



# CINE PARA SER FELIZ

Web de gestión de cine

Víctor Moreno Ruíz

I.E.S. Juan Bosco - Desarrollo de Aplicaciones Web  
2024

1. Introducción.....	3
1.1. Presentación del proyecto .....	3
1.2. Objetivos del proyecto .....	3
1.3. Justificación del proyecto .....	3
2. Análisis de requerimientos .....	3
2.1. Identificación de necesidades y requerimientos .....	3
2.1.1. Necesidades .....	3
2.1.2. Requerimientos.....	4
2.2. Identificación de publico.....	4
2.3. Estudio de mercado y competencia .....	4
3. Diseño y planificación .....	5
3.1. Definición de la arquitectura del proyecto .....	5
3.2. Diseño de la interfaz de usuario.....	5
3.3. Planificación de las tareas y los recursos necesarios .....	6
4. Implementación y pruebas .....	7
4.1. Desarrollo de las funcionalidades del proyecto .....	7
4.2. Pruebas unitarias y de integración .....	7
4.3. Corrección de errores y optimización del rendimiento .....	8
5. Documentación.....	8
5.1. Documentación técnica.....	8
5.2. Documentación de usuario .....	13
5.2.1. Página principal .....	13
5.2.2. Login .....	14
5.2.3. Registro.....	14
5.2.4. Buscar películas .....	15
5.2.5. Detalle de película .....	16
5.2.6. Solicitud de restablecimiento de contraseña .....	17
5.2.7. Restablecimiento de contraseña.....	18
5.2.8. Actores.....	18
5.2.9. Edición de actores .....	19
5.2.10. Creación de actores .....	19
5.2.11. Cines .....	20

5.2.12.	Edición de cines.....	20
5.2.13.	Creación de cines .....	21
5.2.14.	Crear películas .....	21
5.2.15.	Géneros .....	22
5.2.16.	Edición de genero.....	23
5.2.17.	Creación de genero .....	23
5.2.18.	Usuarios.....	23
5.2.19.	Edición de usuario.....	24
5.3.	Manual de instalación y configuración .....	24
6.	Mantenimiento y evolución.....	25
6.1.	Plan de mantenimiento y soporte .....	25
6.2.	Identificación de posibles mejoras y evolución del proyecto.....	25
6.3.	Actualizaciones y mejoras futuras .....	26
7.	Conclusiones .....	27
7.1.	Evaluación del proyecto .....	27
7.2.	Cumplimiento de objetivos y requisitos.....	27
7.3.	Lecciones aprendidas y recomendaciones para futuros proyectos .....	27
8.	Bibliografía y referencias .....	28
8.1.	Fuentes utilizadas en el proyecto .....	28
8.2.	Referencias y enlaces de interés.....	28

# 1. Introducción

## 1.1. Presentación del proyecto

El objetivo principal del proyecto es poder gestionar la información de visualización de películas en distintos cines, de tal manera que la WEB pueda facilitar la búsqueda de las películas que se proyectan en cada cine, además de una información básica de cada una de las películas.

## 1.2. Objetivos del proyecto

Entrando más en detalle de los objetivos que pretende suplir este proyecto podemos encontrar:

- Unificar en una web información de visualización de películas: Actualmente existen muchas webs, cada cine normalmente tiene la suya propia, esto te puede llegar a limitar ya que, si quieres asistir a un cine en una ciudad de la cual no tienes conocimiento de los cines que hay, tendrías que buscar cines en dicha ciudad. Además, cada cine puede visualizar unas u otras películas, con nuestra web realizaríamos una única búsqueda para poder obtener la máxima información posible.
- Visualización de información de películas, sinopsis, trailers, posters, etc.
- En un futuro, se podría implementar diversas integraciones para poder comprar entradas con distintos cines.

## 1.3. Justificación del proyecto

Realizando un estudio de los servicios que se ofrecen de las webs actuales para información de cines, se detectó que se debería realizar una web que facilite la búsqueda de información, compra de entradas en diversos cines, de tal manera que el usuario pueda interactuar con una sola web reduciendo de esta manera la complejidad de poder buscar, comprar entradas para poder visualizar una película en distintos cines.

# 2. Análisis de requerimientos

## 2.1. Identificación de necesidades y requerimientos

Tras analizar las funcionalidades que nuestra web debe aportar, se han detectado las siguientes necesidades.

### 2.1.1. Necesidades

- **Búsqueda de películas:** Se debe de disponer de un buscador de películas, que nos permita encontrar las películas necesarias por el usuario de una manera fácil y rápida.
- **Valoraciones:** Para que un usuario pueda elegir entre diversas películas, se va a implementar un sistema de valoraciones por usuario y película, de esta manera se podrá tener una

valoración promedio que permitirá al usuario saber el nivel de valoración que los usuarios de la web han dado a cada película.

- **Catálogo de películas:** Es necesario implementar un sistema para poder registrar, visualizar películas.
- **Detalle de películas:** Por cada película registrada el usuario podrá visualizar los datos de las películas, entre los cuales podremos encontrar título, resumen, poster, trailer, fecha de lanzamiento, cines en los que se puede visualizar, etc.
- **Integración con diversos cines:** Se necesitaría implementar una compra de entradas para películas de diversos cines, de tal manera que necesitaríamos realizar una integración con la API de cada cine con el que nos queramos integrar.

### 2.1.2. Requerimientos

Para poder ofrecer servicio a los usuarios de la web debemos tener en cuenta los siguientes puntos:

- **Seguridad:** Nuestra web debe contar con medidas de seguridad con el objetivo de proteger la información personal de cada uno de los usuarios.
- **Escalabilidad:** Según los usuarios que consuman nuestra web nos interesará poder asignar recursos para poder suplir dicha necesidad. Además, según la web disponga de usuarios, se agregarán nuevas funcionalidades.
- **Rendimiento:** Para poder atraer a los usuarios debemos distinguirnos del resto, teniendo una mayor velocidad de uso de la web.
- **Facilidad de uso:** Con el objetivo de que los usuarios nos elijan antes que, al resto de webs, debemos ofrecer una interfaz de usuario limpia y fácil de usar, de tal manera que esto nos permita distinguirnos de nuestra competencia.

## 2.2. Identificación de publico

Dicha web va dirigida a usuarios que disfruten del cine y que buscan información sobre ellas antes de asistir a ver una película (trailers, valoraciones, etc). También va focalizado a aquellas personas que les resulta complejo moverse por las diversas webs de los cines de su localidad.

Nuestra web está centrada en todo tipo de públicos desde adolescentes hasta adultos, que disponen de una afición por el mundo del cine y que prefieren buscar información desde casa y no tener que asistir a un cine para poder ver la información de las películas que tienen en cartelera.

## 2.3. Estudio de mercado y competencia

Tras realizar un análisis de la competencia, se han detectado que existen dos competidores directos como son (JustWatch y FilmAffinity).

Referente a los competidores indirectos que podemos encontrar, los más destacados son las redes sociales, ya que los usuarios también pueden ver trailers y encontrar contenido relacionado con películas en plataformas como YouTube, Twitch.

Analizando el mercado hemos apreciado las siguientes tendencias:

- **Opiniones de usuarios:** Los usuarios de las plataformas valoran las opiniones de otros en plataformas de cine, por lo que integrar un sistema de valoraciones es algo esencial.
- **Uso de web:** Actualmente y cada vez más, se hace uso de internet para poder realizar búsquedas más rápidas sobre películas.

### 3. Diseño y planificación

A la hora de realizar la web se han tenido en cuenta las necesidades para ofrecer un buen nivel de calidad al usuario y para estar previstos para futuros cambios de la aplicación en un futuro.

