

Módulo 7

Desarrollo web en entorno servidor

Actividad de Desarrollo (UF3)

Solución de Víctor Tena Palomares



```
function updatePhotoDescription() {  
    if (descriptions.length > (page * 9) + (currentimage.substring(0, 1))){  
        document.getElementById('bigimageDesc').innerHTML = descriptions[page * 9 + currentimage.substring(0, 1)];  
    }  
}  
  
function updateAllImages() {  
    var i = 1;  
    while (i < 10) {  
        var elementId = 'foto' + i;  
        var elementIdBig = 'bigimage' + i;  
        if (page * 9 + i - 1 < photos.length) {  
            document.getElementById( elementId ).src = 'images/' + photos[page * 9 + i - 1];  
            document.getElementById( elementIdBig ).src = 'images/' + photos[page * 9 + i - 1];  
        } else {  
            document.getElementById( elementId ).src = 'images/' + photos[photos.length - 1];  
            document.getElementById( elementIdBig ).src = 'images/' + photos[photos.length - 1];  
        }  
        i++;  
    }  
}
```

EJERCICIO 1

Crear una base de datos en MySQL (*localhost/phpmyadmin*), la cual llamaremos M07. Esta base de datos tiene que contener las siguientes tablas:

Tabla	Campos
Usuarios	Id Nombre Contraseña Email Edad Fecha nacimiento Dirección Código postal Provincia Genero
Noticias	Id Titulo Contenido Autor Hora creación Likes

NOTA: Estas tablas pueden ser creadas con el gestor de contenidos que ofrece MySql, tanto desde el editor SQL como con los formularios del gestor.

Los campos ID tienen que ser las "PRIMARY KEY" de las tablas, y los diferentes campos tienen que tener restricciones de inserción de datos (por ejemplo, el contenido de la noticia limitado a 300 caracteres)

Para crear las tablas se ha elegido el gestor de contenidos phpMyAdmin, a través de él creamos tanto la base de datos como las tablas con los nombres que se piden. A la hora de crear cada campo de las tablas tenemos en cuenta el tipo de dato que va a contener, asignando aquel que se adapte mejor al tamaño y a la lógica de la información que va a almacenar y a la vez restringiendo la inserción al tipo de dato esperado.

Tabla Usuarios:

The screenshot shows the phpMyAdmin interface for the 'usuarios' table. The table structure is as follows:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	Id	smallint(4)		UNSIGNED	No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
2	Nombre	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
3	Contraseña	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
4	Email	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
5	Edad	tinyint(3)		UNSIGNED	No	Ninguna			Cambiar Eliminar Más
6	Fecha_nacimiento	date			No	Ninguna			Cambiar Eliminar Más
7	Dirección	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
8	Código_postal	smallint(6)		UNSIGNED	No	Ninguna			Cambiar Eliminar Más
9	Provincia	varchar(20)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
10	Genero	varchar(10)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más

Below the table structure, there are sections for 'Índices' (Indices) and 'Particiones' (Partitions). The 'Índices' section shows a primary key index on the 'Id' column. The 'Particiones' section is currently empty.

Tabla Noticias:

The screenshot shows the phpMyAdmin interface for the 'noticias' table. The table structure is as follows:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	Id	smallint(3)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
2	Título	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
3	Contenido	varchar(300)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
4	Autor	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
5	Hora_creación	datetime			No	Ninguna			Cambiar Eliminar Más
6	Likes	int(6)			SI	NULL			Cambiar Eliminar Más

Below the table structure, there are sections for 'Índices' (Indices) and 'Particiones' (Partitions). The 'Índices' section shows a primary key index on the 'Id' column. The 'Particiones' section shows a message: '¡Ninguna partición definida!' (No partition defined!).

Ejercicio 2

Crear un archivo PHP (conexión.php) en el cual generaremos la conexión a la base de datos que hemos creado en el ejercicio anterior.

Nota: Este fichero contendrá solamente la conexión a la base de datos, la cual tiene que estar controlando los errores que pueda devolver (por si no tenemos la base datos conectada).

Para realizar la conexión con la base de datos se ha utilizado la extensión PDO, montamos una estructura try/catch. Dentro del try creamos un objeto PDO que almacena las credenciales necesarias para conectar con nuestra base de datos. A continuación, establecemos los atributos de error que le asignamos a esta conexión y la codificación de caracteres. Cerramos el bloque try y en el catch contemplamos lo que el programa debe hacer en caso de que fracase la conexión, en este caso, nos va a informar de lo que ocurre y en qué línea de código ocurre.

```
<?php
try {

    // Conexión mediante la librería PDO
    $connect = new PDO("mysql:dbname=m07;host=localhost", 'root', "");

    // Establecemos atributos para esta conexión
    $connect->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // Establecer encoding en PDO
    $connect->exec("SET CHARACTER SET UTF8");

} catch (Exception $exception) {

    // En caso de error en la conexión matamos el proceso e informamos del tipo de error y de su ubicación
    die("Se ha producido un error en la conexión de tipo: " . $exception->getMessage() .
        "en la línea: " . $exception->getLine());

}
```

Este archivo será incluido en los próximos que necesiten conectar con la base datos, si todo va bien nunca va a imprimir nada, pero si hay un error nos va a dar información muy útil para solucionarlo:

Se ha producido un error en la conexión de tipo: SQLSTATE[HY000] [1045] Access denied for user 'root'@'localhost' (using password: YES) en la línea: 5

Ejercicio 3

En el archivo cuerpo.php tenemos que mostrar dentro de contenedores HTML (tablas, capas, etc...) las noticias que tenemos almacenadas en la base de datos creada, estas noticias tienen que estar ordenadas por fecha (las más actuales al principio) y mostrar solamente los últimas 5 noticias.

A este archivo se le ha cambiado el nombre a index.php para que el sitio web empiece desde aquí.

Lo primero que se hace en este archivo, después de las primeras líneas HTML, es incluir el archivo cabecera.php, que a su vez y como veremos después, tiene la conexión a la base de datos porque va a estar presente en todos los archivos del sitio web. A continuación, realizamos la consulta a la base de datos teniendo en cuenta los criterios que se piden en el ejercicio. La preparamos y la ejecutamos obteniendo como resultado un array asociativo cuyos índices son los nombres de los campos de las tablas. Acto seguido, montamos un bucle foreach que va a imprimir cada noticia con su campo correspondiente.

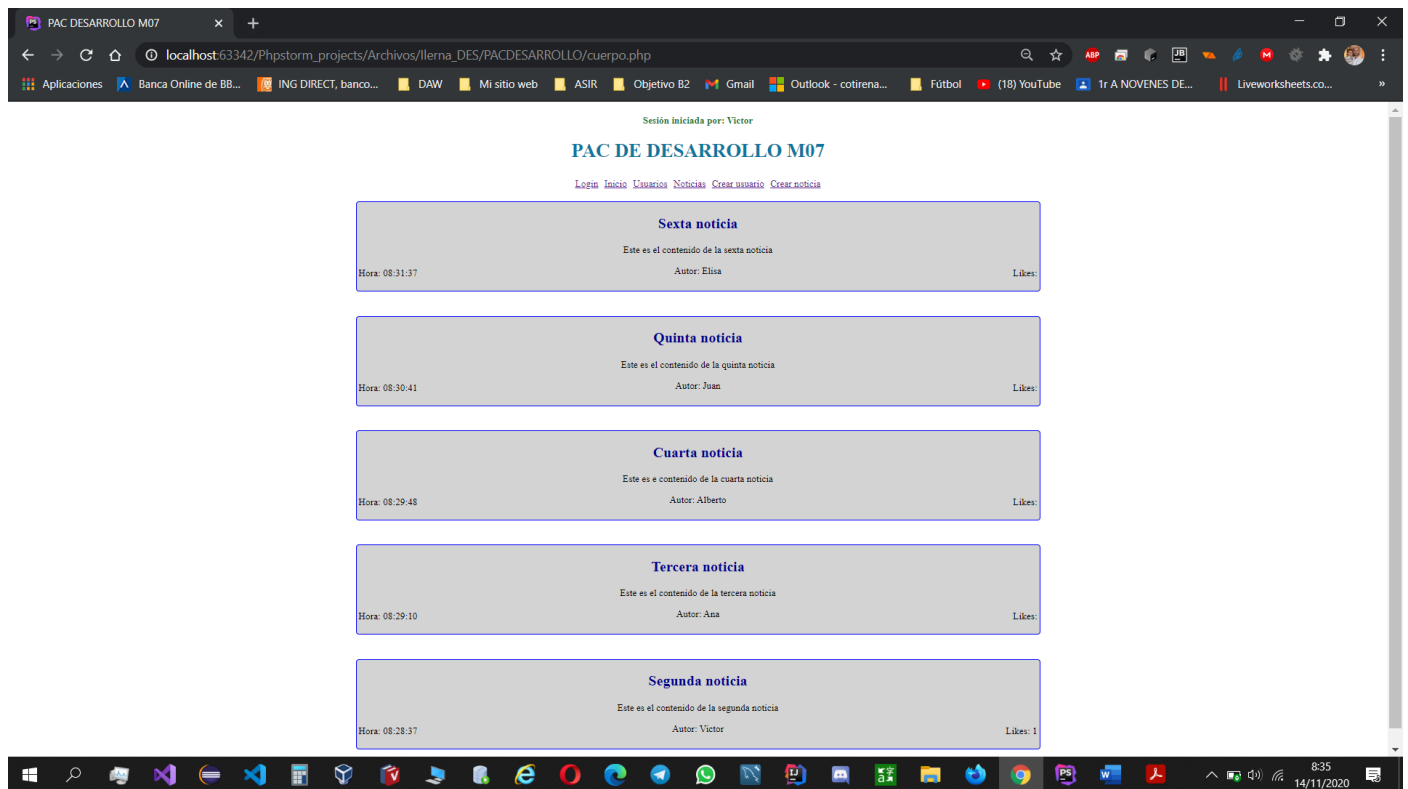
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>PAC DESARROLLO M07</title>
  <link rel="stylesheet" href="estilos.css">
</head>
<?php
  include_once('cabecera.php');

