

CÓMO SE HIZO

Para este trabajo sobre el desarrollo de una página web sobre unos multicines usé HTML y CSS para la primera parte y en esta ocasión para el desarrollo de la práctica 2 he utilizado MySQL, PHP y JavaScript. El objetivo principal de la página web es proporcionar información sobre las películas en cartelera, los horarios de las funciones entre otras cosas.

USUARIO ADMIN: **username = admin**

Password = admin

USUARIO NORMAL: **username = ugr**

Password = ugr

ESTRUCTURA DE CONTENIDOS

Para realizar el primer punto del guion, así como el 3 y 4, he implementado el siguiente código:

```
<?php

session_start();

// Verificar si el usuario ya ha iniciado sesión
if (isset($_SESSION['username'])) {

    // Mostrar mensaje de bienvenida y enlace para desloguearse
    $message = "¡Bienvenido, " . $_SESSION['username'] . "! Rol: " . $_SESSION['rol'] . ".";
    echo '<p class="mensajeBienvenida">' . $message . ' <a id="desconectar" href="desloguear.php">Desloguearse</a></p>';

} elseif ($_SERVER['REQUEST_METHOD'] === 'POST') {

    // Conexión a la base de datos
    $servername = "localhost";
    $username_db = "pwrafaelluque";
    $password_db = "---rafael--";
    $database = "dbrafaelluque_pw2223";
    $conexion = new mysqli($servername, $username_db, $password_db, $database);

    // Verificar si hay un error de conexión
    if ($conexion->connect_error) {die('Error de conexión a la base de datos: '.$conexion->connect_error);}

    // Obtener los datos ingresados por el usuario
    $username = $_POST['username'];
    $password = $_POST['password'];

    // Consulta para verificar las credenciales del usuario
    $consulta = "SELECT * FROM usuarios WHERE username = '$username' AND contrasenia = '$password'";
    $resultado = $conexion->query($consulta);
    $consulta2 = "SELECT rol FROM usuarios WHERE username = '$username' AND contrasenia = '$password'";
    $rol = $conexion->query($consulta2);

    if ($resultado->num_rows === 1) { // Verificar si se encontraron resultados
        // Usuario autenticado correctamente
        $_SESSION['username'] = $username;
```

```

$_SESSION['rol'] = $rol->fetch_assoc()['rol'];

$message = "¡Bienvenido, " . $username . "! Rol: " . $_SESSION['rol'] . ".";

echo '<p class="mensajeBienvenida">' . $message . ' <a id="desconectar" href="desloguear.php">Desloguearse</a></p>';
} else {
    // Usuario no válido

    $message = "Usuario o contraseña incorrectos. Inténtalo nuevamente.";

    echo "<p>$message</p>";

    // Mostrar formulario de inicio de sesión
    ?>

    <form id="formHead" method="POST" action="index.php">

        <label id="form1" for="username">Usuario:</label><br>

        <input type="text" id="username" name="username" value="" required /><br>

        <label id="form2" for="password">Contraseña:</label><br>

        <input type="password" id="password" name="password" value="" required />

        <br>

        <input id="form3" type="submit" id="boton1" value="Enviar" /><br>

        <a id="altaUsuarios" href="altausuarios.php">Registro de usuarios</a>

    </form>

    <?php
}

$conexion->close();
} else {
    // Mostrar formulario de inicio de sesión
    ?>

    <form id="formHead" method="POST" action="index.php">

        <label id="form1" for="username">Usuario:</label><br>

        <input type="text" id="username" name="username" value="" required /><br>

        <label id="form2" for="password">Contraseña:</label><br>

        <input type="password" id="password" name="password" value="" required /><br>

        <input id="form3" type="submit" id="boton1" value="Enviar" /><br>

        <a id="altaUsuarios" href="altausuarios.php">Registro de usuarios</a>

    </form>

    <?php

    // Mostrar mensaje de error si existe

    if (isset($message)) {echo '<p class="mensajeBienvenida">' . $message . '</p>';}

}

?>

```

Este código es muy importante ya que lo uso en todos los archivos .php. Lo que realiza este código es que verifica si el usuario ya ha iniciado sesión, en ese caso muestra el mensaje de bienvenida con su username y el enlace para desloguearse. En caso contrario, si se ha enviado el formulario para iniciar sesión con el método POST, el código php inicia la sesión con la base de datos, comprobando si se ha iniciado correctamente. Realiza la consulta SQL para verificar si el username y la contraseña introducidos existen en la base de datos. Además, hago la consulta para guardar el tipo de rol que es el usuario (admin, o usuario). Si se encontraron usuarios muestro el mensaje de bienvenida y guardo el username de la sesión. En caso de que no se encuentren usuarios con esas credenciales, muestro un mensaje de error y vuelve a aparecer el formulario de inicio de sesión. Por último, se muestra el formulario si no se cumple nada de lo anterior. Para terminar, si hay mensajes, se muestran en la página web.

Además, añadido este código php:

```

<?php
    // Comprueba si el usuario ha iniciado sesión
    if(isset($_SESSION['username']) && $_SESSION['rol'] !== 'admin') {
        $navLink1 = '<li><a href="cambiar_datos.php">Cambiar datos personales</a></li>';
    } else { $navLink1 = ''; }

    ?>

<?php
    // Comprueba si el usuario ha iniciado sesión
    if(isset($_SESSION['username']) && $_SESSION['rol'] === 'admin') {
        $navLink2 = '<li><a href="gestion_admin.php">Administrar películas</a></li>';
    } else { $navLink2 = ''; }

    ?>

<nav id="navegador">
    <ul>
        <li><a href="index.php">Inicio</a></li>
        <li><a href="estrenos.php">Estrenos</a></li>
        <li><a href="cartelera.php">Cartelera</a></li>
        <li><a href="horarios.php">Horarios</a></li>
        <li><a href="tarifas.php">Tarifas</a></li>
        <li><a href="informacion.php">Información</a></li>

        <?php echo $navLink1; ?>
        <?php echo $navLink2; ?>

    </ul>

</nav>

```

Con este código lo que consigo es que si el usuario es de tipo “usuario” se muestre en el nav el enlace para cambiar sus datos personales. En caso de que el usuario sea de tipo “admin” se mostrará en el nav el enlace para administrar las películas. Si no hay ninguna sesión iniciada no se muestran ninguno de estos enlaces.

En cuanto a ***desloguear.php*** es un código muy sencillo que cuando se le llama hace lo siguiente:

```

<?php
session_start();
session_unset();           // Destruir todas las variables de sesión
session_destroy();         // Destruir la sesión
header('Location: index.php'); // Redirigir al formulario de inicio de sesión
exit();
?>

```

Todo este código lo he puesto en todos los archivos .php para que el usuario pueda iniciar sesión o cerrar sesión en cualquier lugar de la página web, y el mensaje de bienvenida también se mostrará en todo momento. Y cuando inicie sesión o cierre la sesión, se le manda al index.php, es decir, al inicio de la página.

Index.php

Con respecto a este documento, he cambiado el diseño para que las películas guardadas en la base de datos se muestren de manera dinámica, esto quiere decir que solamente se muestran las películas que estén guardadas en la base de datos. El diseño está adaptado para que se muestren un máximo de 3 películas por sección (Más

valoradas, Cartelera, etc), en el caso de que no haya películas en la BD o para una sección, el diseño está adaptado para que muestre un mensaje de que no hay películas disponibles. El código sería el siguiente:

```
<?php

    // Conexión a la base de datos

    $servername = "localhost";

    $username_db = "pwrafaelluque";

    $password_db = "---rafael--";

    $database = "dbrafaelluque_pw2223";

    $conexion = new mysqli($servername, $username_db, $password_db, $database);

    // Verificar si hay un error de conexión

    if ($conexion->connect_error) {

        die('Error de conexión a la base de datos: '.$conexion->connect_error);

    }

    // Consulta para obtener las primeras tres películas con la categoría 'Estrenos de la semana'

    $consulta = "SELECT * FROM peliculas WHERE categoria = 'Estrenos de la semana' LIMIT 3;";

    $resultado = $conexion->query($consulta);

    // Mostrar la información de las películas

    if ($resultado->num_rows > 0) {

        for ($i = 0; $i < $resultado->num_rows; $i++) {

            $fila = $resultado->fetch_assoc();

            echo '<article class="seccionPeli" . $i . "> <a href="pele.php?titulo=' . $fila['titulo'] . "> '";

            echo '<img onmouseover="mostrarPopup(event, ' . '&#39; . $fila['titulo'] . '&#39; . ' . ' . '&#39; . $fila['tematica'] . '&#39; .

            ')" onmousemove="actualizarPosicion(event)" onmouseout="ocultarPopup()" class="peliculas" src = "' . $fila['direccionFotol'] . '" alt = "Peli1"

            width="75%"/>';

            echo '<p> . $fila['titulo'] . ' | ' . $fila['director'] . ' | ' . $fila['fecha'] . '&#128198 | ' . $fila['tiempo'] .

            '&#9200</p>';

            echo '<p>(Intérpretes)' . $fila['interpretes'] . '</p>';

            echo '<p> . $fila['valoracionMedia'] . '</p>';

            echo '</a> <a href="pele.php?titulo=' . $fila['titulo'] . "><h2> . $fila['titulo'] . '</h2></a> </article>';

        }

    } else {echo '<p class="noPeliculas">No se encontraron películas.</p>';}

    // Cerrar la conexión a la base de datos

    $conexion->close();

?>
```

Se puede ver como la información de cada película se muestra en un bucle for. Y con sentencias echo que muestran código html.

En este archivo he implementado código JavaScript para realizar el punto 9 del guion, para realizar ventanas emergentes cuando pasa el ratón por encima de la foto de cada película. Como **aspecto innovador**, he realizado el siguiente código:

```
<script>

    var ventanaEmergente = document.createElement('div');

    ventanaEmergente.className = 'popup';

    var ventanaEmergenteTitle = document.createElement('p');

    ventanaEmergenteTitle.id = 'popupTitle';

    ventanaEmergente.appendChild(ventanaEmergenteTitle);

    var ventanaEmergenteCategory = document.createElement('p');

    ventanaEmergenteCategory.id = 'popupCategory';
```

```

ventanaEmergente.appendChild(ventanaEmergenteCategory);

document.body.appendChild(ventanaEmergente);

function mostrarPopup(event, title, category) {
    console.log(event); // Imprimir el objeto event en la consola
    var popupTitle = document.getElementById('popupTitle');
    var popupCategory = document.getElementById('popupCategory');
    popupTitle.innerHTML = title;
    popupCategory.innerHTML = category;
    // Mostrar la ventana emergente
    ventanaEmergente.style.display = 'block';
    // Actualizar la posición inicial de la ventana emergente
    actualizarPosicion(event);
}

function ocultarPopup() {ventanaEmergente.style.display = 'none';}

function actualizarPosicion(event) {
    // Obtener las coordenadas del cursor
    var mouseX = event.pageX;
    var mouseY = event.pageY;
    // Ajustar la posición de la ventana emergente
    ventanaEmergente.style.left = mouseX + 'px';
    ventanaEmergente.style.top = mouseY + 'px';
}
</script>

```

Lo que realiza es 3 funciones a las cuales las llamo en cada imagen cuando las muestro con los echo. Tuve problemas para realizar esto ya que los apostrofes me los cogía mal y no se ejecutaba bien el código php por lo que tuve que poner los apostrofes, pero en lenguaje de html de esta manera: '''. La función mostrarPopup muestra la ventana emergente con el título y la categoría de la película y llama a la función actualizarPosición. Esta función lo que hace es obtener las coordenadas del cursor y actualizar la posición de la ventana emergente para que esté junto a este. La función ocultarPopup simplemente oculta la ventana emergente. Finalmente, este código si me funcionó bien después de tanto esfuerzo y considero que esta parte me ha salido muy bien.

altausuarios.php

En primer lugar, la mayor parte del código de este documento sigue igual. No obstante, he implementado este código en todos los inputs de texto: `pattern="^[^']*?*$"`

Con esto consigo que no se pueda escribir en el formulario apostrofes ya que al insertarlo en la BD me daría error porque se pensaría que es otro campo y pasaría al siguiente atributo. De esta manera me aseguro que no surja este error.

Por otro lado, he implementado un código con JavaScript como **aspecto innovador** en el que consigo que el formulario no se pueda enviar si algún campo de texto está solo rellenado con 1 o varios espacios en blanco. Si esto ocurre salta una alerta y no se envía el audio hasta que se rellene ese campo correctamente. El código es el siguiente:

```

<script>
    // Obtener referencia al formulario
    var form = document.getElementById('formAlta');

    // Agregar un evento de escucha al envío del formulario
    form.addEventListener('submit', function(event) {
        // Obtener todos los campos de entrada del formulario
        var inputs = form.getElementsByTagName('input');

        // Bandera para indicar si hay campos vacíos
        var camposVacios = false;

        // Verificar cada campo de entrada
        for (var i = 0; i < inputs.length; i++) {
            if (inputs[i].value.trim() === '') {
                camposVacios = true;
                break; // Si hay un campo vacío, se detiene el bucle
            }
        }

        // Si hay campos vacíos, se cancela el envío del formulario
        if (camposVacios) {
            event.preventDefault();
            alert('No se pueden enviar campos vacíos.');
        }
    });
</script>

```

Además, me aseguro de que se rellenen todos los campos del formulario ya que todos son obligatorios, si no se rellena alguno salta un mensaje de advertencia y no deja enviar el formulario. Por último, cuando envía el formulario lo redirijo con la función action a:

[gestoralta.php](#)

En este archivo gestiono los datos que se envían del formulario. Implemento el siguiente código:

```

<?php
session_start();

// Obtener los datos enviados desde el formulario
$nombre = $_POST['nombre']; $apellido = $_POST['apellido']; $username = $_POST['username'];
$email = $_POST['email']; $password = $_POST['password']; $confirm_password = $_POST['confirm_password'];
$phone = $_POST['phone']; $birthdate = $_POST['birthdate']; $ciudad = $_POST['ciudad']; $sexo = $_POST['sexo'];

// Validar los datos
if (empty($username) || empty($password) || empty($confirm_password) || empty($email) || empty($nombre) || empty($apellido) ||
empty($phone) || empty($birthdate) || empty($ciudad) || empty($sexo)) {
    $_SESSION['error_message'] = "Por favor, completa todos los campos.";
    header("Location: altausuarios.php");
    exit();
}

```

```

} elseif ($password != $confirm_password) {
    $_SESSION['error_message'] = "Error: contraseñas diferentes";
    header("Location: contraseña_diferente.php");
    exit();
} else {
    // Registrar los datos en una base de datos
    $servername = "localhost"; $username_db = "pwrafaelluque";
    $password_db = "--rafael--"; $database = "dbrafaelluque_pw2223";
    $conn = new mysqli($servername, $username_db, $password_db, $database);

    // Verificar si hay errores de conexión
    if ($conn->connect_error) {die("Error de conexión: " . $conn->connect_error);}

    // Verificar si el usuario ya existe
    $selectQuery = "SELECT * FROM usuarios WHERE username = '$username'";
    $result = $conn->query($selectQuery);

    if ($result->num_rows > 0) {
        $_SESSION['error_message'] = "Ese nombre de usuario ya existe.";
        header("Location: dado_no_alta.php");
        exit();
    } else {
        // Insertar el usuario en la base de datos
        $rol = "usuario";

        $insertQuery = "INSERT INTO usuarios (nombre, apellido, username, email, contrasenia, confirm_password, phone, birthdate,
ciudad, sexo, rol) VALUES (
        '$nombre', '$apellido', '$username', '$email', '$password', '$confirm_password', '$phone', '$birthdate', '$ciudad', '$sexo',
        '$rol')";

        if ($conn->query($insertQuery) === TRUE) {
            header("Location: dado_de_alta.php");
            exit();
        } else {
            $_SESSION['error_message'] = "Error al agregar el usuario: " . $conn->error;
            header("Location: altausuarios.php");
            exit();
        }
    }

    $conn->close();
}

```

Con este código obtengo los datos del formulario y los guardo en variables. Me conecto a la BD y realizo la consulta SQL para comprobar si ese nombre de usuario introducido ya existe en la BD, en caso de que ya exista muestro una alerta de error y mando al usuario a dado_no_alta.php donde se le indica que ese nombre de usuario ya existe. En caso de que las contraseñas no sean iguales también se le indica al usuario de que tienen que ser iguales. En caso contrario realiza la consulta SQL para insertar al usuario en la tabla usuarios con todos los datos extraídos del formulario. En caso de que se agregue bien, nos manda a dado_de_alta.php donde se nos indica que nos hemos registrado correctamente, en caso contrario se indica que ha habido un problema al agregar el usuario.

cambiar_datos.php

En este archivo he usado el mismo formulario que para dar de alta a un usuario. Incluido el código de JavaScript para que no se envíe el formulario en caso de que algún campo contenga solo un espacio en blanco o varios. Comprobando que se rellenan todos los campos y que no se usen apostrofes para rellenar el formulario. Lo que cambia de aquí es que cuando envía el formulario lo redirijo con la función action a:

gestor_cambiar_datos.php

Para este archivo sigo el mismo procedimiento que en todos los archivos “gestor”. En este caso parte del código para este gestor sería así:

```
$userAntiguo = $_SESSION['username'];

if($userAntiguo === $username){

    // Preparar la consulta para insertar los datos en la base de datos

    $rol = "usuario";

    $sql = "UPDATE usuarios SET nombre = '$nombre', apellido = '$apellido', username = '$username', email = '$email',
contrasenia = '$password', confirm_password = '$confirm_password', phone = '$phone', birthdate = '$birthdate', ciudad = '$ciudad', sexo
= '$sexo' WHERE username = '$username'";

    // Ejecutar la consulta
    if ($conn->query($sql) === TRUE) {

        // Mostrar un mensaje de éxito

        header( "Location: datos_cambiados.php" );

        echo "¡Los datos han sido actualizados correctamente!";

        echo "Nombre de usuario: " . $username . "<br>";

    } else {echo "Error al actualizar los datos" . $conn->error;}

}else{

    // Verificar si el username ya existe

    $selectQuery = "SELECT * FROM usuarios WHERE username = '$username'";

    $result = $conn->query($selectQuery);

    if ($result->num_rows > 0) {

        $_SESSION['error_message'] = "Este username ya existe.";

        header("Location: user_no_actualizado.php");

        exit();

    } else{

        // Preparar la consulta para insertar los datos en la base de datos

        $rol = "usuario";

        $sql = "UPDATE usuarios SET nombre = '$nombre', apellido = '$apellido', username = '$username', email = '$email',
contrasenia = '$password', confirm_password = '$confirm_password', phone = '$phone', birthdate = '$birthdate', ciudad = '$ciudad', sexo
= '$sexo' WHERE username = '$userAntiguo'";

        // Ejecutar la consulta

        if ($conn->query($sql) === TRUE) {

            // Mostrar un mensaje de éxito

            header( "Location: datos_cambiados.php" );

            echo "¡Tus datos han sido actualizados correctamente!";

            echo "Nombre de usuario: " . $username . "<br>";

            echo "Correo electrónico: " . $email . "<br>";

        } else {echo "Error al actualizar los datos" . $conn->error;}

    }

}
```


Si el username introducido es el de la sesión iniciada, se cambian los datos en esa fila de la tabla usuarios, sino se comprueba que no haya un username en la tabla igual que el username al que se quiere modificar, si existe se da un mensaje de error y se le manda a `user_no_actualizado.php` indicando que ese username ya existe. En caso contrario se actualiza la información de ese usuario y se le manda a `datos_cambiados.php` para indicarle que sus datos han sido actualizados correctamente.

En los formularios anteriores tanto el de registro como el de actualizar los datos, se comprueban que las contraseñas introducidas sean iguales, en caso contrario se le manda a `contraseña_diferente.php` indicando al usuario que las contraseñas deben de ser iguales.

gestion_admin.php

Ahora, en el caso de que hayamos iniciado sesión como “admin” nos aparecerá este archivo donde tendremos 3 enlaces que nos llevarán a 3 formularios donde podremos añadir películas, modificar películas y eliminar películas.

añadirPeli.php

En este archivo he creado un formulario para rellenar todos los campos sobre una película. Todos los campos son obligatorios, además, todos los campos tienen un máximo de longitud para cuando se guarden en la BD quepan correctamente su tamaño. Por otro lado, el campo de la duración tiene que ser mayor que 0. En los que sean de texto no se pueden incluir apostrofes por el motivo que di anteriormente. Para los inputs lo he implementado de esta manera: `pattern="^[^']*?*$"` Pero para el textarea donde escribo la sinopsis no podía utilizar esto, por lo que tuve que implementar un **aspecto innovador** con JavaScript:

```
<br><br><label class="labelAlta" for="sinopsis">Sinopsis (*) (Máximo 1500 caracteres):</label>
<textarea class="inputAlta" name="sinopsis" strlen() maxlength="1500" oninput="eliminarCaracter(this, '\')"' required></textarea>

<script>
function eliminarCaracter(elemento, caracter) {
    const valor = elemento.value;
    elemento.value = valor.replace(new RegExp(caracter, 'g'), '');
}
</script>
```

Este código elimina el carácter apostrofe o cualquier otro carácter que se le especifique. Además, se puede elegir la categoría y la valoración con la estructura “select y option”.

Por otro lado, he incluido también un código muy parecido al de los formularios de alta de usuario y cambiar datos de usuario, solo que ahora también funciona tanto en los inputs como en los textareas. Este código implementado como **aspecto innovador** no deja mandar el formulario si alguno de los campos está rellenado solamente con un espacio en blanco o varios, saltando una alerta. El código sería el siguiente:

```

<script>
    // Obtener referencia al formulario
    var form = document.getElementById('formAlta');

    // Agregar un evento de escucha al envío del formulario
    form.addEventListener('submit', function(event) {
        // Obtener todos los campos de entrada y textarea del formulario
        var inputs = form.querySelectorAll('input, textarea');

        // Bandera para indicar si hay campos vacíos
        var camposVacios = false;

        // Verificar cada campo de entrada y textarea
        for (var i = 0; i < inputs.length; i++) {
            if (inputs[i].value.trim() === '') {
                camposVacios = true;
                break; // Si hay un campo vacío, se detiene el bucle
            }
        }

        // Si hay campos vacíos, se cancela el envío del formulario
        if (camposVacios) {
            event.preventDefault();
            alert('No se pueden enviar campos vacíos.');
        }
    });
</script>

```

Por otro lado, como **aspecto innovador** he implementado una galería de imágenes para elegir que imágenes se quieren para añadir la película. Lo he hecho con JavaScript con el siguiente código, *imagenesJava.js*:

```

var imagenes = [// Rutas de todas las imágenes];
var currentIndex = 0;
var selectedImages = [];
function showImage() {
    var imageElement = document.getElementById("image");
    imageElement.src = imagenes[currentIndex];
    // Reiniciar clase seleccionada
    var prevSelectedImage = document.querySelector(".selected");
    if (prevSelectedImage) {prevSelectedImage.classList.remove("selected");}
    imageElement.classList.add("selected"); // Agregar clase seleccionada a la imagen actual
}
function previousImage() {
    currentIndex--;
    if (currentIndex < 0) {currentIndex = imagenes.length - 1;}
    showImage();
}
function nextImage() {
    currentIndex++;
    if (currentIndex >= imagenes.length) {currentIndex = 0;}
    showImage();
}

```

```

function selectImage() {
    var selectedImage = images[currentIndex];
    if (selectedImages.length < 4 && !selectedImages.includes(selectedImage)) {
        selectedImages.push(selectedImage);
        addSelectedImageElement(selectedImage);
        updateSelectedImagesInput();

        var selectedImageElements = document.querySelectorAll("#selectedImagesContainer img");
        selectedImageElements.forEach(function(imageElement) {imageElement.classList.remove("selected");});
        var lastSelectedImageElement = selectedImageElements[selectedImageElements.length - 1];
        lastSelectedImageElement.classList.add("selected");
    }
}

function validateForm() {
    if (selectedImages.length !== 4) {
        alert("Debes seleccionar exactamente 4 imágenes.");
        return false; // Evita que se envíe el formulario
    }

    return true; // Permite el envío del formulario si se han seleccionado 4 imágenes
}

function addSelectedImageElement(imagePath) {
    var selectedImagesContainer = document.getElementById("selectedImagesContainer");
    var imageElement = document.createElement("img");
    imageElement.src = imagePath;
    selectedImagesContainer.appendChild(imageElement);
}

function updateSelectedImagesInput() {
    var selectedImagesInput = document.getElementById("selectedImages");
    selectedImagesInput.value = selectedImages.join(",");
}

document.addEventListener("DOMContentLoaded", showImage);

```

Este código tiene la funcionalidad de tener 3 botones, uno para pasar a la anterior imagen, otro para pasar a la siguiente y otro para seleccionar la imagen. El código está implementado para que las imágenes estén en un ciclo, es decir, la primera esté conectada con la última. Además, se deberán seleccionar 4 imágenes obligatoriamente sino saltará una alerta. Por último, las imágenes seleccionadas se mostrarán debajo y no se podrá volver a seleccionar la misma imagen.

gestion_añadir_pelicula.php

En este archivo muy parecido al de añadir un usuario obtenemos los datos del formulario anterior, nos conectamos a la base de datos. Verificamos si la película ya existe y en ese caso se le indica en `pelel_no_añadida.php`. En caso contrario se inserta la película en la BD, si esta acción se realiza correctamente se indica en `pelicula_añadida.php`, y en caso contrario se le indica que ha habido un problema al insertarla.

modificarPeli.php

En este archivo .php realizo el mismo formulario que para añadir una película con todos los requisitos, comprobaciones, y códigos de JavaScript. Además, realizo el siguiente código php para seleccionar la película que se desea modificar. Se muestra una lista con las películas disponibles en la BD en la tabla películas. Esto lo hago para tener en una variable la película que deseo modificar:

```
<label class="labelAlta" for="tituloAntiguo">Elija que película actualizar (*):</label>

<select id="valoracion" name="tituloAntiguo" required>

<option value="">Seleccione una opción</option>

<?php

// Conexión a la base de datos

$servername = "localhost"; $username_db = "pwrafaelluque";

$password_db = "--rafael--"; $database = "dbrafaelluque_pw2223";

$conexion = new mysqli($servername, $username_db, $password_db, $database);

// Verificar si hay un error de conexión

if ($conexion->connect_error) {die('Error de conexión a la base de datos: '.$conexion->connect_error);}

// Consultar las películas en la tabla

$consulta = "SELECT * FROM peliculas";

$resultado = $conexion->query($consulta);

// Mostrar las opciones del select

if ($resultado->num_rows > 0) {

    while ($fila = $resultado->fetch_assoc()) {

        echo "<option value='" . $fila["titulo"] . "'> " . $fila["titulo"] . "</option>";

    }

}

$conexion->close();

?>
```

gestion_cambiar_datos_pelicula.php

Este archivo gestor funciona de manera muy parecida al archivo gestor_cambiar_datos.php en el que se actualizaban los datos del usuario. Si hay algún error al insertarla se le indica en peli_no_actualizada.php, y en caso de que se actualice bien la película se índice en datos_cambiados_pelicula.php.

