

# Documentación Evaluación

Aplicación LikeMovies con  
consumo de la API de TMDB

---

Israel Brea Piñero

Máster MIMO - Tecnologías del  
lado del cliente: HTML5

# Documentación

---

## Tabla de contenido

<b>1. Motivación.....</b>	<b>2</b>
<b>2. Objetivos .....</b>	<b>2</b>
<b>3. Catálogo de requisitos .....</b>	<b>2</b>
<b>3.1. Requisitos funcionales .....</b>	<b>2</b>
<b>3.2. Requisitos no funcionales.....</b>	<b>2</b>
<b>4. Descripción funcional de la aplicación .....</b>	<b>3</b>
<b>5. Descripción técnica de la aplicación .....</b>	<b>7</b>
<b>5.1. Servicios.....</b>	<b>7</b>
5.1.1. Consumo de la API de TMDb (The Movie Database). MoviesapiService.....	7
5.1.2. Autenticación con Firebase Authentication. AuthService.....	8
5.1.3. Persistencia de datos con Ionic Storage.....	9
<b>5.2. LoginPage .....</b>	<b>9</b>
<b>5.3. SignupPage .....</b>	<b>10</b>
<b>5.4. ListadoPage .....</b>	<b>10</b>
<b>5.5. EstrenosPage .....</b>	<b>11</b>
<b>5.6. PerfilPage .....</b>	<b>12</b>
<b>5.7. DetallePelicula .....</b>	<b>13</b>
<b>5.8. PeliculasComponent.....</b>	<b>14</b>
<b>5.9. SwiperPeliculasComponent .....</b>	<b>14</b>
5.9.1. Swiper.js .....	14
<b>5.10. PeliculaComponent.....</b>	<b>14</b>
<b>5.11. Auth(Guard).....</b>	<b>14</b>
<b>5.12. Responsividad.....</b>	<b>14</b>

## 1. Motivación

En la era digital, donde el acceso a la información y el entretenimiento está al alcance de todos, muchas personas enfrentan el desafío de organizar sus intereses y prioridades. En particular, la acumulación de una lista interminable de películas por ver se ha convertido en una experiencia común, creando una necesidad de contar con una herramienta funcional que permita gestionar y encontrar información relevante de manera eficiente. Este proyecto busca resolver este problema al proporcionar una aplicación intuitiva y útil, desarrollada con tecnologías modernas como Angular e Ionic, que ayude a los usuarios a organizar y explorar su catálogo personal de películas.

## 2. Objetivos

La aplicación tiene como propósito ofrecer a los usuarios una plataforma intuitiva para gestionar películas que desean ver, estrenos próximos y sus preferencias personales. Surge de la necesidad de organizar y priorizar listas de películas de manera eficaz, evitando el desorden y facilitando un acceso rápido a información relevante.

## 3. Catálogo de requisitos

### 3.1. Requisitos funcionales

**RF-001. Listado de películas por género:** Ver un listado de las 20 primeras películas obtenidas consumiendo la API, filtradas por género.

**RF-002. Buscar película:** La aplicación permitirá buscar películas mediante una barra de búsqueda.

**RF-003. Detalle película:** La aplicación permitirá mostrar en detalle una película.

**RF-004. Listado de películas sin estrenar:** La aplicación permitirá mostrar TODAS las películas que no se hayan estrenado aún en España.

**RF-005. Log in y sign up:** La aplicación deberá permitirme iniciar sesión mediante email y contraseña. Si no tengo cuenta deberá permitirme crear una nueva.

**RF-006. Añadir o eliminar favoritos:** La aplicación deberá permitir añadir o eliminar una película de favoritos.

**RF-007. Añadir o eliminar a Mi Lista:** La aplicación deberá permitir añadir o eliminar una película de Mi Lista.

**RF-008. Añadir o eliminar a recordatorios:** La aplicación deberá permitir añadir o eliminar una película de recordatorios.