  $sql = "SELECT *FROM Noticias ORDER BY Hora_creación desc limit 5";
  $resultset = $connect->prepare($sql);
  $resultset->execute(array(PDO::FETCH_ASSOC));

  foreach ($resultset

    as $noticia) { ?>
<table id="tabla_inicio">
  <tr>
    <td id="Titulo"><h2><?php echo $noticia['Titulo']; ?></h2></td>
  </tr>
  <tr>
    <td id="Contenido"><?php echo $noticia['Contenido']; ?></td>
  </tr>
  <div id="enLinea">
    <tr id="Autor">
      <td>Autor: <?php echo $noticia['Autor']; ?></td>
    </tr>
    <tr id="texto-uno">
      <td>Hora: <?php echo $noticia['Hora_creación']; ?></td>
    </tr>
    <tr id="texto-dos">
      <td>Likes: <?php echo $noticia['Likes']; ?> <br><br></td>
    </tr>
  </div>
  <?php } ?>
</table>
```

A continuación, se muestra lo que conseguimos con este código, en el momento que se redacta esta memoria, la base de datos consta de 10 noticias en total. Como el ejercicio pide que se muestren las 5 últimas, se pueden visualizar desde la décima a la sexta mostrando primero las más recientes.



Ejercicio 4

Creamos una página que llamaremos `form_usuario.php`, la cual tenemos que adaptar, para que contenga los datos de la tabla usuarios. Este formulario tiene que atacar (crear modificar y eliminar registros) la tabla de usuarios de la base de datos.

También crearemos otra página llamada `form_noticias.php`, la cual tiene que realizar lo mismo que el formulario de usuario, pero atacando a la de noticias.

NOTA: Estos formularios tienen que aportar los datos necesarios para modificar los campos de las tablas de la base de datos, dichos campos deben de tener restricciones y validaciones en el formulario (con el type del input y atributos como `maxlength`).

Para codificar estos archivos se han seguido las instrucciones que plantea el ejercicio a la hora de crear tanto usuarios como noticias. Y, además, como la aplicación también va a permitir modificar, en cada input se ha implementado un condicional que detecta si procedemos del botón crear o del botón modificar. Con este detalle conseguimos aprovechar el mismo formulario para las dos tareas.

En el caso de que vengamos de crear, todos los campos del formulario estarán vacíos a la espera de ser rellenados. Pero si procedemos de modificar, dado que estos campos ya tienen datos asignados, se van a mostrar en el propio formulario para que el usuario tenga que modificar únicamente aquel dato que desee sin necesidad de volver a crear la información completamente desde cero.

Por otro lado, se han utilizado las restricciones necesarias para que a la hora de enviar los formularios no haya conflictos con la base de datos por medio de los atributos HTML `required`, `type`, `maxlength`, etc...

Form_usuario.php:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>PAC DESARROLLO M07</title>
  <link rel="stylesheet" href="estilos.css">
</head>
<?php
  include_once('cabecera.php');
?>

<form method="post" name="form_usuario" action="funciones_bd.php">
  <table>
    <?php
      if (isset($_POST['IdModificar'])) {
        echo '<tr><td><input type="hidden" required name="IdModificar" value="" . $_POST['IdModificar'] .
"></td></tr>';
      } else {
        echo '<tr><td><input type="hidden" required name="IdCrear"></td></tr>';
      }
    ?>
    <tr>
      <?php
        if (isset($_POST['NombreModificar'])) {
          echo '<td> Nombre: <input type="text" required name="NombreModificar" value="" .
$_POST['NombreModificar'] . "" maxlength="50"></td>';
        } else {
          echo '<td> Nombre: <input type="text" required name="Nombre" placeholder="Introduce aquí tu nombre y
apellidos" maxlength="50"></td>';
        }
      ?>
    </tr>
    <tr>
      <?php
        if (isset($_POST['ContraseñaModificar'])) {
          echo '<td> Contraseña: <input type="text" required name="ContraseñaModificar" value="" .
$_POST['ContraseñaModificar'] . "" maxlength="50"></td>';
        } else {
          echo '<td> Contraseña: <input type="password" required name="Contrasenia" placeholder="Aquí tu
contraseña" maxlength="20"></td>';
        }
      ?>
    </tr>
    <tr>
      <?php
        if (isset($_POST['EmailModificar'])) {
          echo '<td> Email: <input type="email" required name="EmailModificar" value="" . $_POST['EmailModificar'] . ""
maxlength="50"></td>';
        } else {
          echo '<td> Email: <input type="email" required name="Email" placeholder="Aquí tu email"
maxlength="50"></td>';
        }
      ?>
    </tr>
    <tr>
      <?php
        if (isset($_POST['EdadModificar'])) {
          echo '<td> Edad: <input type="number" required name="EdadModificar" value="" . $_POST['EdadModificar'] . ""
maxlength="50"></td>';
        } else {
          echo '<td> Edad: <input type="number" required name="Edad" placeholder="Aquí tu edad"
maxlength="2"></td>';
        }
      ?>
    </tr>
  </table>
</form>
```

```

        ?>
    </tr>
    <tr>
        <?php
            if (isset($_POST['FechaModificar'])) {
                echo '<td> Fecha nacimiento: <input type="date" required name="FechaModificar" value=" ' .
$_POST['FechaModificar'] . '" maxlength="50"></td>';
            } else {
                echo '<td>Fecha nacimiento: <input type="date" required name="Fecha" placeholder="Aquí tu fecha de
nacimiento"></td>';
            }
        ?>
    </tr>
    <?php
        if (isset($_POST['DirecciónModificar'])) {
            echo '<td>Dirección: <input type="text" required name="DirecciónModificar" value=" ' .
$_POST['DirecciónModificar'] . '" maxlength="50"></td>';
        } else {
            echo '<td>Dirección: <input type="text" required name="Direccion" placeholder="Aquí tu direccion"
maxlength="50"></td>';
        }
    ?>
</tr>
    <tr>
        <?php
            if (isset($_POST['CódigoModificar'])) {
                echo '<td>Código Postal: <input type="number" required name="CódigoModificar" value=" ' .
$_POST['CódigoModificar'] . '" maxlength="50"></td>';
            } else {
                echo '<td>Código Postal:<input type="number" required name="CP" placeholder="Aquí tu codigo postal"
maxlength="6"></td>';
            }
        ?>
    </tr>
    <tr>
        <?php
            if (isset($_POST['ProvinciaModificar'])) {
                echo '<td>Provincia: <input type="text" required name="ProvinciaModificar" value=" ' .
$_POST['ProvinciaModificar'] . '" maxlength="50"></td>';
            } else {
                echo '<td>Provincia: <input type="text" required name="Provincia" placeholder="Aquí tu provincia"
maxlength="20"></td>';
            }
        ?>
    </tr>
    <tr>
        <?php
            if (isset($_POST['GeneroModificar'])) {
                if ($_POST['GeneroModificar'] == "Masculino") {
                    echo '<td>Género Masculino: <input type="checkbox" name="Masculino" checked></td>';
                    echo '<td>Género Femenino: <input type="checkbox" name="Femenino"></td>';
                } else {
                    echo '<td>Género Masculino: <input type="checkbox" name="Masculino" ></td>';
                    echo '<td>Género Femenino: <input type="checkbox" name="Femenino" checked></td>';
                }
            } else {
                echo '<td>Género Masculino: <input type="checkbox" name="Masculino" ></td>';
                echo '<td>Género Femenino: <input type="checkbox" name="Femenino"></td>';
            }
        ?>
    </tr>
    <tr>
        <td><input type="submit" name="enviar" value="Enviar formulario"></td>
        <td><input type="reset" name="cancelar" value="Resetear campos"></td>
    </tr>

