



Memoria Proyecto de Fin de Ciclo DAW
Aplicación TaquillaWeb

José Alberto Fernández Conde

Memoria Proyecto de Fin de Ciclo DAW
Aplicación TaquillaWeb



Índice

1. Descripción del proyecto y ámbito de implantación

- 1.1. Introducción
- 1.2. Finalidad
- 1.3. Cambios futuros
- 1.4. Parte cliente realizada en TaquillaWeb
- 1.5. Parte servidor realizada en TaquillaWeb

2. Temporalización del proyecto y fases de desarrollo

- 2.1. Temporalización completa
- 2.2. Resumen de aprendizajes

3. Recursos de hardware y software utilizados

- 3.1. Directorio de trabajo y control de versiones
- 3.2. Creación del proyecto React JS taquillaweb_react
- 3.3. Servidor
- 3.4. Especificaciones del equipo de trabajo
- 3.5. Entorno de desarrollo

4. Arquitectura de software y de sistemas

- 4.1. Arquitectura de software
- 4.2. Arquitectura de sistemas

5. Descripción de datos

- 5.1. Explicación detallada del código en la parte servidor, endpoints 5 y 8
- 5.2. Explicación detallada del código en la parte cliente, componente ContainerGetPostCompra.js

Anexo 1. Descripción de la aplicación en funcionamiento

Anexo 2. Endpoints

Anexo 3. Breve descripción de la base de datos

1. Descripción del proyecto y ámbito de implantación

1.1. Introducción

Este Proyecto Final de Ciclo **TaquillaWeb** consiste en la realización del frontend para una aplicación web de venta de entradas para diferentes cines.

La aplicación se apoya en un backend previamente realizado con un servidor **Django** donde se resolvieron 10 endpoints que devolvían respuestas con información bastante detallada por lo que se tomó la decisión de realizar la parte cliente de la aplicación considerando que había trabajo suficiente para cumplir con las exigencias de un Proyecto Final de Ciclo.

Para el desarrollo del frontend solo se han modificado algunas respuestas del servidor con el fin de poder manipularlas más fácilmente en la parte cliente.

1.2. Finalidad

TaquillaWeb permite al usuario registrado inspeccionar la cartelera que ofrecen los cines que hacen uso de la aplicación y si lo desea puede comprar las entradas para la película y el cine elegidos.

Se permite el acceso libre a las páginas Presentación, Descripción general y Contacto. El acceso a las principales funcionalidades solo está permitido a los usuarios registrados.

El frontend necesario para la administración de la página, crear, actualizar y eliminar instancias en la base de datos, no se ha desarrollado en este proyecto.

1.3. Cambios futuros

El proyecto se considera como una primera versión y las principales mejoras previstas a corto y medio plazo son las siguientes entre otras posibles:

- En la página de Compra de entradas (ContainerGetPostCompra.js, Anexo 1 pág. 23) sería conveniente normalizar las opciones en el select de la sala, por ejemplo, en lugar de **Cines Mix - Sala id=14** poner **Cines Mix – Sala 1** y normalizar también el formato de la fecha en el select de la fecha de proyección, por ejemplo, en lugar de **20231127** que aparezca **27-11-2023**
- El desarrollo del proyecto ha llegado hasta la presentación del ticket de compra pero se requieren más pruebas de funcionamiento de los componentes ContainerGetPostCompra.js y ContainerPostRealizarCompra.js para solucionar errores.
- Creación de nuevos componentes funcionales que recibirán sus argumentos mediante props:
 - los formularios de login y de registro
 - la barra de sesión
 - el ticket de compra de entradas
- Desarrollo de endpoints y frontend para el usuario administrador encargado de crear, actualizar y eliminar instancias en las tablas de la base de datos.

1.4. Parte cliente realizada en TaquillaWeb

La parte cliente se desarrolla con **React JS** (ver Anexo 1).

Las páginas realizadas son las siguientes:

1.4.1. Páginas de acceso libre

- Presentación (Inicio)
- Descripción del código
- Registro nuevo usuario
- Abrir sesión

1.4.2. Páginas de acceso limitado a usuarios registrados

- Presentación (como usuario registrado)
- Página de Exploración con enlaces a las siguientes:
 - Todas las películas
 - Estrenos
 - Buscador
 - Películas por cine, con enlace a:
 - Compra de entradas
 - Compra realizada (ticket)
 - Cerrar sesión

1.5. Parte servidor realizada en TaquillaWeb (ver Anexos 2 y 3).

La aplicación hace uso de un servidor local **Django** en <http://localhost:8000>

1.5.1. Base de datos

En este servidor se encuentra la base de datos en MariaDB. La base de datos consta de 5 tablas:

- tusuarios
- tcines
- tsalas
- tpeliculas
- tentradas

1.5.2. Reseña de los endpoints

Funcionalidad | Nº endpoint y método | Endpoint |

Registro nuevo usuario | Endpoint 1 (POST /users) | <http://localhost:8000/users/>

Login de usuario | Endpoint 2 (POST /sessions) | <http://localhost:8000/sessions/>

Todos los cines | Endpoint 3 (GET /cines/) | <http://localhost:8000/cines/>

Todas las películas | Endpoint 4 (GET /peliculas/) | <http://localhost:8000/peliculas/>

Película por su id | Endpoint 5 (GET /peliculas/<id>) | <http://localhost:8000/peliculas/<id>>

Películas por cine | Endpoint 6 (GET /cines/<id>/peliculas) | <http://localhost:8000/cine/<id>/peliculas/>

Películas en estreno | Endpoint 7 (GET /peliculas/estrenos) | <http://localhost:8000/peliculas/estrenos/>

Comprar entradas | Endpoint 8 (POST /entradas/) | <http://localhost:8000/entradas/>

Buscador | Endpoint 9 (GET /buscar_pelicula/<cadena_solicitada>/) |

http://localhost:8000/buscar_pelicula/<cadena_solicitada>/

2. Temporalización del proyecto TaquillaWeb y fases de desarrollo

2.1. Temporalización completa

Se plantea una temporalización completa ideal desde la definición del proyecto hasta la puesta en marcha de **TaquillaWeb**.

Como ya se ha mencionado, el backend y base de datos estaban previamente resueltos así que las fases resueltas en este proyecto serían: la Fase 1, el punto 1 de la Fase 2 y el punto 1 de la Fase 3 en los plazos de tiempo que se indican.

Fase 1: Planificación (2 semanas)

- **1. Definición de Requisitos (Semana 1)**
 - Reuniones con los stakeholders para recopilar requisitos.
 - Identificación de características clave de la aplicación (venta de entradas, búsqueda de películas, pago en línea, etc.).
 - Definición de casos de uso y diagramas de flujo.
- **2. Planificación del Proyecto (Semana 2)**
 - Asignación de recursos y roles.
 - Creación de un plan detallado del proyecto.
 - Establecimiento de hitos y fechas límite.

Fase 2: Diseño (4 semanas)

- **1. Diseño de la Interfaz de Usuario (UI) y Experiencia de Usuario (UX) (Semana 3-4)**
 - Creación de wireframes y prototipos.
 - Obtención de retroalimentación de los stakeholders.
 - Ajuste de la interfaz de usuario y diseño de la experiencia.
- **2. Diseño de la Base de Datos (Semana 5)**
 - Diseño del esquema de la base de datos.
 - Selección de tecnologías de base de datos.

Fase 3: Desarrollo (8 semanas)

- **1. Desarrollo Frontend (Semana 6-8)**
 - Implementación de la interfaz de usuario.
 - Integración de elementos de diseño.
 - Pruebas unitarias del frontend.
- **2. Desarrollo Backend (Semana 9-11)**
 - Implementación de la lógica de negocio.
 - Integración con la base de datos.
 - Desarrollo de funcionalidades de venta de entradas.
- **3. Desarrollo de la Integración de Pagos (Semana 12)**
 - Implementación de pasarelas de pago.
 - Pruebas de transacciones en línea.

Fase 4: Pruebas (4 semanas)

- **1. Pruebas Unitarias (Semana 13)**
 - Verificación de cada componente de manera individual.
- **2. Pruebas de Integración (Semana 14)**
 - Verificación de la interacción entre componentes.
- **3. Pruebas de Aceptación del Usuario (UAT) (Semana 15-16)**
 - Involucramiento de usuarios para probar la aplicación.
 - Corrección de errores y mejoras según la retroalimentación.

Fase 5: Implementación y Puesta en Marcha (2 semanas)

- **1. Implementación en Entorno de Producción (Semana 17)**
 - Configuración del servidor de producción.
 - Despliegue de la aplicación.
- **2. Monitoreo y Ajustes Iniciales (Semana 18)**
 - Monitoreo de la aplicación en producción.
 - Corrección de problemas inesperados.

Fase 6: Post-Lanzamiento (Ongoing)

- **1. Soporte Continuo y Mejoras (Post-Lanzamiento)**
 - Atención a problemas emergentes.
 - Actualizaciones de seguridad y mejoras de características según feedback.

2.2. Resumen de aprendizajes

El diseño completo de backend y frontend de TaquillaWeb me ha permitido tener una visión global del funcionamiento de una aplicación web.

- En el backend he sido consciente de la importancia de un buen diseño de la base de datos y cómo puede ayudar a simplificar el código que se ejecuta en los endpoints.
- En el frontend, la introducción incorrecta de datos por parte del usuario o la ausencia de los mismos me ha permitido ver la cantidad y la importancia del código necesario para gestionar estos casos así como la comunicación que se debe establecer entre aplicación y usuario usando avisos o mensajes informativos.
- Me ha sido útil para seguir practicando el diseño web.
- Me ha permitido seguir mejorando mis conocimientos de React JS y JavaScript.

3. Recursos de hardware y software utilizados

3.1. Directorio de trabajo y control de versiones

Se utiliza Git como sistema de control de versiones y la plataforma GitHub para alojar los repositorios Git de forma remota.

Repositorio local **C:\Users\Alberto Fdez\PFC_DAW** clonado en un repositorio remoto en la dirección:

https://github.com/alberto22-23/PFC_DAW/tree/main

3.2. Creación del proyecto React JS taquillaweb_react

- Es necesaria la instalación de:

- **Node.js** en Windows, en este caso versión 16.13.1
- **npm**, en este caso versión 8.1.2

- Desde la consola Windows:

- Directorio de trabajo **C:\Users\Alberto Fdez\PFC_DAW**.

En esta dirección ejecuto el comando **npx create-react-app taquillaweb_react** de esta forma se crea el proyecto dentro de una nueva carpeta llamada **taquillaweb_react**

C:\Users\Alberto Fdez\PFC_DAW\taquillaweb_react

- También es necesario instalar:

- C:\Users\Alberto Fdez\PFC_DAW> taquillaweb_react> **npm install react-router-dom**
- C:\Users\Alberto Fdez\PFC_DAW> taquillaweb_react> **npm install axios**

- Arrancar el proyecto:

- C:\Users\Alberto Fdez\PFC_DAW>taquillaweb_react> **npm run start**
- Se abre la aplicación en el navegador en **localhost:3000**

Se utiliza el navegador **Microsoft Edge**

3.3. Servidor

En una máquina virtual **Debian** está instalado un servidor local **Django versión 4.1.3** en la dirección **http://localhost/0.0.0.0:8000** donde se gestionan las peticiones a los diferentes endpoints.

En este servidor se encuentra la base de datos en MariaDB **taquillawebbd.sql** obtenida a partir de la importación de **bdtaquillaweb.sql** original realizada usando **phpMyAdmin** en la máquina anfitriona.

3.4. Especificaciones del equipo de trabajo

El sistema operativo de trabajo es Windows 10 Home

Procesador: Intel(R) Core(TM) i7-3537U CPU @ 2.00GHz 2.50 Ghz

RAM: 6,00 GB (5,89 GB usable)

Tipo de sistema: Sistema operativo de 64 bits, procesador basado en x64

3.5. Entorno de desarrollo

Como entorno de desarrollo se ha usado **Microsoft Visual Studio Code**

4. Arquitectura de software y de sistemas

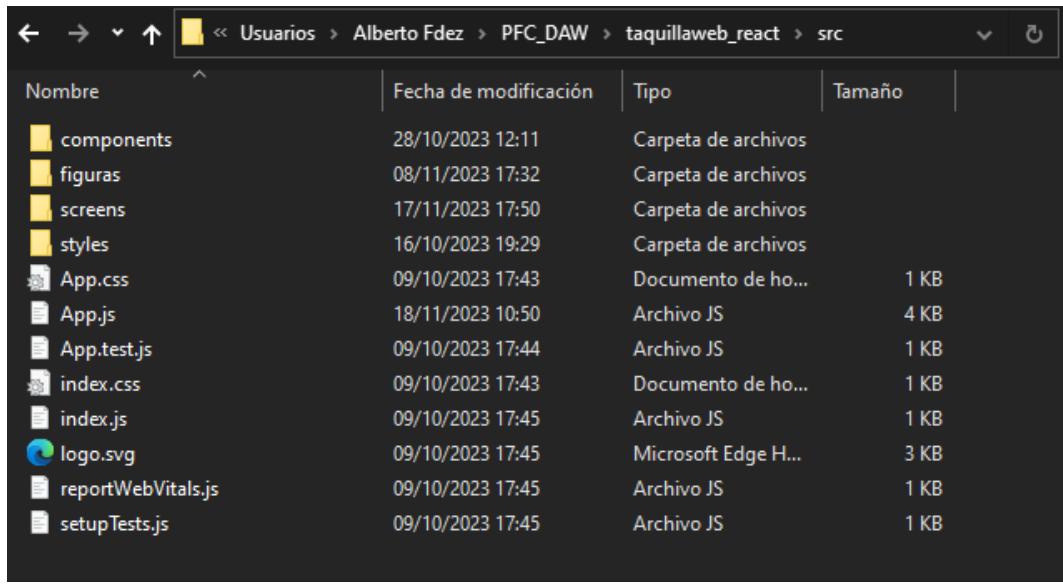
4.1. Arquitectura de software

En el directorio del proyecto **taquillaweb_react** están los archivos necesarios para que el frontend de la aplicación funcione.

4.1.1. Directorios del proyecto



Nombre	Fecha de modificación	Tipo	Tamaño
node_modules	17/10/2023 16:41	Carpeta de archivos	
public	09/10/2023 17:49	Carpeta de archivos	
src	09/10/2023 18:06	Carpeta de archivos	
.gitignore	09/10/2023 17:42	Documento de texto	1 KB
package.json	17/10/2023 16:41	Archivo JSON	1 KB
package-lock.json	17/10/2023 16:41	Archivo JSON	1.234 KB
README.md	09/10/2023 17:45	Archivo MD	4 KB



Nombre	Fecha de modificación	Tipo	Tamaño
components	28/10/2023 12:11	Carpeta de archivos	
figuras	08/11/2023 17:32	Carpeta de archivos	
screens	17/11/2023 17:50	Carpeta de archivos	
styles	16/10/2023 19:29	Carpeta de archivos	
App.css	09/10/2023 17:43	Documento de hoja de estilo	1 KB
App.js	18/11/2023 10:50	Archivo JS	4 KB
App.test.js	09/10/2023 17:44	Archivo JS	1 KB
index.css	09/10/2023 17:43	Documento de hoja de estilo	1 KB
index.js	09/10/2023 17:45	Archivo JS	1 KB
logo.svg	09/10/2023 17:45	Microsoft Edge HTML	3 KB
reportWebVitals.js	09/10/2023 17:45	Archivo JS	1 KB
setupTests.js	09/10/2023 17:45	Archivo JS	1 KB



Nombre	Fecha de modificación	Tipo	Tamaño
BarraSesion.js	28/10/2023 13:07	Archivo JS	1 KB
CardCine.js	28/10/2023 18:03	Archivo JS	2 KB
CardPelicula.js	25/10/2023 9:03	Archivo JS	2 KB
CardPeliculaPorCine.js	15/11/2023 21:15	Archivo JS	2 KB
Footer.js	04/11/2023 11:08	Archivo JS	3 KB
Header.js	14/11/2023 16:40	Archivo JS	1 KB

Nombre	Fecha de modificación	Tipo	Tamaño
ContainerCerrar.js	08/11/2023 17:00	Archivo JS	2 KB
ContainerGetBuscador.js	22/11/2023 16:58	Archivo JS	5 KB
ContainerGetCines.js	22/11/2023 15:50	Archivo JS	7 KB
ContainerGetEstrenos.js	30/10/2023 10:35	Archivo JS	3 KB
ContainerGetPelículas.js	31/10/2023 16:28	Archivo JS	3 KB
ContainerGetPelisCine.js	21/11/2023 9:15	Archivo JS	5 KB
ContainerGetPostCompra.js	26/11/2023 12:39	Archivo JS	17 KB
ContainerPostLogin.js	18/11/2023 11:28	Archivo JS	5 KB
ContainerPostRealizarCompra.js	20/11/2023 18:05	Archivo JS	11 KB
ContainerPostRegistro.js	14/11/2023 16:52	Archivo JS	6 KB
Descriptiva.js	22/11/2023 16:11	Archivo JS	8 KB
Home.js	22/11/2023 15:50	Archivo JS	5 KB
PageWithOutlet.js	27/05/2023 10:51	Archivo JS	1 KB

Nombre	Fecha	Tipo	Tamaño
esquema_relacional.jpg	03/11/2023 12:18	Imagen JPEG	51 KB
facebook.png	18/02/2023 9:59	Imagen PNG	12 KB
gorjeo.png	18/02/2023 10:00	Imagen PNG	10 KB
instagram.png	18/02/2023 10:02	Imagen PNG	17 KB
logoTW_1000x500.jpg	04/11/2023 9:46	Imagen JPEG	65 KB
patio_butacas_antiguo.jpg	07/11/2023 13:48	Imagen JPEG	158 KB
pinterest.png	18/02/2023 10:01	Imagen PNG	12 KB

Listado de directorios y componentes:

C:\Users\Alberto Fdez\PFC_DAW\taquillaweb_react\src

- App.js
 - Retorna el elemento **BrowserRouter** con las rutas de la aplicación y los respectivos componentes asociados.

C:\Users\Alberto Fdez\PFC_DAW\taquillaweb_react\src\components

<u>Componente</u>	<u>Datos que visualiza</u>
CardCine.js	Datos de un cine.
CardPelícula.js	Datos de una película.
CardPelículaPorCine.js	Datos de una película menos la sinopsis de la misma.
Footer.js	Pie de página, común a todas las páginas.
Header.js	Encabezado, común a todas las páginas.

- **ContainerCerrar.js**
 - Sesión cerrada**
 - Route path="/cerrar"

C:\Users\Alberto Fdez\PFC_DAW\taquillaweb_react\src\styles

- estilosTW.css Reglas de estilo de todos los componentes de la aplicación

C:\Users\Alberto Fdez\PFC_DAW\taquillaweb_react\src\figuras

Archivos de imágenes que no están almacenados en la base de datos importados por los componentes de la aplicación.

- | | |
|-----------------------------|--|
| • esquema_relacional.jpg | Esquema relacional de la base de datos |
| • facebook.jpg | Logo Facebook |
| • gorjeo.png | Logo Twitter |
| • instagram.png | Logo Instagram |
| • pinterest.png | Logo Pinterest |
| • logoTW_1000x500.jpg | Logo TaquillaWeb |
| • patio_butacas_antiguo.jpg | Imagen página sesión cerrada |

4.1.2. Descripción genérica del flujo de la aplicación

→ App.js contiene las diferentes rutas (Routes) de la aplicación en el servidor de React, todas muestran al componente **PageWithOutlet**

```

18 function App() {
19   return (
20     <div>
21       <BrowserRouter>
22         <Routes>
23           <Route path="/" element={<PageWithOutlet />}> {/*De esta forma aparece el footer y el header en todas las
24             páginas*/}
25             <Route index element={<Home />}></Route> /*ocupa el <Outlet >*/}
26             <Route path="/descriptiva" element={<Descriptiva />}></Route> /*ocupa el <Outlet >*/}
27             <Route path="/abrir" element={<ContainerPostLogin />}></Route> /*ocupa el <Outlet >*/}
28             <Route path="/cerrar" element={<ContainerCerrar />}></Route> /*ocupa el <Outlet >*/}
29             <Route path="/registro" element={<ContainerPostRegistro />}></Route> /*ocupa el <Outlet >*/}
30             <Route path="/cines-pelis-estrenos" element={<ContainerGetCines />}></Route> /*ocupa el <Outlet >*/}
31             <Route path="/peliculas" element={<ContainerGetPeliculas />}></Route> /*ocupa el <Outlet >*/}
32             <Route path="/estrenos" element={<ContainerGetEstrenos />}></Route> /*ocupa el <Outlet >*/}
33             <Route path="/buscador" element={<ContainerGetBuscador />}></Route> /*ocupa el <Outlet >*/}
34             <Route path="/cine/:parametro_id_cine/:parametro_nombre_cine" element={<ContainerGetPelisCine />}></Route> /*
35               *ocupa el <Outlet >*/}
36             <Route path="/compra/:parametro_id_cine/:parametro_id_pelicula/:parametro_nombre_cine" element=
37               {<ContainerGetPostCompra />}></Route> /*ocupa el <Outlet >*/}
38             <Route path="/realizar-compra/:param_id_cine/:param_id_pelicula/:param_id_sala/:param_num_entradas/
39               :param_sesion/:param_fecha/:param_num_tarjeta" element={<ContainerPostRealizarCompra />}></Route> /*ocupa el
40               <Outlet >*/}
41           </Route>{/*fin path="/"*/}
42         </Routes>
43       </BrowserRouter>
44   );
}
export default App;

```

Captura de App.js

→ El componente **PageWithOutlet.js** contiene a los componentes:

- **Header**
- **Outlet**
- **Footer**

```
src > screens > JS PageWithOutlet.js > [e] default
1  //ok
2  import { Outlet } from "react-router-dom";
3  import Footer from "../components/Footer";
4  import Header from "../components/Header";
5
6
7  function PageWithOutlet(properties) {
8
9    return (
10      <div>
11        |<Header />
12        |<Outlet />
13        |<Footer />
14      </div>
15    );
16  }
17
18  export default PageWithOutlet;
```

Captura de PageWithOutlet.js

Lo que varía de una ruta a otra es el componente **Outlet** cuya posición está ocupada por el respectivo componente **Route**.

