

EJERCICIO PRÁCTICO PASO A PASO

CRUD DE PELÍCULAS CON AUTENTICACIÓN

Este ejercicio consiste en una aplicación en PHP que permite a los usuarios autenticarse, añadir películas, visualizarlas, eliminarlas y darles “like”. El código está dividido en dos partes principales: *la lógica de PHP* y *la presentación con HTML*.

1. Inicio de sesión (debe ser la primera función en el script)

```
<?php
session_start();
?>
```

¿Qué es una sesión?

Las sesiones permiten que el servidor recuerde información sobre un usuario durante varias interacciones. En este caso, se almacena si un usuario ha iniciado sesión o no.

¿Que hace este fragmento de código?

`session_start()` --> activa la gestión de sesiones en PHP. Permite el uso de `$_SESSION` para almacenar y recuperar datos asociados a la sesión actual del usuario (como si inició sesión o no).

Las sesiones permiten que los datos persistan mientras el usuario navega por diferentes páginas o realiza acciones.

2. Conexión a la base de datos

```
$db = new mysqli("localhost", "root", "", "bdPeliculas");
```

Esta línea siempre debe ir cuando se quiere conectar a la base de datos. Donde `localhost` = servidor, `root` = contraseña, `bdPeliculas` = base de datos a la que se conecta.

`new mysqli()` es una forma de crear un nuevo objeto de la clase `mysqli`, permite conectarse a bases de datos y realizar operaciones en ellas.

`localhost` es el servidor donde se conecta la base de datos. En este caso se encuentra en el mismo servidor, si estuviese en otro servidor, se puede indicar la dirección IP o nombre del dominio.

`root` es el usuario predeterminado de la base de datos.

`""` es la contraseña del usuario. En este caso no tiene ninguna contraseña asignada a `root`. `bdPeliculas` es el nombre de la base de datos.

3. Función para mostrar películas

```
function showMovies($db) {
    $q = "SELECT * FROM movies";
    $result = $db->query($q);
    $movies = [ ];
    while ($row = $result->fetch_assoc()) {
        $movies[ ] = $row;
    }
    return $movies;
}
```

Se define una consulta SQL básica (*SELECT * FROM movies*), que selecciona todas las columnas de la tabla *movies*. Para ejecutarla, se utiliza el método *query()*, que es parte de la clase *mysqli*. El resultado se almacena en la variable *\$result*, que va a contener todas las filas devueltas por la consulta.

Con una estructura de ciclo como *while* se recorren todos los resultados devueltos por la consulta, almacenando cada registro al array *\$movies*.

fetch_assoc() es un método que convierte cada fila de la base de datos en un array donde los *nombres* de las columnas se *convierten en claves* del array y los valores de las columnas, en valores del array.

Como en el ejemplo se tienen las columnas *id*, *title*, *year*, *poster* el resultado de *fetch_assoc()* puede verse así:

```
Array(
    "id" => 1,
    "title" => "Inception",
    "year" => 2010,
    "genre" => "Science Fiction"
)
```

Cada vez que se ejecuta el bucle, la función extrae una fila y la almacena en *\$row*, y esta es agregada al array *\$movies*.

4. Inicializar sesión en modo invitado:

```
if (!isset($_SESSION['login'])) {
    $_SESSION['login'] = false;
    $_SESSION['username'] = "invitado";
}
```

En primer lugar se verifica si la variable de sesión, *\$_SESSION['login']*, no está definida. Si no lo está, significa que, o es la primera vez que accede, o la sesión anterior caducó.

En caso de no existir la variable de sesión, se inicializa junto al nombre de usuario, que en este caso se establece como “invitado” ya que al indicar el valor de sesión login en false, se entiende que el usuario aun no ha iniciado sesión como un usuario registrado. *Este valor será el indicador de autenticación.*

5. Iniciar sesión

```
if (isset($_POST['username']) && isset($_POST['password'])) {
    $q = "SELECT * FROM users WHERE username = '" . $_POST['username'] . "'";
    $result = $db->query($q);

    if ($row = $result->fetch_assoc()) { // se recupera la primera fila devuelta
        if ($row['password'] == $_POST['password']) {

            $_SESSION['login'] = true;
            $_SESSION['username'] = $row['username'];
            $info = "Sesión iniciada correctamente";

        } else {
            $info = "Contraseña incorrecta";
        }

    } else {
        $info = "Usuario no encontrado";
    }
}
```

Primero se verifica si se han enviado los campos “username” y “password” y si están definidas en el array superglobal `$_POST` (indica si están registrados o no).

Se realiza una consulta a la base de datos, buscando tanto el usuario como la contraseña introducidos.

`$q = "SELECT * FROM users WHERE username = '" . $_POST['username'] . "'";` // \$q siempre va a definir cual es la consulta a realizar

`$result = $db->query($q);` // Siempre le va a realizar la consulta (**\$q**) a la base de datos (**\$db**).

Posteriormente, con `$row = $result->fetch_assoc()`, se recupera la primera fila devuelta por la consulta y si no hay resultados, significa que el usuario no existe y por lo tanto la función devuelve “false”.

Si por el contrario, existe, se define la variable de sesión a “true” con el nombre de usuario introducido.

6. Cerrar sesión

```
if (isset($_GET['logout'])) {
    $_SESSION['login'] = false;
    $_SESSION['username'] = "invitado";
}
```

¿Qué hace este fragmento de código?

