



UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA
ESCUELA DE COMPUTACION

CICLO 1

GUIA DE LABORATORIO #7

Nombre de la Practica: Utilización de PHP para interactuar con bases de datos MySQL
Lugar de Ejecución: Centro de Cómputo
Tiempo Estimado: 2 horas con 30 minutos
MATERIA: Desarrollo de Aplicaciones Web
con Software Interpretado en el Servidor (DSS404)

I. OBJETIVOS

Que el estudiante:

1. Adquiera la capacidad de realizar validación de formularios antes de enviar los datos ingresados por los usuarios al servidor.
2. Desarrolle la habilidad de conectar con una base de datos MySQL desde un script PHP.
3. Domine las técnicas de concatenación para escribir consultas SQL desde un script PHP.
4. Haga uso de las funciones PHP para enviar las consultas al servidor y
5. Procese de forma adecuada los resultados devueltos por una consulta SQL con un script PHP.

II. INTRODUCCION TEORICA

Validación de formularios con PHP

La validación de datos ingresados por un usuario en un formulario es una necesidad en las aplicaciones web, tanto por razones técnicas, como por aspectos de seguridad. Cuando se envían datos de un formulario erróneos al servidor web, este invierte algún tiempo en procesar los datos, y si estos contienen errores o no están en el formato deseado, invertirá algún tiempo en procesarlos y en determinar si se pueden utilizar esos datos. Cuando esto ocurre con múltiples usuarios conectados simultáneamente al servidor, la demanda de procesamiento en el servidor puede volverse determinante para llegar a la cuenta que es conveniente poner un filtro para los datos. Si a esto sumamos el hecho que, en Internet existen muchos usuarios malintencionados que buscan vulnerabilidad en los sistemas para robar información o para producir daños en los sistemas, resulta que es necesario invertir tiempo de desarrollo en programar rutinas que se encarguen de verificar que los datos que se procesan son correctos y seguros.

Recuperar valores de formulario con PHP

Una vez enviado el formulario al servidor web, la secuencia de comando indicada en el atributo action del elemento FORM se encarga de entrar en acción para procesar los datos. Para ello debe acceder a los valores ingresados en el formulario. Estos valores PHP los recibe haciendo uso de las matrices superglobales asociativas \$_POST y \$_GET, en donde el nombre del campo de formulario se convierte

en la clave y el valor introducido o seleccionado por el usuario el valor que deberá ser procesado en la secuencia de comando. La matriz asociativa que se vaya a utilizar dependerá del método de envío seleccionado en el formulario, si el método fue GET, entonces la matriz superglobal a utilizar será \$_GET y si fue POST entonces deberá utilizar \$_POST.

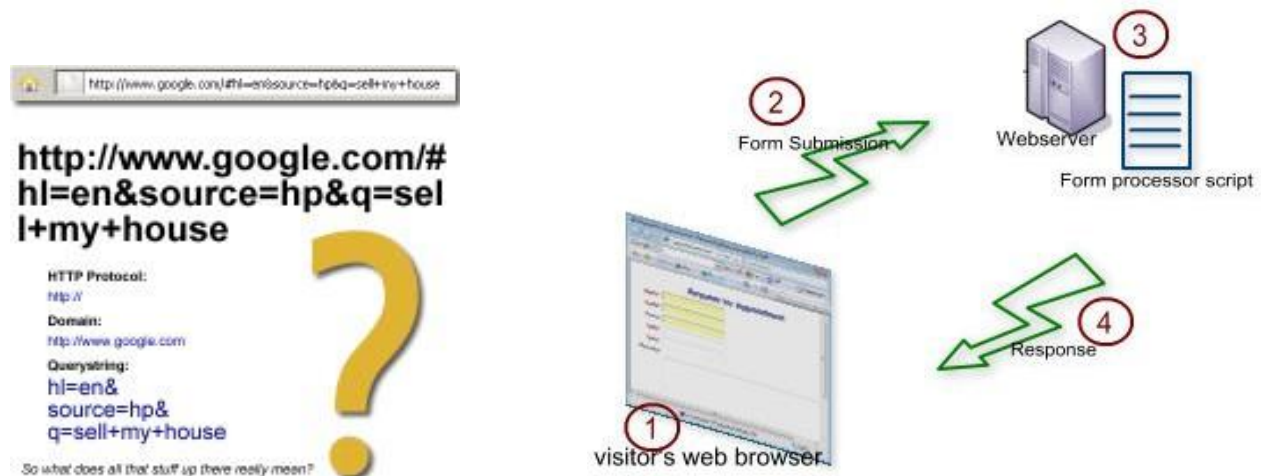
Lo anterior significa que si el elemento HTML es:

```
<input type="text" name="nombre" size="25" maxlength="30" />
```

Y el método de envío empleado ha sido POST, el código PHP para tomar este valor, sería:

```
<?php echo $_POST['nombre']; ?>
```

Cuando el método de envío sea GET, los datos serán enviados en la cadena de consulta que forma parte de la URL que es enviada al servidor. En cambio, cuando se usa el método POST los datos son incorporados en el cuerpo del mensaje HTTP que es enviado desde el navegador al servidor.



Adicionalmente, existe una tercer matriz superglobal, llamada \$_REQUEST, la cual se utiliza cuando no es relevante el método de envío, ya que puede obtener los datos enviados por el método GET o POST, además de los definidos en variables de cookies \$_COOKIES y de envío de archivos, \$_FILES.

Validación de los datos recibidos

Los primeros tipos de datos que se pueden validar, son las cajas de textos y las áreas de texto, en ambos casos se tienen campos en los que el usuario digita directamente los valores a enviar. Lo primero a determinar aquí es qué tipo de dato es el que se va a recibir. La opción más viable para validar un campo de texto es utilizar expresiones regulares. Además, es conveniente evitar que el valor enviado en el campo de formulario esté vacío.

El siguiente código muestra cómo se pueden validar datos provenientes de una caja de texto en la que debe ingresarse una dirección de correo electrónico:

```
<input type="text" name="email" size="50" maxlength="60" />
```

Procesamiento del formulario con PHP:

```

if(isset($_POST['email'])){
    $email = $_POST['email'];
    $regexp = /^[a-zA-Z][\w\.\-]+[a-zA-Z0-9]@[a-zA-Z0-9][a-zA-Z0-9\-]*\.([a-zA-Z]{2,4})$/;
    if(preg_match($regexp, $email)){
        return true;
    }
    else{
        return false;
    }
}

```

Además de la validación, según el tipo de datos, los campos de textos y las áreas de texto, pueden permitir a los usuarios incluir innecesariamente una serie de espacios en blanco antes y después del texto ingresado. Para ello, se pueden utilizar funciones de PHP para limpiar esos espacios. Una de las funciones es trim(), aunque también están las funciones ltrim() y rtrim(), que eliminan los espacios a la izquierda y a la derecha de una cadena, respectivamente, sin embargo, con la función trim() se pueden eliminar ambos al mismo tiempo.

La forma de utilizar esta función si se tiene un campo de formulario en forma de caja de texto es la siguiente:

```
<input type="text" name="nombre" size="30" maxlength="30" />
```

Al enviar el campo de formulario anterior, una secuencia de comandos en la que se desee limpiar el dato ingresado de espacios en blanco, a la derecha y a la izquierda, debería procesar el dato así:

```

if(isset($_POST['nombre'])){
    $nombre = trim($_POST['nombre']);
}

```

La validación de otros tipos de campo de formulario, como casillas de verificación, botones de opción, menús desplegables y cuadros de lista resulta más fácil porque el valor siempre será uno de una lista de valores predefinidos. Si se establece desde la creación del formulario un valor preestablecido, la validación no será necesaria, pero en caso de no hacerlo, lo recomendable es colocar la lista de valores válidos en una matriz.

Observe el siguiente ejemplo:

```

<input type="checkbox" name="input" value="valor1" />
<input type="checkbox" name="input" value="valor2" />
<input type="checkbox" name="input" value="valor3" />
...
<input type="checkbox" name="input" value="valorN" />

<?php
    $options = array('valor1', 'valor2', 'valor3', ... ,valorN);
    $valid = true;
    if(is_array($_GET['input'])){
        $valid = true;
    }

```

```

foreach($_GET['input'] as $input){
    if(!in_array($input, $options)){
        $valid = false;
    }
}

if($valid){
    //Procesar ...
}
}

```

?>

Conectando con MySQL desde PHP

Para trabajar con bases de datos desde PHP se suele utilizar un proceso como el siguiente:

1. Establecer conexión con el gestor de la base de datos. Esta conexión suele requerir que se indique el usuario, la contraseña y el servidor donde está alojada la base de datos.
2. Seleccionar la base de datos con la que se va a trabajar. Este proceso puede estar incluido en el paso anterior si la función a utilizar ya incluye el argumento para la base de datos.
3. Componer una instrucción SQL que se desea ejecutar en el servidor MySQL y almacenarla en una variable de tipo cadena.
4. Enviar la consulta utilizando la variable, o de forma directa.
5. Procesar el resultado obtenido después de ejecutar la consulta.
6. Mostrar el resultado en la página web.
7. Liberar los recursos utilizados en la consulta.
8. Cerrar la conexión con el gestor de base de datos.

Debe considerar que el intérprete de PHP no hace ningún análisis de la sintaxis MySQL con que se construyó la consulta. Para PHP, se trata únicamente de una cadena de caracteres. Será el programador el encargado de revisar su validez como sentencia SQL.

`mysql_connect()`. Esta función tiene la siguiente cabecera:

```
mysql_connect(string Servidor, string Usuario, string Clave);
```

Siendo:

- **Servidor:** Es el nombre (o la dirección IP) del servidor donde está MySQL. El valor por defecto será 'localhost'.
- **Usuario:** Es el nombre del usuario que va a acceder a MySQL
- **Clave:** Es la clave correspondiente al usuario.

Esta función nos devuelve un valor de tipo int que sirve para identificar la conexión con el servidor SQL, así que podremos asignarlo a una variable con la que referirnos a la conexión abierta.

Por ejemplo, si todos pueden acceder al servidor, abríramos la conexión haciendo:

```
$IdConexion = mysql_connect("localhost", "", "");
```

Será útil tener los datos en variables (las posteriores modificaciones son así más sencillas), por ejemplo, en las siguientes líneas guardamos el nombre del servidor, usuario y clave en variables, y abrimos la conexión:

```
<?php
    $Servidor = "localhost";
    $Usuario = "Perico";
    $Clave = "Palotes";
    $IdConexion = mysql_connect($Servidor, $Usuario, $Clave);
?>
```

Además, esto podemos grabarlo en un archivo, por ejemplo, 'ConexionBD.php', y en lugar de escribir lo anterior, poner:

```
<?php include("Utilidades/ConexionBD.php"); ?>
```

Así podremos usar este pequeño script de conexión al servidor de bases de datos en los scripts que lo requieran.

Una vez abierta la conexión, tenemos las siguientes opciones:

- Crear una nueva base de datos
- Borrar una base de datos existente
- Abrir una base de datos existente

Crear una base de datos

Podemos crear una nueva base de datos usando la función:

```
mysql_create_db(string NombreBD, int IdConexion);
```

- NombreBD será el nombre que queramos darle a la base de datos
- IdConexion es el identificador de la conexión abierta

Por ejemplo, teniendo abierta una conexión, cuyo identificador hemos guardado en la variable \$IdConexion, para crear una base de datos cuyo nombre sea "PruebaBD", haremos:

```
<?php
    $NombreBD = "PruebaBD";
    mysql_create_db($NombreBD, $IdConexion);
?>
```

Borrar una base de datos existente

Hay que recordar que borrar una base de datos eliminará toda la información que ésta tuviera, incluida la estructura de las tablas, no dejando rastro alguno de la base de datos en el servidor.