Cada **Route** tiene las propiedades **path** y **element**,

- **element** es un componente **Container** (el cuerpo de la página)
- **path** es la ruta de ese componente dentro de la aplicación React.

→ **Header.js** es el Encabezado de las páginas de la aplicación y contiene los links a las páginas Descripción general y Contacto (no se ha desarrollado el código de la página Contacto).

→ **Footer.js** es el Pie de página en las páginas de la aplicación

→ **Características de los componentes <Container />:**

Solo se importan en **App.js** pues son los componentes a los que apuntan las Routes respectivas.

- En los <Container /> con método GET:

- 1) Se declaran las variable de estado

```
const [variableEst, setVariableEst] = useState(variableEstIni);
```

- 2) Dentro de una función **useEffect()**:

 2.1) se hace la petición GET a la API con **axios.get("http://localhost:8000/...")**

 2.2) dentro de una función Promesa se establecen los valores de las variables de estado asignándoles los datos que devuelve la API con **setVariableEst()**

- 3) Con **return** los <Container /> devuelven un elemento JSX <div>; que sería el cuerpo de la página conteniendo los componentes hijo <Card /> que visibilizan la información obtenida desde la API aplicando la función **.map()** sobre la variable de estado **variableEst.map()**, la variable de estado suele ser un array de objetos, por lo tanto **.map()** pasa mediante **props** cada objeto del array a un componente <Card /> que muestra los datos recibidos en la interfaz de usuario.

- En los <Container /> con método POST:

- 1) Se establecen las variables de estado que formarán parte de la petición al servidor a partir del contenido de los inputs presentes en el elemento JSX, p. ej.
`onChange={(event) => setEmail(event.target.value)}`
- 2) El envío mediante `axios.post("http://localhost:8000/...")` se realiza a través de una función manejadora de evento `onClick` de un botón o bien mediante una manejadora de evento `onSubmit` si los campos están dentro de un formulario

- En los <Container /> sin petición al servidor, por ejemplo ContainerCerrar.js, se mantiene la nomenclatura “Container” para que los elementos que ocupen posiciones similares en la estructura de la aplicación tengan una designación semejante.

→ Paso de valores entre componentes

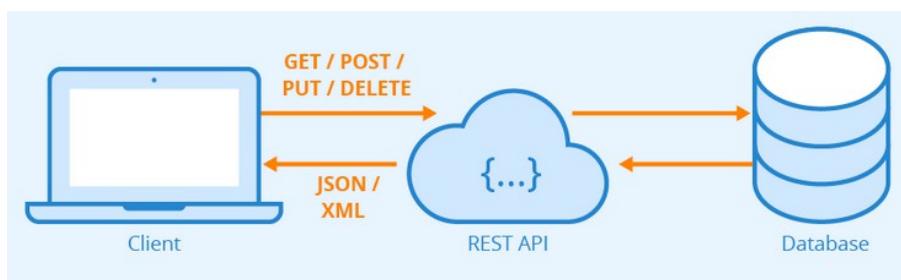
Los componentes funcionales hijo reciben sus argumentos desde el componente padre mediante `props`. Los componentes situados al mismo nivel como los <Container /> utilizan el hook `useParams` para compartir valores entre ellos.

4.2. Arquitectura de sistemas

TaquillaWeb se apoya en una arquitectura **API REST**, esta se entiende como una interfaz común entre frontend y backend. La implementación de la **API REST** se encuentra en el backend, mientras que el frontend utiliza esa interfaz para realizar solicitudes y recibir respuestas. Esta separación facilita la escalabilidad y el mantenimiento de la aplicación, ya que los cambios en el frontend o el backend pueden realizarse de manera independiente, siempre y cuando se respete el contrato definido por la **API REST**.

4.2.1. Definición general de API REST

- **API** (Application Programming Interface) "interfaz de programación de aplicaciones" es un conjunto de reglas que permiten a los programas informáticos interactuar entre sí. Una API es un intermediario entre dos sistemas, que permite que una aplicación se comunique con otra y pida datos o acciones específicas.
- **Protocolo REST** significa Transferencia de estado representacional (Representational State Transfer) y es un conjunto de principios arquitectónicos para diseñar sistemas distribuidos, especialmente en la web. Las aplicaciones que siguen estos principios se consideran "RESTful".



Esquema de API REST en la aplicación TaquillaWeb

4.2.2. Elementos clave de un API REST

- **Recursos:** Son las entidades o datos que se pueden manipular a través de la API.
- **Endpoint (Punto de Acceso):** Es la URL o ruta que se utiliza para acceder a un recurso específico.
- **Métodos HTTP:** Los métodos indican la acción que se desea realizar sobre un recurso. Los métodos más comunes son:
 - **GET:** Obtener datos.
 - **POST:** Crear nuevos datos.
 - **PUT o PATCH:** Actualizar datos existentes.
 - **DELETE:** Eliminar datos.
- **Representación de Recursos:** La representación de un recurso puede ser en formato JSON, XML, o algún otro formato. Es la manera en que los datos del recurso son presentados o enviados al cliente, en el caso de TaquillaWeb se usa el formato JSON
- **Estado del Cliente:** En una arquitectura REST, cada solicitud del cliente al servidor contiene toda la información necesaria para entender y procesar la solicitud. El servidor no guarda información sobre el estado del cliente entre solicitudes.
- **HATEOAS (Hypermedia As The Engine Of Application State):** Este es un principio adicional que sugiere incluir en las respuestas de la API enlaces (hipervínculos) que informan al cliente sobre qué acciones o recursos adicionales están disponibles.

Estos elementos proporcionan una estructura sencilla y eficiente para la comunicación entre clientes y servidores a través de la web.

4.2.3. TaquillaWeb sigue una arquitectura cliente-servidor.

La API REST actúa como el punto de comunicación entre el cliente (frontend) y el servidor (backend).

Elementos que definen el frontend:

- **Cliente (Frontend):** Este es el lado de la aplicación que los usuarios interactúan directamente. El frontend generalmente se ejecuta en el navegador del usuario. Utiliza la API REST para solicitar y enviar datos al servidor.
- **Consumir la API:** El frontend utiliza las rutas (endpoints) definidas en la API para realizar solicitudes al servidor.
- **Métodos HTTP en el Frontend:** En el frontend, se utilizan las mismas convenciones de métodos HTTP que en el backend. Por ejemplo, se puede usar JavaScript para realizar solicitudes GET, POST, PUT, o DELETE según la acción que el usuario desea realizar.
- **Formato de Datos:** El frontend interpreta la representación de los recursos que recibe del servidor. Esto puede ser en formato JSON u otro formato, dependiendo de cómo esté diseñada la API, en el caso de TaquillaWeb es JSON.
- **Actualización de la Interfaz de Usuario:** Despues de recibir los datos del servidor, el frontend actualiza la interfaz de usuario para reflejar la información recién obtenida.
- **Gestión de Estado:** Aunque el estado principal de la aplicación reside en el backend, el frontend también puede tener un estado local para gestionar la interactividad y la presentación de datos sin tener que realizar constantes solicitudes al servidor.

5. Descripción de datos

Como parte del código a explicar elijo la funcionalidad de compra de entradas en la que intervienen:

- En la parte servidor el código que procesa las peticiones a los endpoints intervinientes:
 - **Endpoint 5** (GET /peliculas/<id>): conseguir información de la película con el id introducido.
 - **Endpoint 8** (POST /entradas/): adquirir entradas.

El código se encuentra en el archivo **views.py** en el directorio del servidor Django:

taquillawebAPI/taquillawebapp/views.py

- En la parte cliente el componente:
 - **C:\Users\Alberto Fdez\PFC_DAW\taquillaweb_react\src\screens\ContainerGetPostCompra.js**

Para realizar la compra de entradas necesito crear la petición al servidor con los datos: **id cine** e **id película** más los datos introducidos por el usuario: **sesión, número de entradas, sala y fecha de proyección**, estos datos son gestionados en el archivo **ContainerGetPostCompra.js** que compone la petición de compra que se enviará al endpoint 8.

Si hay aforo suficiente disponible se realiza la petición, esto se lleva a cabo en el componente **ContainerRealizarCompra.js** que recibe mediante **pathParams** los datos necesarios y devuelve el ticket cumplimentado.

5.1. Explicación detallada del código en la parte servidor, endpoints 5 y 8

En principio las respuestas del servidor a los endpoints fueron pensadas ante posibles errores que se podrían producir en la introducción de datos mediante el comando **curl**, al desarrollar el frontend la mayor parte de estos errores ya no se producen pues los datos son gestionados por el código sin posible error humano.

→ **Endpoint 5 (GET /peliculas/<id>): conseguir información de la película con el id introducido**

Líneas 227-237: Autenticación del usuario.

```
226 #----- Endpoint 5 (GET /peliculas/<id>): conseguir información la pelicula con el id introducido.  
227 #-----  
228 @csrf_exempt  
229 def pelicula_por_id_view(request, id_solicitado):  
230     if request.method == 'GET':  
231         # Obtener el valor de la cabecera "Authorization"  
232         authorization_header = request.META.get('HTTP_AUTHORIZATION')  
233  
234         try: # si el token es diferente a los tokens que hay en la BD el usuario no está introduciendo el token  
235             # correcto  
236             usuario = Tusuarios.objects.get(token = authorization_header)  
237         except Tusuarios.DoesNotExist:  
238             return HttpResponseRedirect("La autenticación ha fallado.", status=400)
```

Una vez hecho el frontend con React el valor del token lo gestiona el código.

Línea 242: Obtenemos una lista de objetos, **salas_peli_solicitada**, las salas donde se proyecta la película con el id solicitado, pueden ser salas de diferentes cines.

Línea 244: Declaramos el array **datos_compra = []**

Línea 252: Creamos el objeto **sesionescines** con las sesiones y lo añadimos al array **datos_compra**

```
239     pelicula_solicitada = Tpeliculas.objects.get(id = id_solicitado) # consultamos la base de datos
240
241     #.....Modificación 20230502.....#
242     salas_peli_solicitada = Tsalas.objects.filter(id_pelicula = id_solicitado) # lista de objetos
243
244     datos_compra = [] # pueden ser salas de cines diferentes
245
246     sesionescines = {
247         #'Sesiones para todos los cines': [
248             "sesion_1": "16:30h - 18:30h", "sesion_2": "19:00h - 21:00h", "sesion_3": "21:30h - 23:30h"
249             #####corregido 20231031
250         #}
251     }
252
253     datos_compra.append(sesionescines)
```

Líneas 254-261: Para cada objeto **sala** en **salas_peli_solicitada** creamos un diccionario Python (un diccionario es similar a un JSON) con la id del cine que alberga esa sala, la id de la película que se proyecta en esa sala y la id de la propia sala

(El modelo **Tsalas** en **models.py** es la que nos proporciona la relación entre la id de una sala, su cine y su película)

Línea 263: Añadimos ese diccionario al array **datos_compra**

```
253
254     for sala in salas_peli_solicitada:
255         obj_cine = sala.id_cine
256         obj_pelicula = sala.id_pelicula
257
258         diccionario = {} # creamos un diccionario Python vacío
259         diccionario['id_cine'] = obj_cine.id
260         diccionario['id_pelicula'] = obj_pelicula.id
261         diccionario['id_sala'] = sala.id
262
263         datos_compra.append(diccionario)
264     #.....Fin Modificación 20230502.....#
```

Líneas 266-280: Componemos la respuesta del servidor, un objeto que contiene todos los datos necesarios para realizar la compra de entradas.

```
265
266     respuesta = { #####corregido 20231031
267         "datos_compra_entradas": datos_compra,
268         "#-----": "-----",
269         "datos_para_introducir": "id_cine, id_pelicula, id_sala y Sesión",
270         "datos_introducidos_en": "http://localhost:8000/entradas",
271         "#//////////": "//////////",
272         "id_pelicula": pelicula_solicitada.id,
273         "titulo_pelicula": pelicula_solicitada.titulo,
274         "estreno_pelicula": pelicula_solicitada.estreno,
275         "sinopsis_pelicula": pelicula_solicitada.sinopsis,
276         "precio_pelicula": pelicula_solicitada.peliculaprecio,
277         "cartel_pelicula": pelicula_solicitada.cartel
278     }
279
280     return JsonResponse(respuesta, safe = False, json_dumps_params = {'ensure_ascii':False})
281
282     # Si el método no es GET:
283     return HttpResponseBadRequest()
```

Ejemplo de respuesta:

```
{  
    "datos_compra_entradas": [  
        {  
            "sesion_1": "16:30h - 18:30h",  
            "sesion_2": "19:00h - 21:00h",  
            "sesion_3": "21:30h - 23:30h"  
        },  
        {  
            "id_cine": 3,  
            "id_pelicula": 10,  
            "id_sala": 13  
        },  
        {  
            "id_cine": 3,  
            "id_pelicula": 10,  
            "id_sala": 14  
        }  
    ],  
    "datos_para_introducir": "id_cine, id_pelicula, id_sala y Sesión",  
    "datos_introducidos_en": "http://localhost:8000/entradas",  
    "id_pelicula": 10,  
    "titulo_pelicula": "Memorias de África",  
    "estreno_pelicula": 1,  
    "sinopsis_pelicula": "A principios del siglo XX, Karen (Streep) contrae un matrimonio de conveniencia con el barón Blixen (Brandauer), un mujeriego empedernido. Ambos se establecen en Kenia con el propósito de explotar una plantación de café. En Karen Blixen nace un apasionado amor por la tierra y por las gentes de Kenia.",  
    "precio_pelicula": 8.5,  
    "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/cartel/memorias\_de\_africa.jpg"  
}
```

→ Endpoint 8 (POST /entradas): adquirir entradas

Líneas 383-392: Autenticación del usuario.

```
381 |     #----- Endpoint 8 (POST /entradas/): adquirir entradas. -----#  
382 | @csrf_exempt  
383 | def pedir_entradas_view(request):  
384 |     if request.method == 'POST':  
385 |         # Obtener el valor de la cabecera "Authorization"  
386 |         authorization_header = request.META.get('HTTP_AUTHORIZATION')  
387 |  
388 |         try: # si el token es diferente a los tokens que hay en la BD el usuario no está introduciendo el token correcto  
389 |             usuario = Tusuarios.objects.get(token = authorization_header)  
390 |  
391 |         except Tusuarios.DoesNotExist:  
392 |             return HttpResponse("La autenticación ha fallado.", status=400)  
393 |
```

Líneas 395-404: Recogemos el cuerpo de la petición y almacenamos los datos en variables.

```
394     # recogemos el cuerpo de la petición
395     json_request = json.loads(request.body)
396     # En Django, json.loads es una función que se utiliza para convertir una cadena JSON (JavaScript Object Notation) en un objeto de Python
397
398     pidcine = json_request['idcine'] # cogemos el id del cine
399     pidpelicula = json_request['idpelicula'] # cogemos el id de la película
400     pidsala = json_request['idsala'] # cogemos el id de la sala
401     pcantidadentradas = json_request['cantidadentradas'] # cogemos el número de entradas solicitadas, máximo 5
402     psesion = json_request['sesion'] # cogemos la sesión
403     pfecha = json_request['fecha_yearmonthday'] # cogemos la fecha de proyección
404     pnumero_tarjeta = json_request['numero_tarjeta']
405
```

→ Validación de los campos introducidos (al hacer el frontend ya no hay posibilidad de error en el valor)

- Validación de la id del cine.

Línea 407: Obtenemos objeto cine según la id introducida.

Línea 408: Obtenemos el número de salas de ese cine, es una propiedad del objeto.

Línea 412: Creamos una lista Python **lista_ids_cines** con las id de todos los cines.

Línea 418: Comprobamos que la id introducida aparece una vez en esa lista, si no es así se devuelve mensaje de error.

```
406     #####----- Validación de los campos introducidos -----#
407     cine = Tcines.objects.get(id = pidcine)
408     num_salas_cine = cine.cantidadsalas
409
410     #####----- Cine
411     pidcine_int = int(pidcine)
412     cines_todos = Tcines.objects.all()
413     lista_ids_cines = []
414     for objcine in cines_todos:
415         id_objcine = objcine.id
416         lista_ids_cines.append(id_objcine)
417
418     if lista_ids_cines.count(pidcine_int) == 1:
419         pass
420     else:
421         return HttpResponseRedirect("Error al introducir el id del cine.")
422
```

- Validación de la id de la película (al hacer el frontend ya no hay posibilidad de error en el valor).

Línea 425: Obtenemos las salas del cine que acabamos de validar

Líneas 426-428: De cada objeto sala de ese cine extraemos la película y de ella su id y la almacenamos en la lista **lista_ids_peliculas_cine**

Líneas 432-435: La cantidad de ids de películas de esa lista debe estar entre 1 y el número total de salas de ese cine si no es así se devuelve mensaje de error.

```
423     #####----- Película
424     pidpelicula_int = int(pidpelicula)
425     salas_cine = Tsalas.objects.filter(id_cine = pidcine_int)
426     lista_ids_peliculas_cine = []
427     for obj_sala_cine in salas_cine:
428         objpelicula = obj_sala_cine.id_pelicula
429         id_objpelicula = objpelicula.id
430         lista_ids_peliculas_cine.append(id_objpelicula)
431
432     if 1 <= lista_ids_peliculas_cine.count(pidpelicula_int) <= num_salas_cine:
433         pass
434     else:
435         return HttpResponseRedirect("Error al introducir el id del cine y/o el id de la película.")
```

- Validación de la sala (al hacer el frontend ya no hay posibilidad de error en el valor).

Sala o salas que están determinadas en la tabla Tsalas por el cine y la película anteriores, es decir pertenecen al cine elegido y proyectan la película elegida.

Línea 439: Filtramos en la tabla Tsalas los objetos cuyo valor id_cine coincide con la id del cine ya validado y cuyo campo id_pelicula coincide con la id de la película ya validada y los almacenamos en **salas_cineypelicula**

Líneas 440-443: Extraemos las id de esas salas y las almacenamos en **lista_ids_salas_cineypelicula**

Líneas 447-450: La cantidad de ids de salas de esa lista debe estar entre 1 y el número total de salas de ese cine si no es así se devuelve mensaje de error.

```
436
437     #####----- Sala o salas están determinadas en la tabla Tsalas por el cine y la película anteriores
438     pidsala_int = int(pidsala)
439     salas_cineypelicula = Tsalas.objects.filter(id_cine = pidcine_int, id_pelicula = pidpelicula_int)
440     lista_ids_salas_cineypelicula = []
441     for objsala in salas_cineypelicula:
442         id_objsala = objsala.id
443         lista_ids_salas_cineypelicula.append(id_objsala)
444
445     # cine = Tcines.objects.get(id = pidcine)
446     # num_salas_cine = cine.cantidadsalas
447     if 1 <= lista_ids_salas_cineypelicula.count(pidsala_int) < num_salas_cine:
448         pass
449     else:
450         return HttpResponse("Error al introducir el id de la sala.")
451
```

- Validación de la sesión (al hacer el frontend ya no hay posibilidad de error en el valor).

El valor de la sesión debe estar entre 1 y 3 incluidos si no es así se devuelve mensaje de error.

```
452     #####----- Sesión
453     psesion_int = int(psesion)
454     if 1 <= psesion_int <= 3:
455         pass
456     else:
457         return HttpResponse("Error al introducir el número de sesión.")
458
```

- Validación del número de entradas (al hacer el frontend ya no hay posibilidad de error en el valor).