#### 3.1. Definición de la arquitectura del proyecto

En este proyecto se ha contado con una arquitectura cliente-servidor donde la parte frontend se realizará en React (última versión disponible), con TypeScript, dando la apariencia visual a través de bootstrap. Se tomó la decisión de usar react para poder realizar una SPA (Single Page Application) y por qué el nivel de uso de esta tecnología está muy en alza.

Para la parte backend se hará uso de la tecnología de .NET 8 (versión de larga duración) ya que también es una tecnología muy usada y asentada en la industria. Para realizar la comunicación con la base de datos se hará uso de un ORM (Object Relational Mapping) con objetivo de que dicha comunicación sea lo más separada de la infraestructura de la base de datos, de tal manera que si en un futuro queremos cambiar la base de datos por ejemplo a MySQL nuestra aplicación se vea afectada lo mínimo posible.

Respecto a la base de datos elegida es SQL Server, ya que está muy extendido en el sector y es una base de datos relacional, es el tipo de base de datos más acorde a nuestras necesidades.

#### 3.2. Diseño de la interfaz de usuario

Respecto a la interfaz de usuario se ha buscado siempre cumplir con un diseño sencillo e intuitivo, actualmente se compondrá de las siguientes secciones:

- **Página principal o cartelera:** Se mostrarán todas las películas disponibles, diferenciándose por “en cines” o “próximos estrenos” de tal manera que el usuario pueda navegar a los detalles de cada una de las películas. Además de estas funcionalidades podremos editar y borrar películas si tenemos acceso con un usuario que tiene permisos de administrador (gestión backoffice).
- **Búsqueda de películas:** Una sección de la web dedicada a poder buscar películas por diversos filtros, desde esta página también se podrá acceder al detalle de cada una de las películas.
- **Inicio de sesión:** Sección para ingreso de un usuario, además de poder restablecer contraseñas.
- **Creación de cuentas:** Pagina para poder registrar nuevos usuarios en la web.
- **Solicitud de restablecimiento de contraseña:** Pagina para indicar un correo y solicitar un restablecimiento de contraseña.

- **Cambio de contraseña:** Una vez solicitado un cambio de contraseña, se enviará un correo para poder restablecer esa contraseña, esta página es la encargada de solicitar esa nueva contraseña y activarla.
- **Actores:** Pagina mediante la cual, si eres administrador, tendrás acceso para gestionar los actores de las películas (CRUD).
- **Cines:** Pagina para poder gestionar los cines (CRUD) que se podrán visualizar en la web, solo tendrás acceso si dispones de permisos admin.
- **Creación de películas:** Pagina para poder crear películas para poder visualizar en la web (CRUD), solo se podrá tener acceso a esta página si dispones de permisos de admin.
- **Géneros:** Pagina para poder gestionar los géneros de las películas (CRUD). Solo se podrá tener acceso a esta sección si dispones de permisos admin.
- **Usuarios:** Pagina para poder gestionar los usuarios registrados en la web (CRUD), la parte de creación no se permite desde ese punto ya que no tiene mucho sentido, esta se realiza desde el propio formulario de creación de cuenta.

### 3.3. Planificación de las tareas y los recursos necesarios

Debido a que actualmente solo se compone de un desarrollador, todas las secciones han sido realizadas por este único desarrollador, en caso de disponer de más desarrolladores la planificación pensada para poder realizar las tareas seria la siguiente.

Se realizarían tres equipos:

- **Backend:** Estos recursos se encargarían de realizar la parte de gestión de los datos, conexión con base de datos, integración con otras APIS, etc.
- **Frontend:** Se encargarían de realizar la implementación de la propia web, parte visible y comunicación con la API desarrollada por la parte backend.
- **Testing:** Se encargarían de realizar pruebas de los desarrollos realizados por la parte backend y frontend, en el caso de que encuentren algún fallo lo reportarían a cada uno de los equipos para que creasen la tarea y poder solucionar dicha incidencia. En este punto y al ser el proyecto de pequeña envergadura no se ha planteado la implementación de test automáticos en la parte frontend ni backend.

Para poder realizar las tareas, se implementarían metodologías ágiles, en nuestro caso SCRUM. Definiendo sprints de tiempo reducido que nos permita poder publicar los nuevos desarrollos de forma más rápida, de esta manera también podremos acotar las incidencias que se produzcan y nos resultara más fácil poder solucionarlas.

Además, nos permitirá mejorar la comunicación del equipo y poder motivarlo, para poder tener dicha comunicación se realizarán reuniones diarias para explicar el estado de las tareas y con que esta cada uno en este momento.

## 4. Implementación y pruebas

Durante este bloque se realizará una explicación de cómo se han abordado las pruebas durante el desarrollo del proyecto, además de cómo se podrían afrontar para futuros desarrollos y como se podrían mejorar para realizar pruebas automáticas.

### 4.1. Desarrollo de las funcionalidades del proyecto

En el desarrollo de las funcionalidades se han realizado varias fases:

- Análisis
- Desarrollo
- Pruebas

Vamos a entrar en detalle en la fase de pruebas, una vez realizado el desarrollo se deben definir una batería de pruebas, donde se deben indicar los puntos de la página o del proceso que se deben realizar, un resultado esperado y el resultado real de la prueba.

Una vez se realicen las pruebas de la tarea/ desarrollo se podrá realizar el commit a la rama correspondiente, al finalizar el desarrollo de las tareas del sprint se publicará la web y la API en un entorno de desarrollo donde se volverán a realizar las pruebas de las tareas, pero esta vez lo realizaran los testers y reportaran los bugs encontrados a los equipos de frontend y backend.

### 4.2. Pruebas unitarias y de integración

En este proyecto al no tener una gran envergadura por el momento se ha decidido no realizar pruebas unitarias, integración o end to end, pero si se tiene planteado realizar pruebas para un futuro cuando la web se quiera integrar en producción.

Se tiene planteado aplicar en un futuro test unitario para los casos de usos de la parte backend, de esta manera aseguraremos que todas las nuevas funcionalidades quedan testeadas de forma independiente.

Para las partes compartidas en la parte backend, por ejemplo, Repositorios base se realizarán pruebas de integración, para comprobar que durante el desarrollo de las mejoras no se ha alterado la comunicación de la base de datos.

Respecto a la parte frontend se tiene planteado realizar test e2e (end to end) para asegurar que la aplicación se comporta como se espera durante su desarrollo, comprobando, por ejemplo, un componente se oculta o visualiza según corresponda, poder simular como se debe comportar una página simulando llamadas a la parte backend, etc.



### 4.3. Corrección de errores y optimización del rendimiento

Para la gestión de los errores, se tiene planteado realizar un registro de la tarea indicando la incidencia, este tipo de tareas se realizarán en un sprint urgente, que se publicara cada semana, destacar que solo se integrarán en ese sprint corrección de errores, no tareas.

Para poder gestionar estos errores se debe definir la tarea con el mayor número de detalle para poder reproducirlo.

Referente a las tareas de optimización, se implementarán test de benchmark para poder analizar el tiempo de aquellas consultas a base de datos más costosas, además .NET con EF Core permite mostrar un log con tiempos de cada consulta. Con los tiempos obtenidos podremos analizarlos y ver de dónde viene el problema.

## 5. Documentación

A continuación, se detallará la documentación necesaria para el funcionamiento de este proyecto.

### 5.1. Documentación técnica

Como ya hemos comentado en puntos anteriores, nuestro proyecto estará formado por una página web y una API que nos proveerá los datos, la tecnología seleccionada para la parte web (frontend) ha sido React, ya que actualmente es muy común en proyectos nuevos y además nos permitía poder realizar nuestra web de tipo SPA. Una de las funcionalidades más destacadas de React es la gestión de componentes, esto nos permite separar en trozos muy pequeños y reutilizables nuestra aplicación y comunicarse con el resto de componentes mediante estados y efectos que harán que ese componente se renderice otra vez. Esto nos ayudara a que nuestra página web no tenga que volver a renderizarse entera cada vez que realicemos una interacción con ella, ganando de este modo velocidad.

