

## **Memoria de la práctica final**

En esta entrega final se han completado todas las funcionalidades restantes, es decir, se ha implementado el proceso de compra de entradas, incluyendo el elegir las butacas a reservar. A la vez que hemos mejorado algunas funcionalidades ya completadas en la entrega anterior.

- **Mejoras:**

- Ajustes en los CSS para que la página sea más vistosa.
- Mejoras de la estructura interna, aplicación de JS para mejorar el funcionamiento de las salas y la compra de entradas.

- **Innovaciones:**

- Gestión de compra de entradas completado, incluyendo un historial donde poder ver las entradas compradas y la posibilidad de descargárselas.
- Implementación de código javascript para los formularios y las butacas.
- La generación de un ticket descargable con las entradas compradas.

## **Información del Proyecto**

Parte de actividades:

La X es que ha hecho bastante en el apartado, la / es que ha hecho cosas

Módulos del parte de actividades	Carlos	Lucas	Óscar	Christina
Implementación de sesión	X			
Implementación de las compras	X	X	X	X
Uso de JS	X	/	X	/
Mejoras de CSS		X		/
Retoques, arreglos, documentación	X	X	X	X
Servidor				X

Para la prueba y evaluación del proyecto, te adjuntamos los usuarios existentes

Usuario administrador : admin@ucm.es adminpass

Usuario normal: [user@ucm.es](mailto:user@ucm.es) userpass

También se va a adjuntar un documento que explica el funcionamiento completo de la aplicación. Será una guía de uso que esperamos te ayude a poder comprobar el alcance de la aplicación.

## **Información de la estructura interna de nuestro proyecto**

### **Usuarios:**

#### **Lógica**

- **usuario.php**: aquí se crea la clase Usuario, cuyos métodos más relevantes son:
  - login inicia la sesión.
  - registrar crea un usuario.
  - esAdmin devuelve si ese usuario es administrador.
  - buscaUsuario busca un usuario en la BD a partir de su correo electrónico.
  - insertaUsuario introduce el nuevo usuario en la BD.
- **formularioLogin.php**: clase que hereda de formulario.php implementando sus dos métodos abstractos para mostrar y procesar el formulario de logueo.
- **formularioRegistro.php**: clase que hereda de formulario.php implementando sus dos métodos abstractos para mostrar y procesar el formulario de registro.

#### **Vista**

- **login.php**: página que muestra el formulario de logueo.
- **registro.php**: página que muestra el formulario de registro.
- **logout.php**: página que cierra la sesión del usuario y muestra por pantalla el gif más épico jamás visto en la faz de la Tierra.

### **Películas:**

#### **Lógica**

- **pelicula.php**: aquí se crea la clase Película, cuantos métodos más relevantes son:
  - crear da de alta una nueva película.
  - buscaPelicula busca una película existente en la BD a partir de su id.
  - insertarPelicula sube todos los datos de la nueva película en la BD.
  - actualizaPelicula reemplaza los datos de la película a partir de su id.
  - borrar elimina una película de la base de datos.
  - getPeliculas devuelve películas existentes en la BD para usar en la cartelera.

- **formularioPelicula.php**: clase que hereda de formulario.php implementando sus dos métodos abstractos para mostrar y procesar el formulario de las películas.

## Vista

- **aniadirPelicula.php**: página que muestra el formulario de las películas.
- **buscarPelicula.php**: página que muestra todas las películas contenidas en la BD y muestra una tabla con su nombre, y dos botones para modificarlas o borrarlas.
- **borrarPelicula.php**: página que elimina una película y redirige al usuario a la ventana de administración.
- **modificarPelicula.php**: página que muestra el formulario de las películas con los datos de la película a modificar.

## Sesiones:

### Lógica

- **sesion.php**: aquí se crea la clase Sesión, cuyos métodos más relevantes son:
  - crear da de alta una nueva película.
  - Busca busca una sesión existente en la BD a partir de su id.
  - insertarSesion sube todos los datos de la nueva sesión en la BD.
  - modifica reemplaza los datos de la sesion a partir de sus datos.
  - borrar elimina una sesión de la base de datos.
  - getSesiones devuelve películas existentes en la BD para usar en la cartelera.
- **formularioSesion.php**: clase que hereda de formulario.php implementando sus dos métodos abstractos para mostrar y procesar el formulario de las sesiones.

## Vista

- **aniadirSesion.php**: muestra el formulario para añadir una sesión, sin añadirla.
- **buscarSesion.php**: muestra un formulario para buscar una sesión, sin hacer una búsqueda real.
- **borrarSesion.php**: página que elimina una película y redirige al usuario a la ventana de administración.
- **modificarSesion.php**: página que muestra el formulario de las películas con los datos de la película a modificar.
- **consultaPelicula.php**: página que muestra la información detallada de una película.