El valor del número de entradas debe estar entre 1 y 5 incluidos si no es así se devuelve mensaje de error.

```
459     #####----- Cantidad entradas
460     pcantidadentradas_int = int(pcantidadentradas)
461     if 1 <= pcantidadentradas_int <= 5: # Corregido 20231119
462         pass
463     else:
464         return HttpResponse("Error al introducir el número de entradas (mín.=1, máx.=5).")
465
```

→ Creación del objeto nueva_entrada.

Líneas 468-469: Obtenemos los objetos cine y película elegidos.

Línea 470: la variable **fecha_sesion_id_sala** concatena el valor de la fecha introducida, la sesión introducida y la **id** de la sala introducida, combinación única para cada proyección de esta forma podremos obtener las butacas ocupadas en cada proyección haciendo la suma de los campos **entradacantbutacas** de todos los objetos **Tentrada** que tengan el común el campo **fecha**

Líneas 473-477: guardamos en variable el valor de la propiedad **peliculaprecio** de la película elegida y lo cambiamos a tipo float, lo multiplicamos por el número de entradas introducido.

Líneas 479-486: Creamos el objeto **nueva_entrada** tipo Tentradas y damos valor a sus propiedades.

Líneas 488-489: Creamos la fecha y la pasamos a tipo string.

```
466     # creación del objeto nueva_entrada
467     cine = Tcines.objects.get(id = pidcine)
468     pelicula = Tpeliculas.objects.get(id = pidpelicula)
469     fecha_sesion_idsala = pfecha + "_" + psesion + "_" + pidsala
470     # combinación única para cada proyección, podremos obtener las butacas ocupadas en cada proyección haciendo
471     # la suma
472     # de los campos "entradacantbutacas" de todos los objetos Tentrada que tengan en común el campo "fecha"
473     preciouunit = pelicula.peliculaprecio
474
475     pcantidadentradas_int = int(pcantidadentradas)
476     preciouunit_float = float(preciouunit)
477     preciotot = preciouunit_float * pcantidadentradas_int # precio total
478
479     nueva_entrada = Tentradas()
480     nueva_entrada.id_usuario = usuario # son objetos
481     nueva_entrada.id_cine = cine
482     nueva_entrada.id_pelicula = pelicula
483     nueva_entrada.fecha = fecha_sesion_idsala # ----- campo fecha de proyección con datos añadidos
484     nueva_entrada.entradapreciounitario = preciouunit_float
485     nueva_entrada.entradacantbutacas = pcantidadentradas_int
486     nueva_entrada.entradapreciototal = preciotot
487
488     fecha_actual = datetime.now()
489     fecha_actual_str = fecha_actual.strftime('%d-%m-%Y, %H:%M:%S')
490
```

→ Comprobación de que hay butacas libres suficientes para cubrir el número de entradas solicitado.

Línea 492: Cogemos el objeto sala con la id introducida.

Línea 493: Guardamos el valor de la propiedad “aforosal” en la variable aforo.

Línea 494: Obtenemos una lista de objetos entrada filtrados por el valor de la propiedad “fecha” (este valor “fecha” es único por proyección usando la fecha de proyección, la sesión y el id de la sala), guardamos la lista en la variable **entradas_fecha_sesion_id**

Líneas 496-499: Creamos la lista **listaocupadas** y guardamos en ella el valor del campo

entradacantbutacas (número de butacas compradas en cada entrada para esa proyección) de cada entrada ya almacenada en la base de datos.

Líneas 501-504: Hacemos la suma de los elementos de esa lista que nos da el número de butacas ocupadas hasta el momento para esa proyección, antes de sumar las butacas de la nueva entrada que aun no está almacenada en la base de datos.

De esta forma podemos obtener butacas libres y ocupadas según el aforo de la sala antes y después de ejecutar la compra de la nueva entrada.

```
491     # Comprobar que hay butacas libres suficientes
492     sala = Tsalas.objects.get(id = pidsala) # cogemos la sala con la id introducida
493     aforo = sala.aforosal
494     entradas_fecha_sesion_id = Tentradas.objects.filter(fecha = fecha_sesion_idsala) # es una lista de objetos
495     # entrada con el campo fecha = fecha+sesion+idsala introducidas
496
497     listaocupadas = []
498     for entrada in entradas_fecha_sesion_id:
499         ocupadas = entrada.entradacantbutacas
500         listaocupadas.append(ocupadas)
501
502     totalocupadas_anteriores = sum(listaocupadas)
503     restantes_anteriores = aforo - totalocupadas_anteriores
504     totalocupadas_despues = totalocupadas_anteriores + pcantidadentradas_int
505     restantes_despues = restantes_anteriores - pcantidadentradas_int
```

Líneas 506-508: Obtenemos el id de la sala y el número de sala para incluir en el ticket.

Líneas 510-528: Si el número de plazas ocupadas por la nueva entrada es menor o igual que el aforo no ocupado de la sala (aforo restante antes) almacenamos la nueva entrada en la base de datos y creamos el objeto respuesta con todos los datos que deben aparecer en el ticket.

Líneas 532-535: Si el número de plazas ocupadas por la nueva entrada es mayor que el aforo no ocupado de la sala (aforo restante antes) no se puede realizar la compra y se devuelve un aviso.

```
506     # Obtener id y número de sala para incluir en el ticket:
507     idsala = sala.id
508     numero_sala = sala.numerosala
509
510     if pcantidadadentradas_int <= restantes_anteriores:
511         nueva_entrada.save()
512         respuesta = { #####corregido 20231031
513             "mensaje_compra": "Compra realizada.",
514             "total_ocupadas": totalocupadas_despues,
515             "restantes": restantes_despues,
516             "identificador_ticket": nueva_entrada.id,
517             "cine": cine.cinenombre,
518             "titulo_pelicula": pelicula.titulo,
519             "fecha_proyeccion": pfecha,
520             "sesion": psesion,
521             "id_sala": idsala,
522             "numero_sala": numero_sala,
523             "numero_entradas": pcantidadadentradas,
524             "precio_unitario": preciounit,
525             "precio_total": preciotot,
526             "pago": "Se ha cargado el importe en su tarjeta.",
527             "fecha_compra": fecha_actual_str
528         }
529
530         return JsonResponse(respuesta, safe = False, json_dumps_params = {'ensure_ascii':False})
531         #return HttpResponse("Se ha guardado la entrada.", status=200)
532     else:
533         # pasamos restantes_anteriores a string
534         restantes_anteriores_str = str(restantes_anteriores)
535         return HttpResponse("No se pudo realizar la compra. Solo hay " + restantes_anteriores_str + " butacas disponibles.")
```

```
536
537     # Si el método no es POST:
538     return HttpResponseBadRequest()
539
```

Ejemplo de respuesta:

```
{ "mensaje_compra": "Compra realizada.", "total_ocupadas": 6, "restantes": 94, "identificador_ticket": 385, "cine": "Cines Mix", "titulo_pelicula": "Memorias de Africa", "fecha_proyeccion": "20231127", "sesion": "1", "id_sala": 14, "numero_sala": 6, "numero_entradas": "2", "precio_unitario": 8.5, "precio_total": 17.0, "pago": "Se ha cargado el importe en su tarjeta.", "fecha_compra": "25-11-2023, 10:47:59" }
```

5.2. Explicación detallada del código en la parte cliente, componente ContainerGetPostCompra.js

→ Captura del frontend

The screenshot shows a movie ticket booking interface for 'Memorias de África' at 'Cines Mix'. The movie poster for 'Out of Africa' is displayed on the left. The main form asks for card number input and includes a 'Comprobar datos' button and a check digit field. Navigation links like 'Descripción general' and 'Contacto' are at the top, and 'Volver a Cines' and 'Compra de entradas' are on the left.

Introduzca los campos necesarios.

Cines Mix

MEMORIAS DE AFRICA

A principios del siglo XX, Karen (Streep) contrae un matrimonio de conveniencia con el barón Blixen (Brandauer), un mujeriego empedernido. Ambos se establecen en Kenia con el propósito de explotar una plantación de café. En Karen Blixen nace un apasionado amor por la tierra y por las gentes de Kenia.

Precio unitario: 8.5 €

Elija sesión ▾ Número de entradas ▾
Elija sala ▾ Seleccione fecha ▾

Precio total: 0 €

Introduzca su número de tarjeta

0

.....

Comprobar datos

**

Comprobación datos: 3 - 10 - 0 - 0 - 0 - 0 - 0

Aviso legal

Configuración de cookie

Política de privacidad

Política de cookies

TW
TaquillaWeb

Autor: José Alberto Fernández Conde

¡Síguenos!

f t i p

→ Importaciones necesarias.

```
src > screens > JS ContainerGetPostCompra.js > ContainerGetPostCompra
1 import { ReactDOM } from 'react';
2 import React from 'react';
3 import axios from "axios";
4 import { useEffect, useState } from "react";
5 import { Link } from 'react-router-dom';
6 import { useParams } from 'react-router-dom';
7 //import { useHistory } from 'react-router-dom';
8
9 //Estilos
10 import "../styles/estilosTW.css";
11
```

→ Mediante pathParams llegan a esta página los valores de los parámetros **parametro_id_cine**, **parametro_id_pelicula** y **parametro_nombre_cine**

```
12 const ContainerGetPostCompra = () => {
13
14 //const history = useHistory();
15
16 // RECUPERACIÓN DE LOS PARÁMETROS
17 const { parametro_id_cine, parametro_id_pelicula, parametro_nombre_cine } = useParams("");
18 //Nos llega desde CardPeliculaPorCine, a través de la route definida en App.js. La id del cine no la voy a
19 utilizar pero la envío a ContainerPostRealizarCompra
20 console.log(parametro_id_cine, parametro_id_pelicula, parametro_nombre_cine);
21 const parametro_id_cine_number = Number(parametro_id_cine);
22 const parametro_id_pelicula_number = Number(parametro_id_pelicula);
23 *****/
```

→ Al ejecutarse useEffect se establecen los valores de las variables de estado siguientes:

```
24
25 //VARIABLES DE ESTADO ESTABLECIDAS CON useEffect()
26 const [pelicula_compra, setPeliculaCompra] = useState({}); //pelicula es un objeto
27 const [pelicula_compra_id, setPeliculaCompraId] = useState(0);
28 const [pelicula_compra_titulo, setPeliculaCompraTitulo] = useState("");
29 const [pelicula_compra_sinopsis, setPeliculaCompraSinop] = useState("");
30 const [pelicula_compra_estreno, setPeliculaCompraEstreno] = useState(0);
31 const [pelicula_compra_precio, setPeliculaCompraPrecio] = useState(0);
32 const [pelicula_compra_cartel, setPeliculaCompraCartel] = useState("");
33 const [sesion_1, setSesion1] = useState("");
34 const [sesion_2, setSesion2] = useState("");
35 const [sesion_3, setSesion3] = useState("");
36 const [pelicula_salas, setPeliculaSalas] = useState([]); //es un array vacío
37
```

→ Otras variables de estado: **linkoperable** hace visible el botón que direccióna a la ruta con pathParams **enlace_realizar_compra**, donde se ejecuta la compra de entradas y se genera el ticket cumplimentado con los datos devueltos por el servidor.

→ Las variables de estado que se establecen mediante la acción del usuario a través de elementos “select” o “input”, se ven a continuación:

```

38 //VARIABLES DE ESTADO ESTABLECIDAS CON LOS select
39 const [sesion, setSesion] = useState(0);
40 const [numero_entradas, setNumeroEntradas] = useState(0);
41 const [pelicula_sala, setPeliculaSala] = useState(0);
42 const [fecha_proyeccion, setFechaProyeccion] = useState(0);
43 const [num_tajeta, setNumTarjeta] = useState(0);
44
45 const [enlace_realizar_compra, setEnlaceRealizarCompra] = useState("");
46 const [linkoperable, setLinkOperable] = useState(false);
47 const [aviso, setAviso] = useState("");
48 /*****
49

```

→ El **token** se guarda en la base de datos, cada usuario tiene su token y es devuelto por el servidor al inicio de sesión quedando almacenado en sessionStorage para ser utilizado en todas las peticiones. Al cerrar sesión o al cerrar la pestaña del navegador en la que funciona la aplicación el token del usuario se elimina (se puede ver en el código de ContainerCerrar.js).

```

50 // CREACIÓN DE LA PETICIÓN AL SERVIDOR
51 // Guardamos el token en una variable
52 const tokenUsuActual = sessionStorage.getItem('token');
53
54 // Definimos los datos que queremos enviar en el cuerpo de la solicitud
55 const datos = {};
56
57 // Configuramos las cabeceras de la solicitud
58 const config = {
59   headers: {
60     'Authorization': tokenUsuActual,
61     'Accept': 'application/json'
62   }
63 };
64
65 *****/

```

→ Para la compra de entradas hacemos uso de los datos devueltos por dos endpoints, el endpoint 5 y endpoint 8 vistos anteriormente en la parte servidor, en esta página solo haremos uso del endpoint 5 que devuelve los datos necesarios para compra de entradas y los datos de la propia película.

→ Dentro de **useEffect** con **axios** hacemos la petición GET al endpoint 5.

```

66 //1º EFECTO PARA ESTABLECER LAS VARIABLES DE ESTADO, SE EJECUTA SOLO UNA VEZ, CUANDO SE MONTA EL COMPONENTE EN
67 //EN DOM
68 useEffect(() => {
69   const promesa = axios.get('http://localhost:8000/peliculas/' + parametro_id_pelicula + '/', config);
70   promesa.then((pelicula) => {
71     console.log("Se ha recibido la respuesta");
72     console.log(pelicula.data);           //devuelve la película, que es un objeto js
73     setPeliculaCompra(pelicula.data);
74     //setPeliculaCompraId()
75     setPeliculaCompraTitulo(pelicula.data["titulo_pelicula"]);
76     setPeliculaCompraSinop(pelicula.data["sinopsis_pelicula"]);
77     setPeliculaCompraEstreno(pelicula.data["estreno_pelicula"]);
78     setPeliculaCompraPrecio(pelicula.data["precio_pelicula"]);
79     setPeliculaCompraCartel(pelicula.data["cartel_pelicula"]);
80     //Datos para enviar en la petición al servidor para efectuar la compra: id_cine, id_pelicula, id_sala y
81     //Sesión:
82     //setPeliculaCompraId(pelicula.data["id_pelicula"]);
83     setPeliculaCompraId(pelicula_compra.id_pelicula);
84     setSesion1(pelicula.data["datos_compra_entradas"][0]["sesion_1"]);
85     setSesion2(pelicula.data["datos_compra_entradas"][0]["sesion_2"]);
86     setSesion3(pelicula.data["datos_compra_entradas"][0]["sesion_3"]);

```

→ Para obtener las salas que proyectan la película elegida en el cine elegido:

A partir de la devolución del endpoint que es semejante a:

```
{  
    "datos_compra_entradas": [  
        {  
            "sesion_1": "16:30h - 18:30h",  
            "sesion_2": "19:00h - 21:00h",  
            "sesion_3": "21:30h - 23:30h"  
        },  
        {  
            "id_cine": 3,  
            "id_pelicula": 10,  
            "id_sala": 13  
        },  
        {  
            "id_cine": 3,  
            "id_pelicula": 10,  
            "id_sala": 14  
        }  
    ],  
    "datos_para_introducir": "id_cine, id_pelicula, id_sala y Sesión",  
    "datos_introducidos_en": "http://localhost:8000/entradas",  
    "id_pelicula": 10,  
    "titulo_pelicula": "Memorias de Africa",  
    "estreno_pelicula": 1,  
    "sinopsis_pelicula": "A principios del siglo XX...  
    "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/cartel/memorias\_de\_africa.jpg"  
}
```

podemos obtener un array con las id de todas las salas que proyectan esta película.

```
87 //Obtención de las id de las salas que proyectan la película en este cine  
88 const data_sin_sesiones = pelicula.data["datos_compra_entradas"].slice(1);  
89 console.log(data_sin_sesiones);  
90 console.log(".....");  
91  
92     let array_id_salas = [];  
93     data_sin_sesiones.forEach((element) => {  
94         /*console.log(element.id_pelicula);  
95         console.log(element.id_cine);  
96         console.log(element.id_sala);  
97         console.log(parametro_id_cine_number);  
98         console.log(parametro_id_pelicula_number);  
99         console.log(".....");*/  
100         if (element.id_cine === parametro_id_cine_number && element.id_pelicula ===  
101             parametro_id_pelicula_number) {  
102             console.log(element.id_sala);  
103             console.log(".....");  
104             array_id_salas.push(element.id_sala);  
105         }  
106     });  
107     console.log(array_id_salas);  
108     setPeliculaSalas(array_id_salas);  
109     console.log(pelicula_salas);  
110 }, []); //HOOK DE EFECTO
```

```

112
113     /* Resumen:
114      Endpoint 5 (GET /peliculas/<id>): obtener información de la película con el ID introducido
115      "datos_para_introducir": "id_cine, id_pelicula, id_sala y Sesión", con esto creo la petición http de compra
116      -> obtengo array de las salas donde se proyecta la película seleccionada en el cine actual, si hay más de una
117      sala elijo con un select
118      -> obtengo sesiones, elijo con un select */
119
120  *****/

```

→ Variables de estado que se establecen mediante la acción del usuario a través de elementos “select”, son la sesión de la película, el número de entradas compradas, la sala donde se proyecta la película pues puede haber más de una sala proyectando la misma película, la fecha de proyección y se debe introducir el número de tarjeta en un elemento “input”.

```

120
121  // 2º ESTABLECIMIENTO DE LAS VARIABLES DE ESTADO A TRAVÉS DEL EVENTO onChange DE LOS SELECT
122  const establecerSesion = (event) => {
123      setSesion(event.target.value)
124  }
125
126  const establecerNumEntradas = (event) => {
127      setNumeroEntradas(event.target.value)
128  }
129
130  const establecerPeliculaSalaId = (event) => {
131      setPeliculaSala(event.target.value)
132  }
133
134  const establecerFechaProyec = (event) => {
135      setFechaProyeccion(event.target.value)
136  }
137
138  const establecerNumTarjeta = (event) => {
139      setNumTarjeta(event.target.value)
140  }
141
142  *****/

```

→ Obtenemos un array de 7 fechas desde el día siguiente al actual para colocarlas como opciones en el elemento select de la fecha de proyección.

```
143  /* Obtención de un array de 7 días, son los días de máxima antelación para comprar una entrada:  
144  - desde día siguiente al actual, p. ej. si fuese miércoles, desde el jueves  
145  - hasta miércoles (no incluido) +7 días, el miércoles de la semana siguiente */  
146  
147  
148  // Método para generar la fecha en formato 20230101  
149  function obtenerFormatoFecha(fecha) {  
150      const año = fecha.getFullYear();  
151      const mes = (fecha.getMonth() + 1).toString().padStart(2, '0'); //Rellenamos los espacios libres con 0  
152      const dia = fecha.getDate().toString().padStart(2, '0');  
153      //return `${año}${mes}${dia}`;  
154      return año + mes + dia;  
155  }  
156  
157  //Generar array de 7 fechas  
158  function generarArrayFechas() {  
159      const arrayFechas = [];  
160      const hoy = new Date();  
161  
162      // Avanzar al día siguiente  
163      hoy.setDate(hoy.getDate() + 1);  
164  
165      for (let i = 0; i < 7; i++) {  
166          const fechaFormateada = obtenerFormatoFecha(hoy); //Aplicamos el método anterior  
167          arrayFechas.push(fechaFormateada);  
168  
169          // Avanzar al siguiente día  
170          hoy.setDate(hoy.getDate() + 1);  
171      }  
172  
173      return arrayFechas;  
174  }  
.....  
175  
176  // Uso del array de fechas  
177  const fechas = generarArrayFechas();  
178  console.log(fechas);  
179  
180  /*****
```

→ Con la ejecución del método **comprobarDatos** comprobamos que todos los datos necesarios para almacenar la entrada en la base de datos y para crear el ticket se han introducido correctamente.

Si no hay errores se hace visible el link a la dirección **enlace_realizar_compra**, clicando el link vamos al componente **ContainerPostRealizarCompra.js** donde se ejecuta la compra y se muestra el ticket con los datos devueltos por el servidor.

Si faltan datos por introducir aparece mensaje de aviso.

```
182 const comprobarDatos = () => {
183   if (sesion === 0) {
184     setAviso("Debe elegir una sesión");
185   }
186   else if (numero_entradas === 0) {
187     setAviso("Debe seleccionar un número de entradas");
188   }
189   else if (pelicula_sala === 0) {
190     setAviso("Debe elegir una sala");
191   }
192   else if (fecha_proyeccion === 0) {
193     setAviso("Debe seleccionar una fecha de proyección");
194   }
195   else if (num_tajeta === 0) {
196     setAviso("Debe introducir un número de tarjeta");
197   }else{
198     setEnlaceRealizarCompra("/realizar-compra/" + parametro_id_cine_number + "/" +
199     parametro_id_pelicula_number + "/" + pelicula_sala + "/" + numero_entradas + "/" + sesion + "/" +
200     fecha_proyeccion + "/" + num_tajeta) // Ruta en la aplicación React
201     //history.push(enlace_realizar_compra);
202     setAviso("Ya puede comprar sus entradas");
203     setLinkOperable(true);
204   }
205   //window.location.href = enlace_realizar_compra;
206 }
207 /******
```

→ A partir de aquí hacemos la descripción del elemento JSX <div className='caja-volver-peliculas'> que devuelve el componente funcional ContainerGetPostCompra.