**RF-009. Mostrar favoritos:** La aplicación deberá mostrar todas las películas que están en favoritos.

**RF-010. Mostrar Mi Lista:** La aplicación deberá mostrar todas las películas que están en Mi Lista.

**RF-011. Mostrar recordatorios:** La aplicación deberá mostrar todas las películas que están en recordatorios.

### 3.2. Requisitos no funcionales

**RNF-001. Seguridad:** El sistema debe estar protegido contra el acceso no autorizado.

**RNF-002. Responsividad:** La interfaz debe ser capaz de adaptarse a los diferentes tamaños de dispositivo.

## 4. Descripción funcional de la aplicación

La información sobre las películas ha sido tomada a partir de la API The Movie Database (TMDB).

### Pantalla de inicio de sesión:

- El usuario podrá iniciar sesión utilizando su correo electrónico y contraseña.
- Si no tiene una cuenta, podrá pulsar en “Regístrate aquí” que hará que navegue a una nueva pantalla que le permitirá registrarse.
- Tras iniciar sesión correctamente, el usuario será redirigido a la pantalla principal.

Iniciar Sesión

Bienvenido

Correo Electrónico

Contraseña

INICIAR SESIÓN

¿No tienes una cuenta? [Regístrate aquí](#)

### Pantalla de registro:

- El usuario podrá registrarse utilizando su correo electrónico y contraseña.
- Si ya tiene una cuenta, podrá pulsar en “iniciar sesión” para navegar de nuevo a la pantalla de Inicio de sesión.
- Tras el registro, el usuario será redirigido a la pantalla principal de la aplicación con un cuadro de Registro exitoso.



Registrarse

Crear una cuenta

Correo Electrónico

Contraseña

REGISTRARSE

¿Ya tienes una cuenta? [Inicia sesión aquí](#)

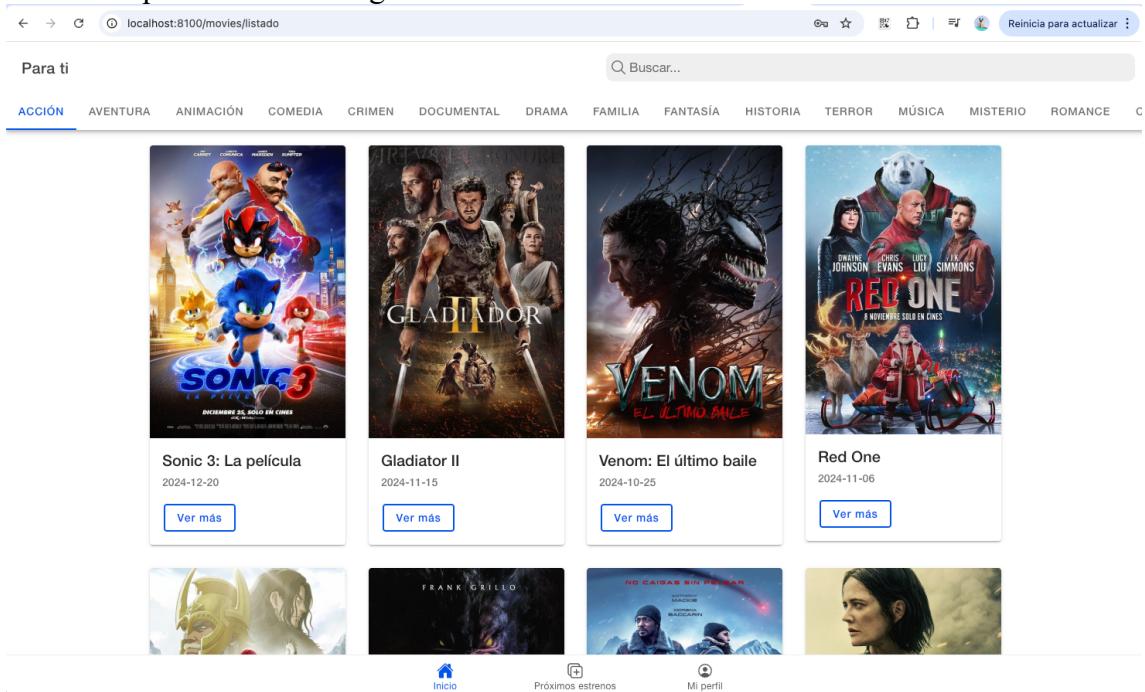
### Sistema de navegación mediante tabs:

- En la parte inferior de la pantalla, se encuentran tres botones (tabs) que permiten navegar entre las tres secciones principales de la aplicación: Para ti, estrenos y novedades y Mi perfil.