```



```
</table>
</form>
```

El archivo form_usuario.php de como resultado:



Form_noticias.php:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>PAC DESARROLLO M07</title>
  <link rel="stylesheet" href="estilos.css">
</head>
<?php
  include_once('cabecera.php');
?>

<form method="post" name="form_noticias" action="funciones_bd.php">
  <table>
    <?php
      if (isset($_POST['IdNoticiaModificar'])) {
        echo '<tr><td><input type="hidden" required name="IdNoticiaModificar" value="' . $_POST['IdNoticiaModificar'] .
"'></td></tr>';
      } else {
        echo '<tr><td><input type="hidden" required name="IdNoticias" ></td></tr>';
      }
    ?>
    <tr>
      <?php
        if (isset($_POST['TituloModificar'])) {
          echo '<td>Título: <input type="text" required name="TituloModificar" value="' . $_POST['TituloModificar'] .
          'maxlength="50"></td>';
        } else {
```

```

        echo '<td>Título: <input type="text" required name="Titulo" placeholder=" Introduce aquí el nombre de la
noticia" size="50" maxlength="50"></td>';
    }
    ?>
</tr>
<tr>
    <?php
        if (isset($_POST['ContenidoModificar'])) {
            echo '<td>Contenido: <textarea required name="ContenidoModificar" placeholder=" ' .
$_POST['ContenidoModificar'] . ' "cols="30" rows="10" maxlength="300" maxlength="50"></textarea> </td>';
        } else {
            echo '<td>Contenido: <textarea required name="Contenido" cols="30" rows="10"
maxlength="300"></textarea>';
        }
    ?>
</tr>
<tr>
    <?php
        if (isset($_POST['AutorModificar'])) {
            echo '<td>Autor: <input required name="AutorModificar" value=" ' . $_POST['AutorModificar'] . ' '
maxlength="50"></td>';
        } else {
            echo '<td>Autor: <input required type="Autor" name="Autor" placeholder="Aquí el nombre del autor"
maxlength="50"></td>';
        }
    ?>
</tr>

<td><input type="submit" name="enviar" value="Enviar formulario"></td>
<td><input type="reset" name="cancelar" value="Resetear campos"></td>
</tr>
</table>
</form>

```

El archivo form_noticias.php de como resultado:

Sesión iniciada por: Victor

PAC DE DESARROLLO M07

[Login](#) [Inicio](#) [Listado de usuarios](#) [Listado de noticias](#) [Cerrar sesión](#)

Título:

Introduce aquí el nombre de la noticia

Contenido:

Autor: Aquí el nombre del autor

Enviar formulario Resetear campos

Ejercicio 5

Creemos un archivo llamado funciones_bd.php, este archivo tiene que contener distintas funciones PHP, las cuales realicen las diferentes acciones que van a realizar los formularios de la página web, es decir, tienen que contener las funciones para crear, modificar y eliminar datos de las tablas de la base de datos.

NOTA: Este archivo tiene que estar compuesto por funciones PHP, que ataquen (INSERT, UPDATE, DELETE) las tablas de la base de datos creadas en los ejercicios anteriores.

En este archivo lo primero que se hace es incluir la conexión, que además pasamos por parámetro a cada una de las funciones PHP. En todos los archivos se utiliza include_once, de esta forma nos aseguramos de que solo se cargue una vez cada archivo durante la ejecución del programa.. A continuación, se han codificado una serie de condicionales que lo que hacen es comprobar si alguna variable global de tipo \$_POST se ha pasado a este archivo procedente de un formulario. Si es así, dependiendo del formulario de procedencia se va a llamar a una función o a otra para procesar esta información e insertarla, modificarla o borrarla de la base de datos según proceda.

Como norma, una vez realizada la tarea de cada función, esta nos va a redirigir al archivo de dónde veníamos, donde podremos comprobar que la acción ha sido completada con éxito.

Además de las funciones que manipulan los formularios, se han incluido otras funciones que se van a encargar de manipular los likes, y de cerrar la sesión.

```
<?php
include_once('conexión.php');

if (isset($_POST['IdCrear'])) {
    insertarDatosUsuario($connect);
}
if (isset($_POST['IdModificar'])) {
    modificarDatosUsuario($connect);
}
if (isset($_POST['IdBorrar'])) {
    borrarUsuario($connect);
}
if (isset($_POST['IdNoticias'])) {
    insertarNoticia($connect);
}
if (isset($_POST['IdNoticiaModificar'])) {
    modificarNoticia($connect);
}
if (isset($_POST['IdBorrarNoticia'])) {
    borrarNoticia($connect);
}
if (isset($_POST['IdNoticiaLike'])) {
    manipulaLikes($connect);
}
if(isset($_GET['cerrarSesion'])){
    cerrarSesion($connect);
}

function insertarDatosUsuario($connect)
{
    $nombre = $_POST['Nombre'];
    $contrasenia = $_POST['Contrasenia'];
    $email = $_POST['Email'];
    $edad = $_POST['Edad'];
```

```

$fecha = $_POST['Fecha'];
$direccion = $_POST['Direccion'];
$cp = $_POST['CP'];
$provincia = $_POST['Provincia'];
$genero = "";
if (isset($_POST['Masculino'])) {
    $genero = 'Masculino';
} else {
    $genero = 'Femenino';
}
$sql = "INSERT INTO Usuarios
(Nombre,Contraseña,Email,Edad,Fecha_nacimiento,Dirección,Código_postal,Provincia,Genero)
values ('{$nombre}','{$contrasenia}','{$email}','{$edad}','{$fecha}','{$direccion}','{$cp}','{$provincia}','{$genero}')";
$resulset = $connect->prepare($sql);
$resulset->execute(array($sql));
$connect = null;
header("Location: list_usuarios.php");
}

```

```

function modificarDatosUsuario($connect)
{
    $IdModificar = $_POST['IdModificar'];
    $nombre = $_POST['NombreModificar'];
    $contrasenia = $_POST['ContraseñaModificar'];
    $email = $_POST['EmailModificar'];
    $edad = $_POST['EdadModificar'];
    $fecha = $_POST['FechaModificar'];
    $direccion = $_POST['DirecciónModificar'];
    $cp = $_POST['CódigoModificar'];
    $provincia = $_POST['ProvinciaModificar'];
    $genero = "";
    if (isset($_POST['GeneroModificar'])) {
        $genero = 'Masculino';
    } else {
        $genero = 'Femenino';
    }
    $sql = "UPDATE Usuarios SET Nombre='{$nombre}',Contraseña='{$contrasenia}',

```

```

Email='{$email}',Edad='{$edad}',Fecha_nacimiento='{$fecha}',Dirección='{$direccion}',Código_postal='{$cp}',Provincia='{$provincia}',Genero='{$genero}'
where Id='{$IdModificar}'";
$resulset = $connect->prepare($sql);
$resulset->execute(array($sql));
$connect = null;
header("Location: list_usuarios.php");
}

```