- Divs con nombre del cine e información de la película seleccionada.

```
208 return (
209   <div className='caja-volver-peliculas'>
210
211     <div className='caja-volver'>
212       <Link to="/cines-pelis-estrenos" className="enlace_enlace-a-pelis-estrenos">Volver a Cines</Link>
213       <h3 className='peliculas-en'>Compra de entradas</h3>
214     </div>
215
216     <div className="caja-compra-entrada">
217       <img src={pelicula_compra_cartel} alt="película" className="img-pelicula" />
218       <div className='datos-compra-entrada'>
219         <div className='div-precio-pelicula info-compra'>{/*pelicula_compra_id*/} Introduzca los campos
220         necesarios.</div>
221         <div className='div-nombre-pelicula info-compra'>{parametro_nombre_cine}</div>
222         <div className='div-nombre-pelicula info-compra'>{pelicula_compra_titulo.toUpperCase()}</div>
223         <div className='div-sinopsis-pelicula info-compra'>
224           <p>
225             {pelicula_compra_sinopsis}
226           </p>
227         </div>
228         <div className='div-precio-pelicula info-compra'>Precio unitario: {pelicula_compra_precio} €</
229         div>
230         {/*<div className='div-nombre-pelicula
231         info-compra'>.....</div>*/}
232       </div>
233     </div>
```

- Elementos JSX select para elegir la sesión y el número de entradas

```
229
230         <div>
231             <select className='select-sesiones info-compra' value={sesion} onChange={establecerSesion}>
232                 <option value={0}>Elija sesión</option>
233                 <option value={1}>1a: {sesion_1}</option>
234                 <option value={2}>2a: {sesion_2}</option>
235                 <option value={3}>3a: {sesion_3}</option>
236             </select>
237             <select className='select-sesiones info-compra' value={numero_entradas} onChange=
238                 {establecerNumEntradas}>
239                 <option value={0}>Número de entradas</option>
240                 <option value={1}>1</option>
241                 <option value={2}>2</option>
242                 <option value={3}>3</option>
243                 <option value={4}>4</option>
244                 <option value={5}>5</option>
245             </select>
```

- Elementos JSX select para elegir la sala y la fecha de proyección

```
244         <select className='select-sesiones info-compra' value={pelicula_sala} onChange=
245             {establecerPeliculaSalaId}>
246                 <option value={0}>Elija sala</option>
247                 {/*Código JS entre llaves*/}
248                 {pelicula_salas.map((idsala, i) => {
249                     return (
250                         <option value={idsala}>{parametro_nombre_cine} - Sala id={idsala}</option>
251                     )
252                 })}
253             </select>
254             <select className='select-sesiones info-compra' value={fecha_proyeccion} onChange=
255                 {establecerFechaProyec}>
256                 <option value={0}>Seleccione fecha</option>
257                 {/*Código JS entre llaves*/}
258                 {fechas.map((fecha, i) => {
259                     return (
260                         <option value={fecha}>{fecha}</option>
261                     )
262                 })
263             </select>
264         </div>
```

- Elementos JSX input para introducir el número de tarjeta donde se hará el cargo y elemento div donde aparecerán los mensajes de aviso

```
264         {/*<div className='div-nombre-pelicula
265             info-compra'>.....</div>*/}
266         <div className='div-precio-pelicula info-compra'>Precio total: {pelicula_compra_precio} *
267             numero_entradas</div>
268         {/*<div className='div-nombre-pelicula
269             info-compra'>.....</div>*/}
270
271         <label htmlFor="password" id="forpassword">Introduzca su número de tarjeta</label>
272         <input type="number" id="num-tarjeta" className="largo" value={num_tajeta} onChange=
273             {establecerNumTarjeta} />
274         <div className='div-nombre-pelicula info-compra'>.....</div>
275
276         <button className="enlace ancho97x100 info-compra" onClick={comprobarDatos}>Comprobar datos</button>
277
278         <div className='div-nombre-pelicula info-compra' style={{color: 'red'}}>- {aviso} -</div>
```

- Se añade un div auxiliar con el texto “Comprobación datos.” donde aparecen los datos a medida que se van introduciendo, este div no debe ser visible en la versión final de la aplicación.
- Elemento Link que se hace visible cuando todos los datos se han introducido correctamente y dirige a la página ContinerPostRealizarCompra.js donde se ejecuta la compra y se muestra el ticket cumplimentado con los datos que devuelve el servidor.

```

276     <div className='div-nombre-pelicula info-compra' style={{color: 'indigo', fontWeight: 'normal'}}>
277       >Comprobación datos: {parametro_id_cine_number} - {parametro_id_pelicula_number} - {sesion} -
278         {numero_entradas} - {pelicula_sala} - {num_tajeta} - {fecha_proyeccion}</div>
279     <Link to={enlace_realizar_compra} className="enlace ancho97x100 info-compra" style={{&
280       visibility: linkoperable ? 'visible' : 'hidden' }}>Comprar entradas</Link>
281   </div>
282 </div>
283 );
284 }
285
286 export default ContainerGetPostCompra;
287

```

Anexo 1. Descripción de la aplicación en funcionamiento

Descripción de la aplicación en funcionamiento

Presentación

[Descripción general](#)

[Contacto](#)

[Registro nuevo usuario](#)

[Abrir sesión](#)

Presentación.

Elige TaquillaWeb.com y disfruta del cine a lo grande.

Haz de **TaquillaWeb** tu aplicación favorita para la compra de entradas en tus cines preferidos. Aprovecha las ventajas de revisar la cartelera conjuntamente con independencia de la distribuidora.

Desde nuestro sitio web puedes acceder cómodamente a la oferta de varias cadenas de distribución con las últimas novedades de la cartelera.

Crea una cuenta de usuario en **TaquillaWeb.com** y aprovecha nuestras ofertas y promociones. Puedes comprar tus entradas con una antelación de hasta 7 días.

¡Te esperamos!

[Aviso legal](#)

[Configuración de cookie](#)

[Política de privacidad](#)

[Política de cookies](#)



Autor: José Alberto Fernández Conde

¡Síguenos!

Descripción general

Descripción general

Contacto

Registro nuevo usuario

Abra sesión con sus datos de usuario.

Abrir sesión

Descripción general de la aplicación.

1. Finalidad y funcionamiento de TaquillaWeb
Esta práctica final de ciclo consiste en la realización del frontend para la aplicación **TaquillaWeb**. El backend que le da soporte está previamente realizado y para la entrega de este ejercicio solo se han modificado algunas respuestas del servidor con el fin de permitir una mejor manipulación de las mismas en la parte cliente. El fin de **TaquillaWeb** es permitirle al usuario inspeccionar la cartelera que ofrecen los cines que hacen uso de la aplicación y, si lo desea, comprar las entradas para la película y el cine elegidos. Se permite el acceso libre a las páginas **Presentación** y **Descripción del código**. El acceso al resto de funcionalidades solo está permitido con el registro previo del usuario.

2. Parte cliente
La parte cliente de la aplicación está hecha con **React JS** y consta de las siguientes páginas:

2.1. Páginas de acceso libre:

- Presentación (Inicio)
- Descripción del código
- Registro nuevo usuario
- Abrir sesión
- Contacto

2.1. Páginas de acceso limitado a usuarios registrados:

1. Presentación (como usuario registrado)
2. Página de Exploración, con enlaces a las siguientes:
 - Todas las películas
 - Estrenos
 - Buscador
 - Películas por cine, con enlace a:
 - Compra de entradas
3. Cerrar sesión

3. Parte servidor
La aplicación hace uso de un servidor local **Django** en <http://localhost:8000> donde se gestionan las peticiones a los diferentes endpoints. En este servidor se encuentra la base de datos en MariaDB **taquillawebdb.sql** obtenida a partir de la importación de **bdtaquillaweb.sql** original, realizada usando phpMyAdmin

3.1. Base de datos
La base de datos consta de 5 tablas:

- **tusuarios**
- **tcines**
- **tsalas**
- **tpeliculas**
- **tentradas**

Esquema relacional de la base de datos.

```
graph TD; tusuarios[tusuarios] --> tcines[tcines]; tusuarios --> tentradas[tentradas]; tcines --> tsalas[tsalas]; tsalas --> tpeliculas[tpeliculas]; tpeliculas --> tentradas
```

3.2. Reseña de los endpoints
Funcionalidad	N° endpoint y tipo	Endpoint
Registro nuevo usuario	Endpoint 1 (POST /users)	<http://localhost:8000/users/>
Login de usuario	Endpoint 2 (POST /sessions)	<http://localhost:8000/sessions/>
Todos los cines	Endpoint 3 (GET /cines/)	<http://localhost:8000/cines/>
Todas las películas	Endpoint 4 (GET /peliculas/)	<http://localhost:8000/peliculas/>
Película por su id	Endpoint 5 (GET /peliculas/<id>)	[http://localhost:8000/peliculas/<id>/](http://localhost:8000/peliculas/<id>)
Películas por cine	Endpoint 6 (GET cines/<id>/peliculas)	<http://localhost:8000/cine/<id>/peliculas/>
Películas en estreno	Endpoint 7 (GET /peliculas/estrenos)	<http://localhost:8000/peliculas/estrenos/>
Comprar entradas	Endpoint 8 (POST /entradas/)	<http://localhost:8000/entradas/>
Buscador	Endpoint 9 (GET /buscar_pelicula/<cadena_solicitada>)	[http://localhost:8000/buscarr_pelicula/<cadena_solicitada>/](http://localhost:8000/buscarr_pelicula/<cadena_solicitada>)

Aviso legal
Configuración de cookie
Política de privacidad
Política de cookies

TW
TaquillaWeb

Autor: José Alberto Fernández Conde

Síguenos!

Anexo 1 – Descripción de la aplicación en funcionamiento

2

Inicio de sesión

[Descripción general](#)

[Contacto](#)

Iniciar sesión

Email

Contraseña

[Entrar](#)

[Aviso legal](#)

[Configuración de cookie](#)

[Política de privacidad](#)

[Política de cookies](#)


Autor: José Alberto Fernández Conde

¡Síguenos!

Inicio de sesión correcto

Descripción general

TW
TaquillaWeb

Contacto

Iniciar sesión

Email
jose@gmail.com

Contraseña

Entrar

Inicio de sesión correcto, status=201

Explorar

Aviso legal

Configuración de cookie

Política de privacidad

Política de cookies

¡Síguenos!

Facebook Twitter Instagram Pinterest

TW
TaquillaWeb

Autor: José Alberto Fernández Conde

Inicio de sesión con error en email

Descripción general

TW
TaquillaWeb

Contacto

Iniciar sesión

Email
ose@gmail.com

Contraseña

Entrar

El email no está registrado, status=400

Aviso legal

Configuración de cookie

Política de privacidad

Política de cookies

¡Síguenos!

Facebook Twitter Instagram Pinterest

TW
TaquillaWeb

Autor: José Alberto Fernández Conde

Inicio de sesión con error en contraseña

[Descripción general](#)

[Contacto](#)

Iniciar sesión

Email

Contraseña

[Entrar](#)

Error: Request failed with status code 401

[Aviso legal](#)

[Configuración de cookie](#)

[Política de privacidad](#)

[Política de cookies](#)


Autor: José Alberto Fernández Conde

¡Síguenos!

Registro nuevo usuario

[Descripción general](#)



[Contacto](#)

Registro nuevo usuario

Nombre de usuario

Email

Contraseña

Confirmar contraseña

[Registrar](#)

[Aviso legal](#)

[Configuración de cookie](#)

[Política de privacidad](#)

[Política de cookies](#)



Autor: José Alberto Fernández Conde

¡Síguenos!



Registro nuevo usuario, campos sin cubrir

Descripción general

TW
TaquillaWeb

Contacto

Registro nuevo usuario

Nombre de usuario

Email

Contraseña

Confirmar contraseña

Registrar

Hay campos sin cubrir, status=400

Aviso legal

Configuración de cookie

Política de privacidad

Política de cookies

Autor: José Alberto Fernández Conde

¡Síguenos!

f t i p

Registro nuevo usuario, error al confirmar contraseña

The screenshot shows a web application for 'TaquillaWeb' with a registration form. A modal dialog box is overlaid on the page, displaying the message: 'localhost:3000 dice Vuelva a introducir la contraseña.' (localhost:3000 says Enter your password again.) with a 'Aceptar' (Accept) button. The main form has fields for 'Nombre de usuario' (User name), 'Email', 'Contraseña' (Password), and 'Confirmar contraseña' (Confirm password). Below the form is a green 'Registrar' (Register) button. The footer includes links for 'Aviso legal', 'Configuración de cookie', 'Política de privacidad', and 'Política de cookies'. It also features the 'TaquillaWeb' logo and social media links for Facebook, Twitter, Instagram, and Pinterest.

Iniciar sesión en tu... Comenzar a usar Fir... Correo: J. A localhost:3000 dice Vuelva a introducir la contraseña. Aceptar Descripción general Contacto TaquillaWeb Registro nuevo usuario Nombre de usuario Usuario X Email xxx@gmail.com Contraseña Confirmar contraseña Registrar Aviso legal Configuración de cookie Política de privacidad Política de cookies TW TaquillaWeb Autor: José Alberto Fernández Conde ¡Síguenos! Facebook Twitter Instagram Pinterest

Creación de usuario correcta

Descripción general

TW
TaquillaWeb

Contacto

Registro nuevo usuario

Nombre de usuario

Email

Contraseña

Confirmar contraseña

Registrar

Link a Inicio sesión

Se ha creado el nuevo usuario, status=201, Usuario X

[Aviso legal](#)
[Configuración de cookie](#)
[Política de privacidad](#)
[Política de cookies](#)

Autor: José Alberto Fernández Conde

¡Síguenos!

Presentación con usuario registrado

[Descripción general](#)

[Contacto](#)

Sesión iniciada.

Cerrar sesión

Presentación.

¡Bienvenid@! ¡Explora TaquillaWeb.com y consigue tus entradas!

[Explorar](#)

Elige TaquillaWeb.com y disfruta del cine a lo grande.

Haz de **TaquillaWeb** tu aplicación favorita para la compra de entradas en tus cines preferidos. Aprovecha las ventajas de revisar la cartelera conjuntamente con independencia de la distribuidora.

Desde nuestro sitio web puedes acceder cómodamente a la oferta de varias cadenas de distribución con las últimas novedades de la cartelera.

Crea una cuenta de usuario en **TaquillaWeb.com** y aprovecha nuestras ofertas y promociones. Puedes comprar tus entradas con una antelación de hasta 7 días.

¡Te esperamos!

[Aviso legal](#)

[Configuración de cookie](#)

[Política de privacidad](#)

[Política de cookies](#)



Autor: José Alberto Fernández Conde

¡Síguenos!

[Descripción general](#)



[Contacto](#)

Sesión iniciada.

[Cerrar sesión](#)

Revisa la cartelera y compra las entradas.

Puedes examinar la cartelera de un vistazo, ver solo las películas en estreno o hacer una búsqueda personalizada.

.....
Después elige un cine y compra tus entradas.

[Ver todas las películas](#)

[Ver estrenos](#)

[Buscador](#)

CINES AUSTIN

Cines Austin

Número de salas: 3

[Ver Películas](#)

CINES TMA

Cines TMA

Número de salas: 5

[Ver Películas](#)

CINES MIX

Cines Mix

Número de salas: 6

[Ver Películas](#)

Sesiones para todos los cines:

1ª: 16:30h - 18:30h ··· 2ª: 19:00h - 21:00h ··· 3ª: 21:30h - 23:30h

[Aviso legal](#)

[Configuración de cookie](#)

[Política de privacidad](#)

[Política de cookies](#)



Autor: José Alberto Fernández Conde

¡Síguenos!



Todas las películas

[Descripción general](#)

TW
TaquillaWeb

[Contacto](#)

[Volver a Cines](#)

[Películas en cartelera](#)

AUSTIN POWERS

Primera aventura de Austin Powers (Mike Myers), un peculiar y atractivo espía de los años sesenta, cuyo principal enemigo es el doctor Maligno (interpretado también por Myers). Tras ser ambos sometidos a un proceso de congelación, se despiertan treinta años después en una sociedad completamente distinta a la que conocían. Sin embargo, ellos siguen siendo los mismos.

Precio: 6.50 €

EL GUATEQUE

Hrundi V. Bakshi es un patoso actor de origen hindú que se encuentra rodando una película en el desierto. Por sus continuas meteduras de pata, es despedido del rodaje. Inesperadamente, recibe una invitación para asistir a una sofisticada fiesta organizada por el productor de su última película. Gracias a Hrundi, en la fiesta se producirán las situaciones más disparatadas.

Precio: 6.50 €

LA VIDA DE BRIAN

Brian nace en un pesebre de Belén el mismo día que Jesucristo. Un cúmulo de desgraciados y tronchantes equivocos le harán llevar una vida paralela a la del verdadero Hijo de Dios. Sus pocas luces y el ambiente de decadencia y caos absoluto en que se haya sumergida la Galilea de aquellos días, le harán vivir en manos de su madre, de una feminista revolucionaria y del mismísimo Pencio Platino, su propia versión del calvario.

Precio: 8.50 €

BLADE RUNNER

Rick Deckard (Harrison Ford) es un blade runner, un agente de policía destinado al retiro de replicantes ilegales. Su misión es dar caza a un grupo de cuatro de estos androides, soldados del Nexus 6 superiores en fuerza e inteligencia a los humanos, pero diseñados para vivir una corta existencia de cuatro años.

Precio: 7.50 €

MATRIX

Thomas Anderson es un brillante programador de una respetable compañía de software. Pero fuera del trabajo es Neo, un hacker que un día recibe una misteriosa visita...

Precio: 7.50 €

TRON

Sam Flynn, un experto en tecnología de 27 años e hijo de Kevin Flynn, investiga la desaparición de su padre y se adentra en un mundo digital distinto al original y creado por su padre, de feroces programas y juegos de arcade, y donde su padre ha estado atrapado durante 20 años.

Precio: 8.50 €

2001 A SPACE ODYSSEY

Hace millones de años, antes de la aparición del "homo sapiens", unos primates descubren un monolito que los conduce a un estadio de inteligencia superior. Millones de años después, otro monolito, enterrado en una luna, despierta a Interés de los científicos. Por ello, durante una misión de la NASA, HAL 9000, una máquina dotada de inteligencia artificial, es encargada de controlar todos los sistemas de una nave espacial tripulada.

Precio: 7.50 €

POLTERGEIST

La película fue el primer gran éxito de Spielberg como productor. La trama gira en torno a los inquietantes sucesos que acontecen en la casa de una familia que vive en los suburbios, y en la que se sospecha que se está produciendo el fenómeno conocido como «poltergeist».

Precio: 7.50 €

REGRESO AL FUTURO

La cinta transcurre en el año 1985, una época en la que el joven Marty McFly lleva una existencia anómala con su novia Jennifer. Los únicos problemas son su familia en crisis y un director al que le encantaría expulsarse del instituto, por lo que deberá hacer todo lo posible en su mayoría para redimirse entre sus amigos y toda normalidad. Amigo del científico profesor Emmett Brown, una noche le acompaña a probar su nuevo experimento: viajar en el tiempo usando un DeLorean modificado...

Precio: 7.50 €

MEMORIAS DE ÁFRICA

A principios del siglo XX, Karen (Streep) contrae un matrimonio drá conveniencia con el barón Blixen (Brendauer), un mujerengo empedernido. Ambos se establecen en Kenia con el propósito de explotar una plantación de café. En Karen Blixen nace un apasionado amor por la tierra y por las gentes de Kenia.

Precio: 8.50 €

[Volver a Cines](#)

[Películas en cartelera](#)

[Aviso legal](#)

[Configuración de cookie](#)

[Política de privacidad](#)

[Política de cookies](#)

TW
TaquillaWeb

Autor: José Alberto Fernández Conde

¡Síguenos!

Anexo 1 – Descripción de la aplicación en funcionamiento

13

Estrenos

[Descripción general](#)



[Contacto](#)

[Volver a Cines](#)

[Estrenos](#)



LA VIDA DE BRIAN

Brian nace en un pesebre de Belén el mismo día que Jesucristo. Un cúmulo de desgraciados y tronchantes equívocos le harán llevar una vida paralela a la del verdadero Hijo de Dios. Sus pocas luces y el ambiente de decadencia y caos absoluto en que se haya sumergida la Galilea de aquellos días, le harán vivir en manos de su madre, de una feminista revolucionaria y del mismísimo Poncio Pilatos, su propia versión del calvario.