### Sección de Para ti:

- Esta es la pantalla por defecto al iniciar sesión.
- En la parte superior se encuentra una barra de herramientas (toolbar) con el nombre de la sección y una barra de búsqueda.
- La barra de búsqueda permite localizar cualquier película existente en la base de datos.
- Debajo de la barra de herramientas hay un segmento deslizante para filtrar películas por géneros.
- Las películas se presentan en formato de cuadrícula. Se mostrarán las primeras 20 películas de cada género.



### Página de Detalle:

- Al seleccionar una película, se abre una vista detallada con información concreta de la película como:
  - Título.
  - Fecha de lanzamiento.
  - Géneros.
  - Duración.
  - Valoración.
  - Descripción.
- Desde esta vista, el usuario podrá:
  - Añadir o eliminar la película de favoritos.
  - Añadir o eliminar la película de su lista de pendientes.

← Mufasa: El rey león



### Mufasa: El rey león (2024)

18/12/2024 • Aventura, Familia, Animación • 118 min

75%

Puntuación de usuarios

#### Vista general

Rafiki debe transmitir la leyenda de Mufasa a la joven cachorro de león Kiara, hija de Simba y Nala, y con Timón y Pumba prestando su estilo característico. Mufasa, un cachorro huérfano, perdido y solo, conoce a un simpático león llamado Taka, heredero de un linaje real. Este encuentro casual pone en marcha un viaje de un extraordinario grupo de inadaptados que buscan su destino.

ELIMINAR DE FAVORITOS

AÑadir a mi lista

## Sección de estrenos y novedades:

- Muestra todas las películas que se estrenarán próximamente en España.
- Cada película tiene la opción de "recordar más tarde" para configurar un recordatorio.
- Los recordatorios de películas se agregarán a la sección de notificaciones.
- Las notificaciones se encuentran en la parte superior derecha de la aplicación y muestran una por cada recordatorio configurado.

localhost:8100/movies/estrenos

Próximos estrenos

Flow, un mundo que salvar  
2025-01-24  
Recordatorio programado

Dirty Angels  
2025-01-24  
Recordármelo

Babygirl  
2025-01-17  
Recordatorio programado

The Brutalist  
2025-01-24  
Recordármelo

El teléfono negro  
La tercera edad

Inicio Próximos estrenos Mi perfil

## Sección de mi perfil:

- Esta sección permite al usuario gestionar su información personal.
- Incluye tres listas principales:
  - Lista de favoritos.
  - Lista de Mi Lista.
  - Lista de recordatorios de futuros estrenos.
- En la parte superior derecha del toolbar se encuentra un menú desplegable que permite cerrar sesión.

- Al cerrar sesión, el usuario es redirigido automáticamente a la pantalla de inicio de sesión.

The image consists of two vertically stacked screenshots of a web-based movie recommendation application. Both screenshots have a header bar at the top with browser controls, a search bar containing 'localhost:8100/movies/perfil', and a user profile icon labeled 'usuario1'. A 'Reinicia para actualizar' button is also present in the top right corner.