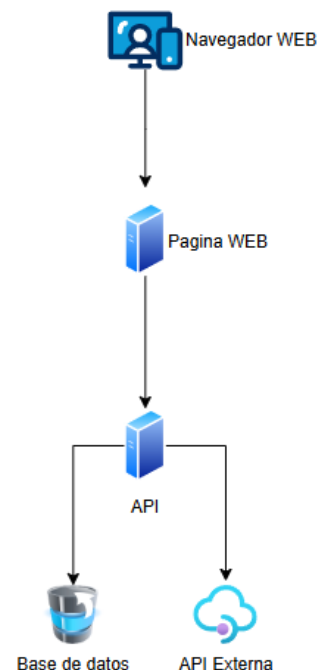
Para la parte de API (backend) hemos elegido .NET 8, ya que es una versión de larga duración de .NET (las versiones pares son de larga duración). Otro motivo por el que hemos apostado por .NET ha sido por su rendimiento y estandarización en el mercado. Existe un amplio mercado de nuget que nos darán la posibilidad de poder reutilizar implementaciones realizadas por terceros y nos aportarán una gran versatilidad a la hora de poder afrontar nuevos retos en nuestra aplicación.

Como base de datos se ha seleccionado SQLServer ya que está muy estandarizado en el mercado, es una base de datos relacional y además aporta un gran rendimiento cuando existe un gran volumen de datos, ya que tenemos planteado tener registrado un gran volumen de películas, cines y usuarios en nuestro proyecto.

El esquema de comunicaciones que necesita nuestra aplicación para poder funcionar correctamente.

Encontraremos las siguientes partes:

- Navegador WEB: Es el navegador WEB desde el cual el usuario ejecutara las llamadas a nuestra aplicación.
- Página WEB: Es la página donde desplegaremos nuestra aplicación, por nuestra parte nos centraríamos en montar un servidor, para poder alojar nuestra página. Nuestro servidor se encontrará alojado en Azure, este también nos proveerá de un nombre de dominio [www.cineparaserfeliz.com](http://www.cineparaserfeliz.com) para poder acceder.
- API: Servidor donde alojaremos nuestra parte backend, nosotros tenemos planteado montarlo en Azure, de este modo contrataríamos un AppService. Desde esta parte backend tendremos acceso a base de datos, que también la tendremos alojada en Azure. En ciertos desarrollos, tal y como tenemos planteados realizarlos en un futuro, necesitaremos realizar una integración con otras APIS, por esta parte no necesitaremos ningún servidor ya que cada integración que queramos realizar ya estará gestionada por la empresa correspondiente.



Para poder dar soporte a la parte frontend se han realizado una serie de endpoints que a continuación se exponen, todos los endpoints expuestos tendrán como inicio de la ruta, por ejemplo, **https://localhost:7290** y a continuación la ruta al endpoint que se necesite.

Para poder dotar una estructura más fácil de entender tanto a la hora de desarrollar como a la hora de poder consumir estos endpoints , se ha realizado una separación agrupando por tema/dominio.

## Actores

Tipo	Ruta	Parámetros
GET	/api/actores	
POST	/api/actores	Nombre - obligatorio – string Biografia – string FechaNacimiento - obligatorio – string Foto - string
GET	/api/actores/{id}	Id – obligatorio - int
PUT	/api/actores/{id}	Id – obligatorio - int Nombre - obligatorio – string Biografia – string FechaNacimiento - obligatorio – string Foto - string
DELETE	/api/actores/{id}	Id – obligatorio - int
GET	/api/actores/buscarPorNombre/{nombre}	Nombre - obligatorio – string

## Cines

Tipo	Ruta	Parámetros
GET	/api/cines	Pagina - obligatorio - int RegistrosPorPagina - obligatorio - int
POST	/api/cines	Nombre - obligatorio – string Latitud – int Longitud – int
GET	/api/cines/{id}	Id – obligatorio - int
PUT	/api/cines/{id}	Id – obligatorio - int
DELETE	/api/cines/{id}	Id – obligatorio - int

## Cuentas

Tipo	Ruta	Parámetros
GET	/api/cuentas/listadoUsuarios	Pagina - obligatorio - int RegistrosPorPagina - obligatorio - int
GET	/api/cuentas/{id}	Id – obligatorio - int
DELETE	/api/cuentas/{id}	Id – obligatorio - int
PUT	/api/cuentas/{id}	Id – obligatorio – int Email - obligatorio - string
POST	/api/cuentas/hacerAdmin	IdUsuario – obligatorio - string
POST	/api/cuentas/quitarAdmin	IdUsuario – obligatorio - string

POST	/api/cuentas/crear	Email - obligatorio - string Password - obligatorio - string
POST	/api/cuentas/login	Email - obligatorio - string Password - obligatorio - string
POST	/api/cuentas/solicitarRestablecimiento	Email - obligatorio - string
POST	/api/cuentas/restablecer	Email - obligatorio - string Password - obligatorio - string Token - obligatorio - string

## Generos

Tipo	Ruta	Parámetros
GET	/api/generos	Pagina - obligatorio - int RegistrosPorPagina - obligatorio - int
POST	/api/generos	Nombre - obligatorio - string
GET	/api/generos/todos	
GET	/api/generos/{id}	Id - obligatorio - int
PUT	/api/generos/{id}	Nombre - obligatorio - string
DELETE	/api/generos/{id}	Id - obligatorio - int

## Películas

Tipo	Ruta	Parámetros
GET	/api/peliculas/landingFilms	
GET	/api/peliculas/{id}	Id - obligatorio - int
PUT	/api/peliculas/{id}	Id - obligatorio - int GenerosIds - int[] CinesIds - int[] Actores - object[], debe contener Id - int y Personaje - string Titulo - obligatorio - string Resumen - string Trailer - string EnCines - bool FechaLanzamiento - string Poster - string
DELETE	/api/peliculas/{id}	Id - obligatorio - int
GET	/api/peliculas /filtrar	Pagina - obligatorio - int RegistrosPorPagina - obligatorio - int Titulo - string Generold - int EnCines - bool ProximosEstrenos - bool
POST	/api/peliculas	Titulo - obligatorio - string Resumen - string

		Trailer – string EnCines – bool FechaLanzamiento – string Poster – IFormFile GenerosIds – int[] CinesIds – int[] Actores – object[], debe contener Id – int y Personaje – string
GET	/api/peliculas/PostGet	
GET	/api/peliculas/PutGet/{id}	Id – obligatorio - int

## Rating

Tipo	Ruta	Parámetros
POST	/api/rating	IdPelicula – obligatorio – int Puntuacion – obligatorio - int