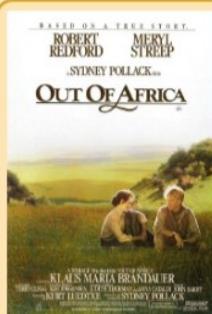
Precio: 8.50 €



TRON

Sam Flynn, un experto en tecnología de 27 años e hijo de Kevin Flynn, investiga la desaparición de su padre y se adentra en un mundo digital distinto al original y creado por su padre, de feroces programas y juegos de arcade, y donde su padre ha estado atrapado durante 20 años.

Precio: 8.50 €



MEMORIAS DE AFRICA

A principios del siglo XX, Karen (Streep) contrae un matrimonio de conveniencia con el barón Blixen (Brandauer), un mujeriego empedernido. Ambos se establecen en Kenia con el propósito de explotar una plantación de café. En Karen Blixen nace un apasionado amor por la tierra y por las gentes de Kenia.

Precio: 8.50 €

[Volver a Cines](#)

[Estrenos](#)

[Aviso legal](#)

[Configuración de cookie](#)

[Política de privacidad](#)

[Política de cookies](#)



Autor: José Alberto Fernández Conde

¡Síguenos!



Buscador

[Descripción general](#)  [Contacto](#)

[Volver a Cines](#) **Buscador de películas**

Introduzca letra, secuencia de letras o palabras incluidas en el título:

[Enviar](#)

[Aviso legal](#)
[Configuración de cookie](#)
[Política de privacidad](#)
[Política de cookies](#)


Autor: José Alberto Fernández Conde

¡Síguenos!
   

Búsqueda realizada

[Descripción general](#)

[Contacto](#)

[Volver a Cines](#)

Buscador de películas

Introduzca letra, secuencia de letras o palabras incluidas en el título:

[Enviar](#)

Se han encontrado 2 títulos con coincidencias.

MATRIX

Thomas Anderson es un brillante programador de una respetable compañía de software. Pero fuera del trabajo es Neo, un hacker que un día recibe una misteriosa visita...

Precio: 7.50 €

TRON

Sam Flynn, un experto en tecnología de 27 años e hijo de Kevin Flynn, investiga la desaparición de su padre y se adentra en un mundo digital distinto al original y creado por su padre, de feroces programas y juegos de arcade, y donde su padre ha estado atrapado durante 20 años.

Precio: 8.50 €

[Aviso legal](#)

[Configuración de cookie](#)

[Política de privacidad](#)

[Política de cookies](#)

TW
TaquillaWeb

Autor: José Alberto Fernández Conde

¡Síguenos!

Búsqueda sin resultado

Descripción general  Contacto

Volver a Cines Buscador de películas

Introduzca letra, secuencia de letras o palabras incluidas en el título:

Enviar

No se han encontrado títulos coincidentes. Vuelva a intentarlo.

Aviso legal Configuración de cookie Política de privacidad Política de cookies


Autor: José Alberto Fernández Conde

¡Síguenos!

Caracteres de búsqueda no introducidos

Descripción general  Contacto

Volver a Cines Buscador de películas

Introduzca letra, secuencia de letras o palabras incluidas en el título:

Enviar

Introduzca los caracteres a buscar.

Aviso legal Configuración de cookie Política de privacidad Política de cookies


Autor: José Alberto Fernández Conde

¡Síguenos!

Ver películas de un cine

- Cines Austin

[Descripción general](#)

[Contacto](#)

[Volver a Cines](#)

Películas en Cines Austin

AUSTIN POWERS

Estreno: No

Precio: 6.50 €

[Comprar entradas](#)

EL GUATEQUE

Estreno: No

Precio: 6.50 €

[Comprar entradas](#)

LA VIDA DE BRIAN

Estreno: Sí

Precio: 8.50 €

[Comprar entradas](#)

[Volver a Cines](#)

Películas en Cines Austin

[Aviso legal](#)

[Configuración de cookie](#)

[Política de privacidad](#)

[Política de cookies](#)

Autor: José Alberto Fernández Conde

¡Síguenos!

**Tw
TaquillaWeb**

[Descripción general](#) [Contacto](#)

[Volver a Cines](#) **Películas en Cines TMA**



BLADE RUNNER

Estreno: No

Precio: 7.50 €

[Comprar entradas](#)



MATRIX

Estreno: No

Precio: 7.50 €

[Comprar entradas](#)



TRON

Estreno: Sí

Precio: 8.50 €

[Comprar entradas](#)



2001 A SPACE ODYSSEY

Estreno: No

Precio: 7.50 €

[Comprar entradas](#)

[Volver a Cines](#) **Películas en Cines TMA**

[Aviso legal](#)

[Configuración de cookie](#)

[Política de privacidad](#)

[Política de cookies](#)



Autor: José Alberto Fernández Conde

¡Síguenos!

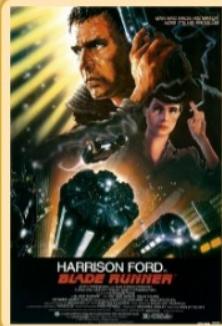


[Descripción general](#)

[Contacto](#)

[Volver a Cines](#)

Películas en Cines Mix



BLADE RUNNER

Estreno: No

Precio: 7.50 €

[Comprar entradas](#)

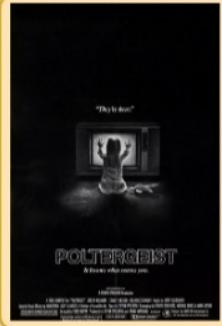


LA VIDA DE BRIAN

Estreno: Sí

Precio: 8.50 €

[Comprar entradas](#)



POLTERGEIST

Estreno: No

Precio: 7.50 €

[Comprar entradas](#)

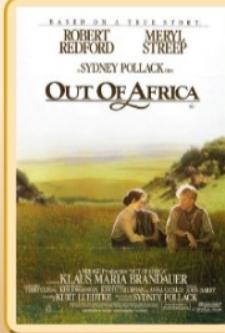


REGRESO AL FUTURO

Estreno: No

Precio: 7.50 €

[Comprar entradas](#)



MEMORIAS DE ÁFRICA

Estreno: Sí

Precio: 8.50 €

[Comprar entradas](#)

[Volver a Cines](#)

Películas en Cines Mix

[Aviso legal](#)

[Configuración de cookie](#)

[Política de privacidad](#)

[Política de cookies](#)



Autor: José Alberto Fernández Conde

¡Síguenos!



Comprar entradas

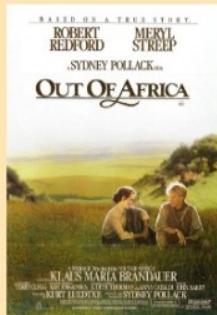
[Descripción general](#)



[Contacto](#)

[Volver a Cines](#)

[Compra de entradas](#)



Introduzca los campos necesarios.

Cines Mix

MEMORIAS DE AFRICA

A principios del siglo XX, Karen (Streep) contrae un matrimonio de conveniencia con el barón Blixen (Brandauer), un mujerengo empoderado.

Ambos se establecen en Kenia con el propósito de explotar una plantación de café. En Karen Blixen nace un apasionado amor por la tierra y por las gentes de Kenia.

Precio unitario: 8.5 €

Elija sesión ▾ Número de entradas ▾

Elija sala ▾ Seleccione fecha ▾

Precio total: 0 €

Introduzca su número de tarjeta

0

.....

Comprobar datos

Comprobación datos: 3 - 10 - 0 - 0 - 0 - 0 - 0

[Aviso legal](#)

[Configuración de cookie](#)

[Política de privacidad](#)

[Política de cookies](#)



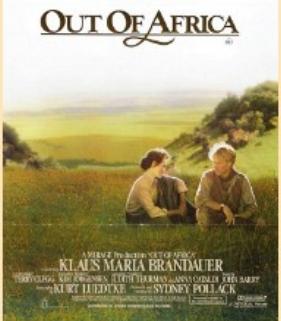
Autor: José Alberto Fernández Conde

¡Síguenos!



Introduciendo datos

- Sesión



Precio unitario: 8.5 €

Elija sesión ▾ Número de entradas ▾

Seleccione fecha ▾

Introduzca su número de tarjeta

total: 0 €

0

Comprobar datos

Comprobación datos: 3 - 10 - 0 - 0 - 0 - 0 - 0 - 0

- Número de entradas



Precio unitario: 8.5 €

1ª: 16:30h - 18:30h ▾ Número de entradas ▾

Elija sala ▾

Precio total:

Introduzca su número de tarjeta

Número de entradas

1
2
3
4
5

0

Comprobar datos

Comprobación datos: 3 - 10 - 1 - 0 - 0 - 0 - 0 - 0

- Sala

 **Precio unitario: 8.5 €**

1ª: 16:30h - 18:30h ▼ 2 ▼
Elija sala ▼ Seleccione fecha ▼

Elija sala
Cines Mix - Sala id=13
Cines Mix - Sala id=14 **Total: 17 €**
Número de tarjeta
0

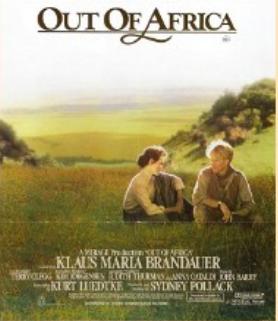
.....

Comprobar datos

.....

Comprobación datos: 3 - 10 - 1 - 2 - 0 - 0 - 0

- Fecha de proyección

 **Precio unitario: 8.5 €**

1ª: 16:30h - 18:30h ▼ 2 ▼
Cines Mix - Sala id=14 ▼ Seleccione fecha ▼

Precio total:
Introduzca su número
0

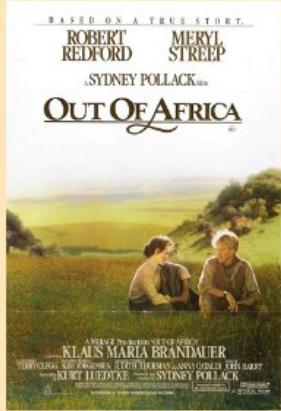
.....

Comprobar

.....

Comprobación datos: 3 - 10 - 1 - 2 - 14 - 0 - 0

- Todos los datos introducidos



Introduzca los campos necesarios.

Cines Mix

MEMORIAS DE AFRICA

A principios del siglo XX, Karen (Streep) contrae un matrimonio de conveniencia con el barón Blixen (Brandauer), un mujerengo empedernido. Ambos se establecen en Kenia con el propósito de explotar una plantación de café. En Karen Blixen nace un apasionado amor por la tierra y por las gentes de Kenia.

Precio unitario: 8.5 €

1ª: 16:30h - 18:30h ▼ 2 ▼
Cines Mix - Sala id=14 ▼ 20231127 ▼

Precio total: 17 €

Introduzca su número de tarjeta

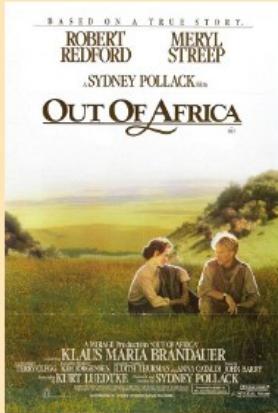
123456

Comprobar datos

..

Comprobación datos: 3 - 10 - 1 - 2 - 14 - 123456 - 20231127

Comprobar datos



Introduzca los campos necesarios.

Cines Mix

MEMORIAS DE AFRICA

A principios del siglo XX, Karen (Streep) contrae un matrimonio de conveniencia con el barón Blixen (Brandauer), un mujeriego empedernido. Ambos se establecen en Kenia con el propósito de explotar una plantación de café. En Karen Blixen nace un apasionado amor por la tierra y por las gentes de Kenia.

Precio unitario: 8.5 €

1ª: 16:30h - 18:30h 2
Cines Mix - Sala id=14 20231127

Precio total: 17 €

Introduzca su número de tarjeta

123456

.....

- Ya puede comprar sus entradas -

Comprobación datos: 3 - 10 - 1 - 2 - 14 - 123456 - 20231127

Comprar entradas → Compra realizada

[Descripción general](#)

[Contacto](#)

[Volver a Cines](#)

Ticket de compra

----- Compra realizada. -----

Ticket TaquillaWeb.com
Cines Mix

Fecha de compra: **22-11-2023, 16:32:18**
Número de ticket: **384**

Título: **Memorias de África**
id sala: **14**
Número de sala: **6**
Sesión: **1ª Sesión**
Fecha proyección: **20231127**

(Sesiones:
1ª 16:30h - 2ª 19:00h - 3ª 21:30h)

Precio unitario: **8.5 €**
Entradas adquiridas: **2**
Precio Total: 17 € IVA incluido

Se ha cargado el importe en su tarjeta.

[Imprimir ticket](#)

En esta proyección:
Total butacas ocupadas: 4 - Total butacas restantes: 96

Gracias por usar TaquillaWeb.com

[Aviso legal](#)
[Configuración de cookie](#)
[Política de privacidad](#)
[Política de cookies](#)


Autor: José Alberto Fernández Conde

¡Síguenos!

Detalle del ticket

----- Compra realizada. -----

Ticket TaquillaWeb.com
Cines Mix

Fecha de compra: **22-11-2023, 16:32:18**
Número de ticket: **384**

Título: **Memorias de África**
id sala: 14
Número de sala: **6**
Sesión: **1ª Sesión**
Fecha proyección: **20231127**

(Sesiones:
1ª 16:30h - 2ª 19:00h - 3ª 21:30h)

Precio unitario: **8.5 €**
Entradas adquiridas: **2**
Precio Total: 17 € IVA incluido

Se ha cargado el importe en su tarjeta.

Imprimir ticket

En esta proyección:
Total butacas ocupadas: 4 - Total butacas restantes: 96

Gracias por usar TaquillaWeb.com

Avisos en la compra de entradas si faltan campos por introducir

- Ningún campo introducido



Precio unitario: 8.5 €

Elija sesión ▾ Número de entradas ▾

Elija sala ▾ Seleccione fecha ▾

Precio total: 0 €

Introduzca su número de tarjeta

0

.....

Comprobar datos

- Debe elegir una sesión -

Comprobación datos: 3 - 10 - 0 - 0 - 0 - 0 - 0

- Introducida solo la sesión



Precio unitario: 8.5 €

2ª: 19:00h - 21:00h ▾ Número de entradas ▾

Elija sala ▾ Seleccione fecha ▾

Precio total: 0 €

Introduzca su número de tarjeta

0

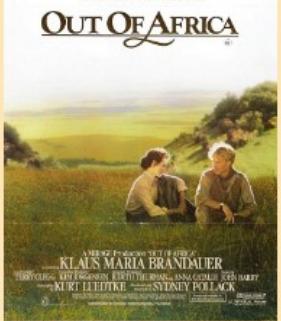
.....

Comprobar datos

- Debe seleccionar un número de entradas -

Comprobación datos: 3 - 10 - 2 - 0 - 0 - 0 - 0

- Introducida sesión y número de entradas



Precio unitario: 8.5 €

2ª: 19:00h - 21:00h ▼ 3 ▼
Elija sala ▼ Seleccione fecha ▼

Precio total: 25.5 €

Introduzca su número de tarjeta

0

.....

Comprobar datos

- Debe elegir una sala -

Comprobación datos: 3 - 10 - 2 - 3 - 0 - 0 - 0

- Introducida sesión, número de entradas y sala



Precio unitario: 8.5 €

2ª: 19:00h - 21:00h ▼ 3 ▼
Cines Mix - Sala id=13 ▼ Seleccione fecha ▼

Precio total: 25.5 €

Introduzca su número de tarjeta

0

.....

Comprobar datos

- Debe seleccionar una fecha de proyección -

Comprobación datos: 3 - 10 - 2 - 3 - 13 - 0 - 0

- Introducida sesión, número de entradas, sala y fecha de proyección

 OUT OF AFRICA

Precio unitario: 8.5 €

2ª: 19:00h - 21:00h



3



Cines Mix - Sala id=13



20231126



Precio total: 25.5 €

Introduzca su número de tarjeta

0

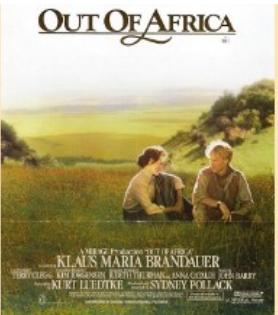
.....

Comprobar datos

- Debe introducir un número de tarjeta -

Comprobación datos: 3 - 10 - 2 - 3 - 13 - 0 - 20231126

- Todos los datos introducidos

 OUT OF AFRICA

Precio unitario: 8.5 €

2ª: 19:00h - 21:00h



3



Cines Mix - Sala id=13



20231126



Precio total: 25.5 €

Introduzca su número de tarjeta

654987

.....

Comprobar datos

- Ya puede comprar sus entradas -

Comprobación datos: 3 - 10 - 2 - 3 - 13 - 654987 - 20231126

Comprar entradas

Cerrar sesión

[Descripción general](#)

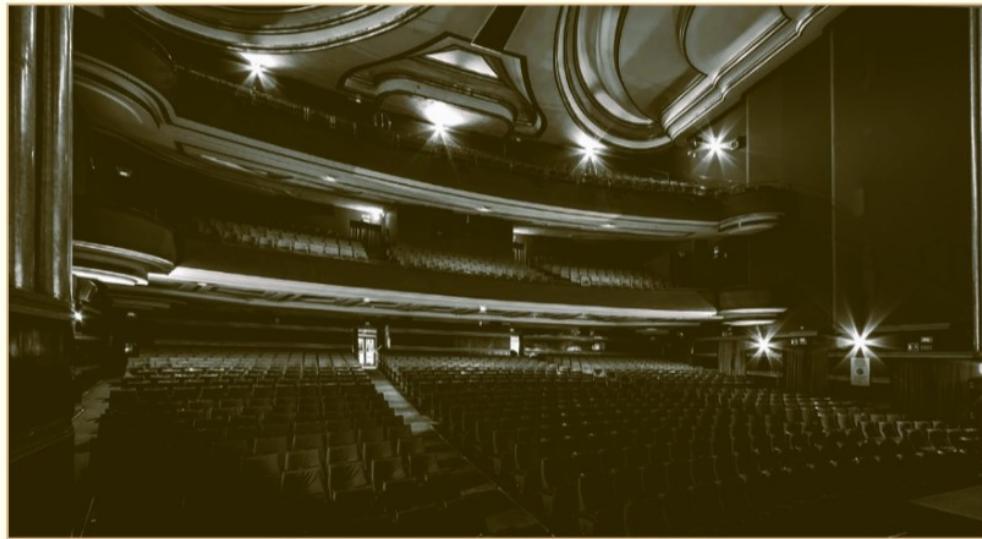


[Contacto](#)

Se ha cerrado la sesión.

¡Hasta la próxima!

