

Práctica 2 – Multicines FAF



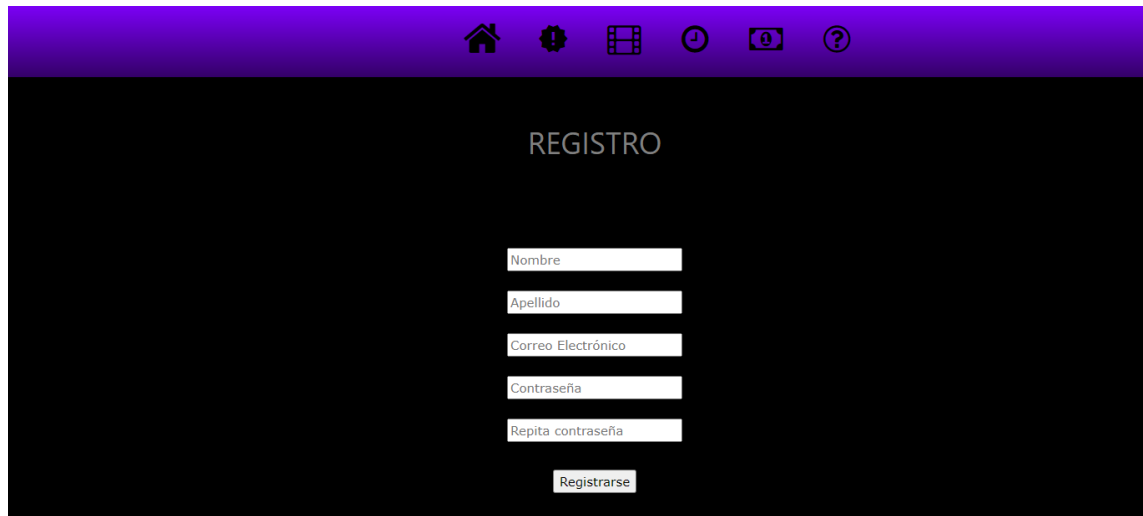
Índice

- Descripción del trabajo → Página 3
- Estructura → Página 3
- Aspectos innovadores → Página 4

1. Descripción del trabajo

1.1. Alta de usuario

En esta sección se ha realizado un formulario de registro de un usuario.



The screenshot shows a web application interface with a dark theme. At the top, there is a purple navigation bar containing several icons: a home icon, a settings gear, a film strip, a clock, a wallet, and a question mark. Below the navigation bar, the main content area is dark gray and features the word 'REGISTRO' in white capital letters. Centered on the page is a registration form with five white input fields stacked vertically, labeled 'Nombre', 'Apellido', 'Correo Electrónico', 'Contraseña', and 'Repita contraseña'. Below these fields is a white button labeled 'Registrarse'.

Aquí se añaden los valores de nombre, apellido, correo electrónico y contraseña. Para el caso de la contraseña se añade un input en el que se debe de repetir la contraseña introducida, esta comprobación se hace con JavaScript:

```
function validacionPass(){
    var passwd = document.getElementById('passwd');
    var rep_passwd = document.getElementById('rep_passwd');

    if(passwd.value != rep_passwd.value){
        alert("Las contraseñas no coinciden");
        return false;
    }
}
```

El formulario sería el siguiente:

```
<form id="formRegUser" name="formRegUser" action="altausuario.php" method="post" onsubmit="return valida"
  <input type="text" id="nombre" name="nombre" placeholder="Nombre" maxlength="50" required><br><br>
  <input type="text" id="apellido" name="apellido" placeholder="Apellido" maxlength="50" required><br><br>
  <input type="email" id="email" name="email" placeholder="Correo Electrónico" maxlength="50" required>
  <input type="password" id="passwd" name="passwd" placeholder="Contraseña" minlength="5" maxlength="26"
  <input type="password" id="rep_passwd" name="rep_contraseña" placeholder="Repita contraseña" minlength="5"
  <input type="submit" value="Registrarse">
</form>
```