Si en la URL aparece el parámetro “logout”, restablece el estado de la sesión a invitado, cerrando la sesión del usuario.

7. Añadir película

```
if (isset($_POST['addMovie'])) {
    if (isset($_POST['title']) && isset($_POST['year'])) {
        if ($_FILES['poster']['name'] != "") {
            $img = $_FILES['poster']['name'];
            move_uploaded_file($_FILES['poster']['tmp_name'], './img/' . $img);
        } else $img = "defecto.jpg";

        $q = "INSERT INTO movies (title, year, poster, likes) VALUES ('" . $_POST['title'] . "', '" .
        $_POST['year'] . "', '" . $img . "', 0)";
        if ($db->query($q)) {
            $info = "Película " . $_POST['title'] . " añadida";
        }
    }
}
```

Se verifica si se ha enviado el formulario de agregar película, comprobando la variable “addMovie” (atributo de un botón *name=“addMovie”*). Después se comprueba que los campos “title” y “year” se hayan proporcionado en el formulario.

Subida de la imagen:

`$_FILES` es una superglobal que se utiliza para manejar los archivos cargados mediante formularios HTML. Utiliza el método POST y el atributo **enctype=“multipart/form-data”**, que permite que el navegador envíe archivos binarios al servidor junto con los datos del formulario.

`$_FILES['poster']['name']` contiene el nombre original del archivo subido.

`$_FILES['poster']['tmp_name']` es la ubicación temporal donde se guarda el archivo en el servidor antes de ser movido a su ubicación final.

`Move_uploaded_file()` mueve el archivo desde su ubicación temporal a una carpeta llamada `img/` en el servidor.

8. Borrar películas

```
if (isset($_GET['del'])) {
    $q = "DELETE FROM movies WHERE id = " . $_GET['del'];
    $db->query($q);
    $info = "Película borrada correctamente";
}
```

De nuevo se verifica que el parámetro “del” ha sido enviado a través de la URL.

Al tener la condición `WHERE id = $_GET['del']`, asegura que solo se eliminará la película cuyo id coincida con el parámetro.

9. Añadir “like” a una película

```
if (isset($_GET['like'])) {
    $q = "UPDATE movies SET likes = likes + 1 WHERE id = " . $_GET['like'];
    $db->query($q);
    $info = "Like añadido a la película.";
    header("Location: pruebaClase.php");
}
```

Después de verificar si el parámetro “like” ha sido enviado a través de la URL, se lanza una consulta a la base de datos con la condición nuevamente de `WHERE id = $_GET['like']`, para asegurar que la operación solo afectará a la película con el id correspondiente.

Por último se realiza una redirección para evitar duplicados con `header("Location: pruebaClase.php");`. De esta forma se evita el envío del parámetro like si se actualiza la página.

Parte visual

1. Formulario de inicio de sesión

```
<form method="post" action="pruebaClase.php">
    <input type='text' name='username' placeholder="Usuario" required />
    <input type='password' name='password' placeholder="Contraseña" required />
    <input type='submit' name="login" value="Login" />
</form>
```

Este es un formulario básico de inicio de sesión, que permite a los usuarios introducir su nombre de usuario y contraseña para autenticarse en la página web.

Method = “post” indica que se enviarán los datos al servidor utilizando el método POST, que es el más indicado para tratar credenciales sensibles.

Action = “pruebaClase.php” especifica la dirección que procesará los datos. Es decir, cuando se presione el botón de envío, todos los datos se redigiran a esta dirección.

name = "username" y name = "password" son los valores a los que se accede en el bloque de PHP para procesar el inicio de sesión.

type = "submit" es el botón que envía los datos mediante el **action** ya mencionado.

name = "login" se utiliza en el servidor para verificar si se envió este formulario. (ejemplo: `isset($_POST['login'])`).

2. Formulario para añadir películas

```
<form method="post" action="pruebaClase.php" enctype="multipart/form-data">
  <input type="text" name="title" placeholder="Título de la película" required />
  <input type="number" name="year" placeholder="Año" required />
  <input type="file" name="poster" placeholder="URL del póster" />
  <input type="submit" name="addMovie" value="Añadir Película" />
</form>
```

`enctype="multipart/form-data"` es el atributo obligatorio cuando el formulario incluye la subida de archivos. Permite al navegador enviar tanto datos como archivos.

Campos de entrada:

name = "title"
name = "year"
name = "poster"
name = "addMovie"

Permiten la interacción entre el usuario y la aplicación. Recogen los datos introducidos por usuarios y los envían al servidor para su procesamiento.

3. Mostrar películas y acciones

```
$movies = showMovies($db);
foreach ($movies as $movie) {
    echo "<img src='../img/' . $movie['poster'] . '" width='50px'" . $movie['title'] . " (Likes: " .
    $movie['likes'] . ") ";
    echo '<a href="pruebaClase.php?like=' . $movie['id'] . '">Like</a> ';
    echo '<a href="pruebaClase.php?del=' . $movie['id'] . '">x </a><br>';
}
```

\$movies = showMovies(\$db) Obtiene todas las películas de la base de datos llamando a la función `showMovies`, definida anteriormente.

Se incluyen enlaces para **"likes"** y **eliminar películas**, que se han definido anteriormente.