[Link a Inicio](#)



[Aviso legal](#)

[Configuración de cookie](#)

[Política de privacidad](#)

[Política de cookies](#)



Autor: José Alberto Fernández Conde

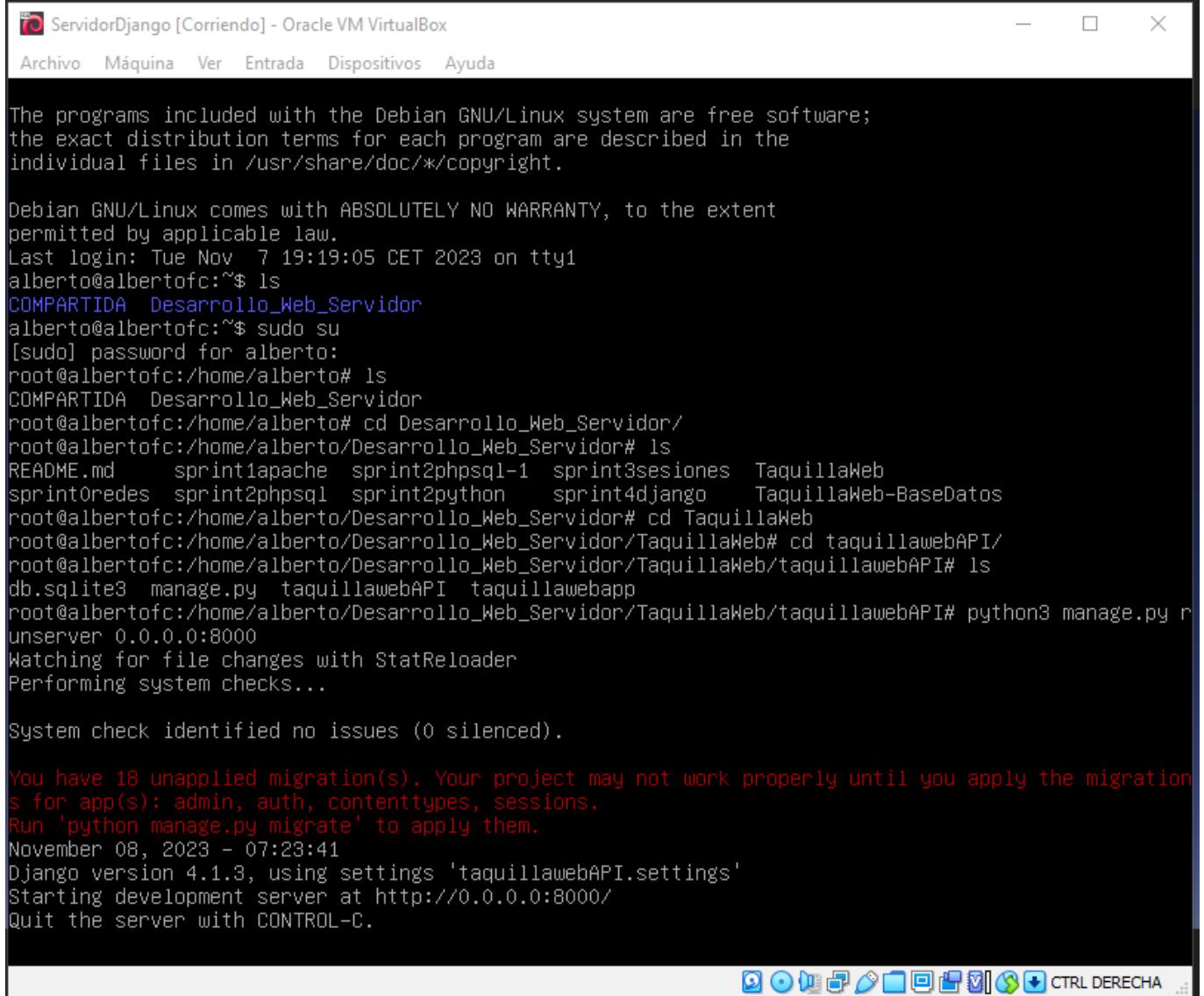
¡Síguenos!



Anexo 2. Endpoints

Endpoints TaquillaWeb

Servidor Django corriendo en <http://localhost:8000>



```
ServidorDjango [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Nov  7 19:19:05 CET 2023 on tty1
alberto@albertofc:~$ ls
COMPARTIDA Desarrollo_Web_Servidor
alberto@albertofc:~$ sudo su
[sudo] password for alberto:
root@albertofc:/home/alberto# ls
COMPARTIDA Desarrollo_Web_Servidor
root@albertofc:/home/alberto# cd Desarrollo_Web_Servidor/
root@albertofc:/home/alberto/Desarrollo_Web_Servidor# ls
README.md  sprint1apache  sprint2pgsql-1  sprint3sesiones  TaquillaWeb
sprint0redes  sprint2pgsql  sprint2python  sprint4django  TaquillaWeb-BaseDatos
root@albertofc:/home/alberto/Desarrollo_Web_Servidor# cd TaquillaWeb
root@albertofc:/home/alberto/Desarrollo_Web_Servidor/TaquillaWeb# cd taquillawebAPI/
root@albertofc:/home/alberto/Desarrollo_Web_Servidor/TaquillaWeb/taquillawebAPI# ls
db.sqlite3  manage.py  taquillawebAPI  taquillawebapp
root@albertofc:/home/alberto/Desarrollo_Web_Servidor/TaquillaWeb/taquillawebAPI# python3 manage.py runserver 0.0.0.0:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migration
s for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
November 08, 2023 - 07:23:41
Django version 4.1.3, using settings 'taquillawebAPI.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
```

Endpoint 1 (POST /users): creación de un nuevo usuario

Nombre	Email	Contraseña
Usuario Y	yyy@gmail.com	yyyy

```
curl -X POST -i -H "Accept: application/json" -H "Content-Type: application/json" -d "{\"usuarioNombre\":\"Usuario Y\",\"password\":\"yyyy\",\"email\":\"yyy@gmail.com\"}"  
http://localhost:8000/users/
```

Captura:

```
Microsoft Windows [Versión 10.0.19045.3693]  
(c) Microsoft Corporation. Todos los derechos reservados.  
  
C:\Users\Alberto Fdez>curl -X POST -i -H "Accept: application/json" -H "Content-Type: application/json" -d "{\"usuarioNombre\":\"Usuario Y\",\"password\":\"yyyy\",\"email\":\"yyy@gmail.com\"}" http://localhost:8000/users/  
HTTP/1.1 200 OK  
Date: Sat, 25 Nov 2023 10:16:55 GMT  
Server: WSGIServer/0.2 CPython/3.9.2  
Content-Type: application/json  
X-Frame-Options: DENY  
Content-Length: 88  
X-Content-Type-Options: nosniff  
Referrer-Policy: same-origin  
Cross-Origin-Opener-Policy: same-origin  
Vary: Origin  
  
{"status": "Se ha creado el nuevo usuario, status=201", "data": {"nombre": "Usuario Y"} }  
C:\Users\Alberto Fdez>
```

HTTP/1.1 200 OK

Date: Sat, 25 Nov 2023 10:16:55 GMT
Server: WSGIServer/0.2 CPython/3.9.2
Content-Type: application/json
X-Frame-Options: DENY
Content-Length: 88
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Vary: Origin

```
{  
    "status": "Se ha creado el nuevo usuario, status=201",  
    "data": {  
        "nombre": "Usuario Y"  
    }  
}
```

Endpoint 2 (POST /sessions): login de usuario

Datos de usuario		
Nombre	Email	Contraseña
José Fernández	jose@gmail.com	0001

```
curl -X POST -i -H "Accept: application/json" -H "Content-Type: application/json" -d
"{"usuarioNombre":"José
Fernández","password":"0001","email":"jose@gmail.com"}"
http://localhost:8000/sessions/
```

Captura:

```
C:\Users\Alberto Fdez>curl -X POST -i -H "Accept: application/json" -H "Content-Type: application/json" -d
"{"usuarioNombre":"José Fernández","password":"0001","email":"jose@gmail.com"}" http://localhost:8000/sessions/
HTTP/1.1 200 OK
Date: Sat, 25 Nov 2023 10:22:34 GMT
Server: WSGIServer/0.2 CPython/3.9.2
Content-Type: application/json
X-Frame-Options: DENY
Content-Length: 236
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Vary: Origin

{"status": "Inicio de sesión correcto, status=201", "data": {"token": "eyJhbGciOiJIUzI1Ni
IsInR5cCI6IkpXVCJ9.eyJlc3Vhcm1vX2lkIjo2LCJ1c3Vhcm1vX25vbWJyZSI6Ikpvclx1MDB1OSBGZXJuXHUwMG
UxbmRleiJ9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0"}}
C:\Users\Alberto Fdez>
```

```
HTTP/1.1 200 OK
Date: Sat, 25 Nov 2023 10:22:34 GMT
Server: WSGIServer/0.2 CPython/3.9.2
Content-Type: application/json
X-Frame-Options: DENY
Content-Length: 236
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Vary: Origin

{
    "status": "Inicio de sesión correcto, status=201",
    "data": {
        "token":
        "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlc3Vhcm1vX2lkIjo2LCJ1c3Vhcm1vX25vbWJyZ
        SI6Ikpvclx1MDB1OSBGZXJuXHUwMGUxbmRleiJ9.xZM51TB2z0CH1UvsWAzoJP-
        U8JUT20A5ceWtAtBfVw0"
    }
}
```

Login de usuario con email incorrecto

```
curl -X POST -i -H "Accept: application/json" -H "Content-Type: application/json" -d
"{"usuarioNombre":"José
Fernández","password":"0001","email":"ose@gmail.com"}"
http://localhost:8000/sessions/
```

Captura:

```
C:\Users\Alberto Fdez>curl -X POST -i -H "Accept: application/json" -H "Content-Type: application/json" -d
"{"usuarioNombre":"José Fernández","password":"0001","email":"ose@gmail.com"}" http://localhost:8000/sessions/
HTTP/1.1 200 OK
Date: Sat, 25 Nov 2023 10:24:42 GMT
Server: WSGIServer/0.2 CPython/3.9.2
Content-Type: application/json
X-Frame-Options: DENY
Content-Length: 54
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Vary: Origin

{"status": "El email no está registrado, status=400"}
C:\Users\Alberto Fdez>
```

```
HTTP/1.1 200 OK
Date: Sat, 25 Nov 2023 10:24:42 GMT
Server: WSGIServer/0.2 CPython/3.9.2
Content-Type: application/json
X-Frame-Options: DENY
Content-Length: 54
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Vary: Origin

{
    "status": "El email no está registrado, status=400"
}
```

Login de usuario con contraseña incorrecta

```
curl -X POST -i -H "Accept: application/json" -H "Content-Type: application/json" -d
"{"usuarioNombre":"José
Fernández","password":"1111","email":"jose@gmail.com"}"
http://localhost:8000/sessions/
```

Captura:

```
C:\Users\Alberto Fdez>curl -X POST -i -H "Accept: application/json" -H "Content-Type: app
lication/json" -d "{\"usuarioNombre\":\"José Fernández\",\"password\":\"1111\",\"email\":
\"jose@gmail.com\"}" http://localhost:8000/sessions/
HTTP/1.1 401 Unauthorized
Date: Sat, 25 Nov 2023 10:27:15 GMT
Server: WSGIServer/0.2 CPython/3.9.2
Content-Type: text/html; charset=utf-8
X-Frame-Options: DENY
Content-Length: 22
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Vary: Origin

Contraseña incorrecta
C:\Users\Alberto Fdez>
```

```
HTTP/1.1 401 Unauthorized
Date: Sat, 25 Nov 2023 10:27:15 GMT
Server: WSGIServer/0.2 CPython/3.9.2
Content-Type: text/html; charset=utf-8
X-Frame-Options: DENY
Content-Length: 22
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Vary: Origin
```

Contraseña incorrecta

Login de usuario con datos sin cubrir

```
curl -X POST -i -H "Accept: application/json" -H "Content-Type: application/json" -d
"{"usuarioNombre":"José Fernández","password": "", "email":"jose@gmail.com"}"
http://localhost:8000/sessions/
```

Captura:

```
C:\Users\Alberto Fdez>curl -X POST -i -H "Accept: application/json" -H "Content-Type: application/json" -d
"{"usuarioNombre":"José Fernández","password": "", "email":"jose@gmail.com"}" http://localhost:8000/sessions/
HTTP/1.1 200 OK
Date: Sat, 25 Nov 2023 10:30:13 GMT
Server: WSGIServer/0.2 CPython/3.9.2
Content-Type: application/json
X-Frame-Options: DENY
Content-Length: 47
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Vary: Origin

{"status": "Hay campos sin cubrir, status=400"}
C:\Users\Alberto Fdez>
```

```
HTTP/1.1 200 OK
Date: Sat, 25 Nov 2023 10:30:13 GMT
Server: WSGIServer/0.2 CPython/3.9.2
Content-Type: application/json
X-Frame-Options: DENY
Content-Length: 47
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Vary: Origin
```

```
{
    "status": "Hay campos sin cubrir, status=400"
}
```

Lista de usuarios registrados			
Nombre	Email	Contraseña	Guardado
Administrador TW	admintw@gmail.com	a1b2c3	Sí
José Fernández	jose@gmail.com	0001	Sí
Paco123	paco@gmail.com	Abc123	Sí
Antonio López	antonio@gmail.com	4321	Sí
Juan Gómez	juan@gmail.com	jg01	Sí
Ana García	ana@gmail.com	ana123	Sí
Usuario X	xxx@gmail.com	xxxx	Sí
Usuario Y	yyy@gmail.com	yyyy	Sí

Endpoint 3 (GET /cines/): obtener información de todos los cines de la base de datos

Token del usuario necesario para las peticiones

```
'eyJhbGciOiJIUzI1NilsInR5cCI6IkpXVCJ9eyJ1c3VhcmhvX2IkIjo2LCJ1c3VhcmhvX25vbWJyZSI6Ikpvclx1MDI  
OSBGZXJuXHUwMGUxbmRleiJ9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0'
```

Petición con el token de José Fernández:

```
curl -X GET -i -H "Accept: application/json" -H "Authorization:  
eyJhbGciOiJIUzI1NilsInR5cCI6IkpXVCJ9eyJ1c3VhcmhvX2IkIjo2LCJ1c3VhcmhvX25vbWJyZSI6Ikpvclx1MDI  
OSBGZXJuXHUwMGUxbmRleiJ9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0"  
http://localhost:8000/cines/
```

Captura:

```
C:\Users\Alberto Fdez>curl -X GET -i -H "Accept: application/json" -H "Authorization: eyJ  
hbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJ1c3VhcmhvX2IkIjo2LCJ1c3VhcmhvX25vbWJyZSI6Ikpvclx1MDI  
OSBGZXJuXHUwMGUxbmRleiJ9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0" http://localhost:8  
000/cines/  
HTTP/1.1 200 OK  
Date: Sat, 25 Nov 2023 10:31:20 GMT  
Server: WSGIServer/0.2 CPython/3.9.2  
Content-Type: application/json  
X-Frame-Options: DENY  
Content-Length: 623  
X-Content-Type-Options: nosniff  
Referrer-Policy: same-origin  
Cross-Origin-Opener-Policy: same-origin  
Vary: Origin  
  
[{"sesion_1": "16:30h - 18:30h", "sesion_2": "19:00h - 21:00h", "sesion_3": "21:30h - 23:  
30h"}, {"id_cine": 1, "nombre_cine": "Cines Austin", "logo_cine": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/logos/logo_Austin.jpg", "cantidadesalas_cine": 3}, {"id_cine": 2, "nombre_cine": "Cines TMA", "logo_cine": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/logos/logo_TMA.jpg", "cantidadesalas_cine": 5}, {"id_cine": 3, "nombre_cine": "Cines Mix", "logo_cine": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/logos/logo_Mix.jpg", "cantidadesalas_cine": 6}]  
C:\Users\Alberto Fdez>
```

HTTP/1.1 200 OK

```
Date: Sat, 25 Nov 2023 10:31:20 GMT  
Server: WSGIServer/0.2 CPython/3.9.2  
Content-Type: application/json  
X-Frame-Options: DENY  
Content-Length: 623  
X-Content-Type-Options: nosniff  
Referrer-Policy: same-origin  
Cross-Origin-Opener-Policy: same-origin  
Vary: Origin
```

```
[  
 {  
     "sesion_1": "16:30h - 18:30h",  
     "sesion_2": "19:00h - 21:00h",  
     "sesion_3": "21:30h - 23:30h"  
 },
```

```
{  
    "id_cine": 1,  
    "nombre_cine": "Cines Austin",  
    "logo_cine": "https://raw.githubusercontent.com/alberto22-  
23/PFC_DAW/main/figurasTW/logos/logo_Austin.jpg",  
    "cantidadsalas_cine": 3  
},  
{  
    "id_cine": 2,  
    "nombre_cine": "Cines TMA",  
    "logo_cine": "https://raw.githubusercontent.com/alberto22-  
23/PFC_DAW/main/figurasTW/logos/logo_TMA.jpg",  
    "cantidadsalas_cine": 5  
},  
{  
    "id_cine": 3,  
    "nombre_cine": "Cines Mix",  
    "logo_cine": "https://raw.githubusercontent.com/alberto22-  
23/PFC_DAW/main/figurasTW/logos/logo_Mix.jpg",  
    "cantidadsalas_cine": 6  
}  
]
```

Petición con token no registrado

```
curl -X GET -i -H "Accept: application/json" -H "Authorization: No_Registrado_eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c3Vhcm1vX2lkIjo2LCJ1c3Vhcm1vX25vbWJyZSI6Ikpvclx1MDB1OSBGZXJuXHUwMGUxbmRleij9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0_No_Registrado" http://localhost:8000/cines/
```

Captura:

```
C:\Users\Alberto Fdez>curl -X GET -i -H "Accept: application/json" -H "Authorization: No_Registrado_eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c3Vhcm1vX2lkIjo2LCJ1c3Vhcm1vX25vbWJyZSI6Ikpvclx1MDB1OSBGZXJuXHUwMGUxbmRleij9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0_No_Registrado" http://localhost:8000/cines/
HTTP/1.1 400 Bad Request
Date: Sat, 25 Nov 2023 10:32:52 GMT
Server: WSGIServer/0.2 CPython/3.9.2
Content-Type: text/html; charset=utf-8
X-Frame-Options: DENY
Content-Length: 29
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Vary: Origin

La autenticación ha fallado.
C:\Users\Alberto Fdez>
```

HTTP/1.1 400 Bad Request

```
Date: Sat, 25 Nov 2023 10:32:52 GMT
Server: WSGIServer/0.2 CPython/3.9.2
Content-Type: text/html; charset=utf-8
X-Frame-Options: DENY
Content-Length: 29
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Vary: Origin
```

La autenticación ha fallado.

Endpoint 4 (GET /peliculas/): obtener información de todas las películas de la base de datos

Petición con el token de José Fernández:

```
curl -X GET -i -H "Accept: application/json" -H "Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c3Vhcm1vX2lkIjo2LCJ1c3Vhcm1vX25vbWJyZSI6Ikpvclx1MDB1OSBGZXJuXHUwMGUxbmRleij9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0"  
http://localhost:8000/peliculas/
```

Captura:

```
C:\Users\Alberto Fdez>curl -X GET -i -H "Accept: application/json" -H "Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c3Vhcm1vX2lkIjo2LCJ1c3Vhcm1vX25vbWJyZSI6Ikpvclx1MDB1OSBGZXJuXHUwMGUxbmRleij9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0" http://localhost:8000/peliculas/  
HTTP/1.1 200 OK  
Date: Sat, 25 Nov 2023 10:34:03 GMT  
Server: WSGIServer/0.2 CPython/3.9.2  
Content-Type: application/json  
X-Frame-Options: DENY  
Content-Length: 5959  
X-Content-Type-Options: nosniff  
Referrer-Policy: same-origin  
Cross-Origin-Opener-Policy: same-origin  
Vary: Origin  
  
[{"id_pelicula": 1, "titulo_pelicula": "Austin Powers", "estreno_pelicula": 0, "sinopsis_pelicula": "Primera aventura de Austin Powers (Mike Myers), un peculiar y atractivo espía de los años sesenta, cuyo principal enemigo es el doctor Maligno (interpretado también por Myers). Tras ser ambos sometidos a un proceso de congelación, se despiertan treinta años después en una sociedad completamente distinta a la que conocían. Sin embargo, ellos siguen siendo los mismos.", "precio_pelicula": 6.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/austin_powers.jpg"}, {"id_pelicula": 2, "titulo_pelicula": "El Guateque", "estreno_pelicula": 0, "sinopsis_pelicula": "Hrundi V. Bakshi es un patoso actor de origen hindú que se encuentra rodando una película en el desierto. Por sus continuas meteduras de pata, es despedido del rodaje. Inesperadamente, recibe una invitación para asistir a una sofisticada fiesta organizada por el productor de su última película. Gracias a Hrundi, en la fiesta se producirán las situaciones más disparatadas.", "precio_pelicula": 6.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/el_guateque.jpg"}, {"id_pelicula": 3, "titulo_pelicula": "La Vida de Brian", "estreno_pelicula": 1, "sinopsis_pelicula": "Brian nace en un pesebre de Belén el mismo día que Jesucristo. Un cúmulo de desgraciados y tronchantes equívocos le harán llevar una vida paralela a la del verdadero Hijo de Dios. Sus pocas luces y el ambiente de decadencia y caos absoluto en que se haya sumergido la Galilea de aquellos días, le harán vivir en manos de su madre, de una feminista revolucionaria y del mismísimo Poncio Pilatos, su propia versión del calvario.", "precio_pelicula": 8.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/la_vida_de_Brian.jpg"}, {"id_pelicula": 4, "titulo_pelicula": "Blade Runner", "estreno_pelicula": 0, "sinopsis_pelicula": "Rick Deckard (Harrison Ford) es un blade runner, un agente de policía destinado al retiro de replicantes ilegales. Su misión es dar caza a un grupo de cuatro de estos androides, sofisticados NEXUS 6 superiores en fuerza e inteligencia a los humanos, pero diseñados para vivir una corta existencia de cuatro años.", "precio_pelicula": 7.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/blade_runner.jpg"}, {"id_pelicula": 5, "titulo_pelicula": "Matrix", "estreno_pelicula": 0, "sinopsis_pelicula": "Thomas Anderson es un brillante programador de una respetable compañía de software. Pero fuera del trabajo es Neo, un hacker que un día recibe una misteriosa visita...", "precio_pelicula": 7.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/matrix.jpg"}, {"id_pelicula": 6, "titulo_pelicula": "Tron", "estreno_pelicula": 1,
```