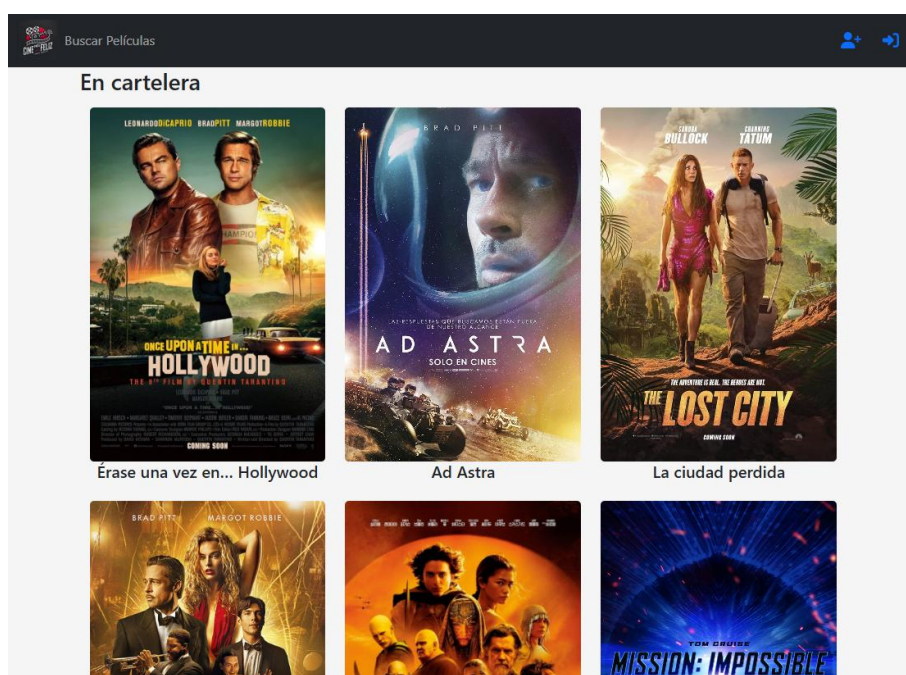
## 5.2. Documentación de usuario

Referente al uso de la aplicación, se va a proceder a explicar todas las páginas que se han realizado para la publicación de la web.

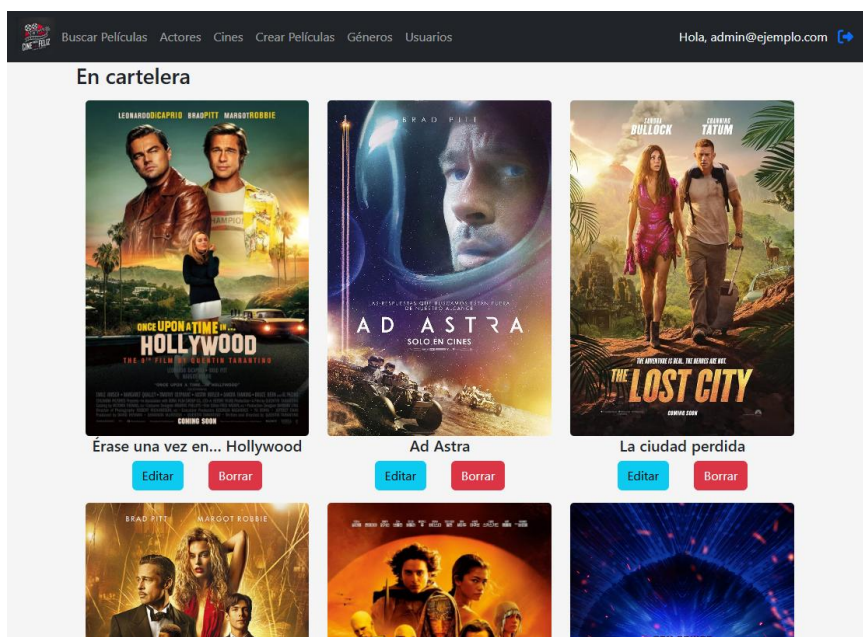
### 5.2.1. *Página principal*

En ella se mostrarán todas las películas dadas de alta en la aplicación, ya se encuentren la cartelera o como próximos estrenos.

Para aquellos usuarios que no son administradores o no están autenticados aparecerá de la siguiente forma, clicando en el poster de cada película se accederá al detalle de cada película.



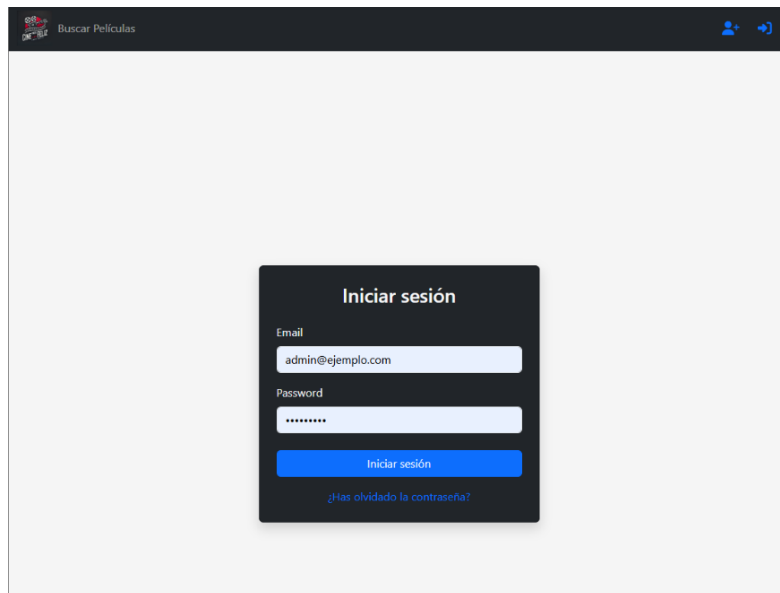
En caso de disponer de permisos de administrador se visualizarán botones para poder editar y borrar la película, además se visualizarán más opciones en el menú superior, como se muestra en la siguiente imagen.



Desde esta página se podrá acceder a la página de login, registro y búsqueda de películas para todos los usuarios.

### 5.2.2. Login

En esta página se dará la funcionalidad para logarse en la aplicación, como función secundaria podremos apreciar que también existe la opción de restablecer contraseña.

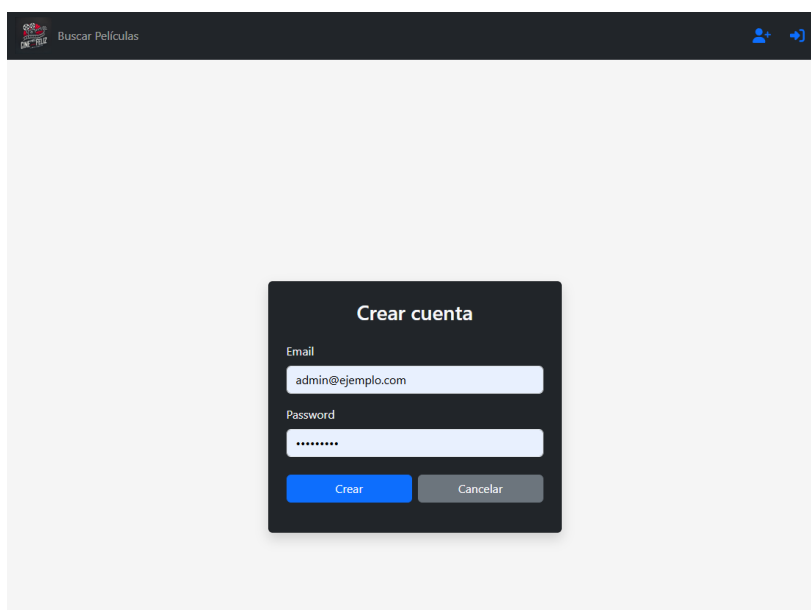


The screenshot shows a web application interface with a dark header bar containing the text 'Buscar Películas' and a user icon. The main content area is light gray. In the center, there is a dark gray modal box titled 'Iniciar sesión'. Inside the modal, there are two input fields: 'Email' with the value 'admin@ejemplo.com' and 'Password' with masked characters '\*\*\*\*\*'. Below the password field is a blue button labeled 'Iniciar sesión'. At the bottom of the modal, there is a link that says '¿has olvidado la contraseña?'.

Una vez introduzcamos las credenciales correctamente esta nos redireccionara a la página principal de la web, que mostrara unas u otras opciones en función del rol que posea el usuario.

### 5.2.3. Registro

Desde esta página podremos crear usuarios para acceder a la aplicación y poder realizar ciertas operaciones a las cuales no tendríamos acceso sino logamos, como puede ser por ejemplo la valoración de películas.