La función para eliminar una base de datos es:

```
mysql_drop_db(string NombreBD, int IdConexion);
```

Por ejemplo, si queremos eliminar la base de datos que acabamos de crear, haríamos:

```
<?php
    mysql_drop_db($NombreBD, $IdConexion);
?>
```

donde `$NombreBD` e `$IdConexion` son las variables que teníamos definidas antes, una con el nombre de la base de datos y otra con el identificador de la conexión.

Abrir una base de datos existente

Por último, podemos escoger sobre qué base de datos de las existentes queremos trabajar, para ello, usaremos la función:

```
mysql_select_db(string NombreBD, int IdConexion);
```

Por ejemplo, si queremos abrir una base de datos cuyo nombre sea "Usuarios", haremos lo siguiente:

```
<?php
    $NombreBD = "Usuarios";
    mysql_select_db($NombreBD, $IdConexion);
?>
```

Siendo, `$IdConexion` el identificador de la conexión abierta.

Dos funciones más que nos van a ser útiles serán las funciones `mysql_errno()`; y `mysql_error()`;

```
mysql_errno(int IdConexion);
```

Esta función devuelve un número de error que se haya dado al hacer alguna operación sobre una base de datos MySQL.

```
mysql_error(int IdConexion);
```

Esta función devuelve un mensaje de error que se haya dado al hacer alguna operación sobre una base de datos MySQL.

mysql_query

`mysql_query` -- Envía una sentencia SQL a MySQL

Descripción

```
mysql_query(string Consulta, int IdConexion);
```

Como se aprecia en la definición, hemos de pasarle como parámetro obligatorio una cadena en la que debe estar almacenada la consulta a realizar y, como parámetro opcional, el identificador de la conexión abierta (si no se especifica, se toma el que esté activo en ese momento). Además, devuelve un identificador de consulta, que podemos almacenar en una variable para referirnos luego a dicha consulta. Lo que hacemos con esta función es enviar la sentencia SQL al servidor MySQL para que ésta sea ejecutada.

Una vez hecha la consulta, hay que saber cómo recuperar los datos; esto es lo que hacemos en las siguientes líneas:

```
$ResultQuery = mysql_fetch_array($IdConsulta);
print("Usuario: ".$ResultQuery["USUARIO"]."<B>");
print("E-mail: ".$ResultQuery[1]."<P>");
```

Tenemos varias funciones para recuperar los datos de una consulta. Una de ellas es, como vemos en el ejemplo, `mysql_fetch_array()`. Esta función devuelve en un array el contenido de la sentencia SELECT, identificando cada elemento con un campo de la base de datos y avanzando a la siguiente fila. Para ello, debemos pasarle el identificador de consulta devuelto por `mysql_query()`. Observamos, además, que como índice podemos especificar tanto su posición numérica (los arrays van del 0 al N-1, donde N es el número de elementos) como,

entre comillas, el nombre del campo. Si queremos recorrer **todas** las filas devueltas por el SELECT, tendremos que ir haciendo la asignación fila a fila hasta que ya no queden filas.

Para saber cuántas filas devuelve un SELECT, tenemos la función `mysql_num_rows()`. Esta función, cuya definición vemos:

```
mysql_num_rows(int $IdConsulta);
```

Devuelve un entero, que será el número de filas del SELECT, de la consulta correspondiente al identificador `$IdConsulta`.

Al igual que tenemos `mysql_fetch_array()`, hay una función que se llama `mysql_fetch_row()`. Hace lo mismo, pero con una salvedad, y es que no se puede usar el nombre del campo en el array que devuelve como si fuera una manera de indexar. Es decir, sólo admite referencias numéricas a los campos de la consulta. Hay que tener en cuenta que el primero es el 0.

Otras funciones que pueden sernos útiles son:

```
mysql_field_type(int $IdConsulta, int $IndiceCampo);
```

Nos devuelve el tipo de dato correspondiente a un campo de la base de datos, perteneciente a la consulta `$IdConsulta` y cuya posición en el SELECT viene dada por `$IndiceCampo`, siendo 0 la primera posición. Pruebe a poner, como muestra, este código en el ejemplo:

```
$TipoDato = mysql_field_type($IdConsulta, 1);  
print("Tipo de dato del campo: ".$TipoDato."<P>");
```

```
mysql_field_name(int $IdConsulta, int $IdCampo);
```

Esta función nos devuelve el nombre del campo que ocupa la posición `$IdCampo` referente a la consulta `$IdConsulta`. Una vez más, hay que decir que los campos se numeran de 0 a N-1, siendo N el número de campos de la consulta. Podemos incluir este fragmento de código en el ejemplo anterior:

```
$NombreDato = mysql_field_name($IdConsulta, 0);  
print("Nombre del campo: ".$NombreDato."<P>");
```

```
mysql_result(int $IdConsulta, int $Fila, int $Campo);
```

Esta función nos devuelve el dato referido a la consulta `$IdConsulta` que ocupa la posición (`$Fila`, `$Campo`). Por ejemplo, si añadimos este código al que teníamos:

```
$Dato1 = mysql_result($IdConsulta, 0, 0);  
print("El valor obtenido en la posición (0,0) " . "es: <B>".$Dato1."</B><P>");
```

Utilización de la extensión MySQLi

La biblioteca MySQLi es una extensión de PHP para trabajar de una forma más rápida y eficiente con el gestor MySQL. Con esta biblioteca es posible utilizar sintaxis orientada a objetos y orientada a procedimientos.

Configurar una conexión con MySQL

PHP 5 proporciona una nueva biblioteca para conectarse con bases de datos de MySQL. Esta biblioteca se denomina `mysqli`. Esta biblioteca permite utilizar una sintaxis orientada a objetos, así como, la sintaxis tradicional basada en procedimientos.

Para conectarse a la base de datos MySQL se utiliza la siguiente sintaxis. Primero se muestra la sintaxis orientada a objetos y después la basada en procedimientos:

```
@ $cn = new mysqli([string host [, string userdb [, string passdb [, string db]]]);  
-----  
@ $cn = mysqli_connect([string host [, string userdb [, string passdb [, string db]]]);
```

En las instrucciones anteriores se ha utilizado el operador de supresión de errores, `@`, para evitar que se muestren los errores reportados por PHP y que usted pueda implementar formas más elegantes para presentar los errores al usuario.

La instrucción orientada a objetos crea una instancia de la clase `mysqli` y una conexión al host especificado en el primer parámetro de forma directa o mediante una variable de tipo de cadena con el nombre del servidor, con el nombre de usuario de la base de datos `userdb` y su respectiva contraseña especificada por el tercer argumento `passdb` y con la base de datos especificada en el cuarto argumento `db`. Al utilizar el enfoque orientado a objetos puede invocar los métodos de este objeto para acceder a la base de datos.

En el caso de utilizar el enfoque basado en procedimientos la función `mysqli_connect()` devuelve un puntero en lugar de un objeto. Además, puede utilizar las funciones que la biblioteca `mysqli` proporciona, recordando que en este caso dichas funciones conservan la misma sintaxis que proporcionaba la biblioteca `mysql_connect`, `mysql_query`, etc. Con la salvedad que ahora tiene que agregar una letra `i`. Así: `mysqli_connect`, `mysqli_query`, etc.

Es recomendable comprobar el resultado del intento de conexión con la base de datos para evitar errores al intentar ejecutar consultas con la base de datos. Puede utilizar cualquier método que estime conveniente para este propósito. A continuación se le presenta una de tantas formas para poder realizar esta comprobación:

```
if(mysqli_connect_errno()){  
    printf("Falló la conexión: %s\n", mysqli_connect_error());  
    exit(0);  
}  
//Acá el código a ejecutar si la conexión se realizó de forma exitosa
```

En este caso el código es exactamente igual independientemente de si se utiliza el enfoque orientado a objetos o basado en procedimientos.

Debe tomar en cuenta que el número de conexiones simultáneas a MySQL viene limitado por el parámetro `max_connections` y `MaxClients` de Apache. Ambos parámetros se pueden reconfigurar en el archivo `httpd.conf`.

Seleccionar la base de datos de trabajo

Notará que en PHP 5 el constructor `mysqli()` indica la base de datos con la que se desea trabajar usando el cuarto argumento del método. Sin embargo, para poder cambiar esta base de datos de trabajo, tendrá que usar la función `mysqli_select_db()` a la que se puede acceder utilizando el enfoque orientado a objetos o basado en procedimientos:


```
$cn->select_db($db);
```

O

```
mysqli_select_db($cn, $db);
```

Realizar consultas a la base de datos

Para realizar una consulta a la base de datos se puede utilizar la función `mysqli_query()`. Esta función recibe como argumento la consulta que se desea ejecutar en el servidor de base de datos. Resulta conveniente configurar esta consulta en una variable de tipo cadena, en lugar de hacerlo directamente dentro de la función. Por ejemplo:

```
$qr = "SELECT * FROM libros WHERE " . $tipobusqueda . " LIKE '%" . $cadenabusqueda . "%'";
```

En la práctica casi siempre se verá obligado a concatenar sentencias SQL con variables de PHP. Estas variables muy probablemente serán establecidas por el ingreso de información de parte del usuario a través de formularios.

El siguiente paso es enviar la consulta al servidor MySQL para que sea ejecutada. Para ello se requiere una instrucción como la siguiente. Se muestra siempre la instrucción orientada a objeto y la basada en procedimientos:

```
$rs = $cn->query()
```

O

```
$rs = mysqli_query($cn, $db);
```

La versión orientada a objetos devuelve un objeto de resultados, mientras que la versión basada en procedimientos, un identificador de resultados. En ambos casos este resultado se almacena en la variable `$rs` (de result) para procesar la información posteriormente. En caso de que falle la función devuelve false.

Recuperar el resultado de la consulta

Existen un conjunto de funciones para dividir los resultados del identificador o del objeto de resultados utilizando diversos métodos o formas. Se pueden utilizar funciones como `mysqli_fetch_assoc()`, `mysqli_fetch_row()`, `mysqli_fetch_array` y `mysqli_fetch_object()`. Es conveniente obtener antes de aplicar estas funciones el número de registros devueltos al ejecutar la consulta. Para esto puede utilizarse la propiedad `num_rows` del objeto de resultados o la función `mysqli_num_rows()`, si utilizó el enfoque basado en procedimientos.

```
$nr = $rs->num_rows();
```

O

```
$nr = mysqli_num_rows();
```

En el enfoque orientado a objetos, el número de filas devuelto se almacena en el miembro o propiedad `num_rows` el objeto de resultados (`$rs`). En tanto que en el enfoque basado en procedimientos, la función `mysqli_num_rows()` indica el número de filas devueltas por la consulta.