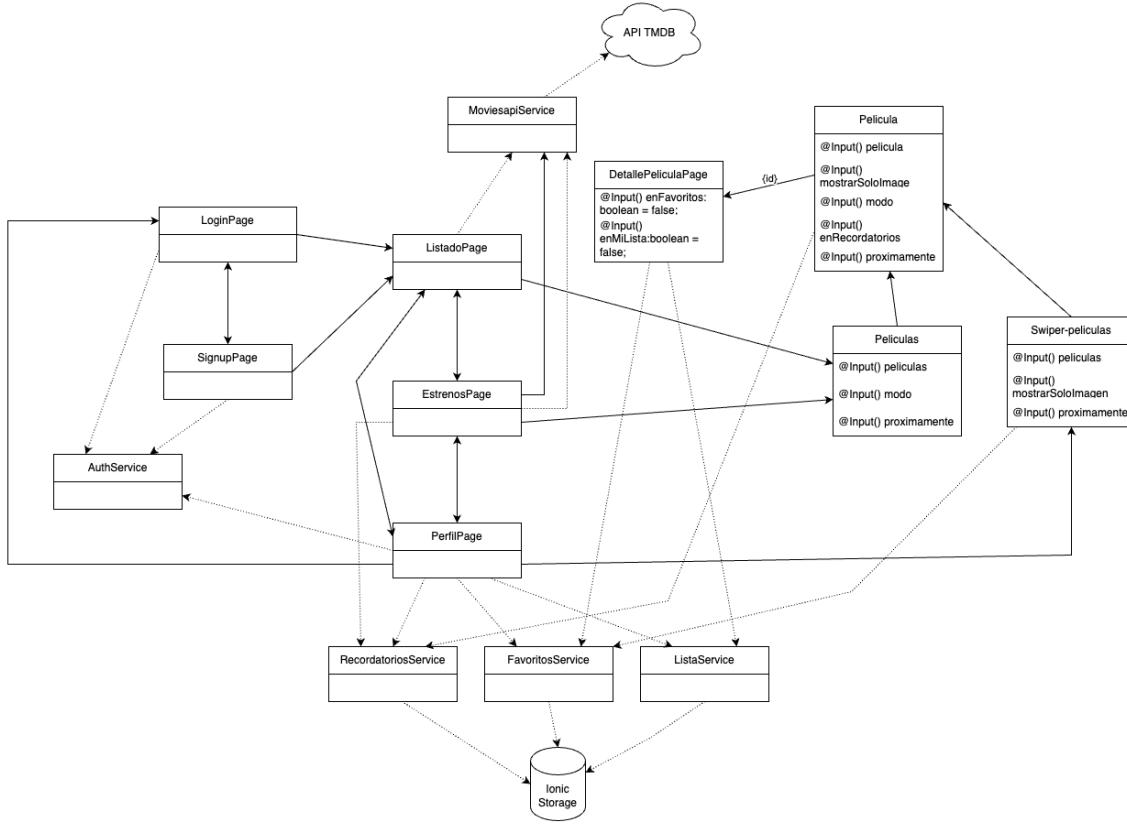
**Screenshot 1 (Top):** This shows the user's profile page ('Mi perfil'). It features a large placeholder profile picture, the user's name 'usuario1' in blue text, and sections for 'Mis favoritos' (Favorite movies) and 'Recordatorios' (Reminders). Under 'Mis favoritos', there are six movie posters: 'WICKED', 'RED ONE', 'MUFASA: EL REY LEON', 'HERETIC', 'SONIC 3', and 'VAIANA 2'. Below these sections are buttons for 'Inicio', 'Próximos estrenos', and 'Mi perfil'.

**Screenshot 2 (Bottom):** This shows the same profile page, but with a modal dialog box centered over the movie posters. The dialog box has a dark background and contains the text 'Cerrar sesión' (Logout) at the top, followed by the question '¿Estás seguro de que quieres cerrar sesión?' (Are you sure you want to log out?). At the bottom of the dialog are two buttons: 'CANCELAR' (Cancel) and 'CERRAR SESIÓN' (Logout).

### Seguridad de acceso:

- Si un usuario intenta acceder a cualquier sección sin estar autenticado, será redirigido a la pantalla de inicio de sesión.
- Este sistema garantiza que solo los usuarios autenticados puedan utilizar las funcionalidades de la aplicación.

## 5. Descripción técnica de la aplicación



El componente Tabs se encuentra en la parte inferior y es la base de la navegación de la aplicación, es accesible desde cualquier vista. Se utiliza para acceder a las diferentes secciones principales de la aplicación: Inicio (Listado de Películas), Próximos Estrenos, y Mi Perfil.

