Universidad Politécnica
de Madrid

Escuela Técnica Superior de
Ingenieros Informáticos

Grado en Ingeniería Informática

Trabajo Fin de Grado

Desarrollo de Aplicación Móvil de Servicios de Streaming

Autor: Cristina Delgado Pardo

Tutor(a): Raúl Alonso Calvo

Madrid, Junio 2024

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado
Grado en Ingeniería Informática
Título: Desarrollo de Aplicación Móvil de Servicios de Streaming
Junio 2024

Autor: Cristina Delgado Pardo

Tutor:
Raúl Alonso Calvo
Lenguajes y Sistemas Informáticos e Ingeniería del Software
ETSI Informáticos
Universidad Politécnica de Madrid

Resumen

Actualmente existen muchas plataformas de streaming en las que se pueden ver películas, como HBO Max, Amazon Prime Vídeo, Netflix, Disney+ y muchas más que van surgiendo con el paso del tiempo. Pero cada plataforma dispone de un contenido diferente, es decir, cada una tiene su propio catálogo de películas disponibles, además de que hoy en día las plataformas quieren sacar más contenido original propio de su plataforma para así atraer a más usuarios.

Esto deriva en que, en ninguna de las plataformas anteriormente mencionadas puedes ver un catálogo con todas las películas que se han estrenado a lo largo de los años. Para ello se quiere desarrollar esta aplicación, llamada StreamList, para que los usuarios tengan una aplicación en la que dispongan de un catálogo completo con todas las películas estrenadas en un único sitio, de forma que sea más cómodo y fácil buscar la próxima película que quieras ver. A parte también se podrán realizar búsquedas de películas por su título en esta aplicación.

Además, en esta aplicación cada vez que te interese una película podrás ver la información disponible de esa película, desde la fecha en la que se estrenó hasta su duración, incluyendo también su descripción, el reparto que ha participado en la película y el equipo que la ha dirigido y desarrollado. También podrás ver la puntuación que le han dado a la película y sus géneros.

Por otro lado, esta aplicación te permitirá guardar en una serie de listas todas las películas que quieras, desde las películas que has visto, hasta las que quieres ver más tarde o en otro momento, incluyendo por supuesto también tus películas favoritas. Esto es muy útil por si en algún momento deseas mirar todas las películas que has visto, que son tus favoritas o que quieres ver, solo tendrías que acceder a esta aplicación para ello, en vez de ir mirando por todas las plataformas de las que dispongas, que películas has visto o cual tienes guardada para ver más tarde en cada una de las plataformas.

El desarrollo de esta aplicación para dispositivos Android está documentada en la presente memoria. Desde las tecnologías utilizadas para su desarrollo hasta el diseño de la aplicación, pasando por supuesto por la arquitectura del sistema, explicando tanto el frontend como el backend de la aplicación y desarrollando también sus funcionalidades.

Para finalizar, se han realizado una serie de pruebas de todas las funcionalidades de las que dispone la aplicación, para comprobar el correcto funcionamiento de ellas. Y por último se han comentado conclusiones finales y líneas futuras de la aplicación.

Abstract

Currently there are many streaming platforms where you can watch movies, such as HBO Max, Amazon Prime Vídeo, Netflix, Disney+ and many more that are emerging over time. But each platform has a different content, that is, each one has its own catalog of movies available, besides that nowadays platforms want to bring out more of their own original content on their platform in order to attract more users.

This derives in that, in none of the previously mentioned platforms you can see a catalog with all the movies that have been released over the years. For this reason, I want to develop this application, called StreamList, so that users have an application in which they have a complete catalog with all the movies released in one place, so that is more convenient and easier to search for the next movie you want to see. In addition, it will also be possible to search for movies by title in this application.

Additionally, in this application every time you are interested in a movie you will be able to see the available information of that movie, from the date it was released to its duration, including also its description, the cast that has participated in the movie and the crew that has directed and developed it. You will also be able to see the vote given to the movie and its genres.

In other hand, this application will allow you to save in different lists all the movies that you want, from the movies you have seen, to the ones you want to watch later or at another time, including of course your favorite movies as well. This is very useful if at some point you want to watch all the movies you have seen, that are your favorites or you want to see, you would only have to access this application for it, instead of going looking through all the platforms that you have, which movies you have seen or which you have saved to watch later on each of the platforms.

The development of this application for Android devices is documented in this report. From the technologies used for its development to the design of the application, including of course the system architecture, explaining both the frontend and the backend of the application and also developing its functionalities.

Finally, a series of test of all the functionalities of the application have been carried out to check their correct operation. And at last, final conclusions and future lines of the application have been commented.

Tabla de contenidos

Índice de Ilustraciones

Índice de Tablas

1 Introducción

1.1 Descripción del problema y solución

El trabajo consiste en desarrollar una aplicación móvil de servicios de streaming, que ayude al usuario con diferentes problemas, que explicaré a continuación.

Hoy en día existen muchas plataformas de streaming como Netflix, HBO Max, Amazon Prime Vídeo, Disney+ y más, donde puedes ver todas las películas que quieras, pero el problema existente es que en cada plataforma tienen unas películas distintas y esto quiere decir que en ninguna de las plataformas aparecen todas las películas estrenadas a lo largo de los años, por eso quiero desarrollar esta aplicación, donde los usuarios dispondrán de un catálogo con todas las películas que se han estrenado, para que así sea más fácil buscar y elegir nuevas películas que el usuario quiera ver.

Por otro lado, como he mencionado antes cada plataforma tiene unas películas diferentes y por eso terminas viendo una película en una plataforma y otra película en otra y así constantemente, el problema que surge es que cuando quieras saber si has visto una película o no te acuerdas si la has visto, deberás ir buscando por todas las plataformas, para ver si en alguna de ellas la has visto, por ello también quiero desarrollar esta aplicación, para que los usuarios dispongan de una lista con todas las películas que ha visto en un único lugar.

Tener todas las películas que has visto en un único lugar también es muy útil, por si en algún momento quieres saber cuáles son todas las películas que has visto a lo largo de los años, solo tienes que acceder a un único sitio para ello, en vez de ir mirando por todas las plataformas de una en una, para ver que películas has visto en cada una. Además, los usuarios dispondrán de otras listas, para guardar películas favoritas o las películas que quieran ver en otro momento.

1.2 Objetivos

Los objetivos del trabajo serían los siguientes:

- Aplicar conocimientos sobre el desarrollo de proyectos software.
- Desarrollar la aplicación para ayudar a los usuarios con los diferentes servicios, como la búsqueda y la organización personalizada de los contenidos disponibles.
- Integración de API necesaria para obtener información de las películas.
- Creación y utilización de bases de datos.
- Disponer de un catálogo con todas las películas estrenadas.
- Disponer de una aplicación donde los usuarios puedan saber todas las películas que han visto, sus favoritas y las que quieren ver en otro momento.

1.3 Tareas y Diagrama de Gantt

Tareas a realizar:

- Investigación del estado del arte
- Estudio del entorno de trabajo
- Estudio del lenguaje de programación
- Desarrollo de requisitos
- Desarrollo de casos de uso
- Diseño de la aplicación móvil, diferentes pantallas
- Desarrollo de la aplicación
- Pruebas del funcionamiento de la aplicación
- Creación de la memoria final del proyecto

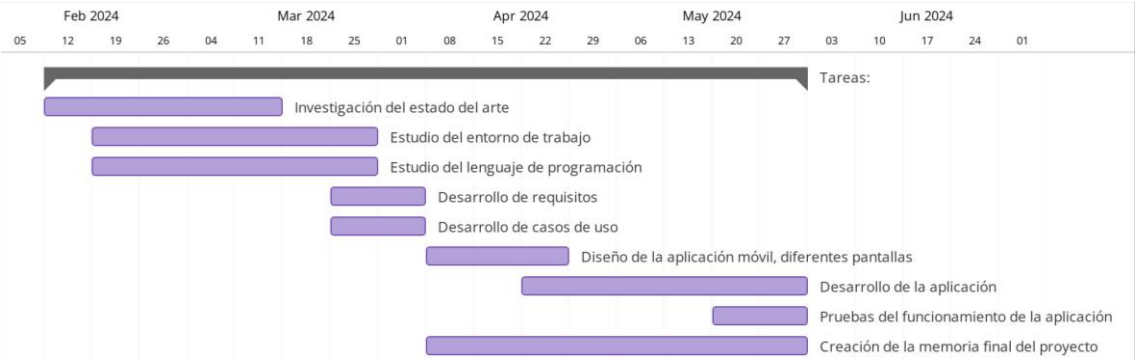

Ilustración 1: Diagrama de Gantt

2 Trabajos previos

En este capítulo me gustaría hablar de algunas aplicaciones que disponen de funcionalidades y características similares a la aplicación que estoy desarrollando.

2.1 CineTrak

La aplicación CineTrak te permite buscar y encontrar las mejores películas y series de televisión, te permite crear listas personalizadas y ver opiniones y valoraciones de varias fuentes. Tiene todo tipo de funciones con las que gestionar las películas que queremos ver o las películas que hemos visto. También puedes recibir recomendaciones de películas o series basadas en tus gustos. [1]

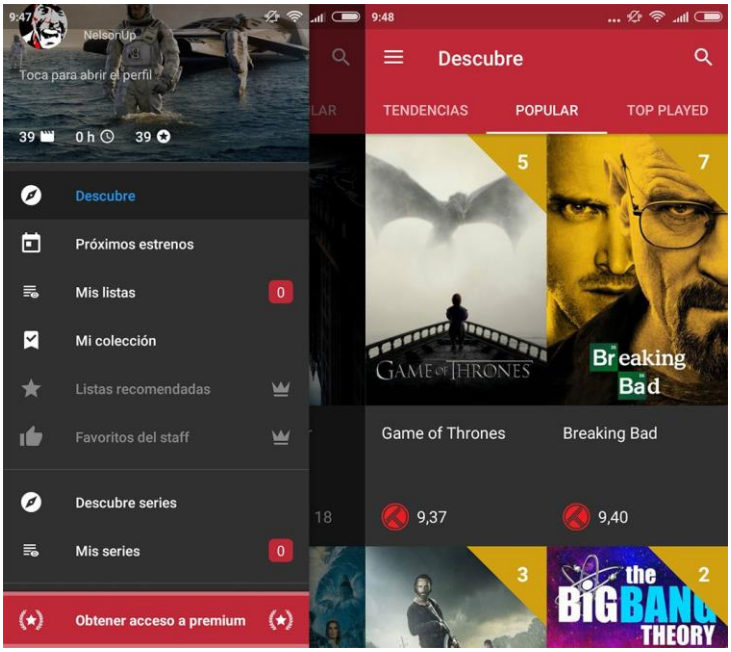

Ilustración 2: Aplicación CineTrak

2.2 MyMovies

Con la aplicación MyMovies puedes catalogar tu colección completa de películas y series de televisión. Dispone de un buscador y filtros de diversas categorías con los que puedes encontrar rápidamente las películas o series por su título original o traducido. Te permite administrar tu colección de películas y series según aquellas que has visto o las que quieras ver. También proporciona información detallada de cada película y serie. [2]

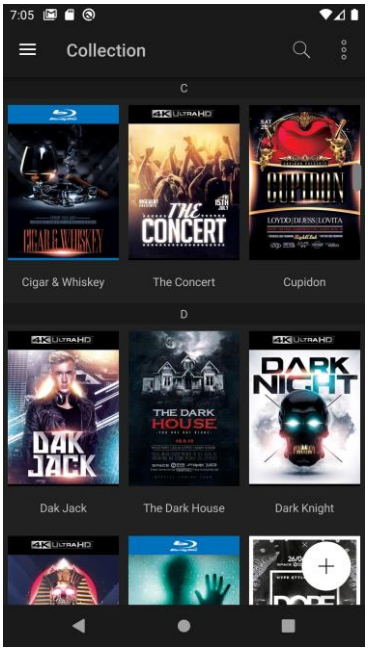

Ilustración 3: Aplicación MyMovies

2.3 SeriesGuide

La aplicación SeriesGuide te permite encontrar series y películas que quieres ver, con ella puedes guardar tus episodios y películas vistas, también puedes recibir notificaciones de próximos episodios.

Te recomendará series o películas similares a las que estás viendo. Podrás explorar películas que estén en cartelera o que hayan sido lanzadas digitalmente o en disco. [3]

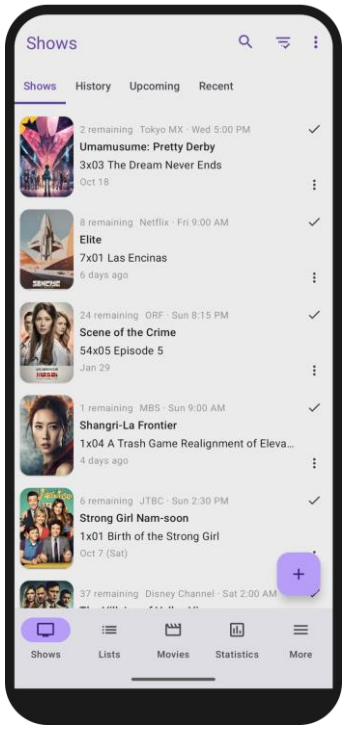

Ilustración 4: Aplicación SeriesGuide

2.4 JustWatch

Con la aplicación JustWatch podrás buscar películas y series de televisión, y te ofrecerá información muy valiosa como el número de temporadas, duración, puntuación de los usuarios, sinopsis y en que plataforma puedes encontrar y ver dicha película o serie. También podrás ver un tráiler. Y te permitirá crear una lista de películas y series favoritas. [4]

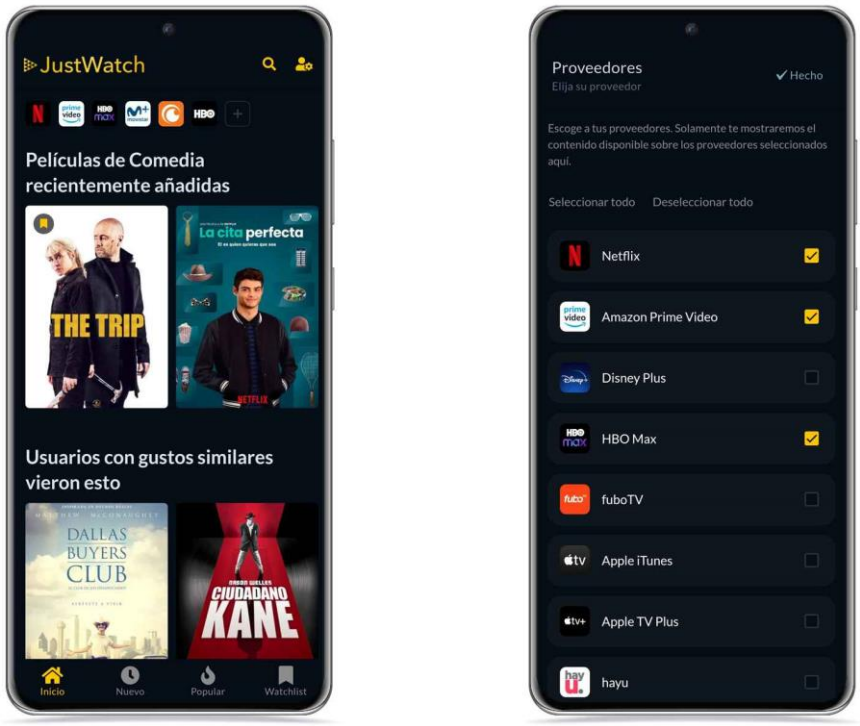

Ilustración 5: Aplicación JustWatch

3 Tecnologías utilizadas

3.1 Entorno de desarrollo - Android Studio

Ilustración 6: Android Studio

El entorno de desarrollo utilizado para realizar la aplicación de este trabajo ha sido Android Studio [5]. Android Studio es el IDE oficial para la plataforma Android y fue diseñado específicamente para el desarrollo de aplicaciones y software de Android, su primera versión estable fue lanzada en 2014 y desde ese momento reemplazó a Eclipse como IDE oficial para el desarrollo de aplicaciones Android.

Android Studio está basado en la herramienta IntelliJ IDEA y fue publicado de forma gratuita mediante la licencia Apache 2.0. Este IDE está disponible para las plataformas Windows, Linux y macOS, lo que es bastante cómodo ya que puedes instalarlo y utilizarlo en casi cualquier dispositivo.

Android Studio admite los lenguajes de programación Java, C++ y Kotlin dependiendo de la versión. Este IDE dispone de un potente editor de códigos y una gran cantidad de funcionalidades que aumentan la productividad a la hora de desarrollar una aplicación.

Algunas de sus funcionalidades más importantes son que realiza renderizados de layouts en tiempo real, lo cual ayuda a ver de forma sencilla y rápida como va quedando la vista que estas desarrollando, también dispone de potentes emuladores integrados que simulan distintos dispositivos, como móviles, tablets, Wear OS y más, para ejecutar la aplicación que estas desarrollando. Se pueden crear emuladores de varios dispositivos con distintas especificaciones como la resolución, el tamaño de la pantalla, la densidad de pixeles, elegir el nivel de API (versión de Android) del dispositivo, etc.

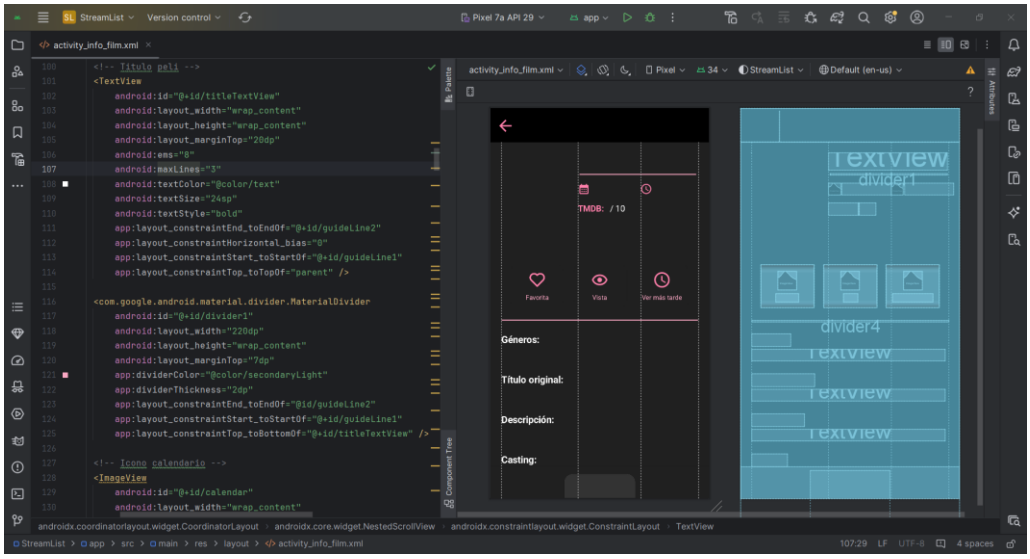

Ilustración 7: Android Studio - Funcionalidades (Renderizado layouts)