## Salas:

### Lógica

- **salas.php:** aquí se crea la clase Sala, cuyos métodos más relevantes son:
  - crear dados los parámetros de una sala la crea y la inserta.
  - buscar busca una sala por su id.
  - modificar actualiza los datos de la sala y de las butacas de ser necesario.
  - borrar elimina una sala de la BD.
  - actualizaButacaAdmin altera el estado de una butaca entre “nulo” y “disponible”.
  - actualizaButacaUsuario altera el estado de una butaca entre “disponible” y “seleccionada”.
  - insertar sube la sala a la BD.
  - modificarButacas reasigna las butacas dado un cambio en el tamaño de la sala.
  - devolverAsiento retorna el estado de una butaca por su id.
- **formularioSala.php:** clase que hereda de formulario.php implementando sus dos métodos abstractos para mostrar y procesar el formulario de las salas.
- **Butaca.js:** procesa el cambio de estado de una butaca entre “nulo” y “disponible”.
- **ActualizarButacas.php:** se encarga de actualizar la base de datos según el usuario vaya realizando los clicks pertinentes a las butacas.

### Vista

- **aniadirSala.php:** página que muestra el formulario de la sala.
- **buscarSala.php:** página que muestra todas las salas contenidas en la BD y muestra una tabla con su nombre, y dos botones para modificarlas o borrarlas.
- **borrarSala.php:** página que elimina una sala y redirige al usuario a la ventana de administración.
- **modificarSala.php:** página que muestra el formulario de las salas con los datos de la película a modificar además de su disposición de butacas.

## Compras:

### Lógica

- **compra.php:** los métodos más relevantes son:
  - crear mira si el usuario tiene una compra pendiente, si no tiene ninguna, se crea una nueva.
  - buscarUsuario busca si el usuario tiene una compra pendiente.

- buscarPorIdYUsuario busca una compra existente en específico.
- buscar busca todas las compras realizadas por el usuario.
- eliminarButacasCaducadas elimina todas las compras pendientes con más de 5 minutos de antigüedad, deseleccionando las butacas y eliminando la compra del usuario.
- procesarCompra una vez realizada la compra, se cambian las butacas seleccionadas a ocupadas y se finaliza la compra (pendiente pasa a ser 0).
- **formularioCompra.php:** clase que hereda de formulario.php implementando sus dos métodos abstractos para mostrar y procesar el formulario de las compras.
- **seleccionButacas.js:** procesa el cambio de estado de una butaca entre “disponible” y “seleccionada”.
- **operacionesButacas.php:** crea una compra con la butaca seleccionada.

## Vista

- **procesarCompra.php:** página que muestra un botón de descargar entradas (generarPDF.php) y otro de volver al menú principal.
- **seleccionDeButacas.php:** muestra las butacas de la sala enumeradas por fila y columna. Verde significa disponible, amarillo seleccionada y rojo ocupada.
- **generarPDF.php:** fichero que genera un pdf con los datos de una compra.
- **historialCompras.php:** muestra un historial de todas las compras realizadas por el usuario, mostrando la película y la fecha y hora de cada compra. (y a lo mejor algo más)

## Otros scripts de lógica:

- **index.php:** página que muestra todas las películas de la base de datos.
- **config.php:** fichero que define las rutas del programa y las constantes de conexión a la BD, e inicia la sesión.
- **formulario.php:** declaración de la clase abstracta encargada de la gestión de los formularios.
- **formulario.js:** procesa la validación de los datos de un formulario
- **utils.php:** fichero que define funciones auxiliares del programa.
- **aplicacion.php:** clase que se encarga de la conexión y desconexión de la BD.

## Otros scripts de vista:

- **estilo.css, cabecera.php, pie.php y plantilla.php:** scripts que definen la vista general del programa.
- **administracion.php:** página que le muestra al usuario el menú de administración.
- **contacto.php:** página que muestra el formulario de contacto.
- **formularioContacto.php:** clase que hereda de formulario.php implementando sus dos métodos abstractos para mostrar y procesar el formulario de contacto.
- **politicaDeCookies.php:** página que muestra nuestra política de cookies.
- **politicaDePrivacidad.php:** página que muestra nuestra política de privacidad.

## Estructura de la Base de Datos

### Base de datos:

Nuestra base de datos consta de varias tablas.

- **Usuario:** almacena la información de los usuarios que van a realizar algún tipo de interacción con el cine, ya sea comprando entradas (si eres un usuario normal) o accediendo a la administración si aplica. La tabla guarda el id, el nombre, la edad, el email, la contraseña y el rol del usuario.
- **Compra:** almacena las compras hechas por los usuarios. Esta tabla contiene el id de la compra, el id del usuario, el id de la película, la fecha de proyección, la hora de inicio de la película, el número de entradas que el usuario ha comprado para esa película, y las butacas obtenidas.
- **Película:** este guarda la información detallada de las películas que hay en el cine. Sus atributos son: el id, el nombre de la película, el género, la edad recomendada para ver la película, la duración en minutos, la sinopsis, y las rutas del póster y el tráiler.
- **Sesión:** que guarda la información de las películas disponibles en ese momento en el cine. Contiene el nombre de las películas que hay en ese momento, la fecha en la que se proyectan, la hora de inicio de las películas, la sala en la que se van a proyectar y un campo de visibilidad, para determinar si ya están visibles para los usuarios normales o no.
- **Salas:** describe las distintas salas de cine y su disposición de asientos. Por cada sala almacena su id, su número de sala, el número de filas y columnas que tiene, y una estructura que detalla la disposición de los asientos.
- **Entrada:** es una tabla donde almacenamos las entradas que han sido compradas para una sesión en particular, permitiendo así la gestión de esas compras.

## Diagrama de la BD:

