



Memoria Práctica 2

PROGRAMACIÓN WEB

JUAN PEDRO MORENO RUIZ

Índice

ÍNDICE	1
INTRODUCCIÓN	2
USUARIOS	3
Registro de Usuario	3
Modificar el Perfil	4
Iniciar Sesión	5
PELÍCULAS	6
Añadir Películas	7
Modificar Películas	10
Eliminar Película	11
COMENTARIOS	12
Crear Comentario	12
PÁGINA PRINCIPAL	13

Introducción

El objetivo de la práctica era actualizar la web hecha para la primera práctica, aunque existía la opción de usar frameworks, solo he usado PHP y JS puro para la entrega.

Las actualizaciones se podrían dividir en 3 grandes bloques: las relacionadas con los comentarios, con las películas y con los usuarios.

Para los comentarios, había que conseguir que los usuarios pudieran mandar comentarios a las películas y estos aparecieran cada uno con su película.

Las películas ahora tenían que ser registradas en una base de datos y tenían que mostrarse en la web de forma dinámica.

Por último, con respecto a los usuarios, el registro de usuario ahora tenía que ser real y una vez que el usuario se hubiera registrado, puede hacer comentarios a las películas y darle una nota. Además, existen 2 tipos de usuario, los administradores y los usuarios normales, los primeros pueden crear, modificar y eliminar películas mientras que los usuarios normales solo pueden hacer comentarios.

Como la práctica estaba enfocada en JS y PHP, solo voy a explicar lo estrictamente necesario de los documentos HTML y el CSS.

Usuarios

Para crear un usuario en el sistema, primero tuve que crear una tabla en la Base de Datos, descrita a continuación:

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	auto_increment
Nombre	varchar(50)	NO		NULL	
Apellido	varchar(50)	NO		NULL	
Fecha_de_nacimiento	date	NO		NULL	
Newsletter	tinyint(1)	NO		0	
Mail	varchar(50)	NO		NULL	
Contraseña	varchar(255)	NO		NULL	
tipo	varchar(50)	YES		cliente	

La clase *usuario.php* contiene todos los métodos que he usado para hacer la práctica y en su archivo se puede ver la implementación, pero a modo de resumen solo indicar que estos métodos son: *obtenerUsuario(\$id)*, *obtenerUsuarioByMail(\$email)*, *modificarUsuario(\$id, \$email, \$nombre, \$apellidos, \$nacimient, \$news)* y *insertarUsuario(\$email, \$nombre, \$apellidos, \$nacimient, \$contra, \$news)*.

Las cosas que un usuario puede hacer son registrarse, modificar su perfil, iniciar sesión y comentar, aunque esto último lo explico en el apartado de Comentarios.

Registro de Usuario

El usuario se puede registrar en la web usando el mismo formulario que usé en la Práctica 1, solo que ahora funciona de verdad:

```
<form class="registro" action="registrarUsuario.php" method="post"
onsubmit="return validarRegistro();">
  <label for="email">E-mail</label><br>
  <input type="email" id="mail" name="email"><br>

  <label for="nombre">Nombre</label><br>
  <input type="text" id="nombre" name="nombre"
pattern="^(?!.*administrador).*$"><br>

  <label for="apellidos">Apellidos</label><br>
  <input type="text" id="apellido" name="apellidos"
pattern="^(?!.*administrador).*$"><br>

  <label for="nacimient">Fecha de Nacimiento</label><br>
  <input type="date" id="nacimient" name="nacimient"><br>

  <label for="contraseña">Contraseña</label><br>
  <input type="password" id="contra" name="contraseña"><br>

  <label for="newsletter">¿Desea suscribirse al
Newsletter?</label><br>
```

```

        <input type="radio" id="newsletter" name="newsletter" value="1"
checked>Si
        <input type="radio" id="newsletter" name="newsletter"
value="0">No<br>

        <input type="submit" id="boton" value="Registrarse en Cines
Moreno"><br>
</form>

```

Al pulsar el botón, primero se comprueba si usando JavaScript si el formulario es correcto y no hay campos vacíos, para luego enviar los datos al script de PHP registrarUsuario.php. En el script solamente se obtienen los datos del formulario y se comprueba si el correo ya está usado con la función *obtenerUsuarioByMail(\$email)*, si lo está se avisa al usuario y si no, se registra al usuario en la Base de Datos usando la función *insertarUsuario(\$email, \$nombre, \$apellidos, \$nacimient, \$contra, \$news)*.