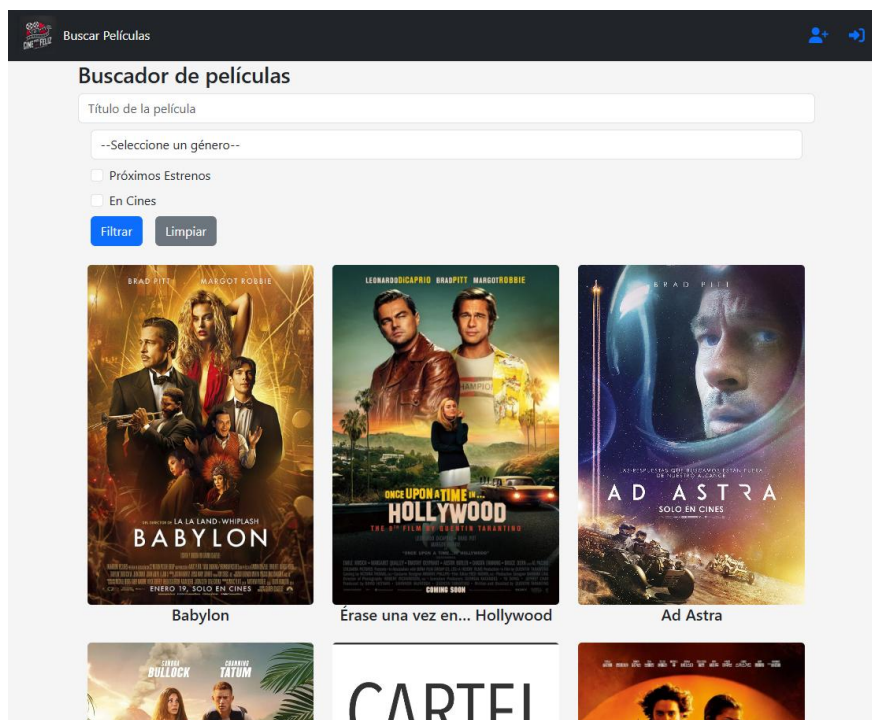
The screenshot shows a web application interface with a dark header bar containing the text 'Buscar Películas' and a user icon. The main content area is light gray. In the center, there is a dark gray modal box titled 'Crear cuenta'. Inside the modal, there are two input fields: 'Email' with the value 'admin@ejemplo.com' and 'Password' with masked characters '\*\*\*\*\*'. Below the password field are two buttons: a blue 'Crear' button and a gray 'Cancelar' button.



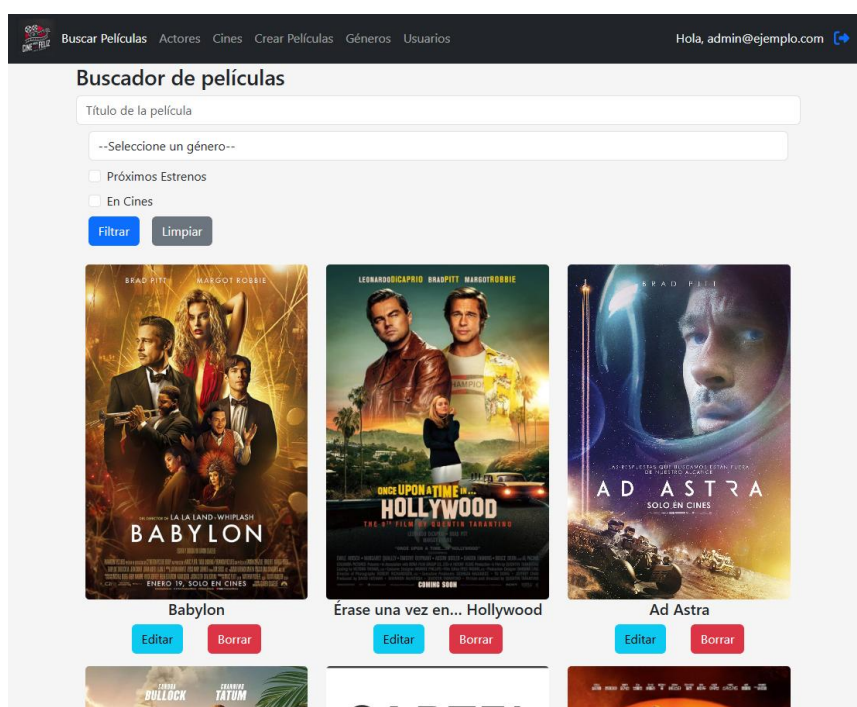
#### 5.2.4. Buscar películas

Desde esta página podremos buscar películas a través de unos filtros en la parte superior, de esta forma facilitaremos al usuario el poder encontrar la película que desea, dicha ventana tiene la misma funcionalidad que la página principal, se puede acceder al detalle de cada película y en el caso de ser administrador se podrá editar y borrar películas.

Este es un ejemplo de cómo se visualizaría con un usuario no administrador o sin estar logado:



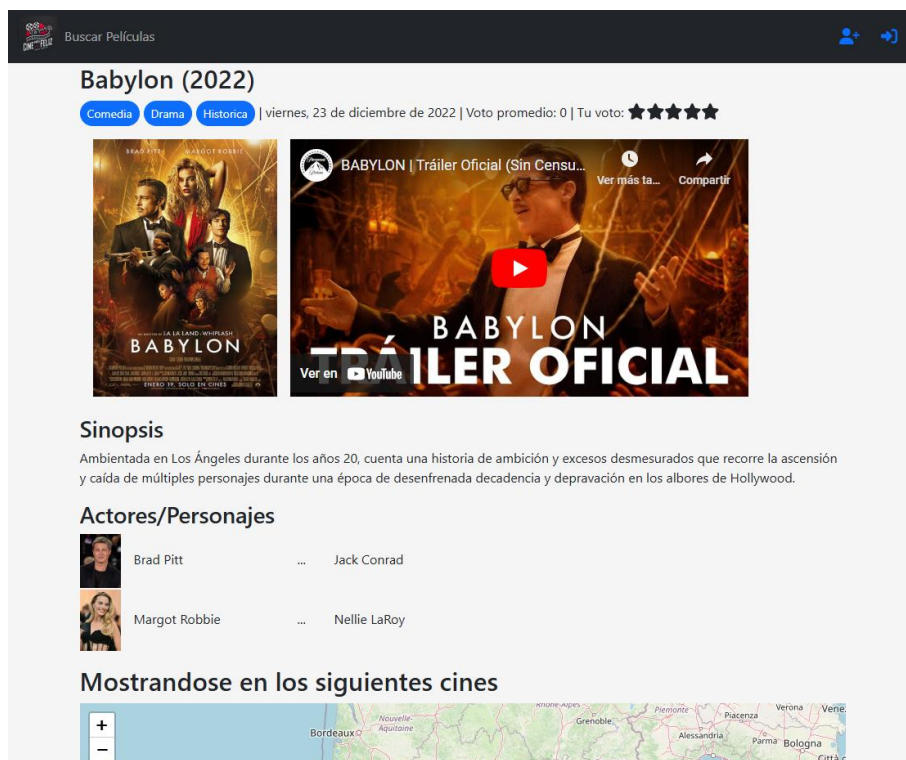
Y la siguiente imagen, muestra como seria de estar logado con un usuario admin:



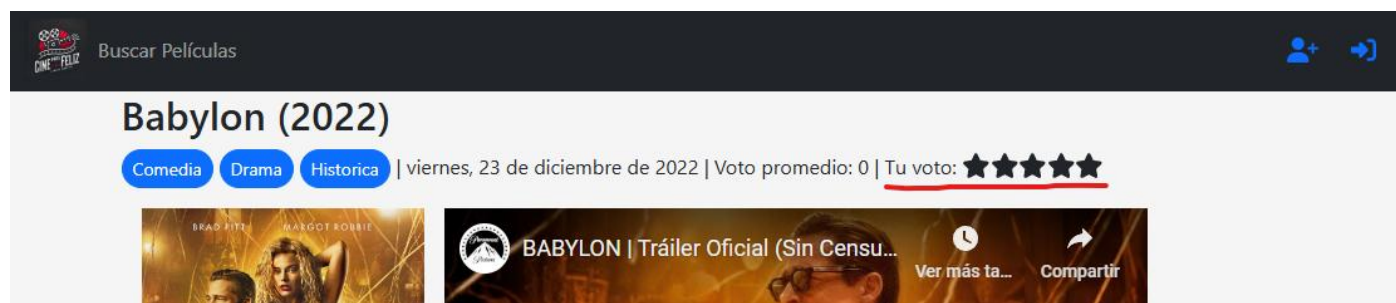


### 5.2.5. Detalle de película

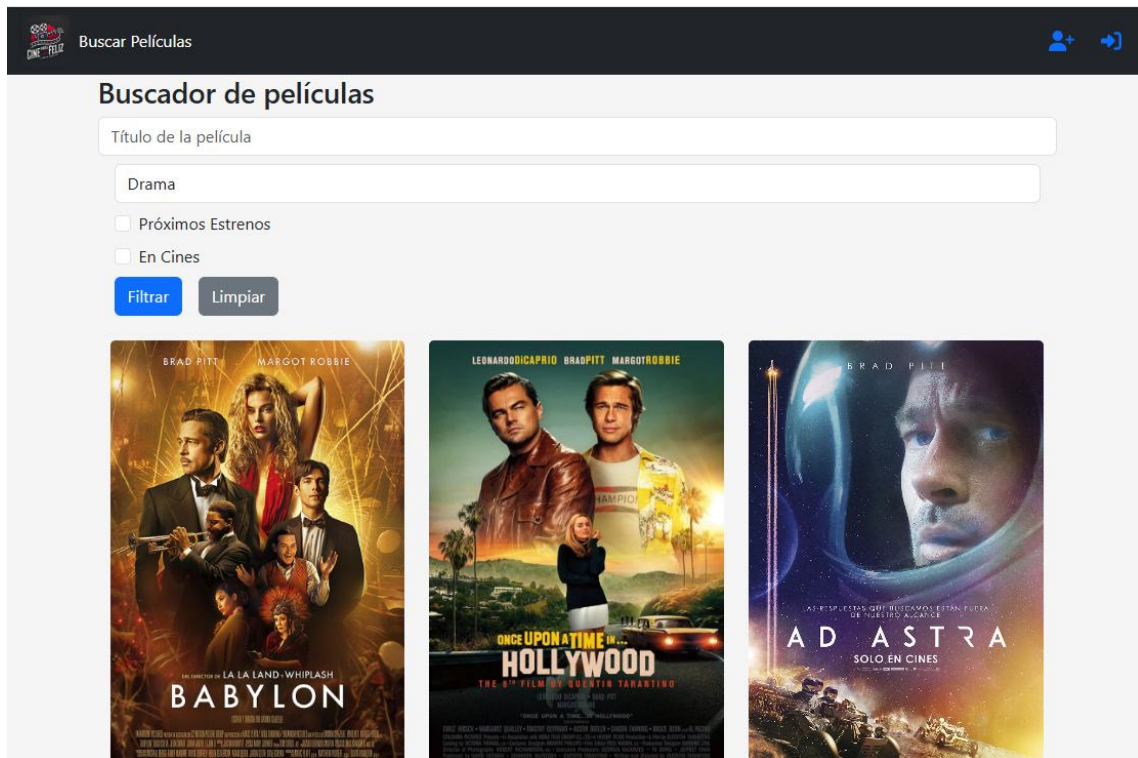
A esta página se accede haciendo clic desde la página de búsqueda o principal de la aplicación, dentro de esta nos mostrará, información de la película, pero sin posibilidad de modificación.



Desde esta podremos establecer una valoración por usuario, pero debemos de estar logado.



También podremos acceder a la búsqueda de películas filtrando por un género seleccionado, por ejemplo, como vemos en la película de la imagen anterior tiene tres géneros Comedia, Drama e Historia, si hacemos clic en cualquiera de esos tres, nos llevara a la búsqueda de películas, pero con un filtro aplicado por defecto del género clicado.



#### 5.2.6. Solicitud de restablecimiento de contraseña

En esta página se podrá indicar un email de cuenta creada, para que se envíe un correo con un link para poder cambiar la contraseña.

### 5.2.7. Restablecimiento de contraseña

En esta página se podrá indicar la nueva contraseña que se quiere establecer en para el email que se había indicado de restablecimiento de contraseña.

The screenshot shows a web application interface with a dark header bar containing a logo, the text 'Buscar Películas', and user icons. The main content area is light gray and features a dark modal box titled 'Crear contraseña'. Inside the modal, there are two input fields: 'Nueva contraseña' and 'Repite contraseña', followed by a blue 'Restablecer' button.

### 5.2.8. Actores

Dicha página nos dotara las funcionalidades necesarias para poder crear, editar, eliminar y visualizar los actores registrados en la aplicación. Además, podremos configurar la paginación que deseamos visualizar en cada momento.

The screenshot displays the 'Actores' management page. The header bar includes navigation links: 'Buscar Películas', 'Actores', 'Cines', 'Crear Películas', 'Géneros', and 'Usuarios'. The user is logged in as 'admin@ejemplo.com'. The page title is 'Actores', and there is a 'Crear Actor' button. Below this is a table with the following data:

		Nombre
Editar	Borrar	Brad Pitt
Editar	Borrar	Brie Larson
Editar	Borrar	Channing Tatum
Editar	Borrar	Charlize Theron
Editar	Borrar	Cliff Curtis
Editar	Borrar	Emma Stone
Editar	Borrar	George Clooney
Editar	Borrar	Hayley Atwell
Editar	Borrar	Iman Vellani
Editar	Borrar	Jason Momoa

At the bottom, there is a pagination control showing 'Anterior', '1' (selected), '2', '3', and 'Siguiente'. Below the pagination is a 'Registros por página' dropdown menu set to '10'.

### 5.2.9. Edición de actores

Desde esta página podremos actualizar la información referente a los actores a través de la interfaz que se muestra en la siguiente imagen:

The screenshot shows the 'Editar Actores' form. At the top, there is a navigation bar with links: 'Buscar Películas', 'Actores', 'Cines', 'Crear Películas', 'Géneros', and 'Usuarios'. The user is logged in as 'Hola, admin@ejemplo.com'. The form title is 'Editar Actores'. It contains three main sections: 'Nombre' with a text input field containing 'Brad Pitt'; 'Fecha Nacimiento' with a date input field containing '18/12/1963' and a calendar icon; and 'Foto' with a file selection button 'Seleccionar archivo' and the text 'Ningún archivo seleccionado'. Below the file selection is a preview of a photo of Brad Pitt.

### 5.2.10. Creación de actores

En esta página tendremos la funcionalidad para poder crear actores con la información que nosotros decidamos.

The screenshot shows the 'Crear Actores' form. It has the same navigation bar as the previous form. The form title is 'Crear Actores'. It contains three main sections: 'Nombre' with an empty text input field; 'Fecha Nacimiento' with a date input field showing the placeholder 'dd/mm/aaaa' and a calendar icon; and 'Foto' with a file selection button 'Seleccionar archivo' and the text 'Ningún archivo seleccionado'. Below the file selection are two large text areas: 'Biografía' and 'Biografía (preview)'. At the bottom of the form are two buttons: 'Guardar' (blue) and 'Cancelar' (red).

### 5.2.11. Cines

Dicha página nos dotara las funcionalidades necesarias para poder crear, editar, eliminar y visualizar los cines registrados en la aplicación. Además, podremos configurar la paginación que deseamos visualizar en cada momento.