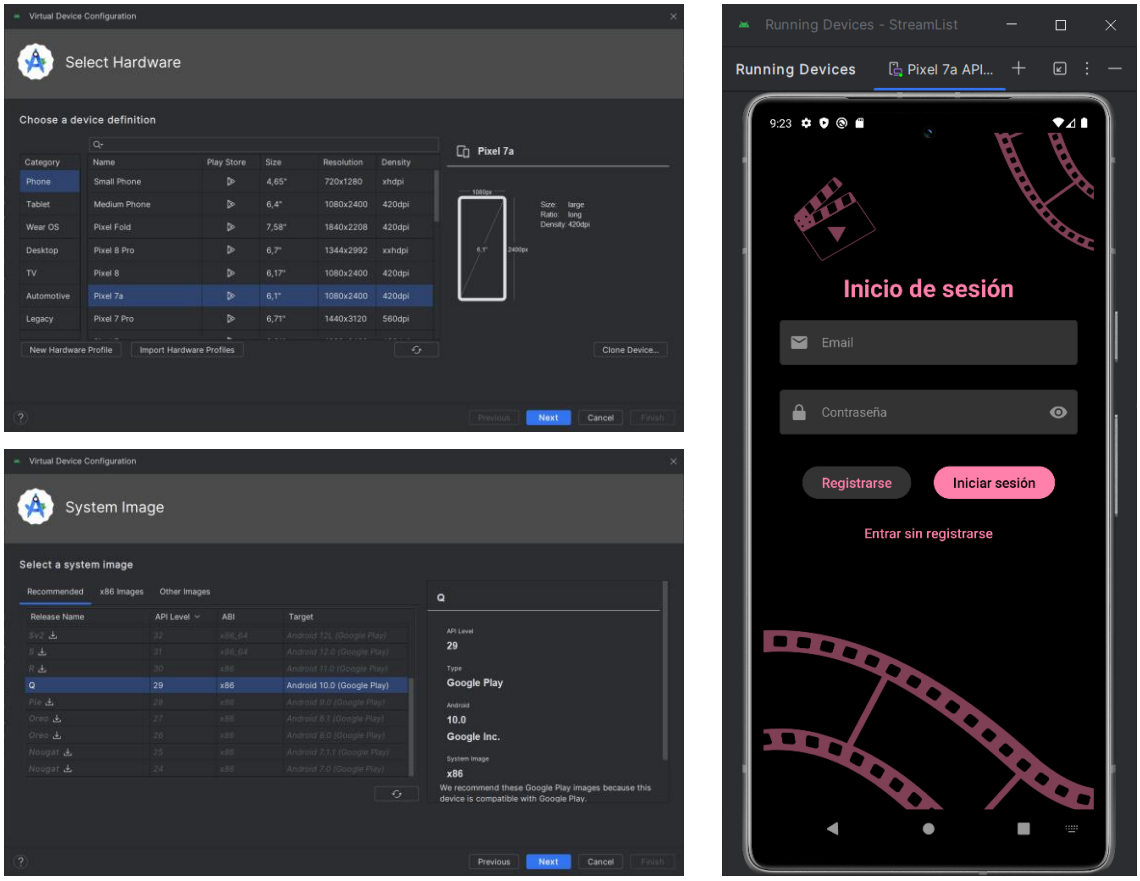

Ilustración 8: Android Studio - Funcionalidades (Emuladores)

3.2 Lenguaje de programación - Kotlin

Ilustración 9: Kotlin

El lenguaje de programación utilizado para desarrollar la aplicación es Kotlin [6]. Kotlin es un lenguaje de programación multiplataforma, de tipado estático y de alto nivel, que puede correr sobre JVM y JavaScript, por lo que es completamente interoperable con código Java, lo que hace posible que código escrito en Java pueda interactuar de forma correcta con código escrito en Kotlin y viceversa.

El lenguaje Kotlin dispone de diversas características, como que es un lenguaje orientado a objetos, pero también trabaja de la mano de la programación funcional. Dispone de corrutinas que optimizan la programación asíncrona y simplifican el trabajo del acceso a bases de datos y de llamadas a la red.

También está estructurado en bloques, es imperativo, declarativo, reflexivo, genérico y flexible, lo que permite que los desarrolladores tengan libertad a la hora de trabajar con él y con el estilo que quieran.

Fue diseñado y desarrollado por la compañía de desarrollo de software JetBrains. En 2019 Google anunció que kotlin se convertiría en el lenguaje

preferido para los programadores, a la hora de desarrollar aplicaciones en Android.

3.3 TMDB API

Ilustración 10: TMDB

The Movie Database (TMDB) [7] es una base de datos de películas y series de televisión como su propio nombre indica, lanzada en 2008. Es una plataforma en línea donde te puedes registrar de forma gratuita y una vez siendo usuario puedes acceder a distintos datos, como las películas populares, las que están en tendencia, también puedes ver datos de cada película como su descripción, la fecha en la que se estrenó, la puntuación, la duración, el reparto, diferentes títulos que tiene la película, y muchos más datos.

TMDB destaca por su interfaz moderna y atractiva, además involucra en la producción de datos y en la revisión de los mismos a los usuarios registrados. También, TMDB es popular entre los desarrolladores de aplicaciones y sitios web que tienen que ver con el cine, ya que dispone de una API abierta con la que puedes integrar su extenso catálogo de información en proyectos y plataformas que vayas a desarrollar.

Esta es la API que he utilizado a la hora de obtener los datos que necesitaba sobre las películas para desarrollar mi aplicación de catálogo de películas.

3.4 Firebase

Ilustración 11: Firebase

Firebase [8] es una plataforma, ubicada en la nube, que se utiliza para desarrollar aplicaciones móviles y aplicaciones web. Esta plataforma se lanzó en 2011 y en 2014 fue adquirida por Google.

Firebase ofrece muchas herramientas y servicios para poder desarrollar aplicaciones web y móviles, por lo que se convierte en una herramienta muy útil para las empresas y los desarrolladores.

Algunas ventajas que tendrán los desarrolladores a la hora de usar esta plataforma son, que se pueden sincronizar de forma fácil datos de sus proyectos sin necesidad de administrar conexiones o de escribir lógica compleja de sincronización, pueden usar un grupo de herramientas multiplataforma ya que se integra de manera fácil, tanto en plataformas web como en aplicaciones móviles y se pueden crear proyectos sin tener que usar un servidor, ya que las herramientas se incluyen en los SDK para los dispositivos web y móviles.

Para el desarrollo de esta aplicación se han usado los siguientes servicios que ofrece Firebase:

- **Firebase Authentication:**

Ilustración 12: Firebase Authentication

Firebase Authentication es un servicio con el que puedes autenticar a los usuarios de tu sistema. Dicha autenticación puedes realizarla mediante diferentes proveedores de inicio de sesión como por ejemplo Twitter, Facebook, Google o mediante métodos más clásicos como correo electrónico y contraseña del usuario, también puedes habilitar que el usuario acceda de forma anónima. Los diferentes proveedores que quieras tener en tu sistema, los podrás habilitar o deshabilitar cuando quieras de forma muy sencilla.

Por otro lado, dispone de un sistema de administración de los usuarios, dónde los desarrolladores podrán ver a los usuarios que han accedido a su sistema y desde ahí podrán realizar varias acciones, como eliminar una cuenta. Gracias a este servicio es más fácil la creación de sistemas de autenticación y a su vez, para los usuarios supone una mejora de incorporación, acceso y seguridad.

- **Firebase Cloud Firestore/Firestore Database:**

Ilustración 13: Firebase Firestore Database

Cloud Firestore o también llamado Firestore Database es un servicio de almacenamiento de datos. Es una base de datos NoSQL que se organiza en colecciones y dentro de cada colección hay un grupo de documentos. En dichos documentos se pueden añadir campos de varios tipos como booleanos, cadenas de texto, números, arrays, puntos geográficos, también puedes crear una colección dentro de un documento, teniendo así subcolecciones.

3.5 Figma

Ilustración 14: Figma

Figma [9] es una página web y un editor de gráficos vectorial, donde puedes generar de forma sencilla prototipos del diseño gráfico del proyecto que estés desarrollando. Puedes hacer prototipos tanto de páginas web, como de aplicaciones móviles. Las funciones de Figma están enfocadas en el diseño de la experiencia del usuario y en el uso de la interfaz del usuario. Esta herramienta será usada para diseñar un prototipo de la aplicación.

4 Desarrollo de la aplicación

En este capítulo se explicará el desarrollo de la aplicación de catálogo de películas, llamada StreamList, para ello empezaré identificando las funcionalidades implementadas en la aplicación y luego mostraré el diseño general seguido para el desarrollo de la aplicación.

4.1 Requisitos

En este primer apartado se identificarán los requisitos del sistema que encontraremos en la aplicación y así determinaremos las principales funcionalidades integradas dentro de la aplicación StreamList.

El apartado está dividido en dos subapartados, ya que identificamos dos tipos de requisitos. Primero veremos los requisitos no funcionales, que son los que definen las características, cualidades y restricciones del sistema, es decir, describen cómo debe ser el sistema, y luego veremos los requisitos funcionales, que son los que detallan las funcionalidades específicas que debe proporcionar el sistema, están centrados en el comportamiento y las funciones del sistema, es decir, describen lo que hace el sistema.

4.1.1 Requisitos No Funcionales

Los requisitos no funcionales o RNF son los que describen cómo es el sistema, definiendo las características y restricciones de dicho sistema. Se centran en la calidad, el rendimiento y la seguridad del sistema, lo que de forma indirecta afecta a la experiencia del usuario y a la calidad del sistema. Los requisitos no funcionales de la aplicación son los siguientes:

- **Disponer de un dispositivo con sistema Android (RNF1)**

Como la aplicación se ha desarrollado con el lenguaje de programación Kotlin y en el entorno de desarrollo Android Studio, es necesario disponer de un dispositivo con sistema operativo Android para poder utilizar la aplicación.

- **Versión Android 7.0 o API 24 (RNF2)**

A parte de disponer de un dispositivo con sistema operativo Android, dicho dispositivo deberá tener como mínimo una versión Android 7.0 (Nougat), o lo que es lo mismo API 24 instalado para que la aplicación pueda funcionar.