La utilidad de obtener el número de filas devuelto está en que al hacerlo facilita el procesamiento de los resultados mediante cualquiera de las formas de lazos o bucles de PHP. Por ejemplo, de la siguiente forma:

```
for($i=0; $i<$nr; $i++){  
    //Procesamiento del resultado  
}
```

En cada iteración adentro del ciclo o lazo se puede utilizar una función como `$rs->fetch_assoc()` (o `mysqli_fetch_assoc()` si está utilizando el enfoque por procedimientos). El bucle no se ejecutará si el conjunto de resultados está vacío. La función `mysqli_fetch_assoc()` obtiene cada fila del conjunto de resultados y devuelve la fila actual en forma de matriz asociativa, con cada clave en forma de nombre de atributo y cada valor con su valor correspondiente en la matriz. Para procesar cada fila del conjunto de resultados se puede utilizar una variable que contendrá los elementos de la matriz devuelta en cada iteración, de la siguiente forma:

```
$row = $rs->fetch_assoc();
```

O

```
mysqli_fetch_assoc($rs);
```

Una vez obtenida una fila mediante alguna de las instrucciones anteriores, será posible obtener cada campo para mostrarlos adecuadamente en una tabla o lista. De la siguiente forma:

```
while($row = $rs->fetch_assoc()){ // o while($row = mysqli_fetch_assoc())
    echo stripslashes($row['autor']);
}
```

Se ha utilizado la función `stripslashes` para quitar las comillas que se han escapado para evitar problemas con las consultas MySQL.

Desconectarse de la base de datos

Se pueden liberar los conjuntos de resultados mediante el llamado al método `$rs->free()` o con `mysqli_free_result()`, si utiliza el enfoque basado en procedimientos.

Después de esto es conveniente cerrar la conexión con la base de datos. Para ello puede utilizar:

```
$cn->close();
```

O

```
mysqli_close($cn);
```

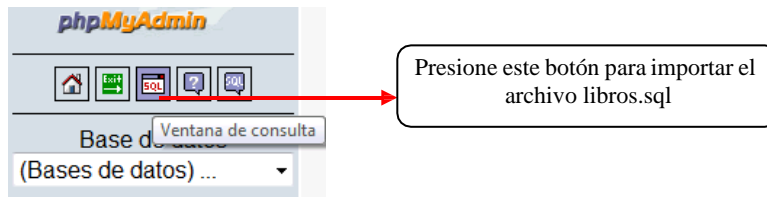
III. MATERIALES Y EQUIPO

Para la realización de la guía de práctica se requerirá lo siguiente:

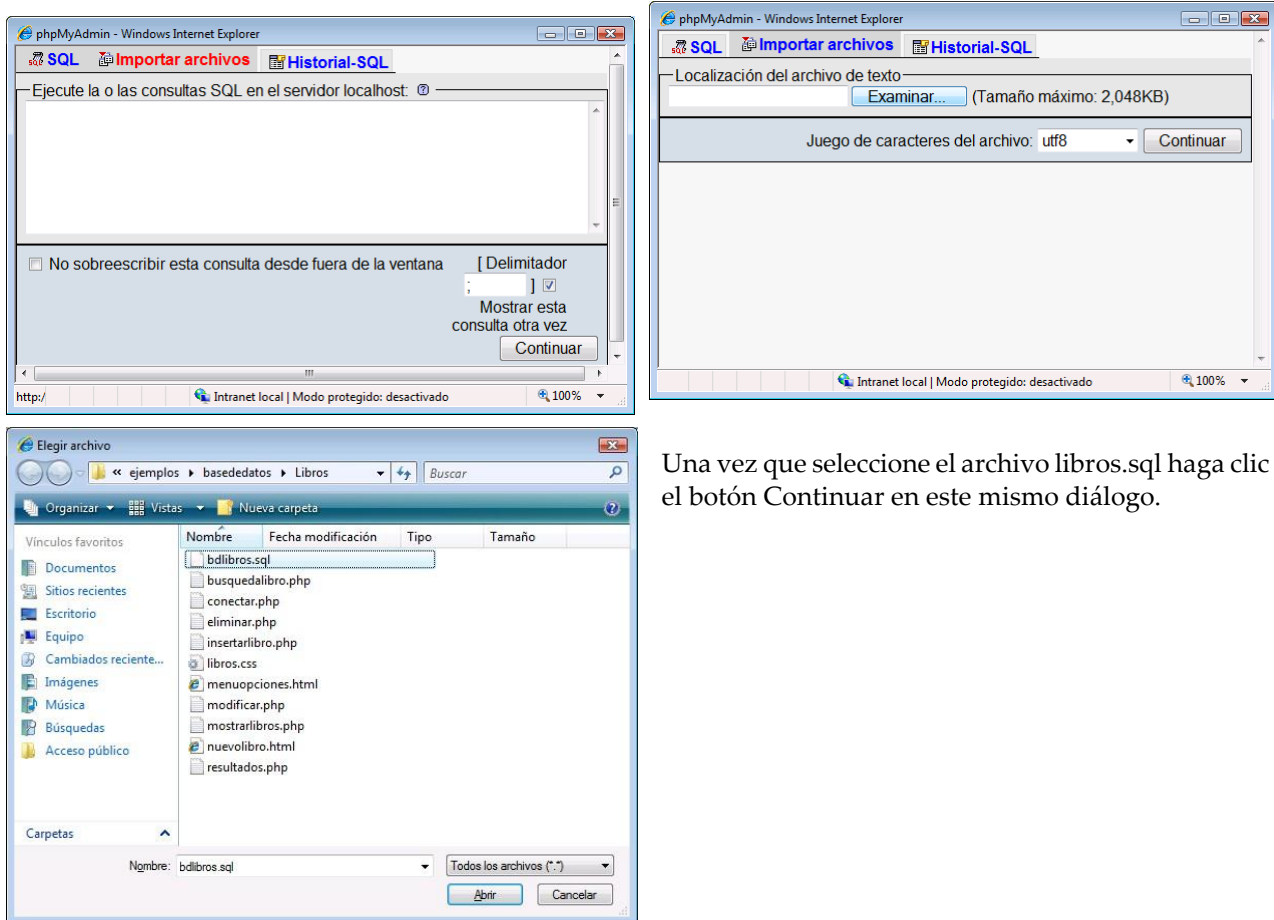
No .	Requerimiento	Cantidad
1	Guía de práctica #9: Utilización de PHP para interactuar con bases de datos MySQL	1
2	Computadora con WampServer y Sublime Text 3 instalado	1
3	Memoria USB	1

IV. PROCEDIMIENTO

Tome el archivo de recursos de la guía 7, denominado recursos-guia-7.zip complementario a esta guía de práctica. Descomprímalo y restaure la base de datos importando la estructura y los registros que vienen en el archivo libros.sql. La creación de la base de datos y su selección ya vienen incluidas en el script. Por tanto, no debe crear la base de datos. Para esto debe importar el script desde los botones del panel izquierdo, como se muestra:



En el diálogo que se aparece hacer clic en la **ficha Importar archivos** y luego hacer clic en el **botón Examinar** para seleccionar el archivo a importar.



Una vez que seleccione el archivo libros.sql haga clic en el botón Continuar en este mismo diálogo.

Si el script se ejecuta con éxito, entonces phpMyAdmin le debe mostrar un mensaje que le indique que la importación se ejecutó exitosamente. La base de datos se creará durante la ejecución del script.

Ejercicio #1. La siguiente aplicación consulta una base de datos de libros y le permite administrarla con las operaciones típicas que se realizan sobre las tablas de la base, como insertar nuevo material, en este caso libros, modificar y eliminar títulos, así como realizar consultas por similitud y búsqueda exacta por uno de los ítems seleccionados, ISBN, autor o título del libro.

Script #1: menuopciones.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8" />
  <title>Sistema de gestión de libros</title>
  <link rel="stylesheet" href="css/vertical-nav.css" />
  <script src="js/modernizr.custom.lis.js"></script>
</head>
<body>
<header>
  <h1>Gestión de libros</h1>
</header>
<section>
<article id="menubuttons">
  <div class="contenedor" id="uno">
    <a href="menuopciones.html" target="_self" title="Menú principal">
      
    </a>
    <p class="texto">Inicio</p>
  </div>
  <div class="contenedor" id="dos">
    <a href="nuevolibro.html" target="_self" title="Nuevo libro">
      
    </a>
    <p class="texto">Nuevo libro</p>
  </div>
  <div class="contenedor" id="cuatro">
    <a href="mostrarlibros.php?opc=modificar" target="_self" title="Modificar libro">
      
    </a>
    <p class="texto">Modificar libro</p>
  </div>
  <div class="contenedor" id="cinco">
    <a href="mostrarlibros.php?opc=eliminar" target="_self" title="Eliminar libro">
      
    </a>
    <p class="texto">Eliminar libro</p>
  </div>
  <div class="contenedor" id="seis">
    <a href="busquedalibro.html" target="_self" title="Consultar libro">
      
    </a>
    <p class="texto">Consultar libro(s)</p>
  </div>
</article>
</section>
</body>
</html>
```

```

    </div>
</section>
</article>
</body>
</html>

```

Script #2: nuevolibro.html

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8" />
    <title>...::: Nuevo libro :::...</title>
    <link rel="stylesheet" href="css/vertical-nav.css" />
    <link rel="stylesheet" href="css/formoid-solid-purple.css" />
    <link rel="stylesheet" href="css/links.css" />
    <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
    <script type="text/javascript">
        if(typeof jQuery === 'undefined') {
            var e = document.createElement("script");
            e.src = "js/modernizr.custom.lis.js";
            document.getElementsByTagName("head")[0].appendChild(e);
        }
    </script>
</head>
<body>
<header>
    <h1 class="3d-text">Nuevo libro</h1>
</header>
<section>
<article>
<form action="insertarlibro.php" method="POST" class="formoid-solid-purple">
    <div class="title">
        <h2>Nuevo libro</h2>
    </div>
    <div class="element-number">
        <label class="title"></label>
        <div class="item-cont">
            <input type="text" name="isbn" placeholder="ISBN" maxlength="18" class="large"
/>
            <span class="icon-place"></span>
        </div>
    </div>
    <div class="element-name">
        <label class="title"></label>
        <div class="nameFirst">
            <input type="text" name="autor" placeholder="Autor del libro" maxlength="50"
class="large" />
            <span class="icon-place"></span>
        </div>

```

```

</div>
<div class="element-input">
  <label class="title"></label>
  <div class="item-cont">
    <input type="text" name="titulo" placeholder="Titulo del libro" maxlength="70"
class="large" />
    <span class="icon-place"></span>
  </div>
</div>
<div class="element-number">
  <label class="title"></label>
  <div class="item-cont">
    <input type="text" name="precio" placeholder="Precio del libro" maxlength="8"
class="large" />
    <span class="icon-place"></span>
  </div>
</div>
<div class="submit">
  <input type="submit" name="enviar" value="Agregar" />
</div>
</form>
<a href="menuopciones.html" class="a-btn">
  <span class="a-btn-symbol">i</span>
  <span class="a-btn-text">Regresar</span>
  <span class="a-btn-slide-text">al menú</span>
  <span class="a-btn-slide-icon"></span>
</a>
</article>
</section>
</body>
</html>

```

Script #3: busquedaalibro.html

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8" />
  <title>...::: Formulario de búsqueda :::...</title>
  <link rel="stylesheet" href="css/vertical-nav.css" />
  <link rel="stylesheet" href="css/formoid-solid-purple.css" />
  <link rel="stylesheet" href="css/links.css" />
  <script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
  <script type="text/javascript">
    if(typeof jQuery === 'undefined') {
      var e = document.createElement("script");
      e.src = "js/modernizr.custom.lis.js";
      document.getElementsByTagName("head")[0].appendChild(e);
    }
  </script>

```

