

# PHP , MYSQL y FORMULARIOS HTML

# Archivos php en localhost

- Recordar que para que funcione el php en local, tenemos que incluir el archivo .php en la carpeta htdocs de xampp. (por ejemplo: prueba.php)
- Ejecutar localhost/prueba.php desde el navegador.
- No olvidéis antes ejecutar xampp con el servidor apache y el de base de datos.

# FORMULARIOS

- Un formulario hace de cada objeto "input" una variable, que tendrá aparejada un dato.  
En el caso siguiente, se generarán dos variables:  
*nombre* (perteneciente al botón tipo "text")  
*enviar* (perteneciente al botón tipo "submit")
- La etiqueta "form" contiene dos parámetros importantes:
  - **method**: define la modalidad de envío de la información.
  - Los valores posibles son *POST* o *GET*;  
En el primer caso, los valores se transmiten de manera oculta (no aparecen en la barra de direcciones del navegador). Este es el que utilizaremos;  
En el segundo caso, aparecería, por ejemplo:  
<http://localhost/processa.php?nombre=pepe&curso=1>
  - **action**: indica el nombre fichero receptor de los datos del formulario.

# Formulario sencillo de envío

- Como se puede observar, contiene un campo de texto (en el que se nos sugiere introducir nuestro nombre) y un botón de envío

```
<html>
<body>
<form method="post" action="resp.php">
<p> Introduce tu nombre:
<input name="nombre" type="text">
<br>
<input type="submit" name="enviar" value="enviar" >
</p>
</form>
</body>
</html>
```

# Página receptora del envío (resp.php)

```
<?php
// Exponemos 2 posibilidades: la primera usa el array $_POST
$op = $_POST['nombre'];
// Esta segunda utiliza variables globales, es insegura
// $op = $nombre;
echo "Hola, .$op.!<br><br>";
echo "¿Te gusta el PHP?";
?>
```

- Este script muestra el contenido de la variable "nombre" originada en el formulario de dos formas.
  - En primer lugar, recurre al array **\$\_POST**
  - En segundo lugar, vuelca su contenido en la variable "op", para mostrar finalmente su contenido por pantalla.



# Formulario: elementos básicos

Elementos básicos de entrada de formulario

- Elementos de tipo INPUT
  - TEXT
    - Texto
  - RADIO
    - Selecciona una opción de un conjunto con forma de círculo
  - CHECKBOX
    - Selecciona una o varias opciones. Forma de cuadrado
  - SUBMIT
    - Envío
  - RESET
    - Borrado
- Elemento SELECT
  - Simple / múltiple
    - Selecciona de un menú desplegable
- Elemento TEXTAREA
  - Texto largo

# Variables

- A través de **variables globales**: cada nombre de variable del formulario (name="...") está disponible como variable global (\$codi). Este es el sistema tradicional y el más simple de implementar, pero también más inseguro, por lo cual se desaconseja.
- A través de tres arrays asociativos que PHP genera en este proceso:
  - \_POST o HTTP\_POST\_VARS**  
Array asociativo que contiene las variables pasadas por el método POST
  - \_GET o HTTP\_GET\_VARS**  
Array asociativo que contiene las variables pasadas por el método GET
  - \_REQUEST o HTTP\_REQUEST\_VARS**  
Array asociativo que contiene las variables pasadas por el método REQUEST

Acceso a estos arrays: **\$\_POST['nombre']** donde "nombre" sería el nombre de la variable y, por tanto, equivalente a *\$nombre*, teniendo en ambos casos el valor introducido en el formulario.

# Ejemplo formulario de envío de datos (actividades.html)

```
<html>
<head>
<title>Practica</title>
</head>
<body>
<h2>Formulario Actividades : </h2>
<form name="form" method="post" action="insertar.php">
<p>Nombre:
<input name="nombre" type="text" id="nombre" maxlength="30">
</p>
<p>Apellido:
<input name="apellido" type="text" id="apellido" maxlength="30">
</p>
<p>Selecciona un deporte:
<select name="deporte" id="deporte">
<option selected>futbol</option>
<option>basket</option>
<option>tenis</option>
<option>padel</option>
</select>
</p>
<p>Sexo:</p>
<p>
<input name="sexo" type="radio" value="masculino" checked>
Masculino</p>+
<p>
<input name="sexo" type="radio" value="femenino">
Femenino</p>
<p>
<input name="conducir" type="checkbox" id="conducir" value="conduce" checked>
¿Te gusta conducir? </p>
<p>Aficiones:</p>
<p>
<textarea name="aficiones" cols="50" rows="5"
id="aficiones"></textarea>
</p>
<p>&nbsp;</p>
<input type="submit" name="enviar" value="enviar" >
</form>
<p>&nbsp;</p>
<p>&nbsp;</p>
</body>
</html>
```