Podemos observar que todos los campos son obligatorios. Ahora pasamos a `altausuario.php`, donde se procesa este formulario.

```
<?php
6 references | 0 implementations
class Formulario
{
    4 references
    private $email;
    4 references
    private $nombre;
    4 references
    private $apellido;
    4 references
    private $passwd;

    6 references | 0 overrides
    function __construct($atributos)
    {
        $this->email = $atributos['email'];
        $this->nombre = $atributos['nombre'];
        $this->apellido = $atributos['apellido'];
        $this->passwd = $atributos['passwd'];
    }

    6 references | 0 overrides
    public function getEmail()
    {
        return $this->email;
    }

    6 references | 0 overrides
    public function getNombre()
    {
        return $this->nombre;
    }

    3 references | 0 overrides
    public function getApellido()
    {
        return $this->apellido;
    }
}
```

```

        public function getPassword()
        {
            return $this->passwd;
        }
    }

    $formulario = new Formulario($_POST);

    require_once('classes/usuario.class.inc');
    $usuario = new User();
    $all_users = $usuario->obtenerUsuarios();

    $validacion = true;
    $nombre = false;
    $email = false;

    foreach($all_users as $user){
        if($user['email'] == $formulario->getEmail()){
            $email = true;
            $validacion = false;
            break;
        }

        if($user['nombre'] == $formulario->getNombre()){
            $nombre = true;
            $validacion = false;
            break;
        }
    }

```

```

    if($validacion){
        session_start();

        $usuario->insertarUsuario($formulario->getNombre(), $formulario->getPassword(), $formulario->getEmail(), $formul
        $_SESSION["nombre"] = $formulario->getNombre();
        $_SESSION["email"] = $formulario->getEmail();
        $_SESSION['rol'] = 'user';
        header("Location: index.php");
    }
    else{
        if($nombre){
            echo('
                <script>
                window.onload = function() {
                    alert("Nombre ya registrado");
                }
            </script>
            ');
        }
        else{
            echo('
                <script>
                window.onload = function() {
                    alert("Email ya registrado");
                }
            </script>
            ');
        }
    }

    exit;
?>

```

Para esto he creado la clase formulario (estructura que seguiré en todos los procesos de registro de la practica) que recibe como argumentos los datos del formulario con \$_POST. Primero se hará una comprobación de si el nombre de usuario o el email ya han sido registrados, en caso afirmativo se denegará el registro y se mostrará una ventana emergente para señalar el error. Para el registro de usuarios he creado una tabla en la base de datos que contendrá todos los datos introducidos además del rol del usuario que por defecto siempre será "user". Una vez registrado los datos del usuario aparecerán en la cabecera de la página.



1.2. Modificar datos personales

En esta sección se abarca el tema de cambio de datos por parte de un usuario. Para esto se hará click en el apartado Cambiar datos en la cabecera (siempre y cuando se esté logueado). Aquí se cogerán los siguientes datos:

The image is a screenshot of a web page with a dark background. At the top, the text 'DATOS PERSONALES' is displayed in a light green, sans-serif font. Below this, the user's details are listed in a light purple font: 'Nombre: Alvaro', 'Apellido: Fernandez', and 'Correo Electrónico: varo12ff@gmail.com'. Further down, the text 'CAMBIAR DATOS PERSONALES' is shown in the same light green font. At the bottom, there are five white input fields stacked vertically, each with a light green placeholder text: 'Nuevo nombre', 'Nuevo apellido', 'Nuevo correo Electrónico', 'Nuevo contraseña', and 'Repita contraseña'.

Se mostrarán los datos del usuario (obviando la contraseña) y un formulario que sigue la misma estructura que para el registro con el cambio de que en este caso los campos no son obligatorios, si uno de los campos está vacío se ignora en el php y no se cambia ese dato.