```

</head>
<body>
<header>
    <h1 class="3d-text">Búsqueda de libro</h1>
</header>
<section>
<article>
<form action="resultados.php" method="POST" class="formoid-solid-purple">
    <div class="title">
        <h2>Catálogo de búsqueda</h2>
    </div>
    <div class="element-select">
        <label class="title"></label>
        <div class="item-cont">
            <div class="large">
                <span>
                    <select name="tema">
                        <option value="autor">Autor</option>
                        <option value="titulo">Título</option>
                        <option value="isbn">ISBN</option>
                    </select>
                    <i></i>
                    <span class="icon-place"></span>
                </span>
            </div>
        </div>
    </div>
    <div class="element-input">
        <label class="title"></label>
        <div class="item-cont">
            <input type="text" name="termino" placeholder="Término de búsqueda"
class="large" />
            <span class="icon-place"></span>
        </div>
    </div>
    <div class="element-checkbox">
        <label class="title">Tipo de búsqueda:</label>
        <div class="column column1">
            <label>
                <input type="checkbox" name="tipobusqueda" value="exacta" / >
                <span>Búsqueda exacta</span>
            </label>
        </div>
        <span class="clearfix"></span>
    </div>
    <div class="submit">
        <input type="submit" name="enviar" value="Buscar" />
    </div>
</form>

```

```

<script src="js/formoid-solid-purple.js"></script>
<a href="menuopciones.html" class="a-btn">
    <span class="a-btn-symbol">i</span>
    <span class="a-btn-text">Regresar</span>
    <span class="a-btn-slide-text">al menú</span>
    <span class="a-btn-slide-icon"></span>
</a>
</article>
</section>
</body>
</html>

```

Script #4: db-mysqli.php

```

<?php
//Datos de conexión
define("DBHOST", "localhost");
define("DBUSER", "root");
define("DBPASS", "");
define("DBDATA", "libros");

//Creando el objeto de conexión a la base de datos con MySQLi
$db = new mysqli(DBHOST, DBUSER, DBPASS, DBDATA);
//Verificar que la conexión se ha realizado o terminar
//el programa o secuencia de comando si no ha sido así
if($db->connect_errno){
    die("No se ha podido conectar a MySQL: (" . $db->connect_errno . ") " . $db->connect_error);
}
//Establecer el conjunto de caracteres para no tener problemas
//con los caracteres especiales del idioma español
$db->set_charset("utf8");

?>

```

Script #5: mostrarlibros.php

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8" />
    <title>Libros de la base de datos</title>
    <link rel="stylesheet" href="css/vertical-nav.css" />
    <link rel="stylesheet" href="css/libros.css" />
    <link rel="stylesheet" href="css/links.css" />
    <script src="js/modernizr.custom.lis.js"></script>
</head>
<header>
    <h1 class="3d-text">Libros disponibles</h1>
</header>

```



```

<body>
<?php
    //Incluir librería de conexión a la base de datos
    include_once("db-mysqli.php");

    //Si se ha llamado esta página desde el formulario
    //para modificar libros ejecutar primero la actualización
    //del registro
    if(isset($_POST['guardar'])) {
        //Creando variables locales con los datos enviados
        //desde el formulario de modificación
        $isbnx = isset($_GET['id']) ? trim($_GET['id']) : "";
        $isbn = isset($_POST['isbn']) ? trim($_POST['isbn']) : "";
        $autor = isset($_POST['autor']) ? trim($_POST['autor']) : "";
        $titulo = isset($_POST['titulo']) ? trim($_POST['titulo']) : "";
        $precio = isset($_POST['precio']) ? trim($_POST['precio']) : "";
        //Verificando que se hayan ingresado datos
        //en todos los controles del formulario
        if(empty($isbn) || empty($autor) || empty($titulo) || empty($precio)) {
            $msg = "Existen campos en el formulario sin llenar. ";
            $msg .= "Regrese al formulario y llene todos los campos. <br>\n";
            $msg .= "[<a href=\"modificar.php?id=\" . $isbnx . "\">Volver</a>]\n";
            echo $msg;
            exit(0);
        }

        $isbnx = addslashes($isbnx);
        $isbn = addslashes($isbn);
        $autor = addslashes($autor);
        $titulo = addslashes($titulo);
        $precio = doubleval($precio);

        //Creando la consulta de actualización con los datos
        //enviados del formulario de modificación de libros
        $consulta = "UPDATE libros SET isbn='" . $isbn . "', autor='" . $autor;
        $consulta .= "', titulo='" . $titulo . "', precio=" . $precio . " WHERE isbn='" .
        $isbnx . "'";

        //Ejecutando la consulta de actualización
        $resultc = $db->query($consulta);
        //Obteniendo el número de registros actualizados
        $num_results = $db->affected_rows;
        echo "<div class=\"query\">\n<t<p>";
        echo "\t\t" . $num_results . " fila(s) actualizada(s)\n";
        echo "\t</p>\n</div>\n";
        $_GET['opc'] = "modificar";
    }

    if(isset($_GET['del']) && $_GET['del'] == "s") {
        $consulta = "DELETE FROM libros WHERE isbn='" . $_GET['id'] . "'";
        $resultc = $db->query($consulta);
    }

```

```

$num_results = $db->affected_rows;
echo "<div class=\"query\">\n\t<p>" . $consulta . "</p>\n";
echo "Se ha eliminado" . $num_results . " registro de isbn = " . $_GET['id'] .
"\n</div>\n";
}

//Haciendo una consulta de todos los libros presentes
//en la tabla libros
$consulta = "SELECT * FROM libros ORDER BY autor";
//Ejecutando la consulta a través del objeto $db
$resultc = $db->query($consulta);
//Obteniendo el número de registros devueltos
$num_results = $resultc->num_rows;
echo "<table class=\"color-table\">\n
\t<colgroup>\n
\t\t<col class=\"isbn\">\n
\t</colgroup>\n
\t<colgroup>\n
\t\t<col class=\"info\">\n
\t\t<col class=\"info\">\n
\t</colgroup>\n
\t<colgroup>\n
\t\t<col class=\"price\">\n
\t</colgroup>\n
\t<colgroup>\n
\t\t<col class=\"action\">\n
\t</colgroup>\n
\t<thead>\n
\t\t<tr id=\"thead\">\n
\t\t\t<th>ISBN</th>\n
\t\t\t<th>AUTOR</th>\n
\t\t\t<th>TÍTULO</th>\n
\t\t\t<th>PRECIO</th>\n
\t\t\t<th>ACCIÓN</th>\n
\t\t</tr>\n
\t</thead>\n
\t<tbody>\n";
while($row = $resultc->fetch_assoc()){
    echo "\t\t<tr class=\"normal\" onmouseover=\"this.className='selected'\"
onmouseout=\"this.className='normal'\">\n";
    echo "\t\t\t<td>\n";
    echo "\t\t\t\t" . $row['isbn'] . "\n";
    echo "\t\t\t</td>\n\t\t\t\t<td>\n";
    echo "\t\t\t\t" . stripslashes($row['autor']) . "\n";
    echo "\t\t\t</td>\n\t\t\t\t<td>\n";
    echo "\t\t\t\t" . stripslashes($row['titulo']) . "\n";
    echo "\t\t\t</td>\n\t\t\t\t<td>$ \n";
    echo "\t\t\t\t" . $row['precio'];
    echo "\t\t\t\t</td>\n\t\t\t\t<td align=\"center\">\n";

```

```

        echo "\t\t\t\t\t[<a href=\"\" . $_GET['opc'] . ".php?id=\" . $row['isbn'] . "\">\n";
        echo "\t\t\t\t\t\t\" . $_GET['opc'] . "\n";
        echo "\t\t\t\t\t\t</a>]\n";
        echo "\t\t\t\t\t</td>\n\t\t\t</tr>\n";
    }
    echo "\t</tbody>\n";
    echo "\t<tfoot>\n";
    echo "\t\t<tr id=\"tfooter\">\n";
    echo "\t\t\t<th colspan=\"5\">\n";
    //Mostrando el número total de registros de la tabla libros
    echo "\t\t\t\tNúmero de registros: \" . $num_results . "\n";
    echo "\t\t\t\t</th>\n";
    echo "\t</tr>\n";
    echo "\t</tfoot>\n";
    echo "</table>\n";
?>
<a href="menuopciones.html" class="a-btn">
    <span class="a-btn-symbol">i</span>
    <span class="a-btn-text">Regresar</span>
    <span class="a-btn-slide-text">al menú</span>
    <span class="a-btn-slide-icon"></span>
</a>
</body>
</html>

```

Script #6: insertarlibro.php

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8" />
    <title>Resultados al agregar libro</title>
    <link rel="stylesheet" href="css/vertical-nav.css" />
    <link rel="stylesheet" href="css/links.css" />
    <script src="js/modernizr.custom.lis.js"></script>
</head>
<body>
<header>
    <h1 class="3d-text">Resultado al agregar libro a la base de datos</h1>
</header>
<section>
<article>
<?php
    //Asignando los datos del formulario
    //a variables locales con nombres cortos
    $isbn = trim($_POST['isbn']);
    $autor = trim($_POST['autor']);
    $titulo = trim($_POST['titulo']);
    $precio = trim($_POST['precio']);

```

```

//Verificando que se hayan ingresado datos
//en todos los controles del formulario
if(empty($isbn) || empty($autor) || empty($titulo) || empty($precio)){
    $msg = "Existen campos en el formulario sin llenar. ";
    $msg .= "Regrese al formulario y llene todos los campos. <br />\n";
    $msg .= "[<a href=\"nuevolibro.html\">Volver</a>]\n";
    echo $msg;
    exit(0);
}

$isbn = addslashes($isbn);
$autor = addslashes($autor);
$titulo = addslashes($titulo);
$precio = doubleval($precio);

//Incluir librería de conexión a la base de datos
include_once("db-mysqli.php");

//Realizando la consulta para insertar
//el nuevo registro a la base de datos
$planconsulta = "INSERT INTO libros (isbn, autor, titulo, precio) ";
$planconsulta .= "VALUES (?, ?, ?, ?)";
$sentencia = $db->prepare($planconsulta);
$sentencia->bind_param("sssd", $isbn, $autor, $titulo, $precio);
$sentencia->execute();
echo "<div class=\"query\">\n\t<p>\n\t\t";
echo $sentencia->affected_rows . " libro(s) agregado(s) a la base de datos\n";
echo "</p>\n</div>\n";
$sentencia->close();

/* $consulta = "INSERT INTO libros (isbn, autor, titulo, precio) ";
$consulta .= "VALUES ('" . $isbn . "', '" . $autor . "', '" . $titulo . "', " . $precio .
")";
$resultc = $db->query($consulta);
if($resultc){
    echo $db->affected_rows . " libro agregado a la base de datos.";
} */
//Cerrar la conexión
$db->close();
?>
<br />
<a href="nuevolibro.html" class="a-btn">
    <span class="a-btn-symbol">i</span>
    <span class="a-btn-text">Agregar</span>
    <span class="a-btn-slide-text">otro libro</span>
    <span class="a-btn-slide-icon"></span>
</a>
<a href="menuopciones.html" class="a-btn">
    <span class="a-btn-symbol">i</span>

```