```

function borrarUsuario($connect)
{
    $IdBorrar = $_POST['IdBorrar'];
    $sql = "DELETE FROM Usuarios where Id='{$IdBorrar}'";
    $resulset = $connect->prepare($sql);
    $resulset->execute(array($sql));
    $connect = null;
    header("Location: list_usuarios.php");
}

```

```

function insertarNoticia($connect)
{
    date_default_timezone_set("Europe/Madrid");
    $titulo = $_POST['Titulo'];
    $contenido = $_POST['Contenido'];

```

```
$autor = $_POST['Autor'];
$hora = date("H:i:s", time());
```

```
$sql = "INSERT INTO Noticias (Titulo,Contenido,Autor,Hora_creación)
VALUES ('{$titulo}','{$contenido}','{$autor}','{$hora}')";
$resulset = $conect->prepare($sql);
$resulset->execute(array($sql));
$conect = null;
header("Location: list_noticias.php");
}
```

```
function modificarNoticia($conect)
{
    date_default_timezone_set("Europe/Madrid");
    $titulo = $_POST['TituloModificar'];
    $contenido = $_POST['ContenidoModificar'];
    $autor = $_POST['AutorModificar'];
    $hora = date("H:i:s", time());

    $sql = "UPDATE Noticias SET Titulo='{$titulo}',Contenido='{$contenido}',Autor='{$autor}',
    Hora_creación='{$hora}' where Id='{$_POST['IdNoticiaModificar']}'";
    $resulset = $conect->prepare($sql);
    $resulset->execute(array($sql));
    $conect = null;
    header("Location: list_noticias.php");
}
```

```
function manipulaLikes($conect)
{
    if (isset($_POST['LeDieronLike'])) {
        if (!isset($_COOKIE["liked" . $_POST['IdNoticiaLike']])) {
            $liked = $_POST['LeDieronLike'];
            $liked++;
            $sql = "UPDATE noticias set Likes=$liked where Id='{$_POST['IdNoticiaLike']}'";
            $resulset = $conect->prepare($sql);
            $resulset->execute(array($sql));
            $conect = null;
            setcookie("liked" . $_POST['IdNoticiaLike'], $liked, time() + 1000000);
            header("Location: list_noticias.php");
        }
    }
}
```

```
function cerrarSesion($conect){
    session_start();
    unset($_SESSION["user1"]);
    unset($_COOKIE[$_COOKIE["liked" . $_POST['IdNoticiaLike']]]);
    session_destroy();
    header("Location: login.php");
    exit;
}
```

```
function borrarNoticia($conect)
{
    $sql = "DELETE FROM Noticias where Id='{$_POST['IdBorrarNoticia']}'";
    $resulset = $conect->prepare($sql);
    $resulset->execute(array($sql));
    $conect = null;
    header("Location: list_noticias.php"); }
}
```

Ejercicio 6

Vamos a generar otra opción en el menú de la web que se llame listado de usuarios, la cual llame a una nueva página que llamaremos lis_usuarios.php, la cual se tiene que incluir en index.php y tiene que mostrar el listado de todos los usuarios que tenemos creados en la base de datos.

Esta página tiene que contener diferentes botones para acceder al formulario de usuarios, para modificar o eliminar el usuario elegido o para crear un nuevo usuario.

En el archivo que hace posible mostrar toda la información de los usuarios, lo primero que hacemos es situar un botón que nos permitirá crear un nuevo usuario, al presionar este botón se nos va a dirigir al archivo de form_usuario.php donde podremos realizar esta operación. Acto seguido, realizamos la consulta sql a la base de datos para que nos muestre todos los usuarios que están registrados en la base de datos, nuevamente utilizamos un foreach para conseguirlo. A cada vuelta de bucle nos imprimirá una tabla con un usuario, dentro de esta tabla habrá dos botones que nos van a permitir modificar o borrar este usuario en concreto, para ello hacemos valer el id que tiene asignado cada usuario en la base de datos.

En el caso de modificar, nos va a llevar al mismo formulario de crear, pero va a pasar toda la información ya insertada consiguiendo así que la modificación sea más cómoda para el usuario.

Al rellenar el formulario con éxito, el método encargado de insertar el nuevo usuario, acaba enviándonos a la página de Loguin para que el nuevo usuario pueda loguearse y utilizar el sitio web sin restricciones.

Para borrar el usuario, simplemente pasamos el id a la función encargada de eliminar usuarios y esta se encarga de realizarlo.

list_usuarios.php:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>PAC DESARROLLO M07</title>
  <link rel="stylesheet" href="estilos.css">
</head>
<?php
  include_once('cabecera.php');

  if (!isset($_SESSION ['user1'])) {
    header('Location: index.php');
  }

  echo "<table id='botonCrear'><tr><td><form action='form_usuario.php' method='post'>
  <input type='submit' value='Crear nuevo usuario'></td></tr></table><br><br><br></form>";

  $sql = 'SELECT * FROM Usuarios';
  $resultset = $conect->prepare($sql);
  $resultset->execute(array(PDO::FETCH_ASSOC));
  foreach ($resultset as $usuario) {
    echo "<table id='tabla_usuarios'>";
    echo "<tr><td>Nombre: {$usuario['Nombre']}</tr></td><br>";
```

```

echo "<tr><td>Contraseña: {$usuario['Contraseña']}</tr></td><br>";
echo "<tr><td>Email: {$usuario['Email']}</tr></td><br>";
echo "<tr><td>Edad: {$usuario['Edad']}</tr></td><br>";
echo "<tr><td>Fecha de nacimiento: {$usuario['Fecha_nacimiento']}</tr></td><br>";
echo "<tr><td>Dirección: {$usuario['Dirección']}</tr></td><br>";
echo "<tr><td>Código postal: {$usuario['Código_postal']}</tr></td><br>";
echo "<tr><td>Provincia: {$usuario['Provincia']}</tr></td><br>";
echo "<tr><td>Genero: {$usuario['Genero']}</tr></td><br>";

```

```

echo "<form action='form_usuario.php' method='post'>
<input type='hidden' name='IdModificar' value='{$usuario["Id"]} '>
<input type='hidden' name='NombreModificar' value='{$usuario["Nombre"]} '>
<input type='hidden' name='ContraseñaModificar' value='{$usuario["Contraseña"]} '>
<input type='hidden' name='EmailModificar' value='{$usuario["Email"]} '>
<input type='hidden' name='EdadModificar' value='{$usuario["Edad"]} '>
<input type='hidden' name='FechaModificar' value='{$usuario["Fecha_nacimiento"]} '>
<input type='hidden' name='DirecciónModificar' value='{$usuario["Dirección"]} '>
<input type='hidden' name='CódigoModificar' value='{$usuario["Código_postal"]} '>
<input type='hidden' name='ProvinciaModificar' value='{$usuario["Provincia"]} '>
<input type='hidden' name='GeneroModificar' value='{$usuario["Genero"]} '>
<tr><td><input type='submit' value='Modificar usuario'></tr></td></form> ";

```

```

echo "<form action='funciones_bd.php' method='post'>
<input type='hidden' name='IdBorrar' value='{$usuario["Id"]} '>
<tr><td><input type='submit' value='Borrar usuario'></tr></td></form><br><br><br>";

```

```

echo "</table>";

```

```

}
$conect = null;

```

El archivo list_usuarios.php da como resultado:

Sesión iniciada por: Victor

PAC DE DESARROLLO M07

[Login](#) [Inicio](#) [Listado de usuarios](#) [Listado de noticias](#) [Cerrar sesión](#)

[Crear nuevo usuario](#)

<p>Nombre: Victor Contraseña: 1234 Email: VictorTTT@gmail.com Edad: 37 Fecha de nacimiento: 1983-10-06 Dirección: calle Xilxes 13 4 G Código postal: 12530 Provincia: Burriana Genero: Masculino</p> <p>Modificar usuario Borrar usuario</p>
<p>Nombre: Enzo Contraseña: 1234 Email: Enzo@gmail.com Edad: 6 Fecha de nacimiento: 2013-11-29 Dirección: C XILXES Código postal: 12530 Provincia: Burriana Genero: Masculino</p> <p>Modificar usuario Borrar usuario</p>
<p>Nombre: CORINNE Contraseña: 1234 Email: cotrena@gmail.com Edad: 35 Fecha de nacimiento: 1985-02-06 Dirección: calle Xilxes 13 4 G Código postal: 12530 Provincia: Burriana Genero: Femenino</p> <p>Modificar usuario Borrar usuario</p>

Ejercicio 7

Tenemos que crear un archivo llamado `list_noticias.php` donde se tiene que mostrar todas las noticias almacenadas en la tabla de noticias de la base de datos.