Modificar el Perfil

Para modificar su perfil, el usuario tiene un botón en la cabecera de la página web que, al pulsarlo, le lleva a un documento con un formulario relleno con sus datos de manera automática:

Tu ID de usuario es: 1

¿Quieres cambiar alguno de los datos de tu usuario?

E-mail

Nombre

Apellidos

Fecha de Nacimiento

¿Desea suscribirse al Newsletter?
☒ Si ☐ No

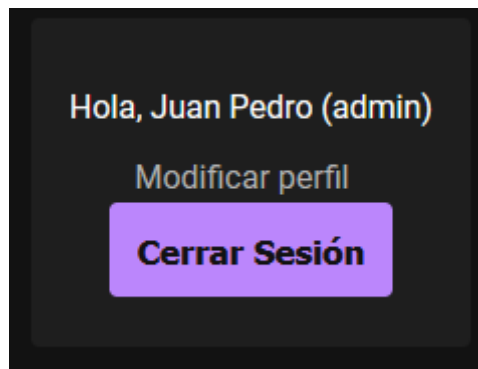
Guardar

Una vez el usuario ha cambiado sus datos y pulsa el botón *Guardar*, se ejecuta un script llamado *modificarPerfil.php*, que, aunque es casi idéntico al de registrar usuarios, en este caso se usa la función de *modificarUsuario(\$id, \$email, \$nombre, \$apellidos, \$nacimient, \$news)*.

Iniciar Sesión

Por último, el usuario es capaz de iniciar sesión el web usando un formulario muy simple en la cabecera, que le pide el correo y la contraseña para después ejecutar el script *iniciarSesion.php*.

Después de obtener los datos del usuario con *obtenerUsuarioByMail(\$email)*, realiza dos comprobaciones importantes, que el usuario exista y si la contraseña introducida es correcta. En el caso de que algunas de estas dos cosas no se cumplan, mostrará al usuario un mensaje para indicarlo, pero si se cumplen las dos, se habrá iniciado sesión correctamente y el usuario lo sabrá ya que la cabecera documento habrá cambiado para mostrar su nombre y su tipo de usuario.



Al pulsar el botón de *Cerrar Sesión*, se ejecuta el script de PHP *cerrarSesion.php*, que solo borra los datos de la sesión y hará que el usuario tenga que volver a identificarse si desea comentar o modificar sus datos.

```
<?php
    session_start();

    // Eliminar todas las variables de sesión
    $_SESSION = array();

    // Destruye la sesión
    session_destroy();

    header("Location: index.php");
    exit();
?>
```

Películas

Para registrar películas también he creado una tabla en la Base de Datos con la siguiente estructura:

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	auto_increment
Titulo	varchar(255)	YES		NULL	
Director	varchar(255)	YES		NULL	
Genero	varchar(50)	YES		NULL	
Duracion	int(11)	YES		NULL	
Sinopsis	text	YES		NULL	
Fecha_de_lanzamiento	date	YES		NULL	
Actores_principales	text	YES		NULL	
ImagenID	int(11)	YES	MUL	NULL	
Categoria	varchar(255)	YES		NULL	

Lo único para destacar es que la imagen de la película se almacena en otra tabla que ya contiene todas las imágenes disponibles, y se hace referencia a estas usando la clave externa ImagenID.