Buscar PelículasActoresCinesCrear PelículasGénerosUsuarios

Hola, admin@ejemplo.com

## Cines

Crear Cines

		Nombre
Editar	Borrar	Cines Callao
Editar	Borrar	Cines Renoir Plaza de España
Editar	Borrar	Cines Verdi Barcelona
Editar	Borrar	Cinesa Diagonal Mar
Editar	Borrar	Kinépolis Valencia
Editar	Borrar	Ocine Granollers
Editar	Borrar	Yelmo Cines Ideal

Anterior1Siguiente

Registros por página  
10

### 5.2.12. Edición de cines

En esta página podremos actualizar la información referente a los cines que se encuentran registrados en nuestra aplicación.

Buscar PelículasActoresCinesCrear PelículasGénerosUsuarios

Hola, admin@ejemplo.com


## Editar Cines

Nombre

Cines Callao

+


-




GuardarCancelar

### 5.2.13. Creación de cines

Página para dar poder crear cines y poder posteriormente asociarlos a películas.

 [Buscar Películas](#) [Actores](#) [Cines](#) [Crear Películas](#) [Géneros](#) [Usuarios](#)


Hola, admin@ejemplo.com 

## Crear Cines

Nombre

+

-





Guardar

Cancelar

### 5.2.14. Crear películas

En esta página podremos registrar películas, asociarle actores, géneros y cines para que posteriormente puedan ser visualizadas por el resto de los usuarios desde el detalle de películas.

 [Buscar Películas](#) [Actores](#) [Cines](#) [Crear Películas](#) [Géneros](#) [Usuarios](#)

Hola, admin@ejemplo.com 


## Crear Película

Título

☐ En cines

Trailer

Fecha Lanzamiento



Poster

Seleccionar archivo

Ningún archivo seleccionado

Síntesis

Síntesis (preview)

Géneros:

Acción

Aventura

Catástrofe

Ciencia Ficción

Comedia

Crimen

Documentales

Drama

Espionaje

Fantasia

Historia

Romántico

Suspense

Terror

Thriller

>>

<<

Cines:

Cines Callao

Cines Renoir Plaza de España

Cines Verdi Barcelona

Cinesa Diagonal Mar

Kinépolis Valencia

Ocine Granollers

Yelmo Cines Ideal

>>

<<

Actores:

Escriba el nombre del actor...

Guardar

Cancelar

- Acción
- Aventura
- Catástrofe
- Ciencia Ficción
- Comedia
- Crimen
- Documentales
- Drama
- Espionaje
- Fantasia
- Historica
- Romantico
- Suspense
- Terror
- Thriller



Cines:

Cines Callao  
Cines Renoir Plaza de España  
Cines Verdi Barcelona  
Cinesa Diagonal Mar  
Kinépolis Valencia  
Ocine Granollers  
Yelmo Cines Ideal

>>



Actores:


Escriba el nombre del actor...

Guardar

Cancelar

### 5.2.15. Géneros

Dicha página nos dotara las funcionalidades necesarias para poder crear, editar, eliminar y visualizar los géneros registrados en la aplicación. Además, podremos configurar la paginación que deseamos visualizar en cada momento.

<div>  <a href="#">Buscar Películas</a> <a href="#">Actores</a> <a href="#">Cines</a> <a href="#">Crear Películas</a> <a href="#">Géneros</a> <a href="#">Usuarios</a> </div>			<div>Hola, admin@ejemplo.com</div>	
<h2>Géneros</h2> <div>Crear Género</div>				
			Nombre	
<div> <div>Editar</div> <div>Borrar</div> </div>			Acción	
<div> <div>Editar</div> <div>Borrar</div> </div>			Aventura	
<div> <div>Editar</div> <div>Borrar</div> </div>			Catástrofe	
<div> <div>Editar</div> <div>Borrar</div> </div>			Ciencia Ficción	
<div> <div>Editar</div> <div>Borrar</div> </div>			Comedia	
<div> <div>Editar</div> <div>Borrar</div> </div>			Crimen	
<div> <div>Editar</div> <div>Borrar</div> </div>			Documentales	
<div> <div>Editar</div> <div>Borrar</div> </div>			Drama	
<div> <div>Editar</div> <div>Borrar</div> </div>			Espionaje	
<div> <div>Editar</div> <div>Borrar</div> </div>			Fantasía	
			<div> <div>Anterior</div> <div>1</div> <div>2</div> <div>Siguiente</div> </div>	
<div>Registros por página</div> <div>10</div>				

Hola, admin@ejemplo.com [\[➔\]](#)

## Géneros

Crear Género

		Nombre
Editar	Borrar	Acción
Editar	Borrar	Aventura
Editar	Borrar	Catástrofe
Editar	Borrar	Ciencia Ficción
Editar	Borrar	Comedia
Editar	Borrar	Crimen
Editar	Borrar	Documentales
Editar	Borrar	Drama
Editar	Borrar	Espionaje
Editar	Borrar	Fantasia

Anterior 1 2 Siguiente

Registros por página

10



### 5.2.16. Edición de género

En esta página se da la posibilidad de modificar los datos relacionados con un género ya registrado en nuestra aplicación.

The screenshot shows the 'Editar Géneros' (Edit Genres) page. At the top, there is a navigation bar with links: 'Buscar Películas', 'Actores', 'Cines', 'Crear Películas', 'Géneros' (active), and 'Usuarios'. The user is logged in as 'admin@ejemplo.com'. The main heading is 'Editar Géneros'. Below it, there is a label 'Nombre' and a text input field containing the word 'Aventura'. At the bottom, there are two buttons: 'Guardar' (blue) and 'Cancelar' (red).

### 5.2.17. Creación de género

Podremos registrar nuevos géneros, para posteriormente poder asociarlos a películas.

The screenshot shows the 'Crear Género' (Create Genre) page. It has the same navigation bar as the previous page. The main heading is 'Crear Género'. Below it, there is a label 'Nombre' and an empty text input field. At the bottom, there are two buttons: 'Guardar' (blue) and 'Cancelar' (red).

### 5.2.18. Usuarios

En esta página podremos visualizar los usuarios registrados en nuestra aplicación, además de poder editar, borrar y poder dar o quitar permisos de administrador a cada uno de los usuarios. Como en los anteriores listados se la da la posibilidad de poder aplicar una paginación a gusto del usuario.

The screenshot shows the 'Usuarios' (Users) page. It has the same navigation bar. The main heading is 'Usuarios'. Below it, there is a table with the following data:

				Nombre
Editar	Borrar	Hacer Admin	Quitar Admin	admin@ejemplo.com
Editar	Borrar	Hacer Admin	Quitar Admin	usuario@ejemplo.com
Editar	Borrar	Hacer Admin	Quitar Admin	vic.mor.rub@gmail.com
Editar	Borrar	Hacer Admin	Quitar Admin	victor_moreno_rubio@hotmail.com

Below the table, there is a pagination control with buttons: 'Anterior', '1' (active), and 'Siguiente'. At the bottom left, there is a label 'Registros por página' and a text input field containing the number '10'.



### 5.2.19. Edición de usuario

Desde esta página podremos editar los datos del usuario.



Editar Usuario

Email

usuario@ejemplo.com

Guardar Cancelar

## 5.3. Manual de instalación y configuración

Con el objetivo de facilitar la configuración y ejecución de la aplicación, se han creado herramientas .ps1 para instalar todas las herramientas necesarias y para la ejecución automática tanto de la parte frontend como backend en la carpeta **tools** que podremos encontrar en el root del repositorio.

Bastara con dirigirnos al terminal de windows o powershell con **permisos de administrador** y ejecutar cada uno de los procesos indicados.

### 1.Instalación de dependencias.ps1

### 2.Ejecutar aplicacion.ps1

En caso de que no funcionen correctamente los scripts, adjunto a continuación las dependencias que se deben de instalar manualmente.

Para poder configurar manualmente y ejecutar la aplicación debemos tener instalado los siguientes elementos:

- Node.js
- Docker
- .NET 8
- React

Una vez instalados dichos componentes para poder ejecutar la parte front debemos ejecutar el comando desde el terminal (situados en la ruta de la carpeta frontend react-peliculas) el comando **npm start**, en caso de que nos indique que dicho comando no existe, debemos ejecutar previamente **npm install**

Para poder ejecutar la parte backend se debe recompilar el proyecto en debug, y ejecutar las migraciones de EFCore para poder lanzar las tablas y los datos de prueba a la base de datos. Para ello ejecutaremos el comando **dotnet build** y posteriormente **dotnet ef database update --project Peliculas.API.csproj --startup-project Peliculas.API.csproj --context**

Una vez hecho esto bastara con arrancar la aplicacion a traves de Visual Studio o cualquier IDE que sea compatible que .NET 8.

## 6. Mantenimiento y evolución

A continuación, se explicarán diversos puntos que se tendrán en cuenta para ofrecer mejoras nuevas a los usuarios y soporte en caso de incidencias.

### 6.1. Plan de mantenimiento y soporte

Para ofrecer a nuestros usuarios el mejor soporte y facilidad para poder comunicarse con nosotros en caso de sugerencias e incidencias, se pretende implementar un numero de WhatsApp y un correo electrónico.

El número de WhatsApp estará disponible de 10:00 – 20:00 de lunes a sábados, de esta manera podremos cubrir un amplio periodo de tiempo en el que los usuarios estarán interaccionando con la aplicación.

Respecto al correo electrónico seguirá el mismo horario indicado anteriormente, pero de esta manera ofrecemos otra alternativa para poder comunicarse con nosotros y de esta manera ampliamos el abanico de disponibilidad.

Además, se dispondrá de una monitorización de WEB, API y base de datos para poder detectar cualquier problema de rendimiento o caída de los servicios, de esta manera ofreceremos un tiempo de parada de nuestros servicios mínimo. Ante una incidencia nos comprometemos a ofrecer una solución en 24-48 horas desde el reporte de la incidencia, evitando de esta manera afectar en gran medida a la experiencia de usuario.

### 6.2. Identificación de posibles mejoras y evolución del proyecto

Para un futuro próximo se tienen planteadas diversas mejoras para poder ofrecer todavía más funcionalidades que permitan al usuario gestionar todos los tramites desde nuestra web.

- Integración de compra de entradas.
- Visualización de horarios de películas en cines.
- Inicio de sesión por redes sociales.
- Diversas optimizaciones de rendimiento.
- Corrección de errores.
- Análisis de incidencias para evitar recurrencias.
- Realización de pruebas rutinarias para evitar fallos.
- Actualización de frameworks para evitar problemas de seguridad e incidencias provenientes de estas.

### 6.3. Actualizaciones y mejoras futuras

Durante la evolución de la aplicación se tienen previsto realizar las siguientes mejoras y actualizaciones, con el objetivo de mejorar tanto a nivel de aplicación como de experiencia de usuario.

- **Actualizaciones regulares:** Se realizarán actualizaciones solucionando errores de forma regular.
- **Notificaciones:** Notificar a los usuarios a través del correo electrónico con objetivo de informar sobre cambios relevantes.
- **Introducción de políticas de seguridad:** Continuar instaurando las políticas de seguridad más estricta con objetivo de ofrecer a los usuarios los menores problemas de seguridad posibles.
- **Actualizaciones de dependencias:** Ya que nuestro proyecto tiene dependencias de terceros, se realizarán actualizaciones regulares para evitar vulnerabilidades y poder aplicar nuevas herramientas disponibles.
- **Integración de compra de entradas:** Actualmente se pueden visualizar películas y los cines donde se pueden encontrar, pero no podemos gestionar reservas a través de nuestra WEB, se quiere realizar diversas integraciones con API de cines, para poder ofrecer la posibilidad de realizar la compra a través de nuestra WEB y no tener que realizarlo desde la propia WEB del cine correspondiente.
- **Visualización de horarios de películas en cines:** Se pretende que en nuestra WEB se puedan visualizar los horarios de la película por cada cine en los que se encuentra disponible. De esta manera los usuarios podrán obtener información muy útil sin salir de nuestra WEB.
- **Inicio de sesión por redes sociales:** En muchas WEB se da la facilidad al usuario de poder logarte con unas credenciales de alguna red social (Facebook, X, LinkedIn, etc) sin tener que estar gestionando una cuenta nueva y su contraseña correspondiente. Con este cambio facilitaremos al usuario el poder entrar en nuestra WEB sin necesidad de tener que recordar otros credenciales.
- **Multilingüe:** Actualmente solo se dispone de traducción en español, en un futuro se quiere realizar dicha aplicación de forma multilingüe para poder ampliar todavía más nuestro abanico de usuarios.

## 7. Conclusiones

### 7.1. Evaluación del proyecto

El desarrollo del proyecto de la web de cine ha sido llevado a cabo cumpliendo las etapas planeadas y siguiendo la metodología establecida. Se ha logrado implementar un sistema funcional que permite a los usuarios buscar información sobre películas y realizar valoraciones. Sin embargo, durante el desarrollo se identificaron áreas que podrían ser mejoradas, como la optimización de tiempos de respuesta en consultas complejas y la posibilidad de poder realizar comunicaciones con APIs de terceros con objetivo de poder realizar la compra de entradas.

### 7.2. Cumplimiento de objetivos y requisitos

Los objetivos principales de la aplicación que se realizaron correctamente fueron:

- Se cumplieron todos los requerimientos iniciales, como la búsqueda avanzada de película, sistema de valoración de películas y gestión de usuarios.
- Se implementó una interfaz intuitiva y atractiva, adaptada para dispositivos móviles y de escritorio.
- El sistema soporta un volumen de usuarios concurrentes dentro de los límites esperados.

### 7.3. Lecciones aprendidas y recomendaciones para futuros proyectos

1. **Gestión del tiempo:** A pesar de realizar una planificación inicial detallada, existieron algunas tareas tomaron más tiempo del esperado. Para futuros proyectos, se recomienda asignar márgenes de tiempo más amplios para imprevistos.
2. **Pruebas automáticas:** Incorporar pruebas automáticas desde etapas tempranas podría haber ayudado a la detección temprana de errores, en nuestro caso esos errores se vieron en un plazo mayor lo cual implicó un tiempo mayor de pruebas.
3. **Adaptabilidad:** Algunos cambios en los requerimientos surgieron a mitad del desarrollo. Contar con un margen de flexibilidad fue crucial para adaptarse a estas nuevas demandas.

## 8. Bibliografía y referencias

### 8.1. Fuentes utilizadas en el proyecto

Durante el proceso de realización del proyecto se han realizado diversas búsquedas, ya sean por dudas de como realizar un proceso o de como integrar herramientas de terceros. Se adjunta la lista de fuentes utilizadas:

1. [Documentación React](#)
2. [Instalación Node.js](#)
3. [Documentación Bootstrap](#)
4. [Documentación Axios](#)
5. [SweetAlert2](#)
6. [Documentación Formik](#)
7. [Información Yup](#)
8. [ASP.NET Core](#)
9. [Documentación Leaflet](#)
10. [Información sobre typeahead](#)

### 8.2. Referencias y enlaces de interés

Con objetivo de estar al tanto de noticias de interés que afectan al proyecto se recomiendan los siguientes enlaces relevantes:

- [Noticias sobre React](#)
- [Novedades .NET 8](#)
- [Información de seguridad .NET 8](#)