Estas noticias tienen que tener un botón para indicar que te gustan, estos datos los tenemos que almacenar en las “cookies”, del navegador y actualizar el campo de me gusta de la tabla en la base de datos.

NOTA (OPCIONAL): Si estamos como un usuario registrado podemos almacenar en las cookies el usuario también, en caso de no estar registrado podemos almacenar los datos con usuario anónimo.

Para programar la funcionalidad de este ejercicio, al igual que hicimos en el de los usuarios, lo primero que hacemos es poner el botón que crea una nueva noticia, que nos lleva al formulario correspondiente.

Luego hacemos la consulta a la base de datos y la ejecutamos tomando como índices los nombres de los campos de la tabla noticias. Acto seguido, dentro del `foreach`, creamos una tabla HTML que contendrá cada noticia con sus correspondientes botones de: me gusta, modificar y borrar noticia.

Para que el funcionamiento del botón “me gusta” tenga un comportamiento lógico, se ha creado una función en el archivo `funciones_bd.php` llamada `manipulaLikes`. Esta función se encarga de actualizar los likes de la noticia en cuestión, también crea una cookie que almacena la sesión de quien ha dado este like. A continuación, devuelve la ejecución el programa al archivo `list_noticias.php` que comprueba si está preparada esta cookie y de ser así cambia la posibilidad de dar otro like por la impresión en negrita de un mensaje con el nombre del usuario informando de que le gusta esta noticia.

De esta forma cada usuario podrá dar un único like cada noticia y además podrá comprobar quién ha dado like a esta u otras noticias.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>PAC DESARROLLO M07</title>
  <link rel="stylesheet" href="estilos.css">
</head>
<?php

  include_once('cabecera.php');

  if (!isset($_SESSION['user1'])) {
    header('Location: index.php');
  }
```



```
// Boton de crear noticia fuera del bucle para que solo esté una vez arriba
echo "<table id='botonCrearNoticia'><tr><td><form action='form_noticias.php' method='post'>
<input type='submit' value='Crear una nueva noticia'></td></tr></table><br><br><br></form>";
```

```
$sql = 'SELECT *FROM Noticias ORDER BY Hora_creación desc';
$resulset = $conect->prepare($sql);
$resulset->execute(array(PDO::FETCH_ASSOC));
```

```
// Bucle con tabla, dentro de ella se encontrará cada noticia cons sus botones
foreach ($resulset as $noticia) {
```

```
//Si no se ha creado la cookie que corresponde al id de esta noticia..
```

```
if (!isset($_COOKIE['liked' . $noticia['Id']])) {
    echo "<table id='tabla_noticias'> ";
    echo "<tr><td id='Titulo'><h2> {$noticia['Titulo']}</h2></td></tr></td><br>";
    echo "<tr><td>{$noticia['Contenido']}</tr></td><br>";
    echo "<tr><td>Autor: {$noticia['Autor']}</tr></td><br>";
    echo "<tr><td>Creada: {$noticia['Hora_creación']}</tr></td><br>";
    echo "<tr><td>Likes: {$noticia['Likes']}</tr></td><br>";
    echo "<form action='funciones_bd.php' method='post'>
<input type='hidden' name='IdNoticiaLike' value='{$noticia['Id']}'>
<input type='hidden' name='LeDieronLike' value='{$noticia['Likes']}'>
<tr><td><input type='submit' value='Me gusta!!!'></tr></td>
</form>";
```

```
//Si se ha creado, esconde la posibilidad de dar más likes e informa con el nombre de la sesión
```

```
} else {
    echo "<table id='tabla_noticias'> ";
    echo "<tr><td id='Titulo'><h2> {$noticia['Titulo']}</h2></tr></td><br>";
    echo "<tr><td>{$noticia['Contenido']}</tr></td><br>";
    echo "<tr><td>Autor: {$noticia['Autor']}</tr></td><br>";
    echo "<tr><td>Creada: {$noticia['Hora_creación']}</tr></td><br>";
    echo "<tr><td><b></b>Likes: {$noticia['Likes']}</b></tr></td><br>";
    echo "<tr><td><b>A {$_COOKIE['liked' . $noticia['Id']] le gusta!</b></tr></td><br>";
}
```

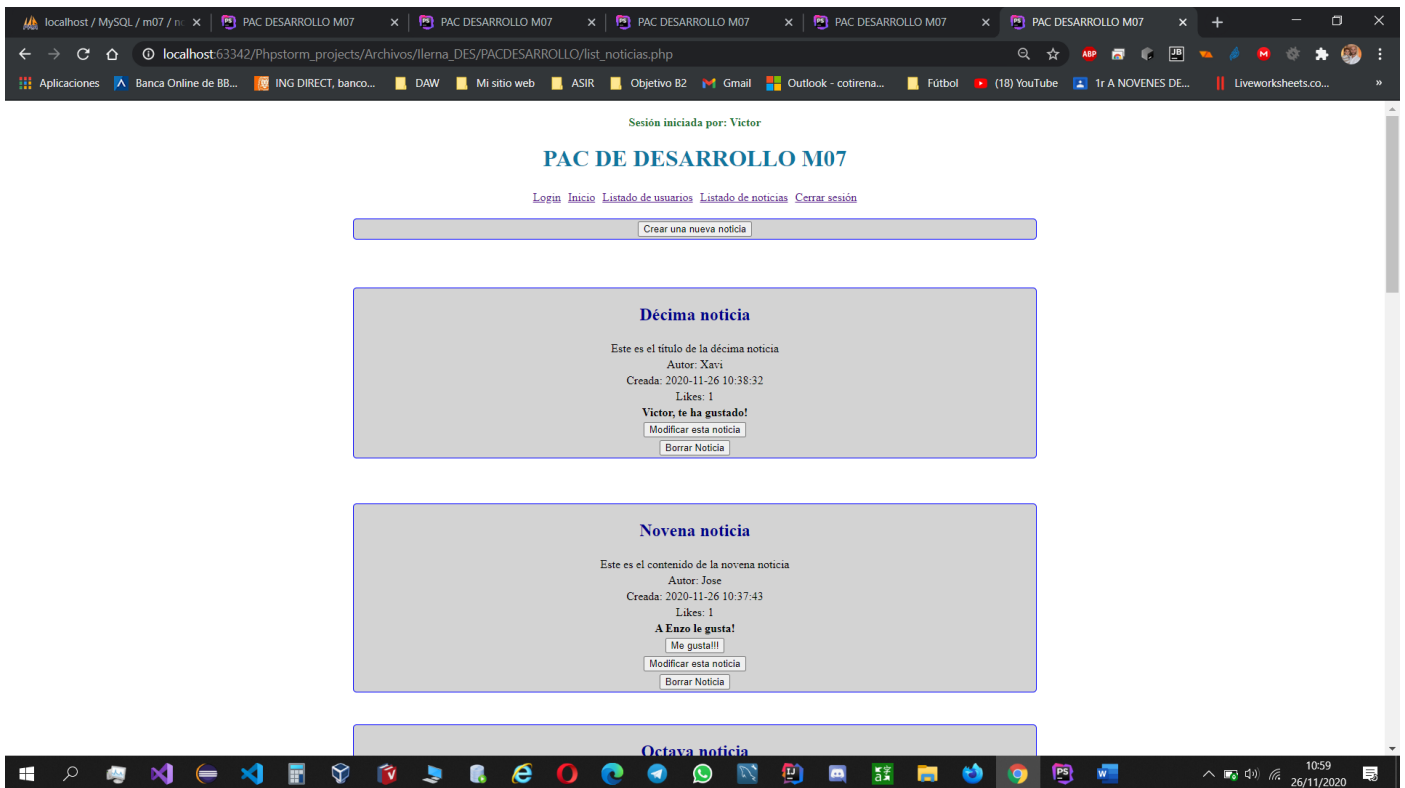
```
// Datos que se pasan al formulario de crear/modificar para no tener que crear los datos desde cero
```