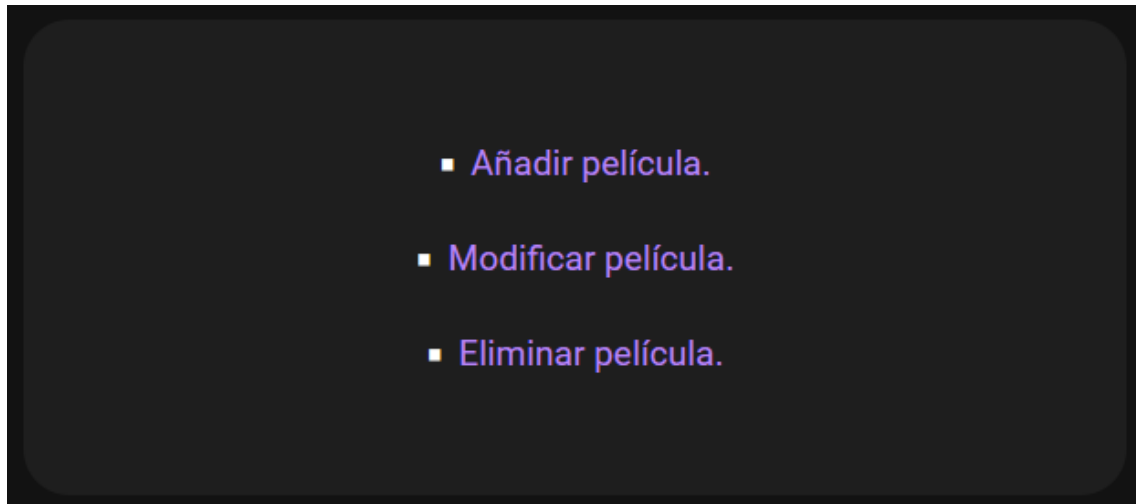
La implementación de la clase Película se encuentra en *pelicula.php*, y contiene los métodos: *obtenerPelicula()*, *modificarPelicula()*, *insertarPelicula()*, *eliminarPelicula()*, *obtenerPeliculas()*, *obtenerValoracionMedia()*.

También decir, que solo los usuarios que sean del tipo *admin* pueden realizar acciones con las películas a través del Panel de Administrador, que aparece en el índice cuando un usuario admin inicia sesión, gracias a este código HTML y PHP en la cabecera de los documentos:

```
<nav class="indice">
    <article><a href="index.php">Inicio</a></article>
    <article><a href="estrenos.php">Estrenos</a></article>
    <article><a
href="cartelera.php">Cartelera</a></article>
    <article><a href="horarios.php">Horarios</a></article>
    <article><a href="tarifas.php">Tarifas</a></article>
    <article><a
href="informacion.php">Información</a></article>
    <?php if (isset($_SESSION['tipo']) && $_SESSION['tipo']
== 'admin') : ?>
        <article><a href="adminpanel.php">Panel de
Administrador</a></article>
    <?php endif; ?>
</nav>
```

Solo si el usuario es Admin, se le agrega al índice el Panel de Administrador que se encuentra en el archivo *adminpanel.php*.

Una vez el administrador ha entrado en su Panel de Administrador, se encuentra con tres enlaces, el primero para añadir películas, el segundo para modificar las películas existentes y el tercero para eliminar películas:



Añadir Películas

Para añadir películas el administrador tiene que rellenar un formulario con los datos, de este formulario solo quiero destacar que a parte del script para comprobar que los datos introducidos son correctos, he creado el siguiente script:

```
// Cuando se cambie la opción del select, cambiar la imagen
document.getElementById('imagen').addEventListener('change',
function() {
    let idElegido = this.options[this.selectedIndex];

    let imagenDatos = idElegido.getAttribute('imagenDatos');

    let imgElement = document.getElementById('imagenSeleccionada');

    imgElement.src = 'data:image/png;base64,' + imagenDatos;
});
```

El objetivo es que, cuando se cambie la imagen de la película en el *select* del formulario, aparezca debajo del mismo la nueva imagen seleccionada, actualizándose de manera dinámica los atributos del siguiente *article*:

```
<article class="imagen_seleccionada"><img id="imagenSeleccionada"
src="" alt="Imagen seleccionada"></article>
```

Una vez que se han rellenado todos los datos y se pulsa el botón de *Registrar*, se ejecuta el script *registrarPelícula.php*, que usa el método *insertarPelícula()* para añadirla a la base de datos y por último, redirige al documento *templatePelícula.php* usando el *ID* de la película recién añadida para modificarlo con sus datos.