### 5.1. Servicios

#### 5.1.1. Consumo de la API de TMDB (The Movie Database). MoviesapiService.

La API de TMDB es un servicio ofrecido por The Movie Database que permite a desarrolladores acceder a una base de datos enorme de contenido relacionado con películas, series de televisión y actores. Esta API es ampliamente utilizada en aplicaciones que requieren información cinematográfica, como catálogos de películas, sistemas de recomendación, aplicaciones de streaming, entre otros.

La API está basada en REST y utiliza JSON como formato de respuesta. Para consumirla, es necesario obtener una clave de API (API Key), que puede ser gratuita para uso personal o con planes pagos para acceso avanzado.

La API de TMDB proporciona datos estructurados sobre:

1. Películas:

- Información básica: título, sinopsis, fecha de lanzamiento, etc.
- Detalles avanzados: presupuesto, duración, calificaciones, etc.

- Géneros asociados.
2. **Series de televisión:**
    - Temporadas, episodios, fechas de emisión, sinopsis y más.
  3. **Personas:**
    - Información sobre actores, directores, guionistas y otros.
  4. **Imágenes y videos:**
    - Pósters, fondos de pantalla, fotos promocionales y trailers.
  5. **Géneros:**
    - Listas predefinidas de géneros cinematográficos y televisivos.
  6. **Tendencias:**
    - Contenido popular y en tendencia.
  7. **Colecciones:**
    - Conjuntos de películas relacionadas, como sagas o franquicias.

Los principales tipos de consultas son:

- El endpoint (/discover/movie) que permite filtrar películas según varios criterios:
  - Género
  - Año de lanzamiento
  - Idioma
  - País de producción
  - Popularidad, calificación promedio, número de votos, etc.
- Este endpoint (/search/movie) que permite buscar películas utilizando una consulta textual (query). Ideal para implementar barras de búsqueda.
- El endpoint (/movie/{movie\_id}) que permite recuperar información detallada de una película específica mediante su ID.

## MoviesapiService

El servicio MoviesapiService encapsula toda la lógica para interactuar con la API de TMDB, garantizando un acceso estructurado a sus diferentes funcionalidades.

- Interactúa con la API de películas para obtener datos como:
  - Películas populares.
  - Películas por género.
  - Detalles de una película específica.
  - Resultados de búsqueda.
  - Películas sin estrenar.
- Utiliza el módulo HttpClient de Angular para realizar peticiones HTTP.
- Para gestionar las peticiones asíncronas usa observables.
- Gestiona la paginación y el filtrado de géneros, manteniendo el estado entre sesiones.

### 5.1.2. Autenticación con Firebase Authentication. AuthService.

Firebase Authentication fue elegido por ser una solución sencilla y confiable para manejar el inicio de sesión y registro de usuarios. Ofrece soporte para múltiples

métodos de autenticación, integración fácil con Angular y asegura el manejo seguro de credenciales sin necesidad de infraestructura adicional.

### **AuthService**

El servicio AuthService centraliza y abstrae la gestión de autenticación del usuario en la aplicación utilizando Firebase Authentication. Proporciona métodos para iniciar sesión, registrarse, cerrar sesión y observar el estado de autenticación del usuario.

Este servicio incluye las siguientes funcionalidades:

- Inicio de sesión.
- Registro de nuevos usuarios.
- Cierre de sesión.
- Obtención del usuario autenticado actual.

El servicio gestiona el estado del usuario de la siguiente manera:

- Utiliza un BehaviorSubject para exponer el estado del usuario autenticado a través de un observable (user\$).
- Actualiza automáticamente el estado del usuario cuando hay cambios en Firebase Authentication mediante onAuthStateChanged.

### **5.1.3. Persistencia de datos con Ionic Storage**

Ionic Storage se seleccionó por su capacidad de gestionar datos localmente de forma eficiente y adaptarse automáticamente según la plataforma (web o móvil). Es fácil de usar, funciona bien con Ionic, y permite guardar información de manera persistente entre sesiones.

### **RecordatoriosService**

El servicio RecordatoriosService se encarga de gestionar las películas que los usuarios desean recordar. Utiliza Ionic Storage para la persistencia de datos y Angular Signals para reactividad, permitiendo actualizar automáticamente las vistas cuando los recordatorios cambian.