```

        <span class="a-btn-text">Regresar</span>
        <span class="a-btn-slide-text">al menú</span>
        <span class="a-btn-slide-icon"></span>
    </a>
</article>
</section>
</body>
</html>

```

Script #7: modificar.php

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8" />
    <title>Modificar un libro</title>
    <link rel="stylesheet" href="css/vertical-nav.css" />
    <link rel="stylesheet" href="css/formoid-solid-purple.css" />
    <link rel="stylesheet" href="css/links.css" />
    <!-- <link rel="stylesheet" href="css/formdesign.css" /> -->
    <script src="js/modernizr.custom.lis.js"></script>
</head>
<body>
<header>
    <h1 class="3d-text">Modificar libro</h1>
</header>
<section>
<article>
<?php
    //Incluir librería de conexión a la base de datos
    include_once("db-mysqli.php");

    //Haciendo una consulta de todos los libros presentes
    //en la tabla libros
    $consulta = "SELECT * FROM libros WHERE isbn='" . $_GET['id'] . "'";
    //echo $consulta . "<br>\n";
    //Ejecutando la consulta a través del objeto $db
    $resultc = $db->query($consulta);
    //Obteniendo el número de registros devueltos
    $num_results = $resultc->num_rows;
    $row = $resultc->fetch_assoc();
?>
<form action="mostrarlibros.php?id=?php echo $_GET['id'] ?>" method="POST" class="formoid-
solid-purple">
    <div class="title">
        <h2>Modificar la información del libro</h2>
    </div>
    <div class="element-number">
        <label class="title"></label>

```

```

        <div class="item-cont">
            <input type="text" name="isbn" value="<?php echo $row['isbn'] ?>" maxlength="18"
placeholder="ISBN" class="large" />
            <span class="icon-place"></span>
        </div>
    </div>
    <div class="element-name">
        <label class="title"></label>
        <div class="nameFirst">
            <input type="text" name="autor" value="<?php echo $row['autor'] ?>" maxlength="50"
placeholder="Autor" class="large" />
            <span class="icon-place"></span>
        </div>
    </div>
    <div class="element-input">
        <label class="title"></label>
        <div class="item-cont">
            <input type="text" name="titulo" value="<?php echo $row['titulo'] ?>"
maxlength="70" placeholder="Titulo" class="large" />
            <span class="icon-place"></span>
        </div>
    </div>
    <div class="element-number">
        <label class="title"></label>
        <div class="item-cont">
            <input type="text" name="precio" value="<?php echo $row['precio'] ?>" maxlength="8"
placeholder="Precio" class="large" />
            <span class="icon-place"></span>
        </div>
    </div>
    <div class="submit">
        <input type="submit" name="guardar" value="Guardar" />
    </div>
</form>
<!--
<form action="mostrarlibros.php?id=<?php echo $_GET['id'] ?>" method="POST">
<fieldset>
<legend><span>Modificar la información de un libro</span></legend>
<ul>
    <li>
        <label for="isbn" class="item">ISBN: </label>
        <div class="campo">
            <input type="text" name="isbn" value="<?php echo $row['isbn'] ?>" size="18"
maxlength="18" placeholder="ISBN" />
        </div>
    </li>
    <li>
        <label for="autor" class="item">Autor: </label>
        <div class="campo">

```

```

        <input type="text" name="autor" value="<?php echo $row['autor'] ?>" size="36"
maxlength="50" placeholder="Autor" />
    </div>
</li>
<li>
    <label for="titulo" class="item">Título: </label>
    <div class="campo">
        <input type="text" name="titulo" value="<?php echo $row['titulo'] ?>" size="36"
maxlength="60" placeholder="T&iacute;tulo" />
    </div>
</li>
<li>
    <label for="precio" class="item">Precio: </label>
    <div class="campo">
        <input type="text" name="precio" value="<?php echo $row['precio'] ?>" size="6"
maxlength="6" placeholder="Precio" />
    </div>
</li>
<li>
    <div class="boton">
        <input type="submit" name="guardar" value="Guardar" />
    </div>
</li>
</ul>
</fieldset>
</form>
-->
<a href="mostrarlibros.php?opc=modificar" class="a-btn">
    <span class="a-btn-symbol">i</span>
    <span class="a-btn-text">Volver</span>
    <span class="a-btn-slide-text">a la tabla de modificación</span>
    <span class="a-btn-slide-icon"></span>
</a>
</article>
</section>
</body>
</html>

```

Script #8: eliminar.php

```

<?php
    //Incluir librería de conexión a la base de datos
    include_once("db-mysqli.php");
    $isbn = $_GET['id'];
    $sql = "SELECT * FROM libros WHERE isbn = '" . $isbn . "'";
    $result = $db->query($sql);
    $row = $result->fetch_assoc();

    $msg = "<script>\n";

```

```

$preg = "Deseas eliminar el libro de: isbn = ";
$preg .= "isbn = " . $row['isbn'] . ",";
$preg .= "autor = " . $row['autor'] . ",";
$preg .= "titulo = " . $row['titulo'] . ",";
$preg .= "precio = " . $row['precio'] . ".";

$msg = "if(confirm(\"" . $preg . "\")){";
$msg .= "location.href=\"mostrarlibros.php?opc=eliminar&del=s&id=" . $isbn . "\";}";
$msg .= "else{location.href=\"mostrarlibros.php?opc=eliminar&del=n\";}</script>";
echo utf8_decode($msg);
?>

```

Script #9: resultados.php

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8" />
    <title>Resultados de la búsqueda</title>
    <link rel="stylesheet" href="css/vertical-nav.css" />
    <link rel="stylesheet" href="css/table-column-options.css" />
    <link rel="stylesheet" href="css/links.css" />
    <!-- <link rel="stylesheet" href="css/libros.css" /> -->
    <script src="js/modernizr.custom.lis.js"></script>
</head>
<body>
<header>
    <h1 class="3d-text">Resultados de la búsqueda</h1>
</header>
<section>
<article>
<?php
    //Asignando los datos ingresados en el formulario
    //a variables locales con nombres cortos
    $tema = trim($_POST['tema']);
    $termino = (isset($_POST['termino']) && is_string($_POST['termino']) &&
    strlen($_POST['termino']) > 0) ? $_POST['termino'] : "";
    $termino = trim($termino);
    $tipobusqueda = isset($_POST['tipobusqueda']) ? $_POST['tipobusqueda'] : "";
    if(empty($tema) || empty($termino)){
        $msg = "No se ha ingresado detalle de la búsqueda. ";
        $msg .= "Regrese al formulario e ingrese los datos en el formulario.<br>";
        $msg .= "[<a href=\"busquedalibro.php\">Volver</a>]";
        echo $msg;
        exit(0);
    }
    $tema = addslashes($tema);
    $termino = addslashes($termino);
    //Incluir librería de conexión a la base de datos

```



```

include_once("db-mysqli.php");

if($tipobusqueda == 'exacta'){
    $consulta = "SELECT * FROM libros WHERE " . $tema;
    $consulta .= " = '" . $termino . "'";
}
else{
    $consulta = "SELECT * FROM libros WHERE " . $tema;
    $consulta .= " LIKE '%" . $termino . "%'";
}

echo "<div class=\"query\">\n\t<p>" . $consulta . "</p>\n\t";
$resultc = $db->query($consulta);
$num_results = $resultc->num_rows;
echo "<p>Número de libros encontrados: . $num_results</p>\n</div>\n";
echo "<table class=\"column-options\">\n";
for($i=0; $i<$num_results; $i++){
    $row = $resultc->fetch_assoc();
    echo "<colgroup>\n";
    echo "<col>\n";
    echo "</colgroup>\n";
    echo "<colgroup>\n";
    echo "<col>\n";
    echo "</colgroup>\n";
    echo "<thead>\n";
    echo "<tr class=\"odd\" onmouseover=\"this.className='selected'\"
onmouseout=\"this.className='odd'\">\n";
    echo "<th colspan=\"2\">Libro " . ($i + 1) . "</th>\n</tr>\n";
    echo "</thead>\n<tbody>\n";
    echo "<tr onmouseover=\"this.className='selected'\"
onmouseout=\"this.className=''\">\n<th>\n";
    echo "Título " . ($i + 1) . ": \n";
    echo "</th>\n<td>\n";
    echo stripslashes($row['titulo']);
    echo "\n</td>\n</tr>\n";
    echo "<tr class=\"odd\" onmouseover=\"this.className='selected'\"
onmouseout=\"this.className='odd'\">\n<th>\n";
    echo "Autor : \n";
    echo "</th>\n<td>\n";
    echo stripslashes($row['autor']);
    echo "\n</td>\n</tr>\n";
    echo "<tr onmouseover=\"this.className='selected'\"
onmouseout=\"this.className=''\">\n<th>\n";
    echo "ISBN: \n";
    echo "</th>\n<td>\n";
    echo stripslashes($row['isbn']);
    echo "\n</td>\n</tr>\n";
    echo "<tr class=\"odd\" onmouseover=\"this.className='selected'\"
onmouseout=\"this.className='odd'\">\n<th>\n";
    echo "Precio: \n";

```