La plantilla que usan las películas obtiene los datos que va a usar así:

```
<?php
session_start();

require_once("pelicula.php");
require_once("imagen.php");
require_once("comentario.php");
require_once("usuario.php");

// Obtener el ID de la película
$id = $_GET["id"];

// Obtener el ID y correo del usuario
if (isset($_SESSION["id"]) && isset($_SESSION["email"])) {
    $idUserio = $_SESSION["id"];
    $correoUsuario = $_SESSION["email"];
}

// Obtener la película desde la base de datos
$pelicula = Pelicula::obtenerPelicula($id);

// Obtener los comentarios de la película
$comentarios = Comentario::obtenerComentariosPelicula($id);
?>
```

Una vez almacenados todos los datos, incluidos los comentarios que explicaré más adelante, solo los muestra en su lugar usando *echo()* de PHP. Los comentarios son comprobados usando JavaScript, no pueden ser vacíos, y solo se mostrará el formulario si el usuario ha iniciado sesión en la web.

```
<?php if (isset($_SESSION['inicio_sesion'])) { ?>
<h2>Enviar comentario</h2>
<form class="formulario_comentario" action="registrarComentario.php"
method="post" onsubmit="return validarComentario();" >

    <label for="comentario">Comentario</label><br>
    <textarea name="comentario" id="comentario" rows="10" cols="40"
placeholder="Escribe tu comentario..."></textarea><br>

    <label for="valoracion">Valoración</label><br>
    <input type="number" id="valoracion" name="valoracion" min="0"
max="5" step="0.5"><br>

    <input type="hidden" name="peliculaID" value="<?php echo $id; ?>">
    <input type="hidden" name="usuarioID" value="<?php echo
$idUsuario; ?>">

    <input type="submit" id="boton" name="Enviar">
```

```

</form>
<?php } else { ?>
    <p>Por favor, inicia sesión para comentar. </p>
<?php } ?>

```

Enviar comentario

Comentario

Escribe tu comentario...

Valoración



Enviar consulta

Comentarios (1)

Juan Pedro

No me ha gustado nada!!!

1/5

Por favor, inicia sesión para comentar.

Por último, sobre esta plantilla, es que las estrellas se actualizan dependiendo de la Valoración Media, que será la media de todas las valoraciones de los comentarios. Para conseguir esto, he usado PHP para mostrar las estrellas con una clase u otra, que será lo que le indique al CSS si la estrella debe tener más o menos opacidad.

```

<?php
// Obtener la valoración media de la película desde la base de
datos
$valoracionMedia = Pelicula::obtenerValoracionMedia($id);

// Mostrar las estrellas de clasificación según la valoración media
for ($i = 1; $i <= 5; $i++) {
    if ($i <= $valoracionMedia) {
        echo '<article class="estrella_llena"></article>';
    } else {
        echo '<article class="estrella_vacia"></article>';
    }
}
?>