4.1.2 Requisitos Funcionales

Los requisitos funcionales o RF describen qué hace el sistema, ya que detallan las funcionalidades que ofrece dicho sistema. Especifican el comportamiento y las funciones de la aplicación, definiendo lo que debe conseguir en términos de resultados y comportamientos. Los requisitos funcionales de la aplicación son los siguientes:

- **Registro del usuario (RF1)**

El usuario puede registrarse, si lo desea, en la aplicación con un correo electrónico, con una contraseña y repitiendo esa contraseña. Cada campo tiene una serie de validaciones, el campo del email no puede estar vacio y debe estar escrito con el formato correcto de un correo (xx@xx.xx), los campos de contraseña y repetir contraseña no pueden estar vacíos, la contraseña tiene que

ser de al menos 6 caracteres y deben ser la misma contraseña en ambos campos, si no se cumple alguna de estas validaciones se mostrarán mensajes de error y no permitirá al usuario registrarse hasta que se cumplan.

También si el usuario intenta registrarse una segunda vez con los mismos datos, saldrá un mensaje de error diciendo que esa cuenta ya existe.

- **Inicio de sesión del usuario (RF2)**

Una vez el usuario se haya registrado la primera vez, podrá acceder las siguientes veces rellenando los campos de correo electrónico y contraseña. Dichos campos tienen las mismas validaciones que a la hora de registrarse, por si el usuario se equivoca al introducir los datos, dichas validaciones son que el campo no puede estar vacio para el email y la contraseña, el email debe estar escrito con el formato correcto y la contraseña debe ser de al menos 6 caracteres.

Por otro lado, al iniciar sesión también aparecerá un mensaje de error si esa cuenta no existe, o si a pesar de haber cumplido las validaciones anteriores, el usuario ha escrito erróneamente alguno de los dos campos y por lo tanto no podrá iniciar sesión hasta que los escriba correctamente.

- **Entrar sin registrarse (RF3)**

El usuario podrá acceder de forma anónima, es decir, sin necesidad de registrarse. Esta es una forma bastante cómoda para los usuarios que solo quieren acceder a la aplicación para mirar películas, ver información de dichas películas o buscar películas.

- **Ver películas por categorías (RF4)**

Una vez el usuario se haya registrado, haya iniciado sesión o haya entrado de forma anónima, accederá a una pantalla de inicio donde podrá ver grupos de películas separadas en cuatro categorías: Películas Actuales, Populares, Mejor Valoradas y Próximas Películas.

- **Ver todas las películas de una categoría (RF5)**

Una vez el usuario haya elegido una categoría, podrá acceder a ver todas las películas de dicha categoría, dando al botón correspondiente de ver más películas. Una vez hecho lo anterior, el usuario podrá ir navegando por las diferentes páginas de dicha categoría avanzando o retrocediendo.

- **Ver información de una película (RF6)**

Una vez el usuario haya seleccionado una película, podrá acceder a la información de dicha película. La información disponible de cada película es su título, la fecha de su estreno, su duración, la puntuación que le han dado los usuarios en TMDB, los géneros de la película, su título original, la descripción de la película, el reparto y el equipo de dirección y desarrollo.

- **Buscar una película (RF7)**

El usuario podrá buscar la película que desee por su título. No es necesario que el usuario recuerde el título entero, ya que la búsqueda ocurre de forma activa, es decir, mientras el usuario va escribiendo van apareciendo películas que coinciden o que es similar el título con el escrito por el usuario. Además, dispondrá de un botón por si el usuario quiere borrar todo lo escrito de forma rápida, para así buscar otra película.

- **Acceder a las listas (RF8)**

Para que el usuario pueda acceder a las listas será obligatorio que se registre o inicie sesión, si ya se ha registrado. Una vez hecho eso, podrá acceder a las listas de películas Favoritas, Vistas o para Ver más tarde. En caso de que el usuario intente acceder a las listas sin haberse registrado o haber iniciado sesión se mostrará un mensaje de error.

- **Guardar película en lista de Favoritas (RF9)**

Si el usuario se ha registrado o ha iniciado sesión, podrá guardar una película en la lista de Favoritas. En el caso de que lo haga sin estar registrado se mostrará un mensaje de error.

- **Eliminar película de lista de Favoritas (RF10)**

Si el usuario se ha registrado o ha iniciado sesión, podrá eliminar una película de la lista de Favoritas. Si el usuario no está registrado no podrá realizar esta acción.

- **Guardar película en lista de Vistas (RF11)**

Si el usuario se ha registrado o ha iniciado sesión, podrá guardar una película en la lista de Vistas. En el caso de que lo haga sin estar registrado se mostrará un mensaje de error.

- **Eliminar película de lista de Vistas (RF12)**

Si el usuario se ha registrado o ha iniciado sesión, podrá eliminar una película de la lista de Vistas. Si el usuario no está registrado no podrá realizar esta acción.

- **Guardar película en lista de Ver más tarde (RF13)**

Si el usuario se ha registrado o ha iniciado sesión, podrá guardar una película en la lista de Ver más tarde. En el caso de que lo haga sin estar registrado se mostrará un mensaje de error.

- **Eliminar película de lista de Ver más tarde (RF14)**

Si el usuario se ha registrado o ha iniciado sesión, podrá eliminar una película de la lista de Ver más tarde. Si el usuario no está registrado no podrá realizar esta acción.

- **Perfil usuario (RF15)**

Si el usuario se ha registrado, podrá acceder a un perfil donde aparecerá su correo electrónico y desde ahí podrá cerrar sesión. Si el usuario no está registrado, podrá ver que está como anónimo y podrá iniciar sesión o registrarse si lo desea.

- **Cierre de sesión del usuario (RF16)**

En el caso de que el usuario se haya registrado o haya iniciado sesión podrá, desde la pantalla de inicio, acceder a su perfil y ahí cerrar su sesión. Si el usuario ha accedido de forma anónima, verá que en su perfil pone “(Anónimo)” y no podrá cerrar sesión, ya que no está registrado.

4.2 Casos de uso

A continuación, explicaré los casos de uso para detallar las funcionalidades desarrolladas en los requisitos funcionales del sistema. Con esto se quiere conseguir que se entienda mejor la implementación de cada funcionalidad que hay en la aplicación.

Los casos de uso constarán de varios campos:

- **ID**

Un identificador único (CU1, CU2…), en la primera fila de la tabla.

- **Nombre**

El nombre del caso de uso en cuestión, también en la primera fila de la tabla.

- **Descripción**

Descripción del caso de uso.

- **Referencias**

Los requisitos funcionales a los que puede hacer referencia cada caso de uso.

- **Actores**

Los actores que participan en la realización del caso de uso.

- **Flujo**

Secuencia que sigue cada caso de uso.

Los casos de uso de la aplicación son los siguientes:

CU1	**El usuario se registra**
Descripción	El usuario se registra en la aplicación
Referencias	RF1
Actores	Usuario, Firebase Authentication
Flujo	1. El usuario accederá a la vista de registrarse 2. El usuario rellenará los campos de correo electrónico, contraseña y repetir contraseña 3. Se realizarán una serie de validaciones a la vez para cada uno de los campos, para el campo del email, se comprobará que el campo no esté vacio o que el correo escrito tenga el formato correcto de un correo, para los campos de contraseña y repetir contraseña se realizarán las mismas validaciones, que el campo no esté vacio, que la contraseña tenga al menos 6 caracteres y que la contraseña sea la misma en ambos campos, si alguna de estas validaciones no se cumple se mostrará un mensaje de error debajo de cada uno de los campos, indicando que no cumple dicho campo y el usuario no se podrá registrar, hasta que los modifique y sean válidos 4. Por el contrario, si los campos son válidos el usuario se podrá registrar, el sistema realizará una llamada a Firebase Authentication para registrar al usuario y guardar sus datos y el usuario accederá a la aplicación, excepto si dicha cuenta con esos mismos datos ya existe, en cuyo caso, saldrá un mensaje de error diciendo que esa cuenta ya existe y por lo tanto no la puede volver a registrar

Tabla 1: CU1 - El usuario se registra

CU2	**El usuario inicia sesión**
Descripción	El usuario inicia sesión en la aplicación
Referencias	RF2
Actores	Usuario, Firebase Authentication
Flujo	1. El usuario accede a la vista de inicio de sesión 2. El usuario rellenará los campos de correo electrónico y contraseña 3. Se realizarán una serie de validaciones a la vez para cada uno de los campos, para el campo email se comprobará que no esté vacio y se comprobará que tenga el formato correcto y para el campo de la contraseña se comprobará que el campo no esté vacio y que tenga como mínimo 6 caracteres de largo, si alguna de estas validaciones no se cumple se mostrará un mensaje de error debajo de cada uno de los campos, indicando que no cumple dicho campo y el usuario no podrá iniciar sesión 4. Por el contrario, si las validaciones se cumplen el usuario podrá iniciar sesión, el sistema realizará una llamada a Firebase Authentication de inicio de sesión del usuario y este accederá a la aplicación, excepto si la cuenta con la que el usuario intenta iniciar sesión no está registrada y no existe, en cuyo caso se mostrará un mensaje de error y también se mostrará un mensaje de error si alguno de los campos es incorrecto y el usuario no podrá iniciar sesión hasta que los escriba correctamente

Tabla 2: CU2 - El usuario inicia sesión

CU3	**El usuario entra sin registrarse**
Descripción	El usuario accede a la aplicación de forma anónima, es decir, sin necesidad de registrarse
Referencias	RF3
Actores	Usuario, Firebase Authentication
Flujo	1. El usuario accede a la pantalla de inicio de sesión o a la pantalla de registrarse 2. El usuario presiona el botón que pone “Entrar sin registrarse” y accede a la aplicación de forma anónima