```

        echo "</th>\n<td>\n";
        echo stripslashes($row['precio']);
        echo "\n</td>\n</tr>\n";
        echo "</tbody>";
    }
    echo "</table>";
    /* $msg = "<p>[<a href=\"busquedalibro.html\">realizar otra búsqueda</a>]&nbsp;&nbsp;&nbsp;";
    $msg .= "[<a href=\"menuopciones.html\">volver al menú</a>]</p>";
    echo $msg; */
    $resultc->free();
    $db->close();
?>
<a href="busquedalibro.html" class="a-btn">
    <span class="a-btn-symbol">i</span>
    <span class="a-btn-text">Realizar</span>
    <span class="a-btn-slide-text">otra búsqueda</span>
    <span class="a-btn-slide-icon"></span>
</a>
<a href="menuopciones.html" class="a-btn">
    <span class="a-btn-symbol">i</span>
    <span class="a-btn-text">Regresar</span>
    <span class="a-btn-slide-text">al menú</span>
    <span class="a-btn-slide-icon"></span>
</a>
</article>
</section>
</body>
</html>

```

En el navegador de su preferencia podrá visualizar lo siguiente:



Nuevo libro

Nuevo libro

123 978-84-415-2507-8

Baron Schawartz

MySQL Avanzado

123 94.32

Agregar

Dar clic en el botón Agregar



Regresar
AL MENÚ

1 libro(s) agregado(s) a la base de datos



Agregar
OTRO LIBRO



Regresar
AL MENÚ

Libros disponibles

ISBN	AUTOR	TÍTULO	PRECIO	ACCIÓN
978-84-415-2595-3	Abraham Gutiérrez	PHP 5 a través de ejemplos	\$ 35.15	[modificar]
978-84-415-2137-7	Andy Budd	CSS Manual avanzado	\$ 52.45	[modificar]
978-84-415-2507-8	Baron Schawartz	MySQL Avanzado	\$ 94.32	[modificar]
978-98-716-0929-1	Christian Cibelli	PHP Programación Web Avanzada para Profesionales	\$ 42.60	[modificar]
978-84-415-2388-3	Danny Goodman	JavaScript, HTML5 y CSS	\$ 68.20	[modificar]
978-84-415-2311-1	Ellie Quigley	PHP y MySQL Práctico para programadores y diseñadores web	\$ 52.30	[modificar]
978-84-481-9814-5	F. Javier Gil Rubio	Creación de sitios web con PHP 5	\$ 36.90	[modificar]
978-84-111-9999-0	Fernando M. Guardado	Creación de páginas web con DHTML	\$ 31.35	[modificar]
978-60-770-7374-1	Gregorio Gil García	El gran libro de Drupal	\$ 19.83	[modificar]
978-84-481-3173-9	Herbert Schildt	Java 2 Manual de referencia	\$ 73.40	[modificar]
978-84-415-2121-6	James Foxall	El libro de Visual C# 2005	\$ 50.25	[modificar]
978-84-415-2217-6	Jason Cranford Teague	Programación CSS, DHTML y AJAX	\$ 72.50	[modificar]
978-84-415-2961-8	Jeff Friesen	Java para Desarrollo Android	\$ 90.60	[modificar]

Búsqueda de libro

Catálogo de búsqueda

Título

JavaScript

Tipo de búsqueda:
☐ Búsqueda exacta

Buscar

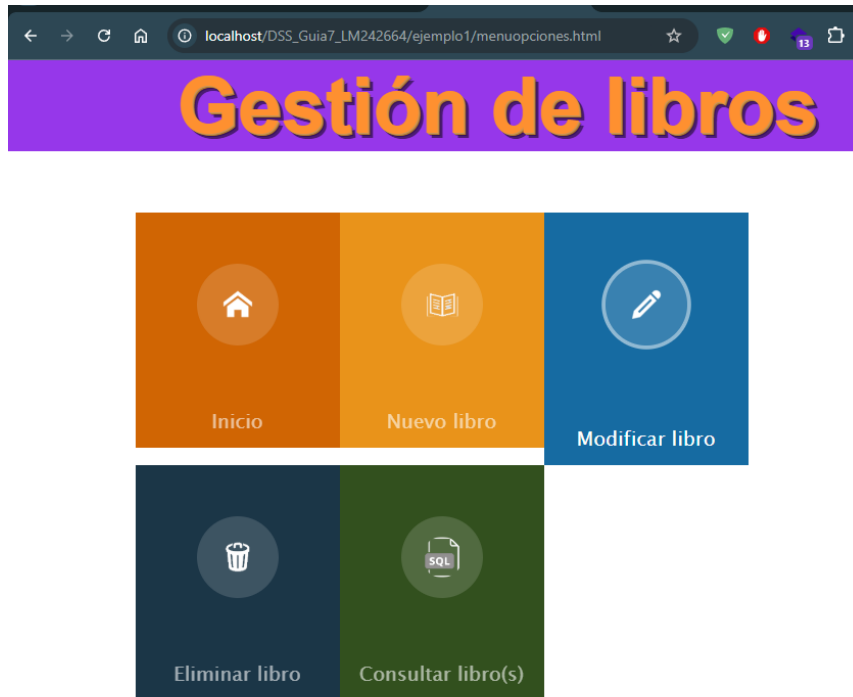
 **Regresar**
AL MENÚ

Resultados de la búsqueda

SELECT * FROM libros WHERE titulo LIKE '%JavaScript%'
Número de libros encontrados: . 6

Libro 1	
Título 1:	JavaScript Edición especial
Autor :	Paul McFedries
ISBN:	978-84-205-3392-6
Precio:	64.35
Libro 2	
Título 2:	JavaScript, HTML5 y CSS
Autor :	Danny Goodman
ISBN:	978-84-415-2388-3
Precio:	68.20
Libro 3	
Título 3:	Ajax, JavaScript y PHP
Autor :	Phil Ballard
ISBN:	978-84-415-2514-6
Precio:	49.25
Libro 4	
Título 4:	JavaScript Ninja
Autor :	John Resig
ISBN:	978-84-415-3397-4
Precio:	59.79
Libro 5	
Título 5:	JavaScript Manual de referencia
Autor :	Thomas Powell
ISBN:	978-84-481-3268-2
Precio:	72.75

Resultado personal:



Estudiante: José Adrián López Medina – LM242664
Técnico en Ingeniería en Computación – Escuela de Computacion Ciclo I – 2025

ISBN	AUTOR	TÍTULO	PRECIO	ACCIÓN
978-84-415-2595-3	Abraham Gutiérrez	PHP 5 a través de ejemplos	\$ 35.15	[modificar]
978-84-415-2137-7	Andy Budd	CSS Manual avanzado	\$ 52.45	[modificar]
978-84-415-2507-8	Baron Schawartz	MySQL Avanzado	\$ 94.32	[modificar]
978-98-716-0929-1	Christian Cibelli	PHP Programación Web Avanzada para Profesionales	\$ 42.60	[modificar]
978-84-415-2388-3	Danny Goodman	JavaScript, HTML5 y CSS	\$ 68.20	[modificar]
978-84-415-2311-1	Ellie Quigley	PHP y MySQL Práctico para programadores y diseñadores web	\$ 52.30	[modificar]

localhost/DSS_Guia7_LM242664/ejemplo1/modificar.php?id=978-84...

Modificar libro

Modificar la información del libro


 **Volver**
A LA TABLA DE
MODIFICACIÓN

Estudiante: José Adrián López Medina – LM242664
Técnico en Ingeniería en Computación – Escuela de Computacion Ciclo I – 2025

localhost/DSS_Guia7_LM242664/ejemplo1/nuevolibro.html

Nuevo libro

Nuevo libro

 **Regresar**
AL MENÚ

Estudiante: José Adrián López Medina – LM242664
Técnico en Ingeniería en Computación – Escuela de Computacion Ciclo I – 2025

Ejercicio #2. El siguiente ejemplo ilustra cómo realizar algo que se conoce como paginación de resultados procedentes de una consulta a una base de datos MySQL. Se crean dos clases, una para la gestión de la base de datos y otra para manejar la paginación de resultados. El script PHP que hace uso de estas clases es mostrarpeliculas.php, en este script se muestra como implementar la autocarga de clases y se crean las instancias para mostrar los resultados en una tabla con diseño CSS.

Script #1: udb_database.class.php

```
<?php

class database {
    private $conexion;
    private $resultc;
    private $sql;
    public static $queries;
    private static $_singleton;
    const HOST = "localhost";
    const USER = "root";
    const PASS = "";
    const DB = "peliculas";

    public static function getInstance(){
        if(is_null (self::$_singleton)) {
            self::$_singleton = new DataBase();
        }
        return self::$_singleton;
    }

    private function __construct(){
        $this->conexion = new mysqli(self::HOST, self::USER, self::PASS, self::DB);
        self::$queries = 0;
    }

    /*public function execute(){
        if(!($this->resource = mysql_query($this->sql, $this->conexion))){
            return null;
        }
        self::$queries++;
        return $this->resource;
    }*/

    public function alter(){
        if(!($this->resource = mysql_query($this->sql, $this->conexion))){
            return false;
        }
        return true;
    }

    public function loadObjectList()
```

```

//Ejecutando la consulta a través del objeto $db
$this->resultc = $this->conexion->query($this->sql);
//Obteniendo el número de registros devueltos
$num_results = $this->resultc->num_rows;

while($row = $this->resultc->fetch_assoc()){
    $array[] = $row;
}

return $array;
}

public function setQuery($sql){
    if(empty($sql)){
        return false;
    }
    $this->sql = $sql;
    return true;
}

public function getNumRows($sql){
    if($this->setQuery($sql)){
        //Ejecutando la consulta a través del objeto $db
        $this->resultc = $this->conexion->query($this->sql);
        //Obteniendo el número de registros devueltos

        while($row = $this->resultc->fetch_assoc()){
            $total = $row['total'];
        }

        return $total;
    }
    else{
        return false;
    }
}
}
?>

```

Script #2: udb_paginacion.class.php

```

<?php
//Definición de la clase
class paginacion {
    //Propiedades de la clase paginacion
    private $numpage;
    private $totalpages;
    private $links;
    private $offset;

```



```

//Métodos de la clase paginacion
//El constructor recibe tres argumentos que son:
//1. $page: El número de la página de resultados que se va a cargar
//(valores desde 1 hasta el número de páginas totales de resultados)
//2. $limit: El número máximo de resultados por página.
//3. $total: El número total de registros obtenidos después de hacer la consulta.
public function __construct(){
    $this->numpage = 1;
    $this->totalpages = 1;
    $this->links = array();
    $this->offset = 0;
}

public function getnumpages($page){
    $this->numpage = (int)$page;
    if($this->numpage < 1){
        $this->numpage = 1;
    }
    return $this->numpage;
}

public function getoffset($limit){
    $this->offset = (($this->numpage)-1) * $limit;
    return $this->offset;
}

public function gettotalpages($total, $limit){
    $this->totalpages = ceil($total / $limit);
    return $this->totalpages;
}

public function showlinkspages($total, $limit){
    $totpages = $this->gettotalpages($total, $limit);
    for($i=1; $i<=$totpages; $i++){
        $this->links[] = ($i == $this->numpage) ? $i : "<a href=\"?npag=$i\">$i</a>";
    }
    return implode(" - ", $this->links);
}
}
?>

```

Script #3: mostrarpeliculas.php

```

<?php
spl_autoload_register(function ($classname){
    require_once("udb_" . $classname . ".class.php");
});

```

```

//Definiendo el número total de registros que se van a mostrar
define("LIMIT", 5);
//Creando objeto de la clase paginacion
$paginacion = new paginacion();

//Obteniendo el número de página de resultados solicitada
$page = isset($_GET['npag']) ? $_GET['npag'] : 1;
$paginacion->getnumpages($page);

//Creando una instancia de la clase database
$db = DataBase::getInstance();
$sql = "SELECT SQL_CALC_FOUND_ROWS titulopelicula, descripcion, ";
$sql .= "nombre, imgpelicula, generopelicula FROM pelicula ";
$sql .= "JOIN genero ON pelicula.idgenero = genero.idgenero ";
$sql .= "JOIN director ON pelicula.iddirector = director.iddirector ";
$sql .= "LIMIT " . $paginacion->getoffset(LIMIT) . ", " . LIMIT;

$sqlTotal = "SELECT FOUND_ROWS() AS total";

//Guardando el set de resultados de la consulta a la base de datos en $pelis
$db->setQuery($sql);
$pelis = $db->loadObjectList();

//Obteniendo el total de registros que se van a paginar de la consulta $sql
// $db->setQuery($sqlTotal);
$regstotal = $db->getNumRows($sqlTotal);
//echo $regstotal . "<br />\n";
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Consultas de varias tablas</title>
    <link rel="stylesheet" href="css/tablas.css" />
    <script src="js/modernizr.custom.lis.js"></script>
</head>
<body>
<section>
<article>
<?php
    //Creando la tabla a mostrar
    $tabla = "<table class=\"tablaps\">\n";
    $tabla .= "<caption>Información de las películas en existencia</caption>\n";
    $tabla
    "<thead>\n<tr>\n\t<th>\nTÍTULO\n</th>\n\t<th>PORTADA</th>\n<th>\nSINOPSIS\n</th>";
    $tabla .= "\n<th>\nDIRECTOR\n</th>\n<th>\nGÉNERO\n</th>\n</tr>\n</thead>\n<tbody>\n";
    $contador = 1;
    foreach($pelis as $pelicula){
        if($contador%2 == 1) $clase = "impar";




```