```

Modificar Películas

El proceso de modificar los datos de una película es el mismo que un usuario sigue para modificar sus datos en el perfil. El formulario se rellena automáticamente con los datos de la película seleccionada, y cambian dinámicamente cuando se elige una nueva película, esto lo hago con el siguiente script de JavaScript:

```
let peliculasData = <?php echo json_encode($peliculas_data); ?>;

let selectPelicula = document.getElementById('pelicula');
let idText = document.getElementById('id');
let tituloText = document.getElementById('titulo');
let directorText = document.getElementById('director');
let generoSelect = document.getElementById('genero');
let duracionText = document.getElementById('duracion');
let sinopsisText = document.getElementById('sinopsis');
let fechaText = document.getElementById('fecha');
let actoresText = document.getElementById('actores');
let imagenSelect = document.getElementById('imagen');
let categoriaSelect = document.getElementById('categoria');

function actualizaValores() {
    let id =
selectPelicula.options[selectPelicula.selectedIndex].value;
    let pelicula = peliculasData[id];

    idText.value = id;
    tituloText.value = pelicula.titulo;
    directorText.value = pelicula.director;
    duracionText.value = pelicula.duracion;
    sinopsisText.value = pelicula.sinopsis;
    fechaText.value = pelicula.fecha;
    actoresText.value = pelicula.actores;
    imagenSelect.value = pelicula.imagen;
    categoriaSelect.value = pelicula.categoria;

    for (let i = 0; i < generoSelect.options.length; i++) {
        if (generoSelect.options[i].value === pelicula.genero) {
            generoSelect.options[i].selected = true;
            break;
        }
    }
}

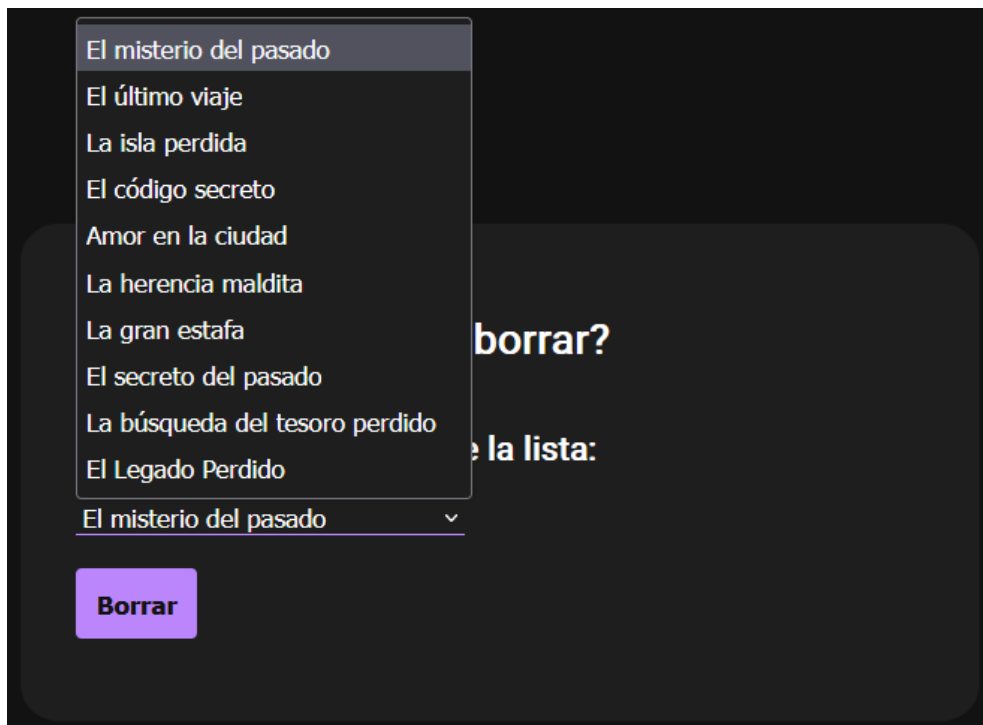
selectPelicula.addEventListener('change', actualizaValores);
window.onload = actualizaValores;
```

Primero guardo todos los elementos que quiero actualizar, y cuando hay un cambio en el *select*, se ejecuta una función que actualiza los valores de todos los campos que he guardado.

Por último, decir que el formulario envía los datos a un script de PHP llamado *modificarPelícula.php*, que recoge los datos del formulario y usa el método *modificarPelícula()* para cambiar los valores de la Base de Datos, y termina redirigiendo al usuario a la página de la película.

Eliminar Película

La última opción que tiene el administrador con respecto a las películas es eliminar una de la base de datos. Para ello, solo tiene que elegir la que quiera de la lista y pulsar el botón *Borrar*.



Una vez pulsa el botón, aparece un mensaje de confirmación que, si el administrador acepta, ejecuta el script de PHP *eliminarPelícula.php*:

```
<?php
require_once("película.php");

if (isset($_GET['id'])) {
    $id = $_GET['id'];

    Pelicula::eliminarPelicula($id);
}

header("Location: adminpanel.php");
exit();
?>
```

Comentarios

Los comentarios se almacenan en la Base de Datos siguiendo la siguiente estructura:

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	auto_increment
ID_usuario	int(11)	NO	MUL	NULL	
ID_pelicula	int(11)	NO	MUL	NULL	
Comentario	text	NO		NULL	
Puntuacion	int(11)	YES		NULL	
Fecha_de_publicacion	datetime	YES		NULL	

Se almacena el comentario, la puntuación, la fecha y dos claves externas, una para el usuario que realiza el comentario y otra para la película donde lo hace.