# Ejemplo archivo recepción de datos (insertar.php)

```
<?php
// Comprobamos si venimos del formulario
if(!isset($_POST['enviar']))
echo "No hay datos";
else {
// Dependiendo del tipo de objeto, utilizamos una estructura de control diferente
echo "Tu nombre es ".$_POST['nombre']." ".$_POST['apellido']."<br>";
if (isset($_POST['conducir']))
echo "Eres un Fernando Alonso<br>";
else
echo "No te gusta el motor<br>";
if($_POST['sexo']=="m")
echo "Eres chico<br>";
else
echo "Eres chica<br>";
//Del deporte elegido haremos un comentario utilizando switch
switch($_POST['deporte']){
case "futbol": echo "Eres futbolero<br>";
break;
case "tenis": echo "Eres tenista<br>";
break;
case "basket": echo "Basquetmaníaco<br>";
break;
case "padel": echo "El pádel está de moda<br>";
break;
}
if($_POST['aficiones']){
echo "Tus aficiones son:<br>";
echo nl2br($_POST['aficiones']); }
else
echo "No tienes aficiones";
}
echo "<p><a href=\"\"actividades.html\">Volver al formulario</a></p>"
?>
```

# MYSQL PHPMYADMIN PHP

# MySQL o MariaDB

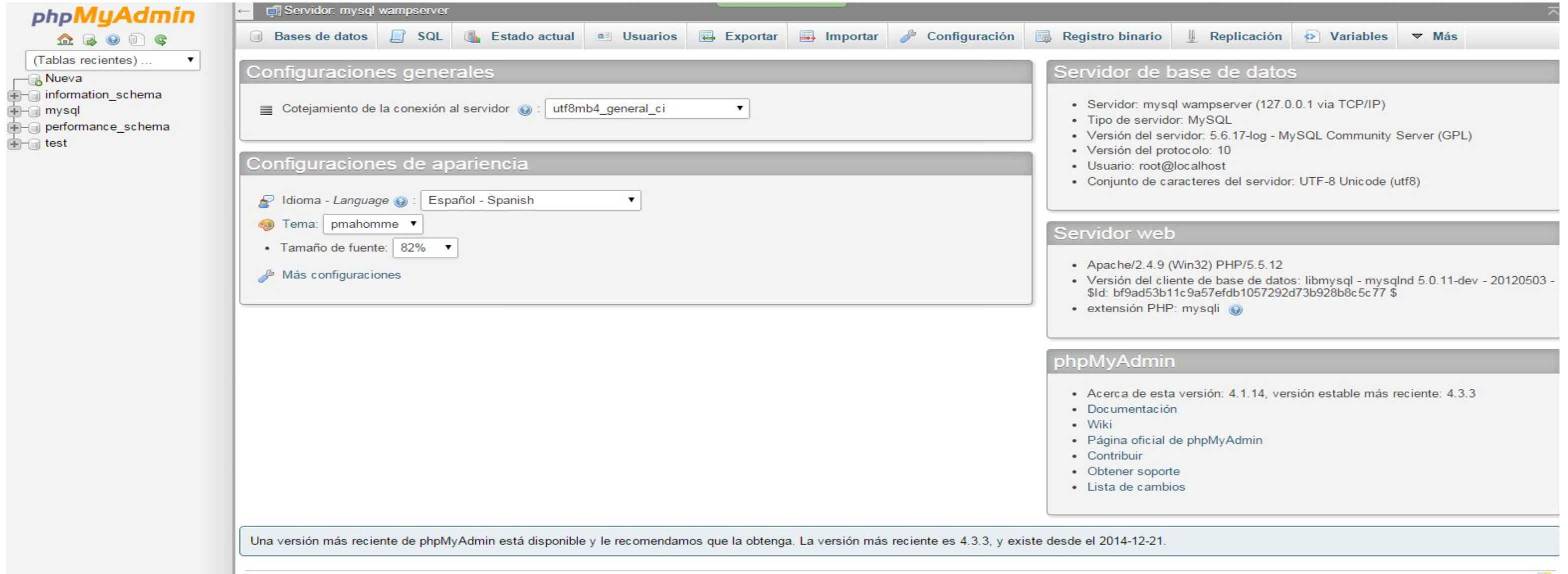
- Es posiblemente uno de los motores de BD más rápidos que se pueden encontrar, además de consumir pocos recursos de la máquina en la que está instalada.
- Fácil de instalar y administrar frente a otros productos del mercado. Dispone de infinidad de utilidades, manuales y documentación que la comunidad de usuarios se ha encargado de realizar desinteresadamente.
- Además dispone de gran variedad de interfaces de programación para su acceso (C, Perl, Java, **PHP**, Python...).