```
echo "<form action='form_noticias.php' method='post'>
<input type='hidden' name='IdNoticiaModificar' value='{$noticia['Id']}'>
<input type='hidden' name='TituloModificar' value='{$noticia['Titulo']}'>
<input type='hidden' name='ContenidoModificar' value='{$noticia['Contenido']}'>
<input type='hidden' name='AutorModificar' value='{$noticia['Autor']}'>
<input type='hidden' name='HoraModificar' value='{$noticia['Hora_creación']}'>
<input type='hidden' name='LikesModificar' value='{$noticia['Likes']}'>
<tr><td><input type='submit' value='Modificar esta noticia'></tr></td>
</form> ";
```

```
// Para borrar la noticia
```

```
echo "<form action='funciones_bd.php' method='post'>
<input type='hidden' name='IdBorrarNoticia' value='{$noticia['Id']}'>
<tr><td><input type='submit' value='Borrar Noticia'></tr></td></form><br><br><br>";
```

```
$conect = null;
}
```



Ejercicio 8

El archivo cabecera.php tiene que contener el menú de navegación para esta página web, donde tendremos las siguientes opciones: Inicio, Usuarios, Noticias, Crear usuario, Crear noticia.

En esta actividad se sigue lo que se pide en el enunciado, pero también, aprovechando que este archivo va a estar presente en todas las páginas del sitio web, le incluimos la conexión a la base de datos con `include_once` por si hay redundancia y también la comprobación de si se ha iniciado sesión. Esto mejora la seguridad de quien está utilizando esta aplicación, ya que en caso de que otra persona copiara la url y la pegara en otro navegador, la aplicación empezaría directamente desde la página index sin las funcionalidades extendidas, obligando a registrarse.

También se ha puesto el nombre del usuario de color verde en la parte superior.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>PAC DESARROLLO M07</title>
  <link rel="stylesheet" href="estilos.css">
</head>
<?php
```

```
include_once('conexión.php');
session_start();
if (!isset($_SESSION['user1'])) {
  header('Location: login.php');
} else {
  echo "<h4 id='exito'>Sesión iniciada por: " . $_SESSION['user1'];
}
?>
<h1>PAC DE DESARROLLO M07</h1>
```

```

<nav>
  <div class="container">
    <ul id="menu">
      <li><a href="login.php">Login</a></li>
      <li><a href="cuerpo.php">Inicio</a></li>
      <li><a href="list_usuarios.php">Listado de usuarios</a></li>
      <li><a href="list_noticias.php">Listado de noticias</a></li>
      <li><a href="form_usuario.php">Crear usuario</a></li>
      <li><a href="form_noticias.php">Crear noticia</a></li>
      <li><a href="funciones_bd.php?cerrarSesion">Cerrar sesión</a></li>
    </ul>
  </div>
</nav>

```

Cabecera con sesión iniciada:

Sesión iniciada por: Victor

PAC DE DESARROLLO M07

[Login](#)
[Inicio](#)
[Listado de usuarios](#)
[Listado de noticias](#)
[Cerrar sesión](#)

Décima noticia

Este es el título de la décima noticia

Hora: 2020-11-26 10:38:32
 Autor: Xavi
 Likes: 1

Cabecera sin sesión iniciada:

Regístrate para tener acceso a todas las funciones!

[Login](#)

Décima noticia

Este es el título de la décima noticia

Hora: 2020-11-26 10:38:32
 Autor: Xavi
 Likes: 1

Ejercicio 9

Aprovechando la tabla de usuarios que hemos creado, vamos a crear una página para acceder a la página web, como un usuario registrado, este acceso lo podemos realizar con contenedores HTML o con una página nueva (por ejemplo Login.php).

Para controlar las sesiones de estos usuarios (\$_SESSION['user1']), vamos a mostrar las opciones de menú a los formularios solo cuando estemos registrados con un usuario.

NOTA: Si la opción de registro la realizamos en una página aparte, se puede añadir una opción de Login en el menú (opcional).

Para realizar este ejercicio se ha decidido crear una página a parte (Login.php). En este archivo jugamos con las variables \$_SERVER['PHP_SELF'] para comprobar dentro del mismo archivo los datos que se introducen tanto para el Login como para el registro.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet" href="estilos.css">
</head>
<body>

<!--Formulario de login-->
<h1>INTRODUCE TUS DATOS</h1>
<form action="<?php $_SERVER['PHP_SELF'] ?>" method="post">
    <table>
        <tr>
            <td class="izq">Login:</td>
            <td class="der"><input type="text" name="login"></td>
        </tr>
        <tr>
            <td class="izq">Password:</td>
            <td class="der"><input type="password" name="password"></td>
        </tr>
        <tr>
            <div>
                <td colspan="3"><input type="submit" name="enviarLogin" value="LOGIN"></td>
            </div>
        </tr>
    </table>
</form>

<!--Botón de registro, al presionarlo recarga la página y aparece el formulario de registro-->
<h1><br>QUIERO REGISTRARME</h1>
<form action="<?php $_SERVER['PHP_SELF'] ?>" method="get">
    <table>
        <tr>
            <div>
                <td colspan="3"><input type="submit" name="enviarRegistro" value="REGISTRAR"></td>
            </div>
        </tr>
    </table>
</form>
</body>
</html>

```

```

<?php

```

```

//Si se ha enviado el formulario de login
if (isset($_POST['enviarLogin'])) {

```

```

    include_once('conexión.php');

```

```

//Sentencia necesaria para comprobar login

```

```

$sql = "SELECT * FROM usuarios WHERE Nombre= :login AND Contraseña= :password";

```

```

//Utilizamos métodos PHP para comprobar los datos con seguridad y evitar inyecciones SQL

```

```

$login = htmlentities(addslashes($_POST['login']));
$password = htmlentities(addslashes($_POST['password']));
$resultado = $conect->prepare($sql);
$resultado->bindValue(":login", $login);
$resultado->bindValue(":password", $password);
$resultado->execute();

```

```

//Contamos los registros afectados, en caso de haberlos, iniciamos sesión asignándole el nombre del Login

```

```

$filas_afectadas = $resultado->rowCount();

```

```

if ($filas_afectadas != 0) {
    session_start();
    $_SESSION["user1"] = $_POST['login'];
    header('Location: cuerpo.php');

    // En caso de no haberlos, Informamos con mensaje
} else {
    echo "<h4 id='error'>No ha sido posible loguearse, por favor, revise su datos y vuelva a intentarlo.";
}
}

```

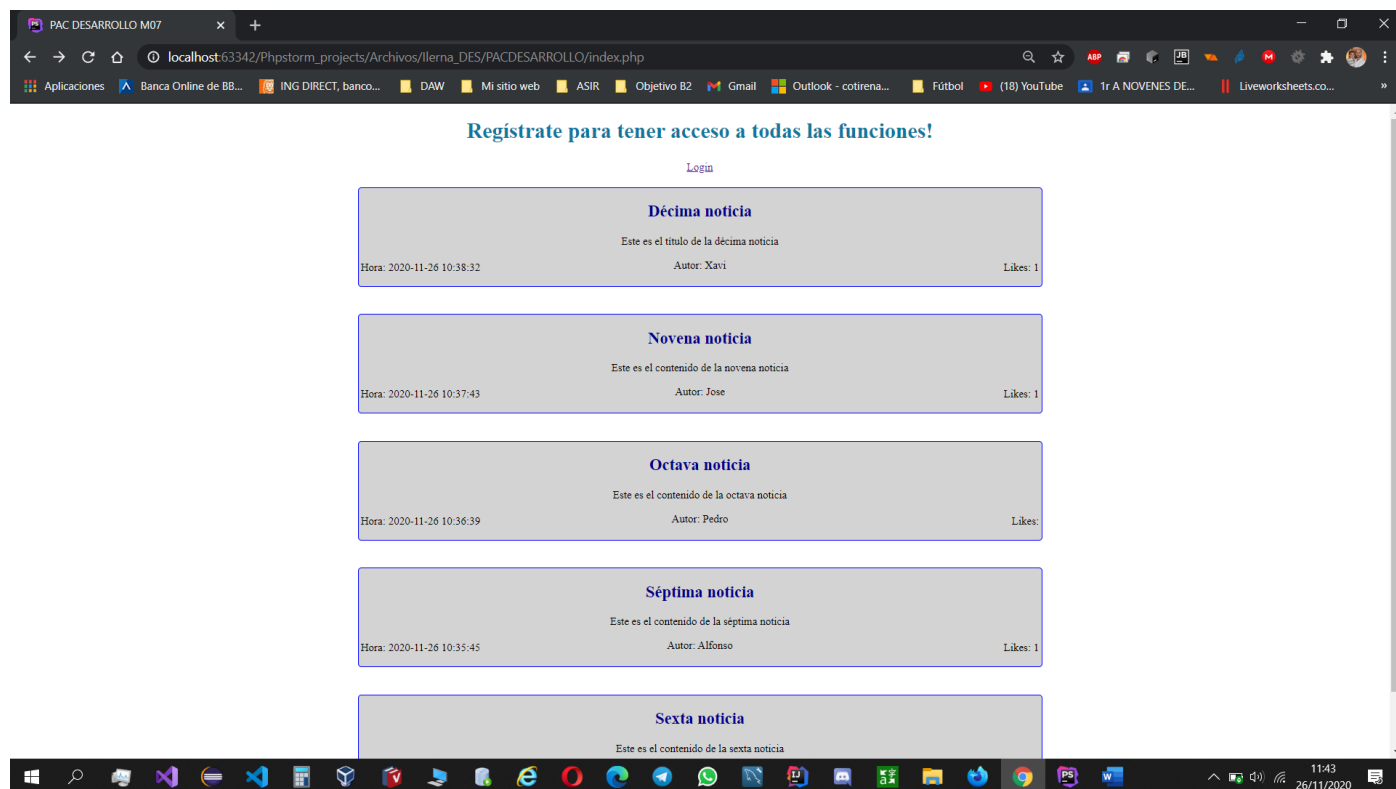
// Si se ha clickado en registro aparecerá este formulario que una vez relleno llamará a la función del "funciones_bd.php"