Por otro lado, la clase que se encarga de tratar los comentarios en la página web se encuentra en el archivo *comentario.php* y contiene los siguientes métodos: *obtenerComentariosPelicula()*, *insertarComentario()* y *eliminarComentariosPelicula()*.

Crear Comentario

Como he comentado en el apartado anterior, un usuario solo podrá poner un comentario y una valoración a una película si está registrado en la página web y ha iniciado sesión con anterioridad.

Una vez que el usuario ha envidado el formulario correctamente, se ejecuta el script *registrarComentario.php*, que solo guarda el comentario en la base de datos y redirige al usuario a la misma película que ha comentado:

```
<?php
require_once("comentario.php");

$comentario = $_POST["comentario"];
$valoracion = $_POST["valoracion"];
$fecha = date('Y-m-d');
$peliculaID = $_POST["peliculaID"];
$usuarioID = $_POST["usuarioID"];

$comentario = Comentario::insertarComentario($peliculaID,
$usuarioID, $comentario, $valoracion, $fecha);

header("Location: templatePelicula.php?id=$peliculaID");
exit();
?>
```

Página Principal

Por último, quiero comentar la manera en la que se muestran las películas en la página principal (que también se aplica de la misma manera a los documentos de las categorías).

Primero, guardo todas las películas por categoría y ordenadas de manera descendente por valoración, con este código en PHP:

```
require_once("pelicula.php");
require_once("imagen.php");

// Obtener las películas de la base de datos
$peliculas = Pelicula::obtenerPeliculas();

// Clasificar películas por categoría
$peliculasCartelera = array();
$peliculasProximamente = array();
$peliculasEstreno = array();

foreach ($peliculas as $pelicula) {
    if ($pelicula->devolverValor("Categoria") == "Cartelera") {
        $peliculasCartelera[] = $pelicula;
    } else if ($pelicula->devolverValor("Categoria") ==
"Proximamente") {
        $peliculasProximamente[] = $pelicula;
    } else if ($pelicula->devolverValor("Categoria") == "Estrenos")
{
        $peliculasEstreno[] = $pelicula;
    }
}

// Guardo los id de las películas ordenadas por valoración media
$peliculasValoradas = array();

foreach ($peliculas as $pelicula) {
    $id = $pelicula->devolverValor("ID");
    $valoracionMedia = Pelicula::obtenerValoracionMedia($id);
    $peliculasValoradas[$id] = $valoracionMedia;
}

arsort($peliculasValoradas);
```

Luego, para cada categoría obtengo 3 películas, que son las que debería haber en cada categoría en la página principal. Cada película será representada por su imagen y su título, usando un bucle que recorre los 3 primeros elementos del array de la categoría.

Por último, he de decir que el punto 9 del guion lo he resuelto usando *tooltips*, dentro del *article* de cada película he incluido el siguiente *span*:

```
<span class="tooltip_texto"><?php echo $peliculaEstreno-  
>devolverValor("Titulo") ?> - <?php echo $peliculaEstreno-  
>devolverValor("Categoria") ?></span>
```

Este *span* se muestra pasando el ratón por encima de la imagen de cada película, para esto he usado JavaScript como se indicaba en el guion:

```
// Script para mostrar el tooltip  
document.addEventListener('DOMContentLoaded', (event) => {  
  // Seleccionar todos los elementos con la clase tooltip  
  const tooltips = document.querySelectorAll('.tooltip');  
  
  // Recorrer los elementos y añadir los eventos  
  tooltips.forEach((tooltip) => {  
    // Añadir el evento mouseover  
    tooltip.addEventListener('mouseover', () => {  
      const tooltipTexto = tooltip.querySelector('.tooltip_texto');  
      tooltipTexto.style.visibility = 'visible';  
      tooltipTexto.style.opacity = 1;  
    });  
  
    // Añadir el evento mouseout  
    tooltip.addEventListener('mouseout', () => {  
      const tooltipTexto = tooltip.querySelector('.tooltip_texto');  
      tooltipTexto.style.visibility = 'hidden';  
      tooltipTexto.style.opacity = 0;  
    });  
  });  
});
```