# PHPMYADMIN

- phpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet.
- Podemos instalar los programas wamp, xampp o easy php por ejemplo.
- Entraremos en phpmyadmin para crear la base de datos.



# localhost/phpmyadmin



The screenshot displays the phpMyAdmin interface for a MySQL server named 'mysql wampserver'. The left sidebar shows a tree view of databases: 'information\_schema', 'mysql', 'performance\_schema', and 'test'. The main content area is divided into two sections: 'Configuraciones generales' and 'Configuraciones de apariencia'. The 'Configuraciones generales' section shows the 'Cotejamiento de la conexión al servidor' set to 'utf8mb4\_general\_ci'. The 'Configuraciones de apariencia' section shows the 'Idioma - Language' set to 'Español - Spanish', the 'Tema' set to 'pmahomme', and the 'Tamaño de fuente' set to '82%'. The right sidebar contains three panels: 'Servidor de base de datos', 'Servidor web', and 'phpMyAdmin'. The 'Servidor de base de datos' panel lists server details: 'Servidor: mysql wampserver (127.0.0.1 via TCP/IP)', 'Tipo de servidor: MySQL', 'Versión del servidor: 5.6.17-log - MySQL Community Server (GPL)', 'Versión del protocolo: 10', 'Usuario: root@localhost', and 'Conjunto de caracteres del servidor: UTF-8 Unicode (utf8)'. The 'Servidor web' panel lists web server details: 'Apache/2.4.9 (Win32) PHP/5.5.12', 'Versión del cliente de base de datos: libmysql - mysqlnd 5.0.11-dev - 20120503 - \$Id: bf9ad53b11c9a57efdb1057292d73b928b8c5c77 \$', and 'extensión PHP: mysqli'. The 'phpMyAdmin' panel lists links: 'Acerca de esta versión: 4.1.14, versión estable más reciente: 4.3.3', 'Documentación', 'Wiki', 'Página oficial de phpMyAdmin', 'Contribuir', 'Obtener soporte', and 'Lista de cambios'. A footer message at the bottom states: 'Una versión más reciente de phpMyAdmin está disponible y le recomendamos que la obtenga. La versión más reciente es 4.3.3, y existe desde el 2014-12-21.'

**Configuraciones generales**

Cotejamiento de la conexión al servidor : utf8mb4\_general\_ci

**Configuraciones de apariencia**

Idioma - Language : Español - Spanish

Tema: pmahomme

Tamaño de fuente: 82%

Más configuraciones

**Servidor de base de datos**

- Servidor: mysql wampserver (127.0.0.1 via TCP/IP)
- Tipo de servidor: MySQL
- Versión del servidor: 5.6.17-log - MySQL Community Server (GPL)
- Versión del protocolo: 10
- Usuario: root@localhost
- Conjunto de caracteres del servidor: UTF-8 Unicode (utf8)

**Servidor web**

- Apache/2.4.9 (Win32) PHP/5.5.12
- Versión del cliente de base de datos: libmysql - mysqlnd 5.0.11-dev - 20120503 - \$Id: bf9ad53b11c9a57efdb1057292d73b928b8c5c77 \$
- extensión PHP: mysqli

**phpMyAdmin**

- Acerca de esta versión: 4.1.14, versión estable más reciente: 4.3.3
- Documentación
- Wiki
- Página oficial de phpMyAdmin
- Contribuir
- Obtener soporte
- Lista de cambios

Una versión más reciente de phpMyAdmin está disponible y le recomendamos que la obtenga. La versión más reciente es 4.3.3, y existe desde el 2014-12-21.



# Acceso base de datos mysql desde php

**`$conex = mysqli_connect ( $host, $usuario, $contraseña,$base_de_datos)`**

Abre una conexión con el servidor MySQL de la base de datos

**`$resul = mysqli_query ($conex, $sql)`**

Envía una instrucción SQL contra la conexión.