Tabla 3: CU3 - El usuario entra sin registrarse

CU4	**El usuario accede a películas por categorías**
Descripción	El usuario accede a la pantalla inicio de la aplicación donde podrá encontrar grupos de películas divididas en cuatro categorías: Películas Actuales, Populares, Mejor Valoradas y Próximas Películas
Referencias	RF4
Actores	Usuario, TMDB API
Flujo	1. El usuario accede a la pantalla inicio de la aplicación 2. El sistema realiza llamadas a la API de TMDB para obtener la información necesaria, dicha información es el poster, el título, el año y la puntuación de cada una de las películas 3. El usuario podrá observar las filas de las películas agrupadas por categorías

Tabla 4: CU4 - El usuario accede a películas por categorías

CU5	**El usuario accede a todas las películas de una categoría**
Descripción	El usuario accede a todas las películas disponibles dentro de una categoría y agrupadas en diferentes páginas
Referencias	RF5
Actores	Usuario, TMDB API
Flujo	1. El usuario accede a una página de todas las películas disponibles de una categoría 2. El sistema realiza una llamada a la API de TMDB para obtener la información necesaria, dicha información es el poster, el título, el año y la puntuación de cada una de las películas 3. El usuario podrá observar la lista de las películas de dicha página 4. El usuario podrá presionar el botón de “Siguiente página” o en el de “Página anterior” para ir navegando por las diferentes páginas y poder ir observando todas las películas 5. Cada vez que al usuario presiona en “Siguiente página” o en “Página anterior” el sistema realiza una llamada a la API de TMDB para obtener los datos necesarios de las películas y que el usuario pueda observar la lista de las películas correctamente

Tabla 5: CU5 - El usuario accede a todas las películas de una categoría

CU6	**El usuario accede a la información de una película**
Descripción	Si el usuario desea saber la información de una película, solo deberá seleccionar dicha película y accederá a la vista con la información de la película deseada
Referencias	RF6
Actores	Usuario, TMDB API
Flujo	1. El usuario accederá a la vista de información de la película seleccionada 2. El sistema realizará una llamada a la API de TMDB para obtener la información de la película, dicha información será el título, la fecha de estreno, la duración, la puntuación, los géneros, el título original, la descripción, el reparto y el equipo de dirección y desarrollo 3. El usuario podrá observar toda la información de la película

Tabla 6: CU6 - El usuario accede a la información de una película

CU7	**El usuario busca una película**
Descripción	El usuario podrá buscar una película por su título o por el trozo que recuerde del título
Referencias	RF7
Actores	Usuario, TMDB API
Flujo	1. El usuario accederá a la vista de búsqueda de una película 2. Mientras el usuario está escribiendo el título, el sistema irá lanzado varias llamadas a la API de TMDB a medida que el usuario va introduciendo o quitando un nuevo carácter 3. Una vez el usuario haya terminado de escribir podrá presionar “enter” para que se oculte el teclado y así observar cómodamente los resultados de la búsqueda 4. El usuario podrá eliminar rápidamente lo que ha escrito presionando un botón con forma de x ubicado en el buscador, para así poder realizar otra búsqueda si lo desea

Tabla 7: CU7 - El usuario busca una película

CU8	**El usuario accede a las listas**
Descripción	El usuario entra a la vista donde se encuentra el acceso a las listas de Favoritas, Vistas y Ver más tarde
Referencias	RF8, CU1, CU2
Actores	Usuario, Firebase Authentication
Flujo	1. Si el usuario está registrado podrá entrar a la vista donde se encuentra el acceso a las listas 2. Por el contrario, si el usuario no está registrado he intenta acceder a esta vista se mostrará un mensaje de error diciendo que se deberá registrar o iniciar sesión y en ese mensaje se mostrará también 3 botones, uno con el que el usuario podrá ir a iniciar sesión, otro con el que el usuario podrá ir a registrarse y otro con el que simplemente se cerrará el mensaje

Tabla 8: CU8 - El usuario accede a las listas

CU9	**El usuario guarda película en lista Favoritas**
Descripción	El usuario guarda una película en la lista de películas Favoritas
Referencias	RF9, CU6
Actores	Usuario, Firebase Firestore Database, TMDB API
Flujo	1. El usuario accede a la vista de información de una película 2. El sistema realizará una llamada a la API de TMDB para obtener la información de la película 3. Si el usuario está registrado y selecciona el botón de Favorita podrá guardar la película, en caso de que no esté registrado y realice dicha acción, se mostrará un mensaje de error diciendo que se debe registrar o iniciar sesión, y en dicho mensaje aparecerán botones para que el usuario pueda realizar cualquiera de las dos acciones. El usuario no podrá guardar la película hasta que se registre 4. El sistema realiza una llamada de guardado a Firestore Database para guardar los datos de la película en la lista de Favoritas, y se modifica y guarda el estado del botón Favorita en Firestore Database 5. El sistema actualiza el botón de Favorita apareciendo ahora seleccionado ya que la película fue guardada, y si el usuario está registrado y vuelve a acceder en otro momento a la vista de información de dicha película, el botón aparecerá seleccionado

Tabla 9: CU9 - El usuario guarda película en lista Favoritas

CU10	**El usuario elimina película de lista Favoritas**
Descripción	El usuario elimina una película de la lista de películas Favoritas
Referencias	RF10, CU6, CU9
Actores	Usuario, Firebase Firestore Database, TMDB API
Flujo	1. El usuario accede a la vista de información de una película 2. El sistema realizará una llamada a la API de TMDB para obtener la información de la película 3. El sistema realizará una llamada a Firestore Database para saber si está guardada la película y marcar como seleccionado el botón correspondiente si lo está (si el usuario no está registrado esta acción no se realiza) 4. Si está seleccionado el botón, el usuario deseleccionará el botón de Favorita y con ello eliminará la película, en el caso de que el usuario no esté registrado no podrá realizar esta acción, ya que en este caso el botón nunca aparecerá seleccionado 5. El sistema realiza una llamada de eliminar a Firestore Database para eliminar los datos de la película de la lista de Favoritas, y se modifica y guarda el estado del botón Favorita en Firestore Database 6. El sistema actualiza el botón de Favorita apareciendo ahora deseleccionado ya que la película fue eliminada, y si el usuario está registrado y vuelve a acceder en otro momento a la vista de información de dicha película, el botón aparecerá deseleccionado

Tabla 10: CU10 - El usuario elimina película de lista Favoritas

CU11	**El usuario guarda película en lista Vistas**
Descripción	El usuario guarda una película en la lista de películas Vistas
Referencias	RF11, CU6
Actores	Usuario, Firebase Firestore Database, TMDB API
Flujo	1. El usuario accede a la vista de información de una película 2. El sistema realizará una llamada a la API de TMDB para obtener la información de la película 3. Si el usuario está registrado y selecciona el botón de Vista podrá guardar la película, en caso de que no esté registrado y realice dicha acción, se mostrará un mensaje de error diciendo que se debe registrar o iniciar sesión, y en dicho mensaje aparecerán botones para que el usuario pueda realizar cualquiera de las dos acciones. El usuario no podrá guardar la película hasta que se registre 4. El sistema realiza una llamada de guardado a Firestore Database para guardar los datos de la película en la lista de Vistas, y se modifica y guarda el estado del botón Vista en Firestore Database 5. El sistema actualiza el botón de Vista apareciendo ahora seleccionado ya que la película fue guardada, y si el usuario está registrado y vuelve a acceder en otro momento a la vista de información de dicha película, el botón aparecerá seleccionado

Tabla 11: CU11 - El usuario guarda película en lista Vistas

CU12	**El usuario elimina película de lista Vistas**
Descripción	El usuario elimina una película de la lista de películas Vistas
Referencias	RF12, CU6, CU11
Actores	Usuario, Firebase Firestore Database, TMDB API
Flujo	1. El usuario accede a la vista de información de una película 2. El sistema realizará una llamada a la API de TMDB para obtener la información de la película 3. El sistema realizará una llamada a Firestore Database para saber si está guardada la película y marcar como seleccionado el botón correspondiente si lo está (si el usuario no está registrado esta acción no se realiza) 4. Si está seleccionado el botón, el usuario deseleccionará el botón de Vista y con ello eliminará la película, en el caso de que el usuario no esté registrado no podrá realizar esta acción, ya que en este caso el botón nunca aparecerá seleccionado 5. El sistema realiza una llamada de eliminar a Firestore Database para eliminar los datos de la película de la lista de Vistas, y se modifica y guarda el estado del botón Vista en Firestore Database 6. El sistema actualiza el botón de Vista apareciendo ahora deseleccionado ya que la película fue eliminada, y si el usuario está registrado y vuelve a acceder en otro momento a la vista de información de dicha película, el botón aparecerá deseleccionado

Tabla 12: CU12 - El usuario elimina película de lista Vistas

CU13	**El usuario guarda película en lista Ver más tarde**
Descripción	El usuario guarda una película en la lista de películas para Ver más tarde
Referencias	RF13, CU6
Actores	Usuario, Firebase Firestore Database, TMDB API
Flujo	1. El usuario accede a la vista de información de una película 2. El sistema realizará una llamada a la API de TMDB para obtener la información de la película 3. Si el usuario está registrado y selecciona el botón de Ver más tarde podrá guardar la película, en caso de que no esté registrado y realice dicha acción, se mostrará un mensaje de error diciendo que se debe registrar o iniciar sesión, y en dicho mensaje aparecerán botones para que el usuario pueda realizar cualquiera de las dos acciones. El usuario no podrá guardar la película hasta que se registre 4. El sistema realiza una llamada de guardado a Firestore Database para guardar los datos de la película en la lista de Ver más tarde, y se modifica y guarda el estado del botón Ver más tarde en Firestore Database 5. El sistema actualiza el botón de Ver más tarde apareciendo ahora seleccionado ya que la película fue guardada, y si el usuario está registrado y vuelve a acceder en otro momento a la vista de información de dicha película, el botón aparecerá seleccionado