```

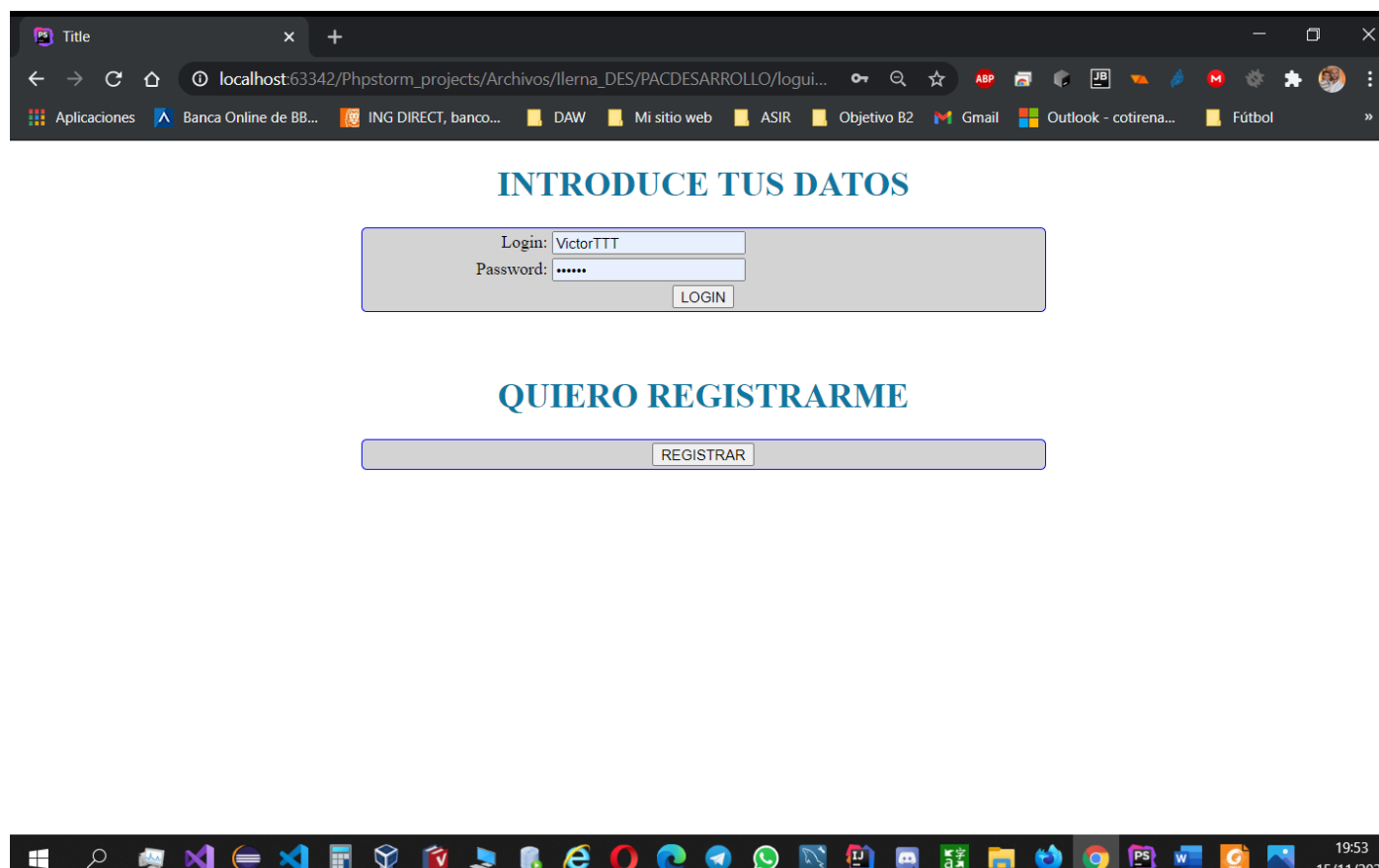
if (isset($_GET['enviarRegistro'])) { ?><br>
<form method="post" name="form_usuario" action="funciones_bd.php">
    <table>
        <tr>
            <td><input type="hidden" required name="IdCrear"></td>
        </tr>
        <tr>
            <td>Nombre: <input type="text" required name="Nombre"
                placeholder="Introduce aquí tu nombre y apellidos"
                maxlength="50"></td>
        </tr>
        <tr>
            <td>Contraseña: <input type="password" required name="Contrasenia" placeholder="Aquí tu contraseña"
                maxlength="20"></td>
        </tr>
        <tr>
            <td>Email: <input type="email" required name="Email" placeholder="Aquí tu email" maxlength="50">
            </td>
        </tr>
        <tr>
            <td>Edad: <input type="number" required name="Edad" placeholder="Aquí tu edad" maxlength="2"></td>
        </tr>
        <tr>
            <td>Fecha nacimiento: <input type="date" required name="Fecha"
                placeholder="Aquí tu fecha de nacimiento"></td>
        </tr>
        <tr>
            <td>Dirección: <input type="text" required name="Direccion" placeholder="Aquí tu direccion"
                maxlength="50"></td>
        </tr>
        <tr>
            <td>Código Postal:<input type="number" required name="CP" placeholder="Aquí tu codigo postal"
                maxlength="6"></td>
        </tr>
        <tr>
            <td>Provincia: <input type="text" required name="Provincia" placeholder="Aquí tu provincia"
                maxlength="20"></td>
        </tr>
        <tr>
            <td>Género Masculino: <input type="checkbox" name="Masculino"></td>
        </tr>
        <tr>
            <td>Género Femenino: <input type="checkbox" name="Femenino"></td>
        </tr>
        <tr>
            <td><input type="submit" name="enviar" value="Enviar formulario"></td>