**`mysqli_fetch_array($resul) o mysqli_fetch_row($resul)`**

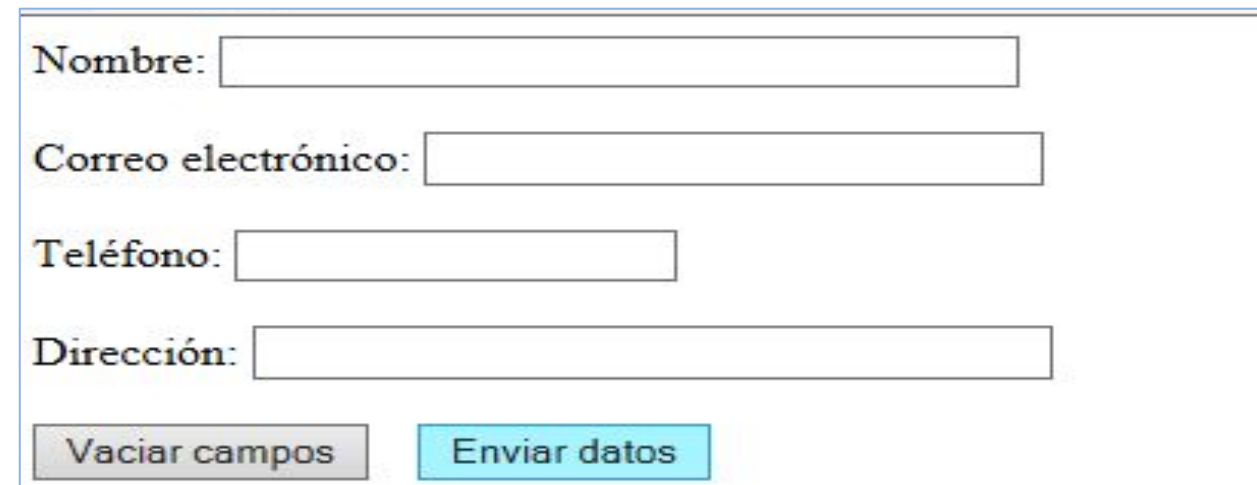
Obtiene resultados de la instrucción SQL.

# Ejercicio básico mysql y php

- ✓ Usando xampp, crear una base de datos en phpmyadmin con los campos siguientes y con el tipo de datos correspondiente:

- Nombre, correo electrónico, teléfono y Dirección

1. Preparar una página web con un formulario que nos pida los valores:



Nombre:

Correo electrónico:

Teléfono:

Dirección:

2. Insertar varios registros a partir del formulario anterior creando un archivo php que lo haga.
3. Visualizar todos los registros con el archivo php correspondiente .

# Creación de base de datos

- Creamos una base de datos que llamaremos club en phpmyadmin.
- Incluiremos una tabla que denominaremos socios con los campos: nombre, correo, telefono y direccion con los tipos de datos adecuados.

# Formulario solución al ejercicio

(llamará al archivo insertar.php)

[illegible]

# Insertar datos (insertar.php)

```
<?php
// Primero comprobamos que ningún campo esté vacío y que todos los campos existan.
if(isset($_POST['nombre']) && !empty($_POST['correo']) && isset($_POST['telefono']) && !empty($_POST['direccion'])) {
// Si entramos es que todo se ha realizado correctamente y conectamos con la base de datos club de mysql
$link = mysqli_connect("localhost","root","","club");
// Con esta sentencia SQL insertaremos los datos en la base de datos
mysqli_query($link,"INSERT INTO socios(nombre,correo,telefono,direccion)
VALUES ('".$_POST['nombre']."','".$_POST['correo']."','".$_POST['telefono']."','".$_POST['direccion']."')");
// Ahora comprobaremos que todo ha ido correctamente
$my_error = mysqli_error($link);
if(!empty($my_error)) {
    echo "Ha habido un error al insertar los valores. $my_error";
} else {
    echo "Los datos han sido introducidos satisfactoriamente";
}
}
else {
echo "Error, no ha introducido todos los datos";
}
?>
```



# Consultar datos

```
<?php
//Conectamos con MySQL y la base de datos club
$link = mysqli_connect("localhost", "root", "", "club");
$resul1 = mysqli_query($link, "SELECT nombre, correo, telefono, direccion FROM socios");
//Creamos la tabla
echo "<table border = '1'> \n";
//incluimos los nombres de los campos
echo "<tr><td>Nombre</td><td>Correo
electrónico</td><td>teléfono</td><td>Dirección</td></tr> \n";
//Incluimos los resultados de la búsqueda
while ($row = mysqli_fetch_row($resul1)){
    echo "<tr><td>$row[0]</td><td>$row[1]</td><td>$row[2]</td><td>$row[3]</td></tr> \n";
}
echo "</table> \n";
?>
```