### **FavoritosService**

El servicio FavoritosService gestiona la funcionalidad de favoritos de la aplicación, permitiendo a los usuarios añadir, eliminar y verificar si una película está en su lista de favoritos. Utiliza el almacenamiento local (Ionic Storage) para persistir los datos entre sesiones y Angular Signals para reactividad.

### **ListaService**

El servicio ListaService gestiona la funcionalidad de añadir a Mi Lista de la aplicación, permitiendo a los usuarios añadir, eliminar y verificar si una película está en su lista. Utiliza el almacenamiento local (Ionic Storage) para persistir los datos entre sesiones y Angular Signals para reactividad.

## **5.2. LoginPage**

Nos encontramos con un componente ion-header en la parte superior que incluye un ion-toolbar con el título de la sección: "Iniciar Sesión".

Dentro del contenido (ion-content), encontramos un ion-card que contiene:

1. Un encabezado (ion-card-header) con el mensaje "Bienvenido".
2. Un formulario con:

- Un campo de entrada para el correo electrónico (ion-input) con una etiqueta flotante.
  - Un campo de entrada para la contraseña, que incluye un botón para mostrar/ocultar la contraseña.
3. Un mensaje de error (ion-text) que se muestra si los datos de inicio de sesión son incorrectos.
  4. Un botón (ion-button) que al hacer clic ejecuta la función de inicio de sesión (login()). Esta función es la que utiliza el servicio de autenticación(AuthService) e indica si se ha iniciado sesión correctamente.

Debajo del formulario, hay un texto que redirige a la página de registro mediante un enlace (routerLink).

### 5.3. SignupPage

**ion-header:** Contiene un ion-toolbar en la parte superior con el título "Registrarse".

**ion-content:** Dentro hay un ion-card con:

1. **Encabezado** (ion-card-header) con el título "Crear una cuenta".
2. **Formulario:**
  - Campo de entrada para correo electrónico (ion-input) con validación asociada. Si el correo no es válido, se muestra un mensaje de error debajo del campo utilizando un ion-text con \*ngIf="emailError".
  - Campo de entrada para contraseña con un botón para mostrar/ocultar la contraseña. Si la contraseña es inválida, se muestra un mensaje de error de forma similar al anterior.
3. **Botón de registro** (ion-button):
  - Ejecuta la función register() al ser presionado.
  - Dentro de esta función:
    - Valida los datos.
    - Si son válidos, llama al método authService.register() para registrar al usuario.
    - Si el registro es exitoso, muestra un mensaje de confirmación con un alertController y redirige al usuario a la página principal usando router.navigate([""]).
    - Si ocurre un error, muestra un mensaje de alerta con los detalles del problema.
4. **Enlace de redirección:** Texto que redirige a la página de inicio de sesión (routerLink="/login").

### 5.4. ListadoPage

**Componentes Usados:**

- **Cabecera (ion-header):**
  - Contiene un título principal, "Para ti", que define el propósito de la página.
  - Incluye un buscador (ion-searchbar) en la parte superior, dentro de la barra de herramientas, permitiendo al usuario buscar películas por su nombre. La barra de búsqueda es interactiva, reaccionando tanto al escribir como al borrar texto.

- Debajo de la barra de herramientas, se encuentra un segmento (ion-segment), que permite filtrar las películas por género. Cada botón dentro del segmento representa un género cinematográfico y está vinculado a un valor que permite cargar las películas correspondientes.
- **Cuerpo Principal (ion-content):**
  - Muestra el componente personalizado app-peliculas, que se encarga de renderizar las películas obtenidas según el género seleccionado o la búsqueda realizada. Este componente encapsula la lógica de presentación, dejando a ListadoPage el trabajo de manejar la obtención de datos.