Captura:

```
rteles/matrix.jpg"}, {"id_pelicula": 6, "titulo_pelicula": "Tron", "estreno_pelicula": 1, "sinopsis_pelicula": "Sam Flynn, un experto en tecnología de 27 años e hijo de Kevin Flynn, investiga la desaparición de su padre y se adentra en un mundo digital distinto al original y creado por su padre, de feroces programas y juegos de arcade, y donde su padre ha estado atrapado durante 20 años.", "precio_pelicula": 8.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/tron.jpg"}, {"id_pelicula": 7, "titulo_pelicula": "2001 A Space Odyssey", "estreno_pelicula": 0, "sinopsis_pelicula": "Hace millones de años, antes de la aparición del \"homo sapiens\", unos primates descubren un monolito que los conduce a un estadio de inteligencia superior. Millones de años después, otro monolito, enterrado en una luna, desperta el interés de los científicos. Por último, durante una misión de la NASA, HAL 9000, una máquina dotada de inteligencia artificial, se encarga de controlar todos los sistemas de una nave espacial tripulada.", "precio_pelicula": 7.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/2001_a_space_odyssey.jpg"}, {"id_pelicula": 8, "titulo_pelicula": "Poltergeist", "estreno_pelicula": 0, "sinopsis_pelicula": "La película fue el primer gran éxito de Spielberg como productor. La trama gira en torno a los inquietantes sucesos que acontecen en la casa de una familia que vive en los suburbios, y en la que se sospecha que se está produciendo el fenómeno conocido como «poltergeist».", "precio_pelicula": 7.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/poltergeist.jpg"}, {"id_pelicula": 9, "titulo_pelicula": "Regreso al Futuro", "estreno_pelicula": 0, "sinopsis_pelicula": "La cinta transcurre en el año 1985, una época en la que el joven Marty McFly lleva una existencia anónima con su novia Jennifer. Los únicos problemas son su familia en crisis y un director al que le encantaría expulsarle del instituto, por lo que deberá hacer todo lo que esté en su mano para revertir esa situación y aparentar total normalidad. Amigo del excéntrico profesor Emmett Brown, una noche le acompaña a probar su nuevo experimento: viajar en el tiempo usando un DeLorean modificado...", "precio_pelicula": 7.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/regreso_al_futuro.jpg"}, {"id_pelicula": 10, "titulo_pelicula": "Memorias de África", "estreno_pelicula": 1, "sinopsis_pelicula": "A principios del siglo XX, Karen (Streep) contrae un matrimonio de conveniencia con el barón Blixen (Brandauer), un mujeriego empedernido. Ambos se establecen en Kenia con el propósito de explotar una plantación de café. En Karen Blixen nace un apasionado amor por la tierra y por las gentes de Kenia.", "precio_pelicula": 8.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/memorias_de_africa.jpg"}]
```

C:\Users\Alberto Fdez>

HTTP/1.1 200 OK

Date: Sat, 25 Nov 2023 10:34:03 GMT
Server: WSGIServer/0.2 CPython/3.9.2
Content-Type: application/json
X-Frame-Options: DENY
Content-Length: 5959
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Vary: Origin

```
[  
 {  
     "id_pelicula": 1,  
     "titulo_pelicula": "Austin Powers",  
     "estreno_pelicula": 0,  
     "sinopsis_pelicula": "Primera aventura de Austin Powers (Mike Myers), un peculiar y atractivo espía de los años sesenta...",  
     "precio_pelicula": 6.5,  
     "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/carteles/austin\_powers.jpg"  
 }
```

```

},
{
    "id_pelicula": 2,
    "titulo_pelicula": "El Guateque",
    "estreno_pelicula": 0,
    "sinopsis_pelicula": "Hrundi V. Bakshi es un patoso actor de origen hindú que se encuentra rodando una película en el desierto...",
    "precio_pelicula": 6.5,
    "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/cartel/figurasTW/cartel/el\_guateque.jpg"
},
{
    "id_pelicula": 3,
    "titulo_pelicula": "La Vida de Brian",
    "estreno_pelicula": 1,
    "sinopsis_pelicula": "Brian nace en un pesebre de Belén el mismo día que Jesucristo. Un cúmulo de...",
    "precio_pelicula": 8.5,
    "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/cartel/figurasTW/cartel/la\_vida\_de\_brian.jpg"
},
{
    "id_pelicula": 4,
    "titulo_pelicula": "Blade Runner",
    "estreno_pelicula": 0,
    "sinopsis_pelicula": "Rick Deckard (Harrison Ford) es un blade runner, un agente de policía...",
    "precio_pelicula": 7.5,
    "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/cartel/figurasTW/cartel/blade\_runner.jpg"
},
{
    "id_pelicula": 5,
    "titulo_pelicula": "Matrix",
    "estreno_pelicula": 0,
    "sinopsis_pelicula": "Thomas Anderson es un brillante programador de una respetable compañía de software. Pero fuera del trabajo es Neo, un hacker que un día recibe una misteriosa visita...",
    "precio_pelicula": 7.5,
    "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/cartel/figurasTW/cartel/matrix.jpg"
},
{
    "id_pelicula": 6,
    "titulo_pelicula": "Tron",
    "estreno_pelicula": 1,
    "sinopsis_pelicula": "Sam Flynn, un experto en tecnología de 27 años e hijo de Kevin Flynn, investiga la desaparición..."
}

```

```

    "precio_pelicula": 8.5,
    "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/carteles/tron.jpg"
},
{
    "id_pelicula": 7,
    "titulo_pelicula": "2001 A Space Odyssey",
    "estreno_pelicula": 0,
    "sinopsis_pelicula": "Hace millones de años, antes de la aparición del \"homo sapiens\", unos primates descubren un monolito...",
    "precio_pelicula": 7.5,
    "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/carteles/2001\_a\_space\_odyssey.jpg"
},
{
    "id_pelicula": 8,
    "titulo_pelicula": "Poltergeist",
    "estreno_pelicula": 0, "sinopsis_pelicula": "La película fue el primer gran éxito de Spielberg como productor. La trama gira en torno a los inquietantes sucesos que acontecen...",
    "precio_pelicula": 7.5,
    "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/carteles/poltergeist.jpg"
},
{
    "id_pelicula": 9,
    "titulo_pelicula": "Regreso al Futuro",
    "estreno_pelicula": 0,
    "sinopsis_pelicula": "La cinta transcurre en el año 1985, una época en la que el joven Marty McFly lleva una existencia anónima...",
    "precio_pelicula": 7.5,
    "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/carteles/regreso\_al\_futuro.jpg"
},
{
    "id_pelicula": 10,
    "titulo_pelicula": "Memorias de África",
    "estreno_pelicula": 1,
    "sinopsis_pelicula": "A principios del siglo XX, Karen (Streep) contrae un matrimonio de conveniencia con el barón Blixen...",
    "precio_pelicula": 8.5,
    "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/carteles/memorias\_de\_africa.jpg"
}
]

```

Endpoint 5 (GET /peliculas/<id>): obtener información de la película con el ID introducido

Petición con el token de José Fernández solicitando película con id=10:

```
curl -X GET -i -H "Accept: application/json" -H "Authorization:  
eyJhbGciOiJIUzI1NilsInR5cCI6IkpXVCJ9.eyJ1c3VhcmlvX2lkIjo2LCJ1c3VhcmlvX25vbWJyZSI6Ikpvclx1MDBI  
OSBGZXJuXHUwMGUxbmRleij9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0"  
http://localhost:8000/peliculas/10/
```

Captura:

```
C:\Users\Alberto Fdez>curl -X GET -i -H "Accept: application/json" -H "Authorization: eyJ  
hbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c3VhcmlvX2lkIjo2LCJ1c3VhcmlvX25vbWJyZSI6Ikpvclx1MDBI  
OSBGZXJuXHUwMGUxbmRleij9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0" http://localhost:8  
000/peliculas/10/  
HTTP/1.1 200 OK  
Date: Sat, 25 Nov 2023 10:39:07 GMT  
Server: WSGIServer/0.2 CPython/3.9.2  
Content-Type: application/json  
X-Frame-Options: DENY  
Content-Length: 910  
X-Content-Type-Options: nosniff  
Referrer-Policy: same-origin  
Cross-Origin-Opener-Policy: same-origin  
Vary: Origin  
  
{"datos_compra_entradas": [{"sesion_1": "16:30h - 18:30h", "sesion_2": "19:00h - 21:00h",  
"sesion_3": "21:30h - 23:30h"}, {"id_cine": 3, "id_pelicula": 10, "id_sala": 13}, {"id_cine": 3, "id_pelicula": 10, "id_sala": 14}], "datos_para_introducir": "id_cine, id_pelicula, id_sala y Sesión", "datos_introducidos_en": "http://localhost:8000/entradas", "id_pelicula": 10, "titulo_pelicula": "Memorias de África", "estreno_pelicula": 1, "sinopsis_pelicula": "A principios del siglo XX, Karen (Streep) contrae un matrimonio de conveniencia con el barón Blixen (Brandauer), un mujeriego empedernido. Ambos se establecen en Kenia con el propósito de explotar una plantación de café. En Karen Blixen nace un apasionado amor por la tierra y por las gentes de Kenia.", "precio_pelicula": 8.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/memorias_de_africa.jpg"}  
C:\Users\Alberto Fdez>
```

HTTP/1.1 200 OK

```
Date: Sat, 25 Nov 2023 10:39:07 GMT  
Server: WSGIServer/0.2 CPython/3.9.2  
Content-Type: application/json  
X-Frame-Options: DENY  
Content-Length: 910  
X-Content-Type-Options: nosniff  
Referrer-Policy: same-origin  
Cross-Origin-Opener-Policy: same-origin  
Vary: Origin
```

```
{  
    "datos_compra_entradas": [  
        {  
            "sesion_1": "16:30h - 18:30h",  
            "sesion_2": "19:00h - 21:00h",  
            "sesion_3": "21:30h - 23:30h"  
        },  
        {  
    
```

```

        "id_cine": 3,
        "id_pelicula": 10,
        "id_sala": 13
    },
    {
        "id_cine": 3,
        "id_pelicula": 10,
        "id_sala": 14
    }
],
"datos_para_introducir": "id_cine, id_pelicula, id_sala y Sesión",
"datos_introducidos_en": "http://localhost:8000/entradas",
"id_pelicula": 10,
"titulo_pelicula": "Memorias de Africa",
"estreno_pelicula": 1,
"sinopsis_pelicula": "A principios del siglo XX, Karen (Streep) contrae un matrimonio de conveniencia con el barón Blixen (Brandauer), un mujeriego empedernido. Ambos se establecen en Kenia con el propósito de explotar una plantación de café. En Karen Blixen nace un apasionado amor por la tierra y por las gentes de Kenia.", "precio_pelicula": 8.5,
"cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/carteles/memorias\_de\_africa.jpg"
}

```

Endpoint 6 (GET cines/<id>/peliculas): obtener información de las películas del cine cuyo ID es el introducido

Petición con el token de José Fernández solicitando películas del cine con id=3:

```
curl -X GET -i -H "Accept: application/json" -H "Authorization:  
eyJhbGciOiJIUzI1NilsInR5cCI6IkpXVCJ9eyJ1c3VhcmlvX2lkIjo2LCJ1c3VhcmlvX25vbWJyZSI6Ikpv...  
OSBGZXJuXHUwMGUxbmRleiJ9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0"
```

<http://localhost:8000/cine/3/peliculas/>

Captura:

```
C:\Users\Alberto Fdez>curl -X GET -i -H "Accept: application/json" -H "Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJ1c3VhcmlvX2lkIjo2LCJ1c3VhcmlvX25vbWJyZSI6Ikpv...  
OSBGZXJuXHUwMGUxbmRleiJ9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0" http://localhost:8000/cine/3/peliculas/  
HTTP/1.1 200 OK  
Date: Sat, 25 Nov 2023 10:40:38 GMT  
Server: WSGIServer/0.2 CPython/3.9.2  
Content-Type: application/json  
X-Frame-Options: DENY  
Content-Length: 1420  
X-Content-Type-Options: nosniff  
Referrer-Policy: same-origin  
Cross-Origin-Opener-Policy: same-origin  
Vary: Origin  
  
[{"cine_solicitado": "Cines Mix", "numero_de_salas": 6}, {"id_pelicula": 4, "titulo_pelicula": "Blade Runner", "estreno_pelicula": 0, "precio_pelicula": 7.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/blade_runner.jpg"}, {"id_pelicula": 3, "titulo_pelicula": "La Vida de Brian", "estreno_pelicula": 1, "precio_pelicula": 8.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/la_vida_de_Brian.jpg"}, {"id_pelicula": 8, "titulo_pelicula": "Poltergeist", "estreno_pelicula": 0, "precio_pelicula": 7.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/poltergeist.jpg"}, {"id_pelicula": 9, "titulo_pelicula": "Regreso al Futuro", "estreno_pelicula": 0, "precio_pelicula": 7.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/regreso_al_futuro.jpg"}, {"id_pelicula": 10, "titulo_pelicula": "Memorias de África", "estreno_pelicula": 1, "precio_pelicula": 8.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/memorias_de_africa.jpg"}, {"id_pelicula": 10, "titulo_pelicula": "Memorias de África", "estreno_pelicula": 1, "precio_pelicula": 8.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/memorias_de_africa.jpg"}]  
C:\Users\Alberto Fdez>
```

HTTP/1.1 200 OK

```
Date: Sat, 25 Nov 2023 10:40:38 GMT  
Server: WSGIServer/0.2 CPython/3.9.2  
Content-Type: application/json  
X-Frame-Options: DENY  
Content-Length: 1420  
X-Content-Type-Options: nosniff  
Referrer-Policy: same-origin  
Cross-Origin-Opener-Policy: same-origin  
Vary: Origin
```

[

```
{  
    "cine_solicitado": "Cines Mix",  
    "numero_de_salas": 6  
},  
{
```

```

        "id_pelicula": 4,
        "titulo_pelicula": "Blade Runner",
        "estreno_pelicula": 0,
        "precio_pelicula": 7.5,
        "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/carteles/blade\_runner.jpg"
    },
    {
        "id_pelicula": 3,
        "titulo_pelicula": "La Vida de Brian",
        "estreno_pelicula": 1,
        "precio_pelicula": 8.5,
        "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/carteles/la\_vida\_de\_Brian.jpg"
    },
    {
        "id_pelicula": 8,
        "titulo_pelicula": "Poltergeist",
        "estreno_pelicula": 0,
        "precio_pelicula": 7.5,
        "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/carteles/poltergeist.jpg"
    },
    {
        "id_pelicula": 9,
        "titulo_pelicula": "Regreso al Futuro",
        "estreno_pelicula": 0,
        "precio_pelicula": 7.5,
        "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/carteles/regreso\_al\_futuro.jpg"
    },
    {
        "id_pelicula": 10,
        "titulo_pelicula": "Memorias de Africa",
        "estreno_pelicula": 1,
        "precio_pelicula": 8.5,
        "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/carteles/memorias\_de\_africa.jpg"
    },
    {
        "id_pelicula": 10,
        "titulo_pelicula": "Memorias de Africa",
        "estreno_pelicula": 1,
        "precio_pelicula": 8.5,
        "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/carteles/memorias\_de\_africa.jpg"
    }
]

```

Endpoint 7 (GET /peliculas/estrenos): obtener las películas que son estrenos

Petición con el token de José Fernández para películas que son estreno:

```
curl -X GET -i -H "Accept: application/json" -H "Authorization:  
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c3VhcmlvX2lkIjo2LCJ1c3VhcmlvX25vbWJyZSI6Ikpv  
c1x1MDBl0SBGZXJuXHUwMGUxbmRleiJ9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0"  
http://localhost:8000/peliculas/estrenos/
```

Captura:

```
C:\Users\Alberto Fdez>curl -X GET -i -H "Accept: application/json" -H "Authorization: eyJ  
hbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c3VhcmlvX2lkIjo2LCJ1c3VhcmlvX25vbWJyZSI6Ikpv  
c1x1MDBl0SBGZXJuXHUwMGUxbmRleiJ9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0" http://localhost:8  
000/peliculas/estrenos/  
HTTP/1.1 200 OK  
Date: Sat, 25 Nov 2023 10:42:21 GMT  
Server: WSGIServer/0.2 CPython/3.9.2  
Content-Type: application/json  
X-Frame-Options: DENY  
Content-Length: 1806  
X-Content-Type-Options: nosniff  
Referrer-Policy: same-origin  
Cross-Origin-Opener-Policy: same-origin  
Vary: Origin  
  
[{"numero_total_peliculas_en_estreno": 3, "id_pelicula": 3, "titulo_pelicula": "La Vida de Brian", "estreno_pelicula": 1, "sinopsis_pelicula": "Brian nace en un pesebre de Belén el mismo día que Jesucristo. Un cúmulo de desgraciados y tronchantes equívocos le harán llevar una vida paralela a la del verdadero Hijo de Dios. Sus pocas luces y el ambiente de decadencia y caos absoluto en que se haya sumergida la Galilea de aquellos días, le harán vivir en manos de su madre, de una feminista revolucionaria y del mismísimo Poncio Pilatos, su propia versión del calvario.", "precio_pelicula": 8.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/la_vida_de_Brian.jpg"}, {"id_pelicula": 6, "titulo_pelicula": "Tron", "estreno_pelicula": 1, "sinopsis_pelicula": "Sam Flynn, un experto en tecnología de 27 años e hijo de Kevin Flynn, investiga la desaparición de su padre y se adentra en un mundo digital distinto al original y creado por su padre, de feroz programas y juegos de arcade, y donde su padre ha estado atrapado durante 20 años.", "precio_pelicula": 8.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/tron.jpg"}, {"id_pelicula": 10, "titulo_pelicula": "Memorias de África", "estreno_pelicula": 1, "sinopsis_pelicula": "A principios del siglo XX, Karen (Streep) contrae un matrimonio de conveniencia con el barón Blixen (Brandauer), un mujeriego empedernido. Ambos se establecen en Kenia con el propósito de explotar una plantación de café. En Karen Blixen nace un apasionado amor por la tierra y por las gentes de Kenia.", "precio_pelicula": 8.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/memorias_de_africa.jpg"}]  
C:\Users\Alberto Fdez>
```

HTTP/1.1 200 OK

```
Date: Sat, 25 Nov 2023 10:42:21 GMT  
Server: WSGIServer/0.2 CPython/3.9.2  
Content-Type: application/json  
X-Frame-Options: DENY  
Content-Length: 1806  
X-Content-Type-Options: nosniff  
Referrer-Policy: same-origin  
Cross-Origin-Opener-Policy: same-origin  
Vary: Origin
```

```
[
  {
    "numero_total_peliculas_en_estreno": 3
  },
  {
    "id_pelicula": 3,
    "titulo_pelicula": "La Vida de Brian",
    "estreno_pelicula": 1,
    "sinopsis_pelicula": "Brian nace en un pesebre de Belén el mismo día que Jesucristo. Un cúmulo de desgraciados y tronchantes equívocos le harán llevar una vida paralela a la del verdadero Hijo de Dios. Sus pocas luces y el ambiente de decadencia y caos absoluto en que se haya sumergida la Galilea de aquellos días, le harán vivir en manos de su madre, de una feminista revolucionaria y del mismísimo Poncio Pilatos, su propia versión del calvario.",
    "precio_pelicula": 8.5,
    "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/cartel/la\_vida\_de\_Brian.jpg"
  },
  {
    "id_pelicula": 6,
    "titulo_pelicula": "Tron",
    "estreno_pelicula": 1,
    "sinopsis_pelicula": "Sam Flynn, un experto en tecnología de 27 años e hijo de Kevin Flynn, investiga la desaparición de su padre y se adentra en un mundo digital distinto al original y creado por su padre, de feroz programas y juegos de arcade, y donde su padre ha estado atrapado durante 20 años.",
    "precio_pelicula": 8.5,
    "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/cartel/tron.jpg"
  },
  {
    "id_pelicula": 10,
    "titulo_pelicula": "Memorias de África",
    "estreno_pelicula": 1,
    "sinopsis_pelicula": "A principios del siglo XX, Karen (Streep) contrae un matrimonio de conveniencia con el barón Blixen (Brandauer), un mujeriego empedernido. Ambos se establecen en Kenia con el propósito de explotar una plantación de café. En Karen Blixen nace un apasionado amor por la tierra y por las gentes de Kenia.", "precio_pelicula": 8.5,
    "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/cartel/memorias\_de\_africa.jpg"
  }
]
```

Endpoint 8 (POST /entradas/): adquirir entradas

Con curl entrada para el usuario José Fernández:

```
idcine: 3; idpelicula: 10; idsala: 14; entradas: 2; sesión: 1
curl -X POST -i -H "Accept: application/json" -H "Content-Type: application/json" -H
"Authorization:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c3Vhcm1vX2lkIjo2LCJ1c3Vhcm1vX25vbWJyZSI6Ikpvc1x1MDBlOSBGZXJuXHUwMGUxbmRleiJ9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0" -d
 "{\"idcine\":\"3\", \"idpelicula\":\"10\", \"idsala\":\"14\", \"cantidadentradas\":\"2\", \"sesion\":\"1\", \"fecha_yeardmonthday\":\"20231127\", \"numero_tarjeta\":\"123456\"}"
http://localhost:8000/entradas/
```