1.3. Permitir iniciar sesión

Como ya se ha visto antes una vez logueado se visualiza en index.php los datos del usuario para esto se ha hecho el siguiente formulario:

```
<?php
require_once('classes/usuario.class.inc');

$usuario = new User();
session_start();
if (!isset($_SESSION['email'])) {
    echo ( '
        <form action="login.php" method="post" name="formLog">
        <input type="text" id="email" name="email" placeholder="Correo Electrónico"><br><br>

        <input type="password" id="passwd" name="passwd" placeholder="Contraseña"><br><br>

        <input type="submit" value="Iniciar sesión">
        </form>

        <a href="altausuario.html">Registro de usuarios</a>
        ');
} else {
    echo ( ' <p class="usuario">' . $_SESSION["nombre"] . ' </p>
        <p class="usuario">' . $_SESSION['rol'] . ' </p>
        <a href="datos.php">Cambiar datos</a>

        <a href="logout.php">Cerrar sesión</a>
        ');
    if ($_SESSION['rol'] == 'admin') {
        echo ( ' <a href="pelis_config.php">Modificar películas</a> ');
    }
}
```

donde si la sesión ha sido iniciada se muestran los datos y si no se piden los datos para el formulario login.php:

```

$form = new FormularioLog($_POST);

require_once ('classes/usuario.class.inc');

$user = new User();

$usuariodb = $user->obtenerUsuario($form->getEmail());

if($form->getEmail() == $usuariodb['email'] && $form->getPassword() == $usuariodb['passwd']){

    session_start();

    $_SESSION['email'] = $usuariodb['email'];
    $_SESSION['nombre'] = $usuariodb['nombre'];
    $_SESSION['rol'] = $usuariodb['rol'];

    header("Location: index.php");
}
else{
    header("Location: index.php");
    echo('

        <script>
            window.onload = function() {
                alert("email o contraseña incorrectos");
            }
        </script>

    ');
}

exit;

```

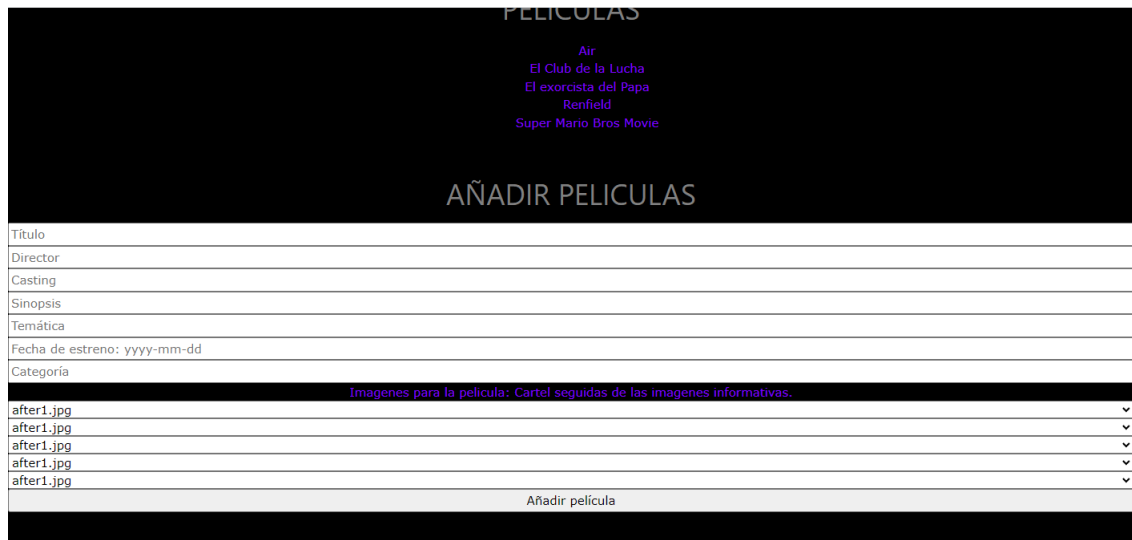
Como podemos ver obtiene el email y la contraseña introducidos por el usuario y consigue los datos de ese correo y/o contraseña para ver si son correctos, en caso erróneo se muestra una ventana de alerta con el mensaje visto en la imagen.

1.4. Mostrar películas dadas de alta

Para esto simplemente he obtenido todas las películas de la categoría deseada de la base de datos y se muestran en la página.

1.5. Crear nuevos filmes y modificarlos/eliminarlos

Como se ha visto en apartados anteriores si se está logueado se ve en la cabecera los datos del usuario más una sección donde cambiar sus datos y en solo en caso de ser “admin” en lugar de “user” una sección para modificar películas:



Aquí se verán las películas ya disponibles y un formulario para añadir una nueva. Solo se podrá añadir una película con el mismo título ya que fue seleccionada como clave primaria para la tabla películas en la base de datos. Se comprobará que la fecha esté en el formato correcto y además se meta en una de las categorías disponibles: próximas, cartelera, mas_valoradas y estrenos. Como aspecto innovador he usado unas listas desplegables dinámicas hechas con php donde se mostrarán las imágenes disponibles en el servidor:

```
<?php
$folderPath = 'img/';
$files = scandir($folderPath);
$files = array_diff($files, array('.', '..'));
?>
<p>Imágenes para la película: Cartel seguidas de las imágenes informativas.</p>
<select name="cartel">
  <?php foreach ($files as $file): ?>
    <option value="<?php echo $file; ?>"><?php echo $file; ?></option>
  <?php endforeach; ?>
</select>
<?php
$folderPath = 'img/';
$files = scandir($folderPath);
$files = array_diff($files, array('.', '..'));
?>
```

De esta manera también se evita que se pueda introducir una ruta no válida. En cuanto al php que se encarga del registro de una película es muy similar al del usuario.

En cuanto al archivo película.php sigue la siguiente estructura:

```
if (isset($_SESSION['email'])) {
    if ($_SESSION['rol'] == 'admin') {
        echo ('<a href=eliminar_peli.php?film=' . $dato . ' onclick="return confirmarAccion()">Eliminar</a>');
    }
}

$_SESSION['titulo'] = $peliculadb['titulo'];

echo ('

    <h1>' . $peliculadb['titulo'] . '</h1>

    <section class="imgs">
        <img src = ' . $peliculadb['img1'] . ' alt = "Imagen 1"/>
        <img src = ' . $peliculadb['img2'] . ' alt = "Imagen 2"/>
        <img src = ' . $peliculadb['img3'] . ' alt = "Imagen 3"/>
        <img src = ' . $peliculadb['img4'] . ' alt = "Imagen 4"/>
    </section>

    <section class="datos">
        <p>Título: ' . $peliculadb['titulo'] . '</p>
        <p>Directed by: ' . $peliculadb['directed'] . '</p>
        <p>Cast: ' . $peliculadb['cast'] . '</p>
        <p>Sinopsis: ' . $peliculadb['sinopsis'] . '</p>
        <p>Temática: ' . $peliculadb['tema'] . '</p>
        <p>Fecha de estreno: ' . $peliculadb['fecha'] . '</p>
        <p>Valoración: ' . $val_media . '/5</p>

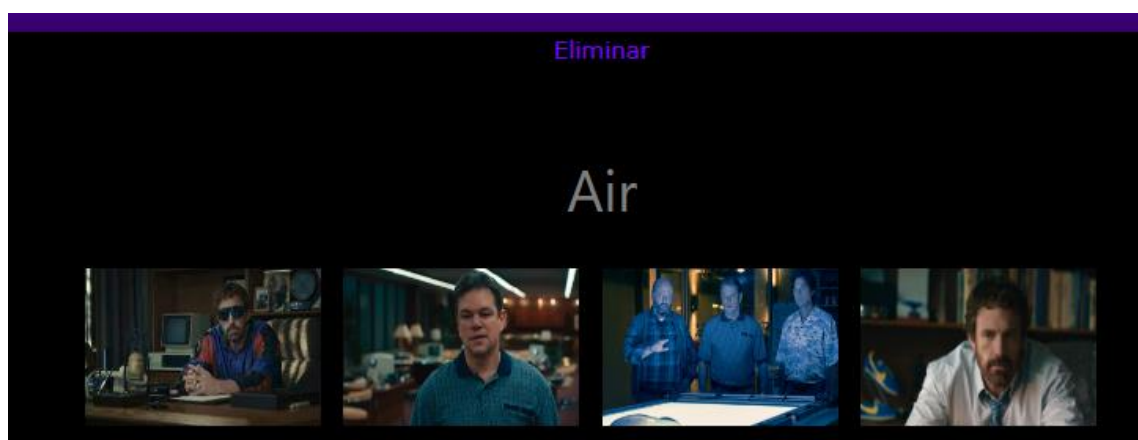
    ');
?>
```

Aquí vemos como se ven los datos de cualquier película con un solo archivo usando un método \$_GET en la etiqueta <a> codificando el título de la película cambiando los espacios por un '+':

```
$pelicula = new Pelicula();

$peliculadb = $pelicula->obtenerPelicula($_GET['film']);
$dato = str_replace(' ', '+', $peliculadb['titulo']);
```

En cuanto a la modificación y eliminación de películas serán dos secciones que solo aparecerán en caso de ser admin:



MODIFICAR PELICULAS	
Título	
Director	
Casting	
Sinopsis	
Temática	
Fecha de estreno: yyyy-mm-dd	
Categoría	
Imágenes para la película: Cartel seguidas de las imágenes informativas.	
Cartel: img/imagen.jpg	
Imagen 1: img/imagen.jpg	
Imagen 2: img/imagen.jpg	
Imagen3: img/imagen.jpg	
Imagen4: img/imagen.jpg	
Añadir película	

Para modificar película se hará de igual manera que para el usuario. Todos los datos que no se introduzcan serán ignorados. Para el caso de eliminarla antes de confirmar la acción aparecerá una ventana emergente que nos pida la confirmación de esta acción.

1.6. Comentarios y valoración

En esta sección veremos como he hecho que los usuarios puedan hacer comentarios y valorar con un formulario de estrellas.

Para ambos casos he creado una tabla comentarios y valoración. En estas se almacenan los datos respectivos a cada acción:

- Comentarios: id, usuario que ha hecho el comentario, película donde la ha hecho y fecha del comentario.
- Valoración: id, usuario que ha hecho la valoración, película donde la ha hecho y el valor entre 1 y 5.

Para los comentarios simplemente extraigo todos los comentarios de la base de datos según la película que se esté visualizando y se muestran y para la valoración antes de realizar una se comprueba que un usuario no haya hecho ya una valoración en esa película. En caso de haberla hecho se actualiza el valor, en caso contrario se introducen los datos.

1.7. Ventana emergente con onmouseover

Para esto he usado JavaScript en el cual se obtienen los elementos por id y el título que tengan en la cartelera añadiendo el texto a un <p> con position: absolute; para que se pueda ver por encima. El hecho de llamar a la función de JS con onmouseover es para que solo se active al pasar por encima con el cursor. Con onmouseout llamo a la otra función correspondiente para que elimine la ventana emergente.

```
function showPopup(title) {
    var popup = document.getElementById('popup');
    var popupTitle = document.getElementById('popupTitle');

    popupTitle.innerHTML = title;
    popup.style.display = 'block';
}

function hidePopup() {
    var popup = document.getElementById('popup');
    popup.style.display = 'none';
}
```