```

        else $clase = "par";
        $tabla .= "<tr class=\"\" . $clase . \"\">\n<td>\n" . $pelicula['titulopelicula'] .
"</td>\n";
        $tabla .= "<td>\n<img src=\"\" . $pelicula['imgpelicula'] . \"\" alt=\"\" .
$pelicula['nombre'] . \"\" /></td>\n";
        $tabla .= "<td>\n" . $pelicula['descripcion'] . "</td>\n";
        $tabla .= "<td>\n" . $pelicula['nombre'] . "</td>\n";
        $tabla .= "<td>\n" . $pelicula['generopelicula'] . "</td>\n</tr>\n";
        $contador++;
    }
    $tabla .= "</tbody>\n<tfoot>\n<tr>\n<th colspan=\"5\">\n" . $paginacion-
>showlinkspages($regstotal, LIMIT) . "</th>\n</tr>\n</tfoot>\n";
    $tabla .= "</table>\n";
    echo $tabla;
?>
</article>
</section>
</body>
</html>

```

Al visualizar el script en el navegador de su preferencia podrá observar:

Información de las películas en existencia				
TÍTULO	PORTADA	SINOPSIS	DIRECTOR	GÉNERO
Percy Jackson y el Ladrón del Rayo		La historia narra la vida de un estudiante que descubre ser hijo de Poseidón, a raíz de esto se ve envuelto en una carrera contra el tiempo para impedir que los dioses griegos inicien una guerra que tiene como campo de batalla el continente americano de hoy en día.	Chris Columbus	Acción
Los Vengadores 2 - La era de Ultrón		El destino del planeta pende de un hilo cuando Tony Stark intenta hacer funcionar un inactivo programa para mantener la paz. Las cosas le salen mal y los héroes más poderosos, incluyendo Iron Man, Capitán América, la Viuda Negra, Thor, el Increíble Hulk y Ojo de Halcón, se ven enfrentados a la prueba definitiva. Cuando el villano Ultrón aparece, es tarea de Los Vengadores el detenerle antes de que lleve a cabo sus terribles planes para el mundo. Inesperadas alianzas y acción por doquier sientan las bases para una épica aventura global.	Joss Whedon	Acción
Batman v Superman: El origen de la justicia		Superman se ha convertido en la figura más controvertida del mundo. Mientras que muchos siguen creyendo que es un emblema de esperanza, otro gran número de personas lo consideran una amenaza para la humanidad. Para el influyente Bruce Wayne, Superman es claramente un peligro para la sociedad, su poder resulta imprudente y alejado de la mano del gobierno. Por eso, ante el temor de las acciones que pueda llevar a cabo un superhéroe con unos poderes casi divinos, decide ponerse la máscara y la capa para poner a raya al superhéroe de Metrópolis. Mientras que la opinión pública debate sobre el interrogante de cuál es realmente el héroe que necesitan, el Hombre de Acero y Batman, enfrentados entre sí, se sumergen en una contienda el uno contra el otro. La rivalidad entre ellos está alimentada por el rencor y la venganza, y nada puede disuadirlos de librar esta guerra. Hostigados por el multimillonario Lex Luthor, Batman y Superman se ven las caras en una lucha sin precedentes.	Zack Snyder	Acción

Resultado personal:

PD. Te Amo		La vida de Holly (Hilary Swank) se ve truncada cuando su marido, Gerry (Gerard Butler), muere. Incapaz de salir adelante por sí misma, su madre y sus amigos intentan animarla. Un día, después de su 30 cumpleaños, Holly recibe una carta de Gerry animándola a salir, a divertirse, a seguir adelante. Cada mes recibirá una carta firmada con un "Posdata: Te amo", que le devolverán las ganas de vivir.	Richard LaGravenese	Comedia Romántica
Efecto mariposa		Evan Treborn, un joven que se está esforzando por superar unos dolorosos recuerdos de su infancia, descubre una técnica que le permite viajar atrás en el tiempo y ocupar su cuerpo de niño para poder cambiar el curso de su dolorosa historia. Sin embargo también descubre que cualquier mínimo cambio en el pasado altera enormemente su futuro.	Eric Bress	Suspense

Estudiante: José Adrián López Medina - LM242664

Técnico en Ingeniería en Computación - Escuela de Computacion Ciclo I - 2025

Ejercicio #3. El siguiente ejemplo muestra cómo utilizar la extensión PDO para realizar las operaciones básicas con registros sobre una tabla de usuarios de una base de datos de prueba. En el ejemplo, se ha utilizado el framework Bootstrap para mejorar la apariencia de los formularios de una forma rápida.

Script #1: index.php

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8" />
    <title>CRUD con PDO</title>
    <link rel="stylesheet" href="css/bootstrap.min.css" />
</head>
<body>
<div class="content">
    <div class="row">
        <div class="col-md-12">
            <div class="row">
                <div class="col-offset-2 col-md-8">
```

```

        <h1>Crear, Obtener, Actualizar y Borrar (CRUD) con
PDO.</h1>
        </div>
    </div>
</div>
</div>
<div class="row">
    <div class="col-offset-2 col-md-8">
        <div class="row">
            <div class="col-md-offset-2 col-md-8">
                <a href="create.php" class="btn btn-primary">Create</a>
            </div>
        </div>
    </div>
</div>
<div class="row">
    <div class="col-md-offset-2 col-md-8">
        <table class="table table-bordered">
            <thead>
                <tr>
                    <th class="text-center">#</th>
                    <th class="text-center">Nombre</th>
                    <th class="text-center">Apellido</th>
                    <th class="text-center">Operaciones</th>
                </tr>
            </thead>
            <tbody>
                <?php
                    include 'connection.php';
                    $pdocon = Database::connect();
                    $sql = ('SELECT * FROM usuario ORDER BY idusuario
DESC');

                    foreach ($pdocon->query($sql) as $row) {
                        echo '
                        <tr>
                            <td class="text-center">'. $row["idusuario"]. '</td>
                            <td class="text-center">'. $row["nombre"]. '</td>
                            <td class="text-center">'. $row["apellido"]. '</td>
                            <td class="text-center">
                                <a
                                    href="read.php?id='. $row["idusuario"]. '" class="btn btn-default">Obtener</a>
                                <a
                                    href="update.php?id='. $row["idusuario"]. '" class="btn btn-success">Modificar</a>
                                <a
                                    href="read.php?id='. $row["idusuario"]. '" class="btn btn-danger">Eliminar</a>
                                </td>
                            </tr>';
                    }
                </tbody>
            </table>
    </div>
</div>

```

```

        }
        ?>
    </tbody>
</table>

</div>
</div>
</div>
<script src="http://code.jquery.com/jquery-1.11.2.min.js"></script>
<script src="js/bootstrap.min.js" ></script>
</body>
</html>

```

Script #2: connection.php

```

<?php
class Database {
    private static $dbName = 'tienda' ;
    private static $dbHost = 'localhost' ;
    private static $dbUsername = 'root';
    private static $dbUserPassword = '';
    private static $cont = null;

    //Constructor de la clase
    public function __construct() {
        die('Init function is not allowed');
    }

    public static function connect() {
        //Una sola conexión para toda la aplicación
        if (null == self::$cont) {
            try {
                self::$cont = new
PDO("mysql:host=".self::$dbHost.";". "dbname=".self::$dbName.";charset=utf8",
self::$dbUsername, self::$dbUserPassword);
            } catch(PDOException $e) {
                die($e->getMessage());
            }
        }
        return self::$cont;
    }

    public static function disconnect() {
        self::$cont = null;
    }
}
?>

```

Script #3: create.php

```

<!DOCTYPE html>

```

```

<html lang="es">
<head>
  <meta charset="utf-8" />
  <title>Ingresar nuevo usuario con PDO</title>
  <link rel="stylesheet" href="css/bootstrap.min.css" />
</head>
<body>
<div class="content">
  <div class="row">
    <div class="col-md-12">
      <div class="row">
        <div class="col-offset-2 col-md-8">
          <h1>Crear un nuevo usuario</h1>
        </div>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-offset-2 col-md-8">
      <div class="row">
        <div class="col-md-offset-2 col-md-8">
          <a href="index.php" class="btn btn-default">Back</a>
        </div>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-md-offset-3 col-md-6">
      <form action="create.php" method="POST">
        <div class="form-group">
          <label for="-">Nombre</label>
          <input type="text" class="form-control" placeholder="Nombre" name="nombre"
id="nombre" />
        </div>
        <div class="form-group">
          <label for="">Apellido</label>
          <input type="text" class="form-control" placeholder="Apellido"
name="apellido" id="apellido" />
        </div>
        <div class="form-group">
          <label for="password">Contraseña</label>
          <input type="password" class="form-control" id="Password"
placeholder="password" name="password" id="password" />
        </div>
        <div class="form-group">
          <label for="edad">Edad</label>
          <input type="text" class="form-control" placeholder="Edad" name="edad"
id="edad" />
        </div>
        <div class="form-group">

```



```

        <label for="genero">Género</label>
        <select name="genero" id="genero" class="form-control">
            <option value="M">Masculino</option>
            <option value="F">Feminino</option>
        </select>
    </div>
    <div class="form-group">
        <label for="-">Ciudad</label>
        <input type="text" class="form-control" placeholder="Ciudad" name="ciudad"
id="ciudad" />
    </div>
    <button type="submit" class="btn btn-success">Enviar</button>
    </form>
</div>
</div>
</div>
<script src="http://code.jquery.com/jquery-1.11.2.min.js"></script>
<script src="js/bootstrap.min.js" ></script>
</body>
</html>
<?php
include 'connection.php';
if (!empty($_POST)) {
    $nombre = trim($_POST['nombre']);
    $apellido = trim($_POST['apellido']);
    $password = md5(trim($_POST['password']));
    $edad = trim($_POST['edad']);
    $genero = trim($_POST['genero']);
    $ciudad = trim($_POST['ciudad']);
    $cn = Database::connect();
    $cn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $query = $cn->prepare("INSERT INTO usuario(nombre, apellido, codigo, edad, genero,
ciudad) VALUES(?, ?, ?, ?, ?, ?)");
    $query->execute(array($nombre, $apellido, $password, $edad, $genero, $ciudad));
    Database::disconnect();
}
?>

```

Script #4: update.php

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8" />
    <title>Actualizar información del usuario con PDO</title>
    <link rel="stylesheet" href="css/bootstrap.min.css" />
</head>
<body>

```

```

<div class="content">
    <div class="row">
        <div class="col-md-12">
            <div class="row">
                <div class="col-offset-2 col-md-8">
                    <h1>Actualizar</h1>
                </div>
            </div>
        </div>
    </div>

    <div class="row">
        <div class="col-offset-2 col-md-8">
            <div class="row">
                <div class="col-md-offset-2 col-md-8">
                    <a href="index.php" class="btn btn-default">Back</a>
                </div>
            </div>
        </div>
    </div>

    <div class="row">
        <div class="col-md-offset-2 col-md-8">
            <form action="update.php" method="POST">
<?php
$id=null;
if (!empty($_GET)) {
$id=$_GET['id'];
include 'connection.php';
$cn = Database::connect();
$cn->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
$query = $cn->prepare("SELECT * FROM usuario where idusuario = ?");
$query->execute(array($_GET['id']));
$data = $query->fetch(PDO::FETCH_ASSOC);
echo '
    <div>
        <label for="-">-</label>
        <input type="text" value="'. $data["idusuario"].'" class="cod"
readonly="readonly" name="idusuario">
    </div>
    <div>
        <label for="-">-</label>
        <input type="text" value="'. $data["nombre"].'" placeholder="Nombre cuenta"
name="nombre">
    </div>
    <div>
        <label for="-">-</label>
        <input type="text" value="'. $data["apellido"].'" placeholder="Descripcion"
name="apellido">
    </div>
    <div>
        <label for="-">-</label>

```