Captura:

```
C:\Users\Alberto Fdez>curl -X POST -i -H "Accept: application/json" -H "Content-Type: application/json" -H "Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c3Vhcm1vX2lkIjo2LCJ1c3Vhcm1vX25vbWJyZSI6Ikpvc1x1MDBlOSBGZXJuXHUwMGUxbmRleiJ9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0" -d "{\"idcine\":\"3\", \"idpelicula\":\"10\", \"idsala\":\"14\", \"cantidadentradas\":\"2\", \"sesion\":\"1\", \"fecha_yeardmonthday\":\"20231127\", \"numero_tarjeta\":\"123456\"}" http://localhost:8000/entradas/
HTTP/1.1 200 OK
Date: Sat, 25 Nov 2023 10:47:59 GMT
Server: WSGIServer/0.2 CPython/3.9.2
Content-Type: application/json
X-Frame-Options: DENY
Content-Length: 409
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Vary: Origin

{"mensaje_compra": "Compra realizada.", "total_ocupadas": 6, "restantes": 94, "identificador_ticket": 385, "cine": "Cines Mix", "titulo_pelicula": "Memorias de Africa", "fecha_proyeccion": "20231127", "sesion": "1", "id_sala": 14, "numero_sala": 6, "numero_entradas": "2", "precio_unitario": 8.5, "precio_total": 17.0, "pago": "Se ha cargado el importe en su tarjeta.", "fecha_compra": "25-11-2023, 10:47:59"}
C:\Users\Alberto Fdez>
```

HTTP/1.1 200 OK

```
Date: Sat, 25 Nov 2023 10:47:59 GMT
Server: WSGIServer/0.2 CPython/3.9.2
Content-Type: application/json
X-Frame-Options: DENY
Content-Length: 409
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Vary: Origin
```

```
{
  "mensaje_compra": "Compra realizada.",
  "total_ocupadas": 6,
  "restantes": 94,
  "identificador_ticket": 385,
  "cine": "Cines Mix",
  "titulo_pelicula": "Memorias de Africa",
```

```
"fecha_proyeccion": "20231127",
"sesion": "1",
"id_sala": 14,
"numero_sala": 6,
"numero_entradas": "2",
"precio_unitario": 8.5,
"precio_total": 17.0,
"pago": "Se ha cargado el importe en su tarjeta.",
"fecha_compra": "25-11-2023, 10:47:59"
}
```

Endpoint 9 (GET /buscar_pelicula/<cadena_solicitada>/): buscador de películas

Petición con el token de José Fernández solicitando película en cuyo título figure la cadena introducida:

```
curl -X GET -i -H "Accept: application/json" -H "Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c3VhcmlvX2lkIjo2LCJ1c3VhcmlvX25vbWJyZSI6Ikpv c1x1MDB10SBGZXJuXHUwMGUxbmRleiJ9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0"  
http://localhost:8000/buscar\_pelicula/Matr/
```

Captura:

```
C:\Users\Alberto Fdez>curl -X GET -i -H "Accept: application/json" -H "Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c3VhcmlvX2lkIjo2LCJ1c3VhcmlvX25vbWJyZSI6Ikpv c1x1MDB10SBGZXJuXHUwMGUxbmRleiJ9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0" http://localhost:8000/buscar_pelicula/Matr/  
HTTP/1.1 200 OK  
Date: Sat, 25 Nov 2023 16:56:37 GMT  
Server: WSGIServer/0.2 CPython/3.9.2  
Content-Type: application/json  
X-Frame-Options: DENY  
Content-Length: 473  
X-Content-Type-Options: nosniff  
Referrer-Policy: same-origin  
Cross-Origin-Opener-Policy: same-origin  
Vary: Origin  
  
[{"Coincidencias": "Se han encontrado 1 títulos con coincidencias.", "id_pelicula": 5, "titulo_pelicula": "Matrix", "estreno_pelicula": 0, "sinopsis_pelicula": "Thomas Anderson es un brillante programador de una respetable compañía de software. Pero fuera del trabajo es Neo, un hacker que un día recibe una misteriosa visita...", "precio_pelicula": 7.5, "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/matrix.jpg"}]  
C:\Users\Alberto Fdez>
```

HTTP/1.1 200 OK

```
Date: Sat, 25 Nov 2023 16:56:37 GMT  
Server: WSGIServer/0.2 CPython/3.9.2  
Content-Type: application/json  
X-Frame-Options: DENY  
Content-Length: 473  
X-Content-Type-Options: nosniff  
Referrer-Policy: same-origin  
Cross-Origin-Opener-Policy: same-origin  
Vary: Origin
```

```
[  
 {  
     "Coincidencias": "Se han encontrado 1 títulos con coincidencias."  
 },  
 {  
     "id_pelicula": 5,  
     "titulo_pelicula":  
     "Matrix",  
     "estreno_pelicula": 0,  
     "sinopsis_pelicula": "Thomas Anderson es un brillante programador de una respetable compañía de software. Pero fuera del trabajo es Neo, un hacker
```

```

        que un día recibe una misteriosa visita...",
        "precio_pelicula": 7.5,
        "cartel_pelicula": "https://raw.githubusercontent.com/alberto22-23/PFC\_DAW/main/figurasTW/cartel/matrix.jpg"
    }
]

```

.....

Sin títulos coincidentes

```

curl -X GET -i -H "Accept: application/json" -H "Authorization:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c3VhcmlvX2lkIjo2LCJ1c3VhcmlvX25vbWJyZSI6Ikpvc1x1M
c1x1MDBl0SBGZXJuXHUwMGUxbmRleiJ9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0"
http://localhost:8000/buscar\_pelicula/xxx/

```

Captura:

```

C:\Users\Alberto Fdez>curl -X GET -i -H "Accept: application/json" -H "Authorization: ey
JhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c3VhcmlvX2lkIjo2LCJ1c3VhcmlvX25vbWJyZSI6Ikpvc1x1M
DBl0SBGZXJuXHUwMGUxbmRleiJ9.xZM51TB2z0CH1UvsWAzoJP-U8JUT20A5ceWtAtBfVw0" http://localhost:8000/buscar_pelicula/xxx/
HTTP/1.1 200 OK
Date: Sat, 25 Nov 2023 17:00:43 GMT
Server: WSGIServer/0.2 CPython/3.9.2
Content-Type: text/html; charset=utf-8
X-Frame-Options: DENY
Content-Length: 64
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Vary: Origin

No se han encontrado títulos coincidentes. Vuelva a intentarlo.
C:\Users\Alberto Fdez>

```

HTTP/1.1 200 OK

```

Date: Sat, 25 Nov 2023 17:00:43 GMT
Server: WSGIServer/0.2 CPython/3.9.2
Content-Type: text/html; charset=utf-8
X-Frame-Options: DENY
Content-Length: 64
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Vary: Origin

```

No se han encontrado títulos coincidentes. Vuelva a intentarlo.

Endpoints correspondientes al usuario administrador

En TaquillaWeb no se ha realizado el frontend para el usuario administrador. Se han realizado estos endpoints con el fin de facilitar las pruebas de funcionamiento de la aplicación.

Endpoint Extra 01

Actualización de los aforos de las salas de Cines Austin en la base de datos.

Se han reducido los aforos de las salas de Cines Austin en la base de datos para hacer las comprobaciones de funcionamiento con más comodidad.

Token del usuario con nombre Administrador TW

```
'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c3VhcmlvX2lkIjoxMSwidXN1YXJpb19ub21icmUiOiJBZG1pbmlzdHJhZG9yIFRXIn0.rxziY85rdSdliUew1XcWaBxRN18uYxNzDHoP4dKjKJY'
```

Con curl el usuario Administrador TW deberá introducir los siguientes campos:

```
curl -X POST -i -H "Accept: application/json" -H "Content-Type: application/json" -H "Authorization:  
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c3VhcmlvX2lkIjoxMSwidXN1YXJpb19ub21icmUiOiJB  
ZG1pbmlzdHJhZG9yIFRXIn0.rxziY85rdSdliUew1XcWaBxRN18uYxNzDHoP4dKjKJY" -d  
>{"aforo_salaid_1": "10", "aforo_salaid_2": "12", "aforo_salaid_3": "15"}  
http://localhost:8000/admin\_austin\_aforos/
```

Endpoint Extra 02

Borrado de todos los registros en la tabla entradas de la base de datos por si fuese necesario.

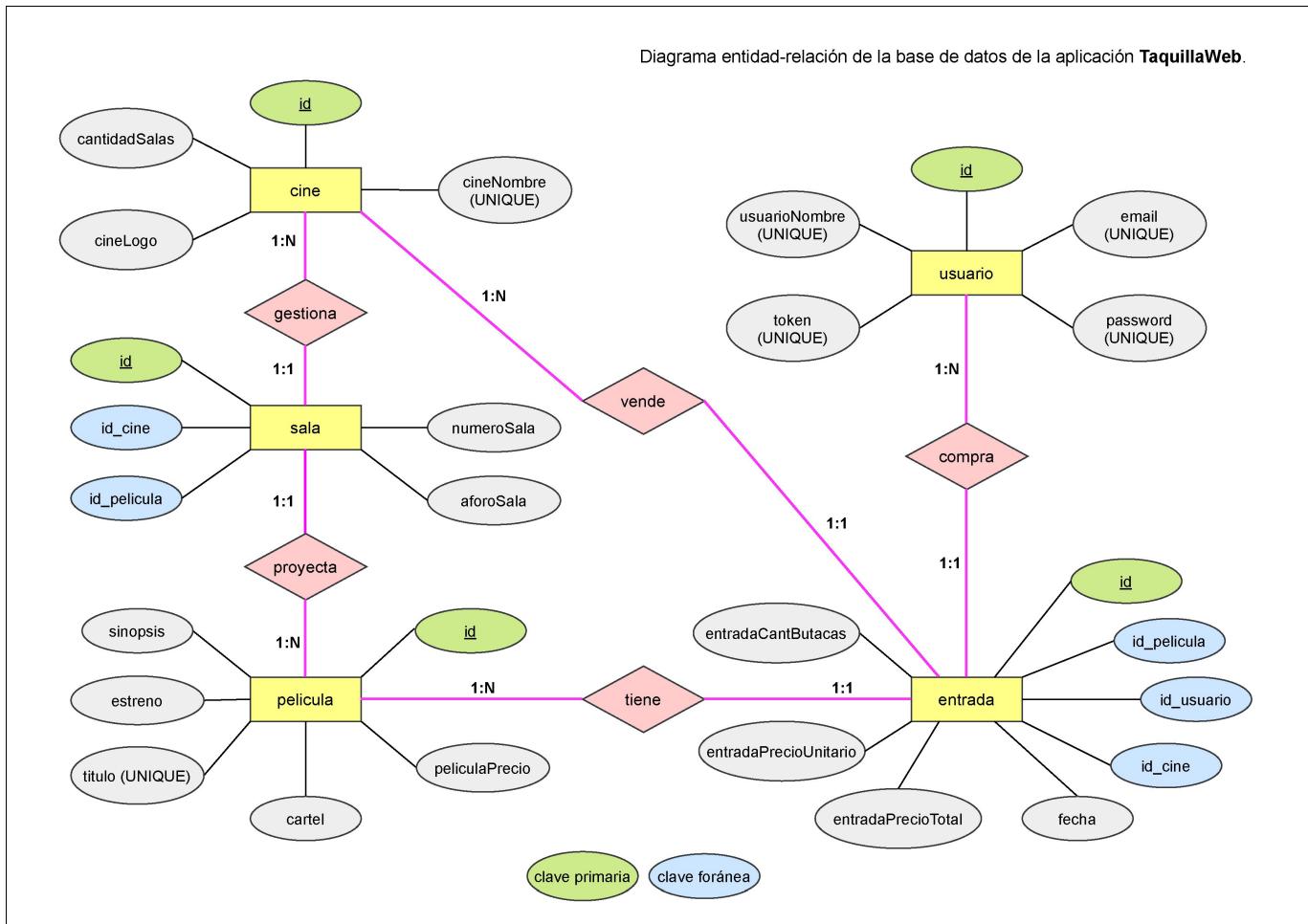
Con curl el usuario Administrador TW deberá introducir los siguientes campos:

```
curl -X POST -i -H "Accept: application/json" -H "Content-Type: application/json" -H "Authorization: eyJhbGci0... " -d "{\"borrar_registros_tentradas\": \"Aceptar\"}"  
http://localhost:8000/admin\_borrado\_entradas/
```

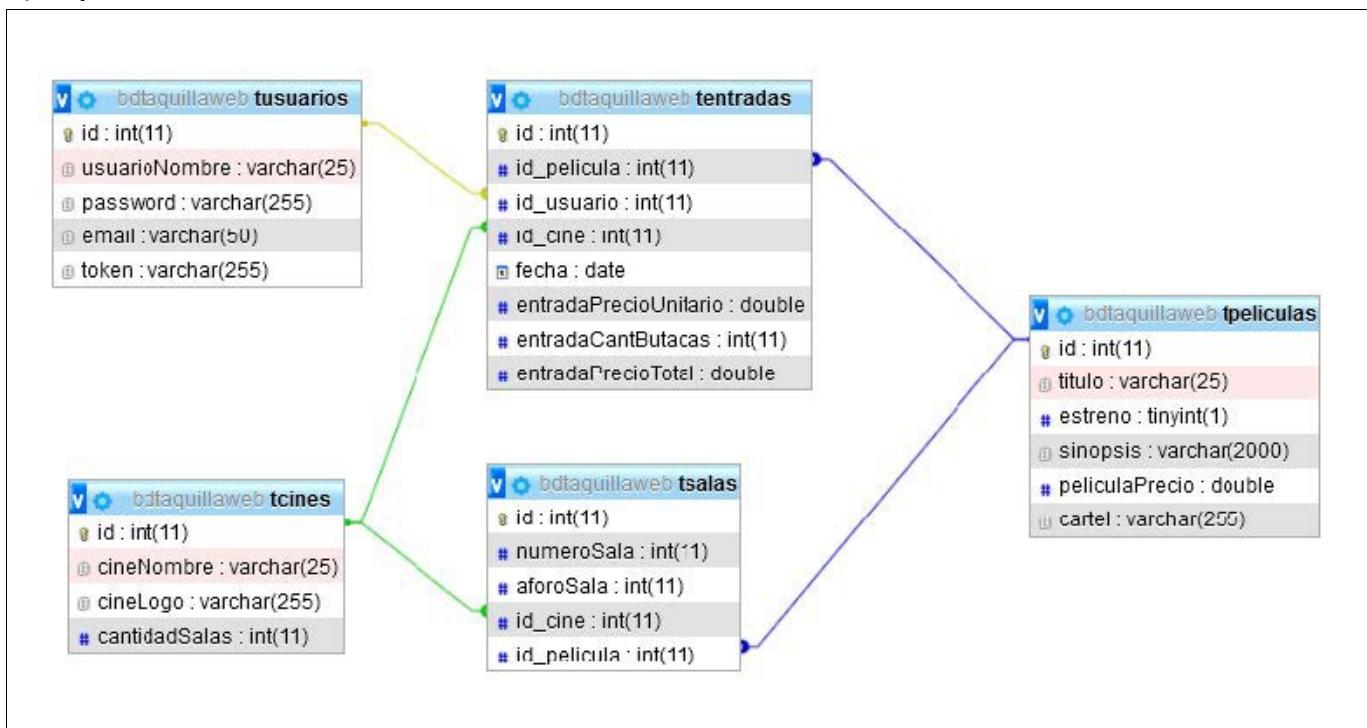
Anexo 3. Breve descripción de la base de datos.

Breve descripción de las tablas en la base de datos

a) Diagrama entidad-relación



b) Esquema relacional de la base de datos



c) Tablas y datos

Tabla tUsuarios				
id	usuarioNombre	password	email	token

Tabla tCines			
id	cineNombre	cantidadSalas	cineLogo
1	Cines Austin	3	
2	Cines TMA	5	
3	Cines Mix	6	

Cines Logos:

https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/logos/logo_Austin.jpg

https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/logos/logo_TMA.jpg

https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/logos/logo_Mix.jpg

Cines Austin (id=1)	
Hay 3 películas y 3 salas en Cines Austin:	Aforo Cines Austin: <ul style="list-style-type: none">• Austin Powers (id=1) en sala 1 (id=1)• El Guateque (id=2) en sala 2 (id=2)• La Vida de Brian (id=3) en sala 3 (id=3) <ul style="list-style-type: none">• sala 1 (id=1) → 50 butacas• sala 2 (id=2) → 60 butacas• sala 3 (id=3) → 70 butacas

Con el fin de realizar las pruebas de funcionamiento los aforos de Cines Austin se pueden modificar.

Cines TMA (id=2)	
Hay 4 películas y 5 salas en Cines TMA:	Aforo cines TMA: <ul style="list-style-type: none">• Blade Runner (id=4) en sala 1 (id=4)• Matrix (id=5) en sala 2 (id=5)• Tron (id=6) en sala 3 (id=6)• 2001 A Space Odissey (id=7) en sala 4 (id=7)• Tron (id=6) en sala 5 (id=8) <ul style="list-style-type: none">• sala 1 (id=4) → 65 butacas• sala 2 (id=5) → 75 butacas• sala 3 (id=6) → 90 butacas• sala 4 (id=7) → 80 butacas• sala 5 (id=8) → 80 butacas

Cines Mix (id=3)	
Hay 5 películas y 6 salas en Cines Mix:	Aforo Cines Mix: <ul style="list-style-type: none">• Blade-Runner (id=4) en sala 1 (id=9)• La Vida de Brian (id=3) en sala 2 (id=10)• Poltergeist (id=8) en sala 3 (id=11)• Regreso al Futuro (id=9) en sala 4 (id=12)• Memorias de África (id=10) en sala 5 (id=13)• Memorias de África (id=10) en sala 5 (id=13) <ul style="list-style-type: none">• sala 1 (id=9) → 70 butacas• sala 2 (id=10) → 70 butacas• sala 3 (id=11) → 80 butacas• sala 4 (id=12) → 80 butacas• sala 5 (id=13) → 100 butacas• sala 6 (id=14) → 100 butacas

Tabla tSalas				
Id (AI)	id_cine	id_pelicula	numeroSala	aforoSala
1	1	1	1	50
2	1	2	2	60
3	1	3	3	70
4	2	4	1	65
5	2	5	2	75
6	2	6	3	90
7	2	7	4	80
8	2	6	5	80
9	3	4	1	70
10	3	3	2	70
11	3	8	3	80
12	3	9	4	80
13	3	10	5	100
14	3	10	6	100

Tabla tPeliculas					
id	titulo	estreno	sinopsis	peliculaPrecio	cartel
1	Austin Powers	False		6,5	
2	El Guateque	False		6,5	
3	La Vida de Brian	True		8,5	
4	Blade Runner	False		7,5	
5	Matrix	False		7,5	
6	Tron	True		8,5	
7	2001 A Space Odissey	False		7,5	
8	Poltergeist	False		7,5	
9	Regreso al Futuro	False		7,5	
10	Memorias de Africa	True		8,5	

Tabla tPeliculas, carteles y sinopsis:

https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/austin_powers.jpg

Primera aventura de Austin Powers (Mike Myers)...

https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/el_guateque.jpg

Hrundi V. Bakshi es un patoso actor de origen hindú...

https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/la_vida_de_Brian.jpg

Brian nace en un pesebre de Belén el mismo día que Jesucristo. Un cúmulo de...

https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/blade_runner.jpg

Rick Deckard (Harrison Ford) es un blade runner, un agente de policía...

https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/matrix.jpg

Thomas Anderson es un brillante programador de una respetable compañía de software...

https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/tron.jpg

Sam Flynn, un experto en tecnología de 27 años e hijo de Kevin Flynn, investiga...

https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/2001_a_space_odyssey.jpg

Hace millones de años, antes de la aparición del "homo sapiens", unos primates descubren...

https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/poltergeist.jpg

La película fue el primer gran éxito de Spielberg como productor. La trama gira en torno a los...

https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/regreso_al_futuro.jpg

La cinta transcurre en el año 1985, una época en la que el joven Marty McFly lleva una existencia...

https://raw.githubusercontent.com/alberto22-23/PFC_DAW/main/figurasTW/carteles/memorias_de_africa.jpg

A principios del siglo XX, Karen (Streep) contrae un matrimonio de conveniencia con el barón...

Tabla tEntradas							
Id (AI)	id_pelicula	id_usuario	id_cine	fecha	entradaCant Butacas	entradaPrecio Unitario	entradaPrecioT otal