Tabla 13: CU13 - El usuario guarda película en lista Ver más tarde

CU14	**El usuario elimina película de lista Ver más tarde**
Descripción	El usuario elimina una película de la lista de películas para Ver más tarde
Referencias	RF14, CU6, CU13
Actores	Usuario, Firebase Firestore Database, TMDB API
Flujo	1. El usuario accede a la vista de información de una película 2. El sistema realizará una llamada a la API de TMDB para obtener la información de la película 3. El sistema realizará una llamada a Firestore Database para saber si está guardada la película y marcar como seleccionado el botón correspondiente si lo está (si el usuario no está registrado esta acción no se realiza) 4. Si está seleccionado el botón, el usuario deseleccionará el botón de Ver más tarde y con ello eliminará la película, en el caso de que el usuario no esté registrado no podrá realizar esta acción, ya que en este caso el botón nunca aparecerá seleccionado 5. El sistema realiza una llamada de eliminar a Firestore Database para eliminar los datos de la película de la lista de Ver más tarde, y se modifica y guarda el estado del botón Ver más tarde en Firestore Database 6. El sistema actualiza el botón de Ver más tarde apareciendo ahora deseleccionado ya que la película fue eliminada, y si el usuario está registrado y vuelve a acceder en otro momento a la vista de información de dicha película, el botón aparecerá deseleccionado

Tabla 14: CU14 - El usuario elimina película de lista Ver más tarde

CU15	**El usuario accede al perfil**
Descripción	El usuario accede al perfil para ver con que cuenta se ha registrado o para ver si ha iniciado de forma anónima
Referencias	RF15
Actores	Usuario
Flujo	1. El usuario accede a la pantalla inicio de la aplicación 2. El usuario seleccionará el botón de perfil 3. Si el usuario está registrado podrá ver el correo electrónico con el que está registrado y podrá cerrar sesión desde ahí si lo desea, si el usuario no está registrado, al acceder al perfil podrá ver que pone “(Anónimo)” ya que ha entrado de forma anónima y desde ahí podrá ir a iniciar sesión o a registrarse.

Tabla 15: CU15 - El usuario accede al perfil

CU16	**El usuario cierra la sesión**
Descripción	El usuario cierra la sesión de la aplicación
Referencias	RF16
Actores	Usuario, Firebase Authentication
Flujo	1. El usuario accede a la pantalla de inicio de la aplicación 2. El usuario seleccionará el botón de perfil 3. Si el usuario está registrado verá en su perfil el correo electrónico con el que está registrado actualmente y podrá desde ahí cerrar sesión, por otro lado, si el usuario no está registrado simplemente verá que está de forma anónima y por lo tanto no podrá cerrar sesión

Tabla 16: CU16 - El usuario cierra la sesión

4.3 Arquitectura del sistema

En este apartado se explicará la arquitectura del sistema desarrollando las dos partes que componen la aplicación StreamList. Dichas dos partes son el frontend y el backend de la aplicación.

Para ello explicaré primero la arquitectura seguida a la hora de desarrollar la aplicación y después explicaré tanto el desarrollo del frontend, como el del backend teniendo en cuenta todos los componentes implicados.

4.3.1 Arquitectura MVVM

Primero explicaré el patrón de arquitectura seguido para el desarrollo de la aplicación, dicho patrón ha sido MVVM o Model-View-ViewModel, se caracteriza por tratar de separar la parte lógica del sistema, de la parte gráfica, es decir, de la interfaz del usuario, es un patrón muy utilizado a la hora de desarrollar aplicaciones.

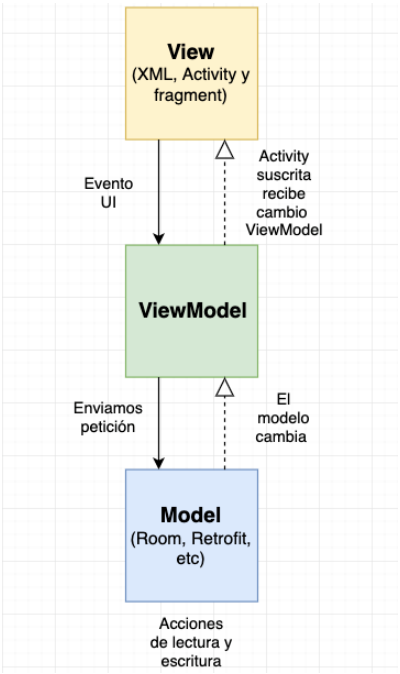

Ilustración 15: Arquitectura MVVM [12]

Este patrón consta de tres partes distintas como su propio nombre indica, que son las siguientes:

- **Model**: este módulo representa la parte de los datos y/o la lógica de negocio, es decir, se encargará de recuperar los datos necesarios de un servicio web o de una base de datos, para posteriormente almacenar esa información en modelos de datos. En esta aplicación, se encargará de obtener la información de la base de datos Firestore Database y también de TMDB API. Los diferentes modelos de datos de la aplicación se explicarán en el apartado 4.3.3.
- **View**: este módulo se encarga de representar la información mediante los elementos visuales que la componen, es la parte de la interfaz del usuario. La View estará suscrita de forma activa al ViewModel para que este último le envíe los datos que necesita para pintarlos por pantalla, cada vez que los datos se modifiquen el ViewModel se lo notificará a la View y esta actualizará la información de la pantalla con los nuevos datos. La View también notificará al ViewModel cuando el usuario realice una acción como pulsar un botón. Las diferentes vistas de la aplicación se explicarán más adelante en los apartados 4.3.2 y 4.4.
- **ViewModel**: este es un módulo intermediario que conecta la View con el Model. Como he explicado en el párrafo anterior la View se suscribirá a su correspondiente ViewModel y este último se encargará de enviarle los datos a la View o se encargará de notificar a la View cada vez que los datos hayan sido modificados enviándole los nuevos datos. Los diferentes ViewModels que conforman esta aplicación son los siguientes:

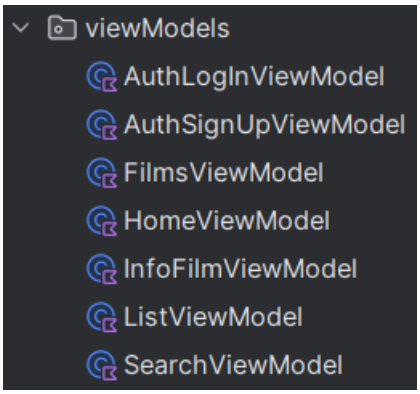

Ilustración 16: ViewModels de StreamList

1. **AuthLogInViewModel:** se encarga de enviar los datos que introduce el usuario al iniciar sesión, a Firebase Authentication, para comprobar que sean correctos y por lo tanto el usuario pueda iniciar la sesión. También se encarga de permitir acceder al usuario sin necesidad de registrarse, es decir, de forma anónima, mediante una solicitud a Firebase Authentication.
2. **AuthSignUpViewModel:** se encarga de enviar los datos que introduce el usuario, a Firebase Authentication, para poder registrar dicho nuevo usuario. También se encarga de permitir acceder al usuario sin necesidad de registrarse, es decir, de forma anónima, mediante una solicitud a Firebase Authentication.
3. **FilmsViewModel:** se encarga de enviar los datos necesarios de las películas, pedidos a TMDB API, a la vista de todas las películas de una categoría.
4. **HomeViewModel:** se encarga de enviar los datos necesarios de las películas, pedidos a TMDB API, a la vista de inicio, es decir, la vista donde están las

películas agrupadas en cuatro categorías. También se encarga de solicitar el cierre de sesión del usuario, mediante una solicitud a Firebase Authentication.

5. **InfoFilmViewModel:** se encarga de enviar los datos necesarios de la película seleccionada, pedidos a TMDB API, a la vista de información de una película. También se encarga de enviar los datos de la película, para guardarlos en la base de datos, en la lista correspondiente a la seleccionada por el usuario y, por lo tanto, también se encarga de solicitar que se elimine la película de la base de datos, de la correspondiente lista seleccionada por el usuario. Además, se encarga de solicitar que se guarde el estado de los tres botones Favorita, Vista y Ver más tarde cada vez que se modifique uno de ellos al guardar o eliminar una película de la lista correspondiente.
6. **ListViewModel:** se encarga de enviar los datos necesarios de las películas guardadas en una lista, pedidos a Firebase Firestore Database, a la vista de dicha lista.
7. **SearchViewModel:** se encarga de enviar los datos necesarios de las películas, pedidos a TMDB API, que coinciden con el título escrito por el usuario en el buscador, a la vista de búsqueda de una película.

4.3.2 Desarrollo del Frontend

En este apartado explicaré el desarrollo de la parte del frontend del sistema. La parte del frontend es la que el usuario puede observar en su dispositivo al acceder a la aplicación. En Android las vistas tienen dos partes el activity, que es la parte lógica de la vista y el layout o el xml, que es la parte visual de la vista.