#### Comunicación entre Componentes:

- **Navegación Interna:**
  - Al seleccionar un género en el segmento, el evento (ionChange) activa el método cambioGenero(). Este método actualiza la lista de películas mediante el servicio MoviesapiService.
  - Al escribir en el buscador, el evento (ionInput) llama al método buscarPelícula(), que realiza una búsqueda de películas en el servicio. Si el campo de búsqueda se vacía, se vuelve a cargar la lista del género seleccionado actualmente.
- **Interacción entre Componentes:**
  - La comunicación entre ListadoPage y el componente PeliculasComponent se realiza mediante filiación: PeliculasComponent recibe la lista de películas como entrada a través de la propiedad [peliculas].
  - La obtención de películas según género o búsqueda utiliza el servicio MoviesapiService, que conecta la página con la API de TMDb.

## 5.5. EstrenosPage

#### Describir las Partes de la Pantalla Relacionadas con los Componentes:

- **Cabecera (ion-header):**
  - Incluye un título, "Próximos estrenos", indicando el propósito de la página.
  - Un botón de notificaciones en el lado derecho muestra un ícono con un badge (ion-badge) que indica la cantidad de recordatorios actuales.
  - Al pulsar el botón de notificaciones, se despliega un popover (ion-popover) que muestra una lista de recordatorios. Si no hay recordatorios, se informa al usuario mediante un mensaje.
- **Cuerpo Principal (ion-content):**
  - Contiene el componente app-peliculas, que presenta las películas con la funcionalidad adicional de permitir al usuario añadir recordatorios. Esta funcionalidad se controla mediante la entrada [modo]=""recordar"".
  - Incluye un componente de scroll infinito (ion-infinite-scroll), que permite cargar más datos al desplazarse hacia abajo. Esta funcionalidad mejora la experiencia de usuario al presentar más películas sin necesidad de recargar la página.

#### Comunicación entre Componentes:

- **Navegación Interna:**

- Al interactuar con el componente de scroll infinito, el evento (ionInfinite) llama al método onIonInfinite(), que utiliza el servicio MoviesapiService para cargar más películas populares.
- Al interactuar con los recordatorios (dentro del popover), no hay navegación explícita, pero el usuario puede identificar las películas listadas y buscar más detalles si lo desea.
- **Interacción entre Componentes:**
  - Similar a ListadoPage, la comunicación con PeliculasComponent se realiza mediante **filiación**, utilizando las propiedades [peliculas] y [modo].
  - El manejo de recordatorios se realiza mediante el servicio RecordatoriosService, que actúa como intermediario entre la lógica de la página y el almacenamiento de recordatorios. La propiedad computada numItemsRecordatorios obtiene el número total de recordatorios directamente desde este servicio.

## 5.6. PerfilPage

Describir las partes de la pantalla relacionadas con los componentes:

1. **Cabecera (ion-header):**
  - Incluye un título: "Mi perfil".
  - Un ion-chip en el lado derecho muestra el nombre del usuario y un avatar predeterminado. El nombre se calcula a partir del email del usuario o muestra "Invitado" si no hay sesión activa.
  - Un botón de menú (ion-menu-button) en el lado derecho despliega un menú lateral con opciones.
  - El menú lateral (ion-menu):
    - Contiene un encabezado con el título "Opciones".
    - Una lista de opciones que incluye un único elemento: "Cerrar sesión". Este activa un ion-alert para confirmar la acción.
2. **Cuerpo principal (ion-content):**
  - **Sección de perfil:**
    - Incluye un avatar centralizado y el nombre del usuario.
  - **Favoritos:**
    - Muestra el componente app-swiper-peliculas con las películas favoritas del usuario, obtenidas desde el servicio FavoritosService.
    - Si no hay películas favoritas, se muestra el mensaje "No hay favoritos".
  - **Recordatorios:**
    - Presenta el componente app-swiper-peliculas para los recordatorios, gestionados por el servicio RecordatoriosService.
    - Muestra un mensaje "No hay recordatorios" si la lista está vacía.
3. **Alert de cierre de sesión (ion-alert):**
  - Se despliega al seleccionar "Cerrar sesión" en el menú lateral.
  - Incluye dos botones: "Cancelar" (cierra el alert) y "Cerrar sesión" (ejecuta el cierre de sesión).