# Buscar datos (formulario que llamará a buscar.php)

*Formulario sencillo para buscar por nombre:*

```
<form method="POST" action="buscar.php">
```

```
Búsqueda por nombre:<input maxlength="30" size="30" name="cadena"><br>
```

```
<input name="Enviar" value="Envía datos" type="submit">
```

```
</form>
```

Búsqueda por nombre:

# Buscar datos (buscar.php)

```
<?php
if (!isset($_POST['cadena'])){
    echo "Debe especificar una cadena a bucar";
    exit;
}
$link = mysqli_connect("localhost", "root", "", "club");
$result = mysqli_query($link, "SELECT * FROM socios WHERE nombre LIKE '%".$_POST['cadena']."' ORDER BY nombre");
if ($row = mysqli_fetch_array($result)){
    echo "<table border = '1'> \n";
    //Mostramos los nombres de las tablas
    echo "<tr> \n";
    while ($field = mysqli_fetch_field($result)){
        echo "<td>$field->name</td> \n";
    }
    echo "</tr> \n";
    do {
        echo "<tr> \n";
        // echo "<td>".$row["id"]."</td> \n";
        echo "<td>".$row["nombre"]."</td> \n";
        echo "<td>".$row["correo"]."</td> \n";
        echo "<td>".$row["telefono"]."</td> \n";
        echo "<td>".$row["direccion"]."</td> \n";
        echo "</tr> \n";
    } while ($row = mysqli_fetch_array($result));
    echo "</table> \n";
} else {
    echo "¡ No se ha encontrado ningún registro !";
}
?>
```

# Eliminar datos (formulario que llama a eliminar.php)

```
<?php
//Conectamos con MySQL y la base de datos club
$link = mysqli_connect("localhost", "root", "", "club");
$resul1 = mysqli_query($link, "SELECT nombre, correo, telefono, direccion FROM socios");
//Creamos la tabla
echo "<table border = '1'> \n";
//incluimos los nombres de los campos
echo "<tr><td>Nombre</td><td>Correo
electrónico</td><td>teléfono</td><td>Dirección</td></tr> \n";
//Incluimos los resultados de la búsqueda
while ($row = mysqli_fetch_row($resul1)){
    echo "<tr><td>$row[0]</td><td>$row[1]</td><td>$row[2]</td><td>$row[3]</td></tr> \n";
}
echo "</table> \n";
?>
```

```
<form method="POST" action="eliminar.php">
Elimina por nombre:<input maxlength="30" size="30" name="cadena"><br>
<input name="Enviar" value="Elimina datos" type="submit">
</form>
```

# Eliminar (eliminar.php)

```
<?php
if (!isset($_POST['cadena'])){
    echo "Debe especificar una cadena a bucar";
    exit;
}
$link = mysqli_connect("localhost", "root", "", "club");
$result = mysqli_query($link, "DELETE FROM socios WHERE nombre LIKE '%".$_POST['cadena']."' ORDER BY nombre");
$result2 = mysqli_query($link, "SELECT nombre, correo, telefono, direccion FROM socios");
//Creamos la tabla
echo "<table border = '1'> \n";
//incluimos los nombres de los campos
echo "<tr><td>Nombre</td><td>Correo
electrónico</td><td>teléfono</td><td>Dirección</td></tr> \n";
//Incluimos los resultados de la búsqueda
while ($row = mysqli_fetch_row($result2)){
    echo "<tr><td>$row[0]</td><td>$row[1]</td><td>$row[2]</td><td>$row[3]</td></tr> \n";
}
echo "</table> \n";
?>
```



# Modificación (formulario que llama a modificar.php)

[illegible]

# Modificación (modificar.php)

```
<?php
// Primero comprobamos que ningún campo esté vacío y que todos los campos existan.
if(isset($_POST['nombre']) && !empty($_POST['correo']) && isset($_POST['telefono']) && !empty($_POST['direccion'])) {
// Si entramos es que todo se ha realizado correctamente y conectamos con la base de datos club de mysql
$link = mysqli_connect("localhost","root","","club");
// Con esta sentencia SQL insertaremos los datos en la base de datos
mysqli_query($link,"UPDATE SOCIOS SET correo='{$_POST['correo']}', telefono='{$_POST['telefono']}', direccion='{$_POST['direccion']}'
WHERE nombre='{$_POST['nombre']}'");
// Ahora comprobaremos que todo ha ido correctamente
$my_error = mysqli_error($link);
if(!empty($my_error)) {
    echo "Ha habido un error al insertar los valores. $my_error";
} else {
    echo "Los datos han sido introducidos satisfactoriamente";
}
}
else {
echo "Error, no ha introducido todos los datos";
}
?>
```