A continuación, explicaré las vistas que componen la aplicación y con el diagrama de las vistas se podrá ver cómo están conectadas entre ellas, pero el detalle y diseño de la interfaz gráfica de cada una de las vistas lo explicaré en el apartado 4.4. Las vistas, es decir, las activitys y sus correspondientes layouts, que componen la aplicación son las siguientes:

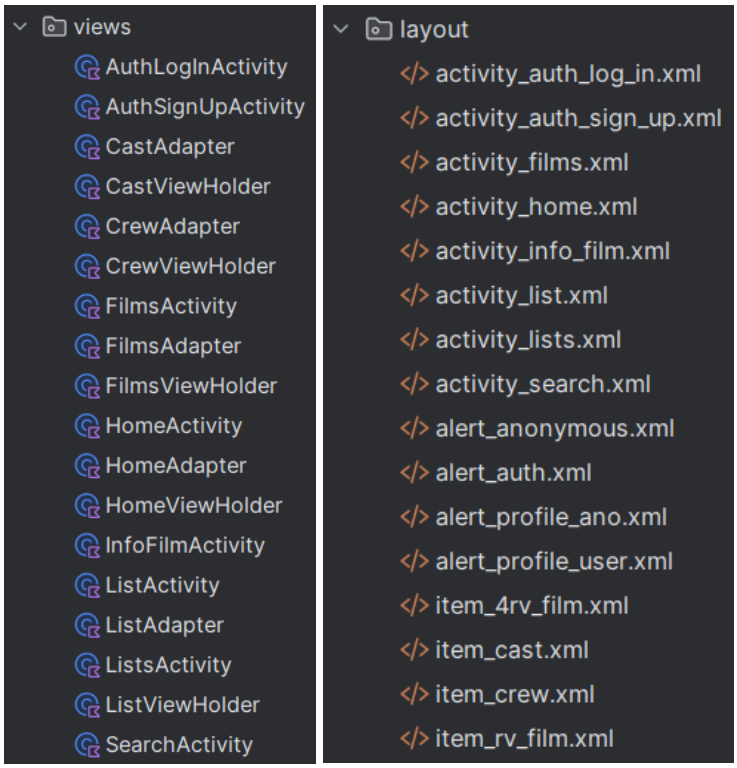

Ilustración 17: Views de StreamList

1. **AuthLogInActivity:** en esta vista el usuario podrá iniciar sesión o podrá acceder de forma anónima. Y su correspondiente layout es activity_auth_log_in.
2. **AuthSignUpActivity:** en esta vista el usuario podrá registrarse o podrá acceder de forma anónima. Y su correspondiente layout es activity_auth_sign_up.
3. **FilmsActivity:** en esta vista el usuario podrá visualizar todas las películas disponibles de una categoría. Y su correspondiente layout es activity_films.
4. **HomeActivity:** en esta vista el usuario podrá ver grupos de películas organizadas en cuatro categorías: Películas Actuales, Populares, Mejor Valoradas y Próximas Películas. Y su correspondiente layout es activity_home.
5. **InfoFilmActivity:** en esta vista se encontrará toda la información disponible de una película, seleccionada previamente por el usuario. Y su correspondiente layout es activity_info_film.
6. **ListActivity:** esta vista pintará las películas guardadas en cada una de las listas de Favoritas, Vistas y Ver más tarde dependiendo de cual haya seleccionado el usuario. Y su correspondiente layout es activity_list.
7. **ListsActivity:** esta vista contendrá el menú donde el usuario podrá seleccionar a que lista desea acceder, para ver las películas que tenga guardadas en dicha lista. Y su correspondiente layout es activity_lists.
8. **SearchActivity:** en esta vista el usuario podrá buscar una película por su título y posteriormente visualizar el resultado de la búsqueda. Y su correspondiente layout es activity_search.

Los otros ficheros que se ven en la ilustración 17, por un lado, en la carpeta de views, con nombres terminados en Adapter y en ViewHolder, son ficheros auxiliares que ayudan a las vistas a pintar los diferentes recyclerViews que componen la aplicación. Un recyclerView es un elemento gráfico de Android que permiten mostrar listas de elementos mediante una disposición gráfica establecida, en este caso son listas de películas y de personas, el reparto y el equipo. Por otro lado, en la carpeta de layout, aparecen varios ficheros que comienzan con alert, que son el diseño de los diferentes mensajes que aparecen en la aplicación y hay otros ficheros que comienzan por item, que son los diseños de los diferentes ítems de los recyclerViews.

4.3.2.1 Diagrama de las vistas

Una vez enumeradas y explicadas las vistas que componen la aplicación, podemos ver como el usuario puede ir navegando a través de las diferentes pantallas gracias al siguiente diagrama.

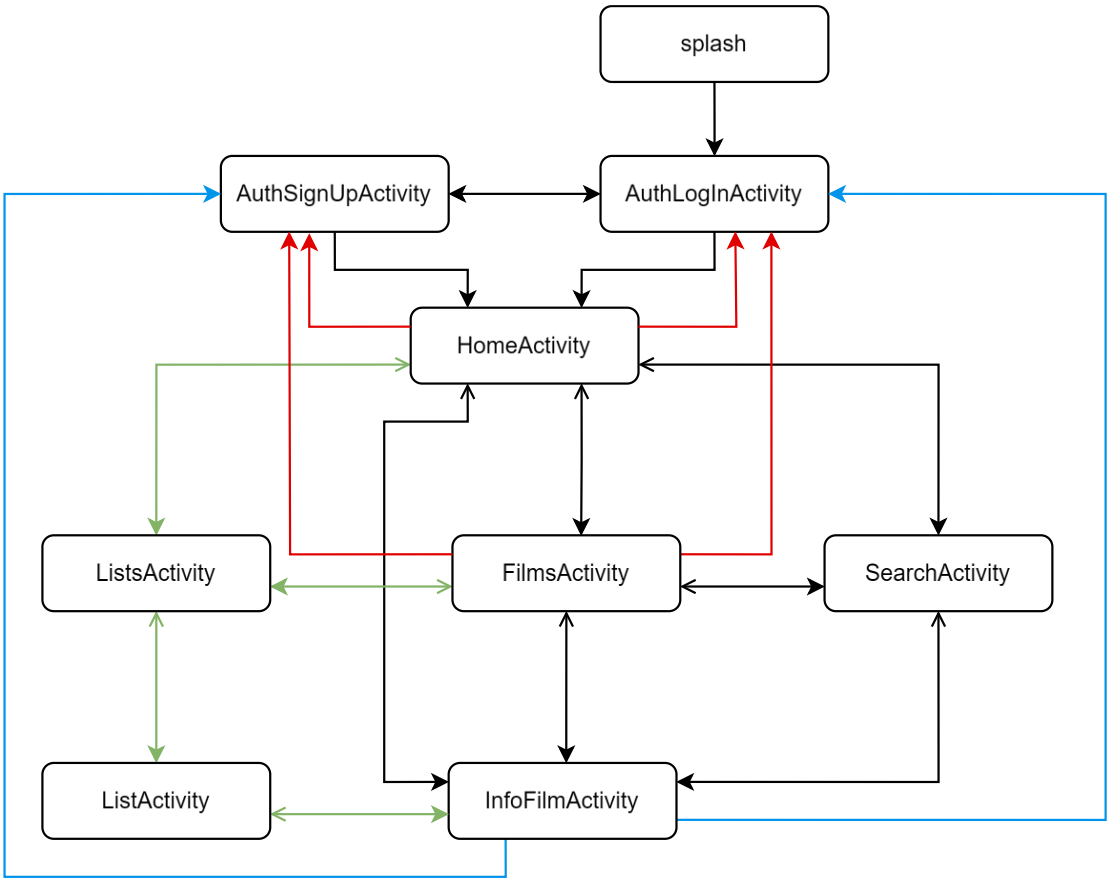

Ilustración 18: Diagrama de las vistas

Dado que en la aplicación pueden acceder usuarios registrados y usuarios no registrados, tendremos que tener en cuenta varias cosas, todos los usuarios, tanto los registrados como los no registrados, podrán acceder por las pantallas siguiendo las flechas negras, luego, solo los usuarios registrados podrán acceder a las vistas por las que se llega con una flecha verde, las flechas rojas y azules indican que los usuarios no registrados, podrán ir a registrarse o iniciar sesión si desean acceder a las listas o si desean guardar una película en alguna de las listas. Las flechas rellenas implican ir (ej. de HomeActivity vas a FilmsActivity) y las flechas no rellenas implican volver (ej. de FilmsActivity vuelves a HomeActivity). La navegación es lineal, es decir, por el camino que vas es por el que vuelves.

Una vez explicado el esquema por encima, pasaré ahora a detallar hacia donde el usuario puede ir en cada una de las vistas presentes.

Cuando se inicia la aplicación aparece el splash de carga y luego accedes al AuthLogInActivity, desde donde puedes ir a AuthSignUpActivity o puedes acceder a HomeActivity.

Desde AuthSignUpActivity puedes ir a AuthLogInActivity o puedes acceder a HomeActivity.

Cuando estás en la pantalla de HomeActivity puedes acceder a casi todas las vistas, desde aquí el usuario puede ir a FilmsActivity y volver, puede ir a InfoFilmActivity y puede volver, y puede ir a SearchActivity y volver, en el caso de que el usuario esté registrado podrá ir a ListsActivity y volver, si el usuario no está registrado podrá ir a AuthLogInActivity o a AuthSignUpActivity para registrarse y así poder acceder a ListsActivity.

Desde FilmsActivity puedes ir a InfoFilmActivity y volver, puedes ir a SearchActivity y volver, y si el usuario está registrado podrá acceder a ListsActivity y volver, en caso contrario podrá acceder a AuthLogInActivity o a AuthSignUpActivity para registrarse y así poder acceder a ListsActivity.

Estando en SearchActivity podrás ir a InfoFilmActivity y volver. Desde ListsActivity podrás ir a ListActivity y volver. Y desde ListActivity el usuario podrá ir a InfoFilmActivity y volver.

Desde InfoFilmActivity indico con flechas azules que los usuarios que no estén registrados pueden ir a AuthLogInActivity o a AuthSignUpActivity a registrarse, para así poder guardar una película, o eliminar en caso de que ya la tuvieran guardada.

4.3.3 Desarrollo del Backend

En este apartado explicaré el desarrollo de la parte backend del sistema. Veremos los servicios utilizados para proveer la información necesaria a la aplicación, explicaré los diseños y las llamadas, a la API de TMDB y a las bases de datos, en los diferentes apartados.