Comunicación entre componentes:

1. **Navegación interna:**

- Al confirmar el cierre de sesión, el usuario es redirigido a la página de inicio de sesión (/login).
- 2. Interacción entre componentes:**
- Los datos de favoritos y recordatorios se obtienen de los servicios FavoritosService y RecordatoriosService, respectivamente.
  - La suscripción al observable user\$ en AuthService permite obtener información en tiempo real del usuario autenticado.
  - El alert utiliza eventos para gestionar el estado del diálogo y desencadenar la acción de logout.

## 5.7. DetallePelícula

Describir las partes de la pantalla relacionadas con los componentes:

1. **Cabecera (ion-header):**
  - Incluye un botón de navegación hacia atrás (ion-back-button), configurado para volver a la página anterior.
  - Muestra el título de la película seleccionada.
2. **Cuerpo principal (ion-content):**
  - **Poster de la película:**
    - Se presenta el póster de la película utilizando la URL base de TMDb concatenada con poster\_path.
  - **Información de la película:**
    - Título, fecha de estreno, géneros y duración.
    - Una descripción general (overview) de la película.
    - Una puntuación visual que muestra el porcentaje basado en el vote\_average.
  - **Botones de acción:**
    - Un botón para añadir o eliminar la película de favoritos. El texto cambia dinámicamente según si ya pertenece a favoritos o no.
    - Un segundo botón "Añadir a Mi Lista", actualmente sin funcionalidad implementada.
3. **Mensaje de carga (ng-template):**
  - Mientras se cargan los datos de la película, se muestra un mensaje "Cargando detalles de la película...".

Comunicación entre componentes:

1. **Navegación interna:**
  - Al cargar la página, el ID de la película se obtiene de la URL mediante ActivatedRoute.
  - El servicio MoviesapiService se utiliza para obtener los detalles completos de la película.
2. **Interacción entre componentes:**
  - El estado de favoritos se controla a través del servicio FavoritosService. Igual con el de MiLista.
  - Los botones de acción interactúan directamente con este servicio para agregar o eliminar películas de la lista de favoritos.
  - Propiedades computadas (formattedReleaseDate, formattedGenres) transforman datos de la API para su presentación en un formato amigable.

## **5.8. PeliculasComponent**

Este componente organiza y muestra una lista de películas en una cuadrícula. Cada película se ve como una tarjeta individual. Es ideal para mostrar muchas películas al mismo tiempo de forma ordenada.

## **5.9. SwiperPeliculasComponent**

Este componente usa un carrusel deslizante para mostrar películas. Es perfecto para destacar películas en una presentación interactiva, adaptándose a diferentes tamaños de pantalla. También puede mostrar un mensaje si no hay favoritos.

### **5.9.1. Swiper.js**

**Swiper.js** es una biblioteca ampliamente utilizada para crear carruseles y sliders con una experiencia de usuario fluida y altamente personalizable. En este caso, se emplea en el componente app-swiper-peliculas para mostrar listas de películas de manera visualmente atractiva y responsive.

## **5.10. PeliculaComponent**

Este es el componente que representa cada película individualmente como una tarjeta. Puede mostrar solo la imagen o incluir detalles como el título, la fecha y opciones para verla o añadirla a recordatorios.

## **5.11. Auth(Guard)**

AuthGuard proporciona una capa de seguridad en la aplicación, restringiendo el acceso a ciertas rutas si el usuario no está autenticado. Funciona interceptando las solicitudes de navegación y verificando el estado de autenticación del usuario mediante el servicio AuthService.

Esto se consigue mediante el observable user\$ de AuthService que determina si hay un usuario autenticado. Si el usuario existe devuelve true y permite acceso a la ruta. En caso de que devuelva false, se redirigirá a la pantalla de login.

## **5.12. Responsividad**

Durante esta aplicación se ha buscado garantizar su completa responsividad, buscando que sea capaz de adaptarse a los diferentes tamaños de pantalla.

Para lograr esto, se han utilizado técnicas de diseño web responsive, como el uso de **media queries** en CSS, diseño basado en **flexbox** y **grid**.