            <td><input type="reset" name="cancelar" value="Resetear campos"></td>
        </tr>
    </table>
</form>
<?php
} ?>

```

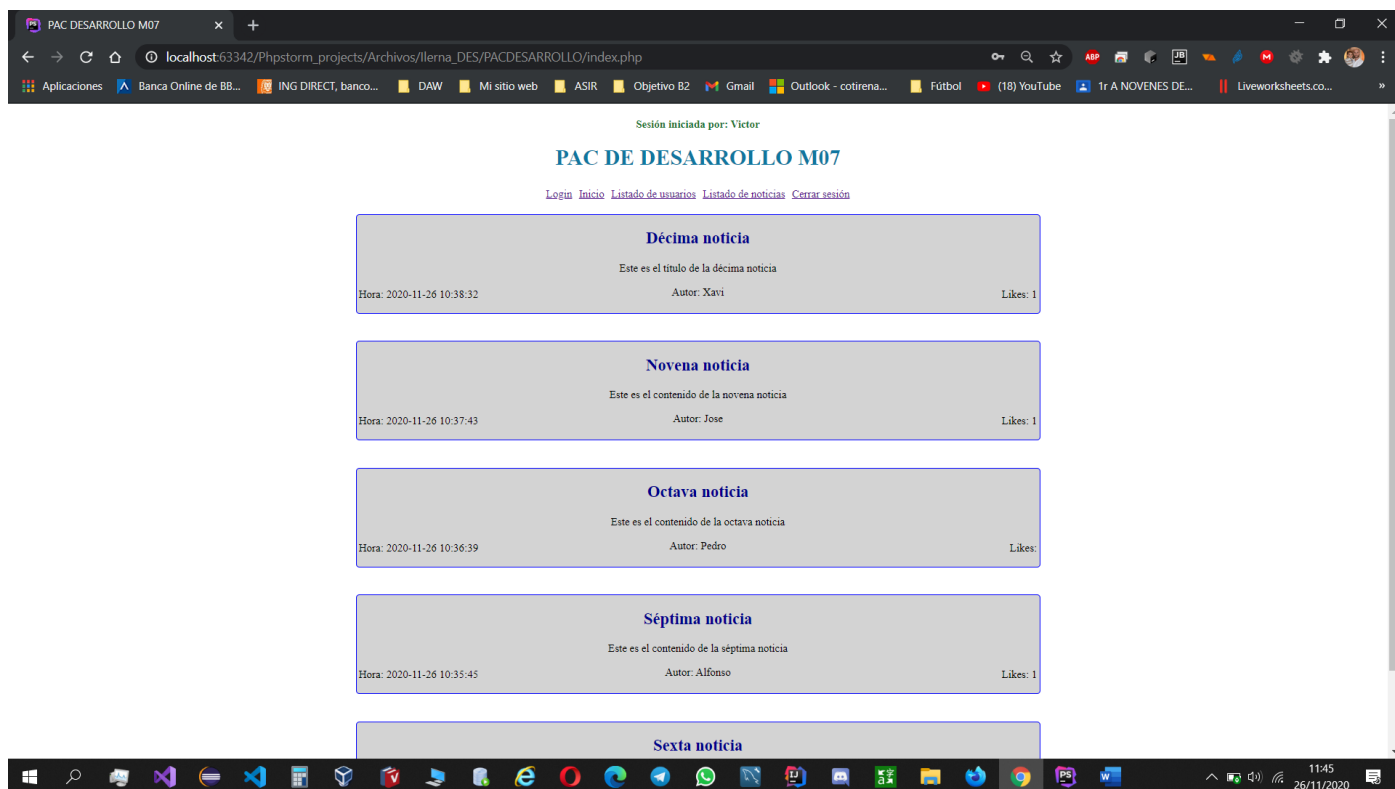
A raíz de lo que se pide en el ejercicio, se sobreentiende que este sitio web va a empezar desde el archivo index.php. Si no estamos registrados, o se ha cerrado la sesión, podremos ver las noticias que hay en el sitio web sin poder interactuar con ellas (likes, modificar o borrar). La página nos informa de la posibilidad de loguearnos para más funcionalidad y las opciones del menú de navegación, se reducen únicamente a el login.



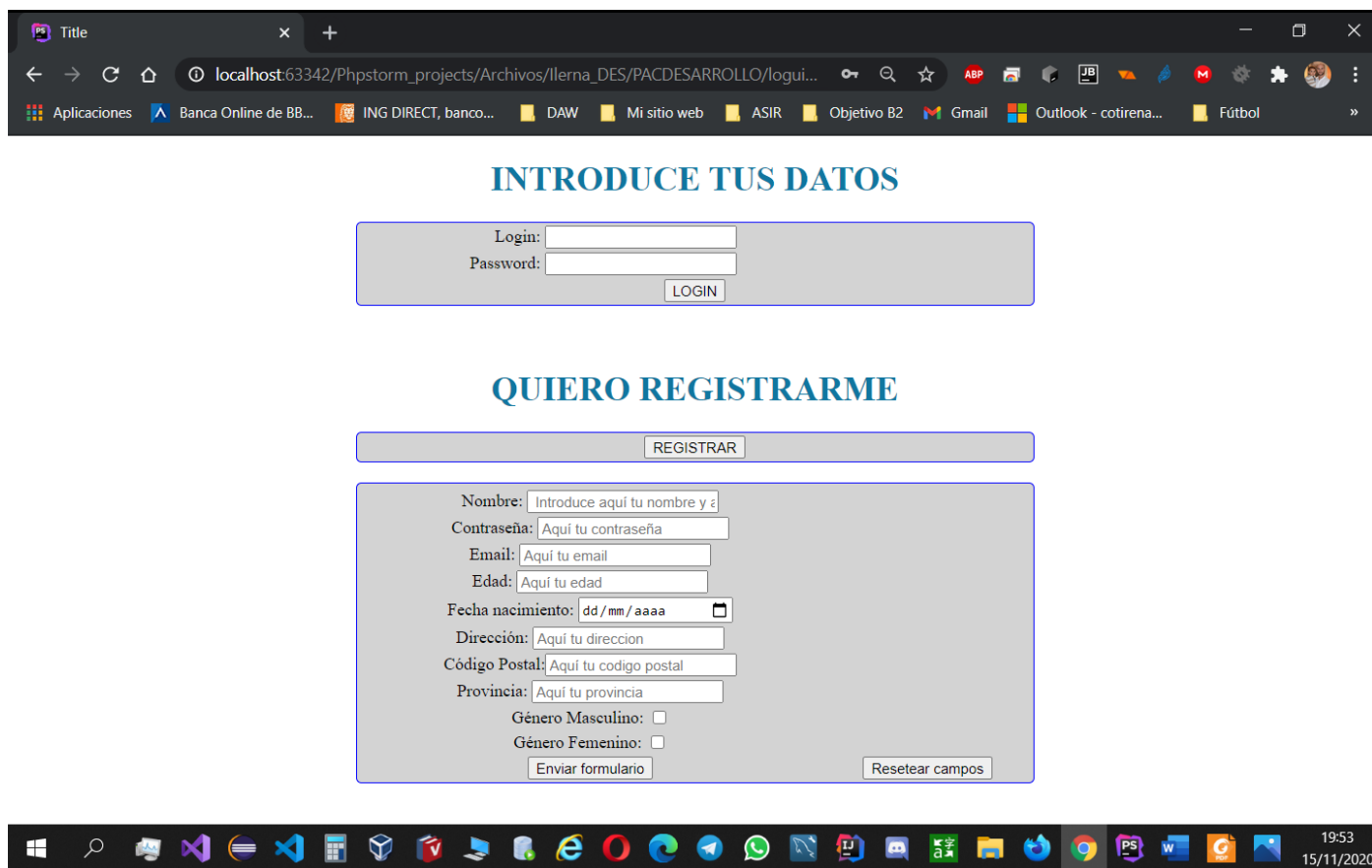
Cuando decidimos ir al login, pueden ocurrir 2 cosas, que estemos registrados y solo necesitemos loguearnos o que no lo estemos y tengamos que hacerlo.



Si se completa el Login con las credenciales correctas, el sitio nos envía de vuelta a la página principal, pero en esta ocasión ya tenemos disponible toda la funcionalidad, además de nuestra sesión iniciada.



En caso de que debamos registrarnos, al pulsar el botón aparece el formulario de registro que nos permite hacerlo. Una vez completado el registro, la página se recarga para que podamos loguearnos con nuestros datos de usuario y poder así disfrutar de la funcionalidad del sitio web.



Bibliografía/Webgrafía:

Para hacer los ejercicios de esta PAC se ha utilizado el material didáctico de la asignatura, las ideas facilitadas en las videotutorías y los recursos colgados en el apartado de contenidos. También se ha extraído información de la documentación oficial de PHP en el sitio web:

<https://www.php.net/manual/es/index.php>.

También se han aplicado los conocimientos adquiridos en el curso de programación en PHP del canal Píldoras informáticas de you tube:

[Curso PHP MySQL. Presentación. Vídeo 1 - YouTube](#)