eliminarPeli.php

Este archivo .php sirve para seleccionar que película eliminar de la BD. Se muestra mediante un formulario todas las películas disponibles en la BD de la misma manera que se mostraban las películas en el archivo modificarPeli.php mostrado anteriormente. Por lo tanto, no cabe la posibilidad de eliminar una película que no exista en la BD. Se envía el formulario al siguiente gestor mediante la acción action del formulario.

gestor_eliminar_pelicula.php

Este gestor elimina la película enviada por el formulario mediante la siguiente consulta SQL:

```
$sql = "DELETE FROM peliculas WHERE titulo = '$titulo';";
```

Si todo ha salido bien se indica en datos_pelicula_eliminada.php, y en caso contrario se muestra un mensaje de error.

Hasta aquí estaría todo lo relacionado con los formularios y archivos relacionados con los datos de los usuarios y con la administración del admin para añadir, modificar y eliminar películas. A continuación, voy a explicar la parte del archivo `peli.php` junto con la parte de comentarios y explicar cómo he implementado de manera dinámica los archivos de `cartelera.php`, `próximas.php`, etc.

`peli.php`

En primer lugar, obtengo el título de la película desde la URL y obtengo todos los datos de esa película guardados en la BD. En el caso de que no se encuentre la película saldrá un mensaje de error.

```
// Obtener la información de la película de la base de datos
$title = $_GET["titulo"]; // Obtener el ID de la película desde la URL

$sql = "SELECT * FROM peliculas WHERE titulo = '$title'";
$result = $conn->query($sql);
```

Una vez que tengo guardados todos los datos simplemente es mostrarlos mediante un `echo`. De esta manera ya he implementado el código para que muestra la información de las películas de forma dinámica, en vez de tener 20 archivos `.html` como en la práctica anterior.

De una manera parecida muestro de manera dinámica los posibles comentarios que haya guardados en la BD de esa película, de la siguiente manera:

```
// Consulta para obtener las primeras tres películas con la categoría 'Estrenos de la semana'
$consulta = "SELECT * FROM comentarios WHERE titulo = '{$_GET['titulo']}'";
$resultado = $conexion->query($consulta);

// Mostrar la información de las películas
$_SESSION['titulo'] = $title;
if ($resultado->num_rows > 0) {
    for ($i = 0; $i < $resultado->num_rows; $i++) {
        $fila = $resultado->fetch_assoc();
        echo '<li><h3 class="nombreU">Nombre de usuario: </h3><p>' . $fila['username'] . '</p>';
        echo '<h3 class="comentarioU">Comentario del usuario: </h3><p>' . $fila['comentario'] . '</p>';
        echo '<section class="rating"><h3 class="valoracionU">Valoración: </h3>';
        echo '<article class="star">' . $fila['valoracionMedia'] . '</article></section></li>';
    }
} else {
    echo 'No se encontraron comentarios.';
}
```

Con el bucle `for` se mostrarán todos los comentarios existentes que haya de esa película, en caso de que no haya comentarios se muestra que no se encontraron comentarios.

Por otro lado, he implementado que el formulario para añadir un comentario solo aparezca a los usuarios de tipo "usuario" que estén con la sesión iniciada. Esto lo hago de esta manera:

```
// >php
// Comprueba si el usuario ha iniciado sesión
if(isset($_SESSION['username']) && $_SESSION['rol'] !== 'admin') {
    $navLink3 = '<form id="formComentarios" action="gestorComentarios.php" method="POST">

<label for="comentario">Comentario (Máximo 200 caracteres)</label>
<textarea id="comentario" name="comentario" strlen() maxlength="200" oninput="eliminarCaracter(this, \' . \'&#39;\' . \'&#39;\' . \'&#39;\' . \'&#39;\' . \') "
required></textarea>

<label for="valoracionMedia">Valoración:</label>
<select id="valoracionMedia" name="valoracionMedia" required>
    <option value="">Seleccione una opción</option>
    <option value="#11088 #10025 #10025 #10025 #10025">#11088 #10025 #10025 #10025 #10025</option>
    <option value="#11088 #11088 #10025 #10025 #10025">#11088 #11088 #10025 #10025 #10025</option>
    <option value="#11088 #11088 #11088 #10025 #10025">#11088 #11088 #11088 #10025 #10025</option>
    <option value="#11088 #11088 #11088 #11088 #10025">#11088 #11088 #11088 #11088 #10025</option>
    <option value="#11088 #11088 #11088 #11088 #11088">#11088 #11088 #11088 #11088 #11088</option>
</select>

<button type="submit">Enviar</button>

</form>;
} else {
    $navLink3 = '';
}
}>

<script>
function eliminarCaracter(elemento, caracter) {
    const valor = elemento.value;
    elemento.value = valor.replace(new RegExp(caracter, 'g'), '');
}
</script>
```

