>Matrícula añadida del alumno ".$_row['nombre']. " en <
la asignatura ".$nombre_asig."</p>";
//se coloca el enlace que permite volver al listado
echo "<p><A HREF='ver_matriculas.php?id=".$_GET['id']. "'> <
Volver</A></p>";
}else{
// si no nos han metido campos
```



```
        echo "<p>Vuelva a intentarlo</p>";
// se vuelve a colocar el formulario con los mismos campos que ahora, la
diferencia es que se pone tantas veces como campos haya en los VALUE el
$_POST['nombre_de_campo'] para que no haya que volver a teclearlo
        echo "<form method=\"post\" action=\"matricular.php\">";
//aqui se deben poner tantas líneas echo como campos haya
//cambiando el echo "Nombre : <input type=\"text\" name=\"nombre\" maxlength=
=200 size=40><br>";
// por echo "Nombre : <input type=\"text\" name=\"nombre_campo\" maxlength=
longitud_campo size=tamaño_caja_texto><br>";
//se muestra el código del alumno
echo "<p>Código: ".$row["id"]."</p>";
//se muestra el nombre del alumno
echo "<p>Nombre: ".$row["nombre"]."</p>";

//se coloca la caja oculta con el id del alumno
        echo "id : <input type=\"hidden\" name=\"id_alum\" value=\"".$row['id
'].\"\"><br>";
//inicia la combo de las firmas
echo "<select name=\"asignatura\">";
//rellena la sql para buscar las asignaturas
$sql2="Select * from asignaturas;";
//re realiza la consulta para meter en $result2 las asignaturas
$result2=mysql_query($sql2,$conexion);
//coloca en $row2 la primera asignatura
$row2= mysql_fetch_array($result2);
DO
{
    //coloca la asignatura en el listado
    echo "<option>".$row2['nombre'].</option>";
}
//comprueba si hay más asignaturas
while ($row2= mysql_fetch_array($result2));
//si no hay mas asignatutas cierra el listado de asignaturas
echo "</select> ";
////se coloca el botón de guardar
        echo "<input type=\"submit\" name=\"nuevo\" value=\"Guardar\"></form
><br>";
    }
}
?>
</body>
</html>
```

borrar_matricula.php alumnos

```
<?php
//incluimos el fichero config.php para coger los datos de la conexión
include_once('../config.php');
//cogemos las variables de las conexión para que se puedan utilizar en el script
global $server, $database, $user, $passwd;
//abrimos la conexión
$conexion = mysql_connect($server,$user,$passwd);
//elegimos la BBDD
mysql_select_db ($database, $conexion) OR die ("No se puede conectar");
//rellenamos la consulta, cambiar alumnos por el nombre de la tabla a listar
$sql="Select * from matriculas WHERE id_mat=".$_GET['id'].";";
//re realiza la consulta
$result=mysql_query($sql,$conexion);
//print $sql;
//se coge el registro del alumno
$row= mysql_fetch_array($result);
```

```
?>
<html>
  <head>
    <TITLE>Borrar Alumno</TITLE>
    <LINK REL="stylesheet" TYPE="text/css"
      HREF="../../../estilo.css" />
  </head>
  <body>
    <?php
      // se rellena la consulta que borra el alumno se substituye alumno por el ←
      nombre de la tabla
      $sql="DELETE from `matriculas` WHERE `id_mat` = ".$_GET['id']." LIMIT 1 ;";
      //echo $sql;
      // se ejecuta la consulta se borrado
      $result=mysql_query($sql,$conexion);
      // se informa de que se ha eliminado un alumno
      echo "<p>Matrícula eliminada</p>";
      // se coloca el enlace que permite volver al listado
      echo "<p><A HREF='ver_matriculas.php?id=".$_GET['id_alum']."'>Volver</A></p> ←
        >";

    ?>
  </body>
</html>
```

4.2. Conclusiones

Mediante estos ultimos ficheros se ha podido ver cómo relacionar entidades de una manera sencilla, tómelos como ejemplo y haga su primer propia aplicación web ;) después mejor la estructura XD

Apéndice A

GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent

copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- Ñ. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.