```

        <input type="text" value="'. $data["edad"].'" placeholder="Tipo" name="edad">
    </div>
    <div>
        <label for="-">-</label>
        <input type="text" value="'. $data["genero"].'" placeholder="Er"
name="genero">
    </div>
    <div>
        <label for="-">-</label>
        <input type="text" value="'. $data["ciudad"].'" placeholder="Er"
name="ciudad">
    </div>
';
Database::disconnect();
}
?>

        <div>
            <input type="submit" class="btn btn-success" value="Actualizar">
        </div>
    </form>
</div>
</div>

<script src="http://code.jquery.com/jquery-1.11.2.min.js"></script>
<script src="js/bootstrap.min.js" ></script>
</body>
</html>
<?php
if (!empty($_POST)) {
    include 'connection.php';
    $id = trim($_POST['idusuario']);
    $nombre = trim($_POST['nombre']);
    $apellido = trim($_POST['apellido']);
    $edad = trim($_POST['edad']);
    $genero = trim($_POST['genero']);
    $ciudad = trim($_POST['ciudad']);
    $cnu = Database::connect();
    $cnu->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $query = $cnu->prepare("UPDATE usuario SET nombre = ?, apellido = ?, edad = ?, genero
= ?, ciudad= ? WHERE idusuario = ?");
    $query->execute(array($nombre, $apellido, $edad, $genero, $ciudad, $id));
    Database::disconnect();
    header("Location: index.php");
}
?>

```

Script #5: read.php

```

<!DOCTYPE html>
<html lang="es">

```

```

<head>
    <meta charset="utf-8">
    <title>Obtener información del usuario con PDO</title>
    <link rel="stylesheet" href="css/bootstrap.min.css">
</head>
<body>
<div class="content">
    <div class="row">
        <div class="col-md-12">
            <div class="row">
                <div class="col-offset-2 col-md-8">
                    <h1>Mostrar información del usuario</h1>
                </div>
            </div>
        </div>
    </div>
    <div class="row">
        <div class="col-offset-2 col-md-8">
            <div class="row">
                <div class="col-md-offset-2 col-md-8">
                    <a href="index.php" class="btn btn-default">Back</a>
                </div>
            </div>
        </div>
    </div>
    <div class="row">
        <div class="col-md-offset-2 col-md-8">
            <table class="table table-bordered" >
                <thead >
                    <tr>
                        <th class="text-center">#</th>
                        <th class="text-center">Nombre</th>
                        <th class="text-center">Apellido</th>
                        <th class="text-center">Edad</th>
                        <th class="text-center">Género</th>
                        <th class="text-center">Ciudad</th>
                    </tr>
                </thead>
                <tbody>
<?php
if (!empty($_GET)) {
//echo $_GET['id'];
include 'connection.php';

$cn = Database::connect();
$cn->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
$query = $cn->prepare("SELECT * FROM usuario where idusuario = ?");
$query->execute(array($_GET['id']));
$data = $query->fetch(PDO::FETCH_ASSOC);
echo '

```

```

        <tr>
            <td class="text-center">'. $data["idusuario"]. '</td>
            <td class="text-center">'. $data["nombre"]. '</td>
            <td class="text-center">'. $data["apellido"]. '</td>
            <td class="text-center">'. $data["edad"]. '</td>
            <td class="text-center">'. $data["genero"]. '</td>
            <td class="text-center">'. $data["ciudad"]. '</td>
        </tr>
    ';
}
else{
    echo "nada ha venido";
}
?>

                </tbody>
            </table>

        </div>
    </div>
</div>
<script src="http://code.jquery.com/jquery-1.11.2.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</body>
</html>

```

En el navegador debería visualizar lo siguiente al cargar la página index.php
Crear, Obtener, Actualizar y Borrar (CRUD) con PDO.

Create

#	Nombre	Apellido	Operaciones
13	Marcos Antonio	Villalta Cortez	<div style="display: flex; justify-content: space-around; padding: 5px;"> <div style="border: 1px solid #ccc; padding: 2px 5px; background-color: #f3f3f3;">Obtener</div> <div style="background-color: #27ae60; color: white; padding: 2px 5px;">Modificar</div> <div style="background-color: #c0392b; color: white; padding: 2px 5px;">Eliminar</div> </div>
12	Luis Alexander	Díaz Sánchez	<div style="display: flex; justify-content: space-around; padding: 5px;"> <div style="border: 1px solid #ccc; padding: 2px 5px; background-color: #f3f3f3;">Obtener</div> <div style="background-color: #27ae60; color: white; padding: 2px 5px;">Modificar</div> <div style="background-color: #c0392b; color: white; padding: 2px 5px;">Eliminar</div> </div>
11	Erika María	Landaverde Ochoa	<div style="display: flex; justify-content: space-around; padding: 5px;"> <div style="border: 1px solid #ccc; padding: 2px 5px; background-color: #f3f3f3;">Obtener</div> <div style="background-color: #27ae60; color: white; padding: 2px 5px;">Modificar</div> <div style="background-color: #c0392b; color: white; padding: 2px 5px;">Eliminar</div> </div>
9	Diana Lisette	Benítez Contreras	<div style="display: flex; justify-content: space-around; padding: 5px;"> <div style="border: 1px solid #ccc; padding: 2px 5px; background-color: #f3f3f3;">Obtener</div> <div style="background-color: #27ae60; color: white; padding: 2px 5px;">Modificar</div> <div style="background-color: #c0392b; color: white; padding: 2px 5px;">Eliminar</div> </div>
8	Julio Amílcar	Durán Umaña	<div style="display: flex; justify-content: space-around; padding: 5px;"> <div style="border: 1px solid #ccc; padding: 2px 5px; background-color: #f3f3f3;">Obtener</div> <div style="background-color: #27ae60; color: white; padding: 2px 5px;">Modificar</div> <div style="background-color: #c0392b; color: white; padding: 2px 5px;">Eliminar</div> </div>
7	Lucía Carolina	Duarte	<div style="display: flex; justify-content: space-around; padding: 5px;"> <div style="border: 1px solid #ccc; padding: 2px 5px; background-color: #f3f3f3;">Obtener</div> <div style="background-color: #27ae60; color: white; padding: 2px 5px;">Modificar</div> <div style="background-color: #c0392b; color: white; padding: 2px 5px;">Eliminar</div> </div>
5	Mario Alberto	Cáceres	<div style="display: flex; justify-content: space-around; padding: 5px;"> <div style="border: 1px solid #ccc; padding: 2px 5px; background-color: #f3f3f3;">Obtener</div> <div style="background-color: #27ae60; color: white; padding: 2px 5px;">Modificar</div> <div style="background-color: #c0392b; color: white; padding: 2px 5px;">Eliminar</div> </div>

Si intentamos modificar:

Actualizar

Back

- 11
- Erika María
- Landaverde Castro
- 32
- M
- Er

Actualizar

Resultado personal:

localhost/DSS_Guia7_LM242664/ejemplo3/

Crear, Obtener, Actualizar y Borrar (CRUD) con PDO.

Create

#	Nombre	Apellido	Operaciones
17	Lilian Adrina	Cortéz Montalvo	Obtener Modificar Eliminar
16	Mónica Lucía	Arévalo Chinchilla	Obtener Modificar Eliminar
15	Alexander Ismael	Contreras	Obtener Modificar Eliminar
14	Julian Alberto	Alvarado Ruiz	Obtener Modificar Eliminar
13	Marcos Antonio	Villalta Cortez	Obtener Modificar Eliminar
12	Luis Alexander	Díaz Muñoz	Obtener Modificar Eliminar
11	Erika María	Landaverde Castro	Obtener Modificar Eliminar
9	Diana Lisette	Benítez Contreras	Obtener Modificar Eliminar
8	Julio Amílcar	Durán Umaña	Obtener Modificar Eliminar
7	Lucía Carolina	Duarte	Obtener Modificar Eliminar
5	Mario Alberto	Cáceres	Obtener Modificar Eliminar

Estudiante: José Adrián López Medina - LM242664

Técnico en Ingeniería en Computación - Escuela de Computacion Ciclo I - 2025

Mostrar información del usuario

Back

#	Nombre	Apellido	Edad	Género	Ciudad
17	Lillian Adrina	Cortéz Montalvo	28	F	San Vicen

Estudiante: José Adrián López Medina - LM242664

Técnico en Ingeniería en Computación - Escuela de Computacion Ciclo I - 2025

Actualizar

Back

-

14

-

Julian Alberto

-

Alvarado Ruíz

-

36

-

M

-

Mejicanos

Actualizar

Estudiante: José Adrián López Medina - LM242664

Técnico en Ingeniería en Computación - Escuela de Computacion Ciclo I - 2025

V. DISCUSION DE RESULTADOS

1. Modifique el script de eliminación de registros para que en lugar de funcionar como lo hace actualmente, funcione exactamente igual al de modificar registros, mostrando previamente un formulario que muestre los datos del registro a eliminar y luego, permita mediante dos botones de formulario, confirmar la eliminación del registro o cancelarla.

Resultado personal:



The screenshot shows a web browser window with the address bar displaying `localhost/DSS_Guia7_LM242664/ejemplo3/delete.php?id=7`. The page title is "Eliminar Usuario". Below the title, there is a "Volver" button. The form contains the following fields:

- ID: 7
- Nombre: Lucía Carolina
- Apellido: Duarte
- Edad: 28
- Género: F
- Ciudad: San Salvador

At the bottom of the form, there are two buttons: "Eliminar" (red) and "Cancelar" (blue).

3. Implemente en el ejemplo 3 la operación de eliminar que no está realizada en este ejemplo de uso de la extensión PDO. Solicite confirmación antes de proceder a eliminar el registro y muestre en esa página de confirmación los datos que se van a borrar.

Resultado personal:



← → ↻ 🏠 ⓘ localhost/DSS_Guia7_LM242664/ejemplo3/delete.php?id=7

Eliminar Usuario

Volver

ID: 7

Nombre: Lucía Carolina

Apellido: Duarte

Edad: 28

Género: F

Ciudad: San Salvador

Eliminar Cancelar

VI. BIBLIOGRAFIA

- Matt Doyle. Fundamentos PHP Práctico. 1ra. Edición. Editorial Anaya Multimedia. Madrid, España 2010.
- John Coggeshall. LA BIBLIA DE PHP 5. 1ra. Edición. Editorial Anaya Multimedia. Madrid, España 2005.
- Welling, Luke / Thomson, Laura. Desarrollo web con PHP y MySQL. Traducción de la 3ra Edición en inglés. Editorial Anaya Multimedia. 2005. Madrid, España.
- Gil Rubio, F. Javier / Villaverde, Santiago Alonso. Creación de sitios web con PHP 5. 1a edición. Editorial McGraw Hill. 2006. Madrid, España.

- Gutierrez, Abraham / Bravo, Ginés. PHP 5 a través de ejemplos. 1ra Edición. Editorial Alfaomega. Junio 2005. México.
- DuBois, Paul. Programación MySQL. Traducción de la 3ra Edición. Editorial Anaya Multimedia. Madrid, España 2005.
- <http://dev.mysql>.