De esta manera el formulario también aparecerá de forma dinámica `<?php echo $navLink3; ?>`. Como ya hice en otros formularios, para este textarea he implementado de nuevo como **aspecto innovador** el script con JavaScript para que no se puedan escribir apostrofes en este textarea y así no de errores al introducir los datos en la BD. Este formulario se envía a:

[gestorComentarios.php](#)

Este gestor se en carga de añadir los comentarios a la BD en la tabla comentarios. Esta tabla no tiene clave primaria, de esta manera cualquier usuario podrá escribir todos los comentarios que él quiera en cualquier película. La consulta SQL sería la siguiente:

```
// Preparar la consulta para insertar los datos en la base de datos
$sql = "INSERT INTO comentarios (username, comentario, valoracionMedia, titulo) VALUES('$username', '$comentario', '$valoracionMedia', '$titulo')";

// Ejecutar la consulta
if ($conn->query($sql) === TRUE) {
    // Mostrar un mensaje de éxito
    header("Location: peli.php?titulo=" . $titulo);
    echo "¡El comentario ha sido añadida correctamente!";
} else {
    echo "Error al añadir el comentario" . $conn->error;
}
```

En caso de que se añada correctamente se vuelve a la página de esa película, y en caso de que no se haya añadido correctamente se indica con un mensaje de error.

Por último, para los archivos [cartelera.php](#), [estrenos.php](#), [masvaloradas.php](#) y [próximas.php](#) he implementado que se vayan mostrando las películas que haya en la BD de forma dinámica mediante el siguiente código:

```
<?php

    // Conexión a la base de datos

    $servername = "localhost"; $username_db = "pwrafaelluque";

    $password_db = "--rafael--"; $database = "dbrafaelluque_pw2223";

    $conexion = new mysqli($servername, $username_db, $password_db, $database);

    // Verificar si hay un error de conexión

    if ($conexion->connect_error) {die('Error de conexión a la base de datos: '.$conexion->connect_error);}

    // Consulta para obtener las primeras tres películas con la categoría 'Estrenos de la semana'
```

```

$consulta = "SELECT * FROM peliculas WHERE categoria = 'Cartelera';";
$resultado = $conexion->query($consulta);
// Mostrar la información de las películas
if ($resultado->num_rows > 0) {
    for ($i = 0; $i < $resultado->num_rows; $i++) {
        $fila = $resultado->fetch_assoc();
        echo '<div class="movie"><article class="movie_info">';
        echo '<a href="peli.php?titulo=' . $fila['titulo'] . '"><h2 class="titPeli"> . $fila['titulo'] . '</h2></a> ';
        echo '<p class="infoPeli"> . $fila['titulo'] . ' | ' . $fila['director'] . ' | ' . $fila['fecha'] . '&#128198 | ' .
        $fila['tiempo'] . '&#9200</p>';
        echo '<p class="infoPeli">(Intérpretes) ' . $fila['interpretes'] . '</p>';
        echo '<p class="infoPeli"> . $fila['valoracionMedia'] . '</p></article>';
        echo '<article class="cuadroPeliculas"><a href="peli.php?titulo=' . $fila['titulo'] . '">';
        echo '<img src = "" . $fila['direccionFoto1'] . '" alt = "Peli1"/></a></article></div>';
    }
} else {echo '<p class="noPeliculas">No se encontraron películas.</p>';}
$conexion->close();
?>

```

Obtengo todas las películas con la categoría pertinente, compruebo si hay películas. En caso de que no haya muestro un mensaje de que no se encontraron películas. Y en caso de que si haya las muestro todas una debajo de otra mediante un bucle for. Toda la información de las películas las muestro mediante código html en sentencias echo. De esta manera se mostrarían todas las películas que hubiera en la BD de esa categoría de forma dinámica.

Hasta aquí llegaría la explicación de esta práctica, si tienes alguna duda más no dudes en preguntarme. Estaré encantado de resolver tus dudas. Saludos.