Pero primero explicaré los diferentes diseños de modelos de datos que tiene la aplicación y que gracias a ellos podemos obtener y almacenar la información tanto de la API, como de la base de datos y también veremos las clases que se encargan de realizar las funciones de llamadas y demás. Por lo tanto, los diferentes modelos son los siguientes:

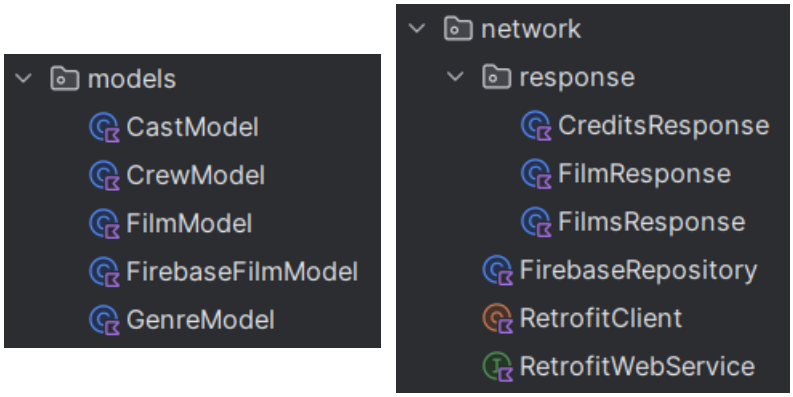

Ilustración 19: Models de StreamList

- **FilmsResponse:** es el modelo de datos que se encarga de almacenar la lista de las películas pedidas a la API TMDB, dicha lista utiliza otro modelo de datos llamado FilmModel
- **FilmModel:** es el modelo de datos que tiene la información de una película, es decir, tiene el id de la película, la ruta de la imagen, la fecha de estreno, el título y la puntuación. Este modelo y el anterior serán los encargados de almacenar la información de las películas que se pintan

en los recyclerView de las pantallas FilmsActivity, HomeActivity y SearchActivity.

- **FilmResponse:** es el modelo de datos que almacenará la información de la película que se mostrará en la pantalla de InfoFilmActivity. Dichos datos son una lista de géneros que será otro modelo de datos llamado GenreModel, el id de la película, el título original, la descripción, la ruta de la imagen, la fecha de estreno, la duración, el título y la puntuación.
- **GenreModel:** es un modelo de datos con el que podremos almacenar los géneros de una película.
- **CreditsResponse:** este modelo tendrá una lista del reparto de la película que utilizará otro modelo que es CastModel y también tendrá una lista del equipo que desarrolló y dirigió la película que utilizará otro modelo llamado CrewModel. Esto se utilizará para pintar los dos recyclerView de reparto y equipo que se encuentran en la vista InfoFilmActivity.
- **CastModel:** modelo que contendrá datos de las personas del reparto, que son el departamento, el nombre, la foto y el personaje.
- **CrewModel:** modelo que contendrá datos de las personas del equipo, que son el departamento, el nombre, la foto y el trabajo que realizó.
- **FirebaseFilmModel:** es el modelo con el que almacenaremos los datos de las películas que se piden a la base de datos, ya que han sido guardadas ahí debido a que el usuario la ha guardado en alguna de las listas existentes, este modelo se usará para pintar el recyclerView de la vista de ListActivity y contendrá los siguientes datos necesarios: el id de la película, la ruta de la imagen, la fecha de estreno, el título y la puntuación.

Por otro lado, están las clases RetrofitClient y RetrofitWebService las cuales se encargarán de las llamadas a la API TMDB y luego está la clase FirebaseRepository que se encargará de las llamadas y las funciones de Firebase Authentication y de Firebase Firestore Database.

4.3.3.1 Llamadas a la API TMDB

A continuación, explicaré las llamadas que se realizan a la API de TMDB junto con el resultado que se obtiene de cada llamada.

Las llamadas a la API para obtener las listas de películas separadas por categorías, para la vista de HomeActivity y para la de FilmsActivity con una categoría, son las siguientes:

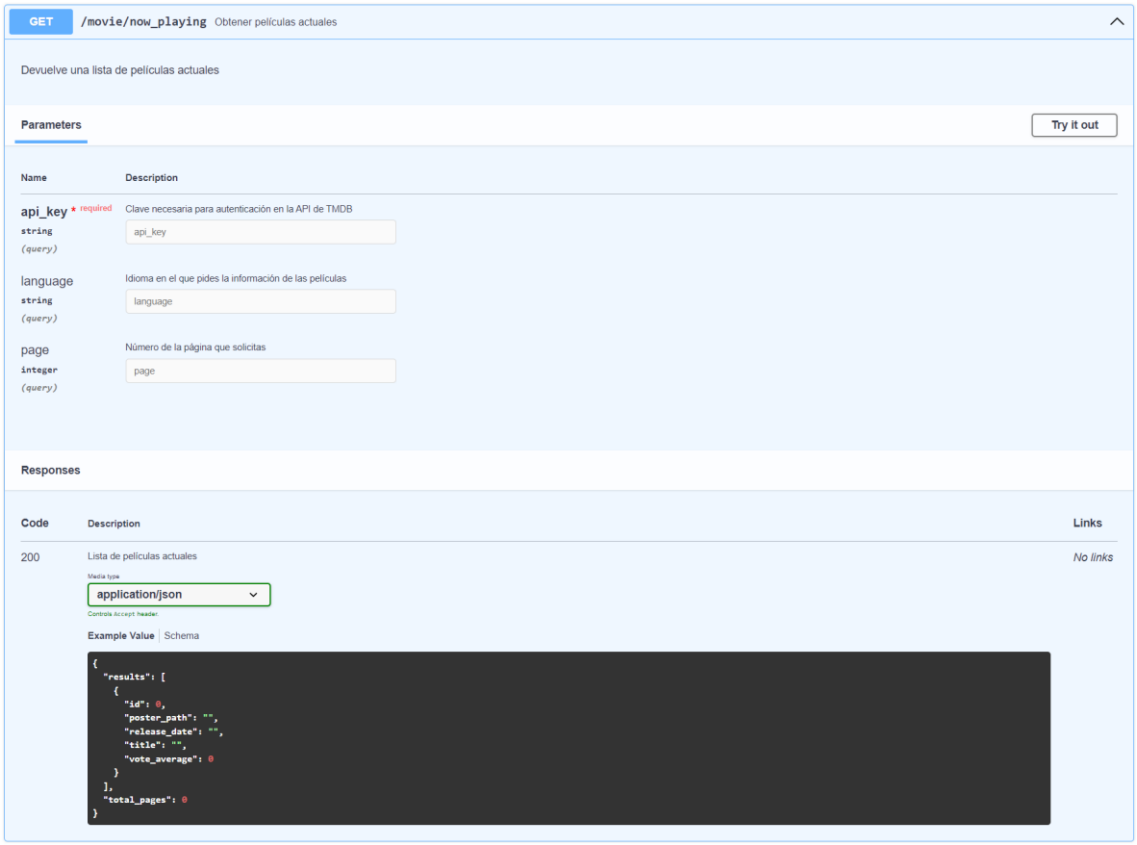

Ilustración 20: Llamada a TMDB API de Películas Actuales

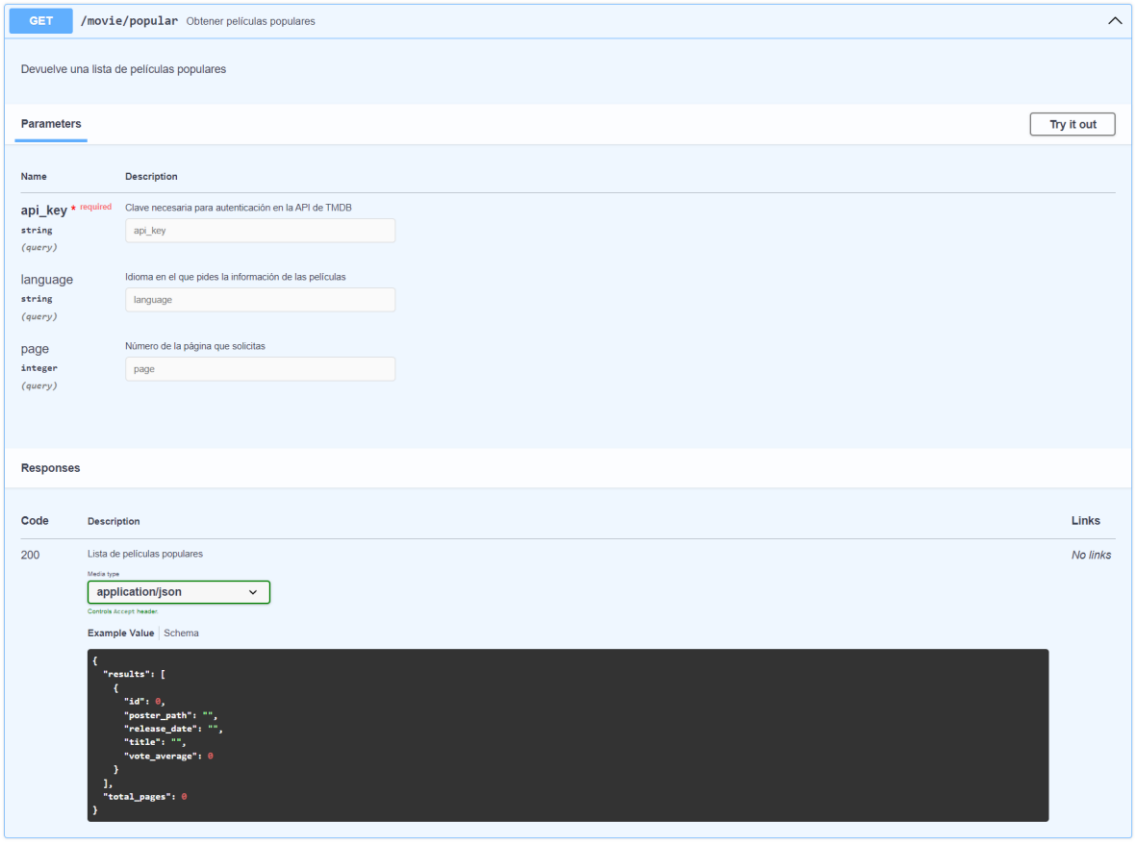

Ilustración 21: Llamada a TMDB API de Populares

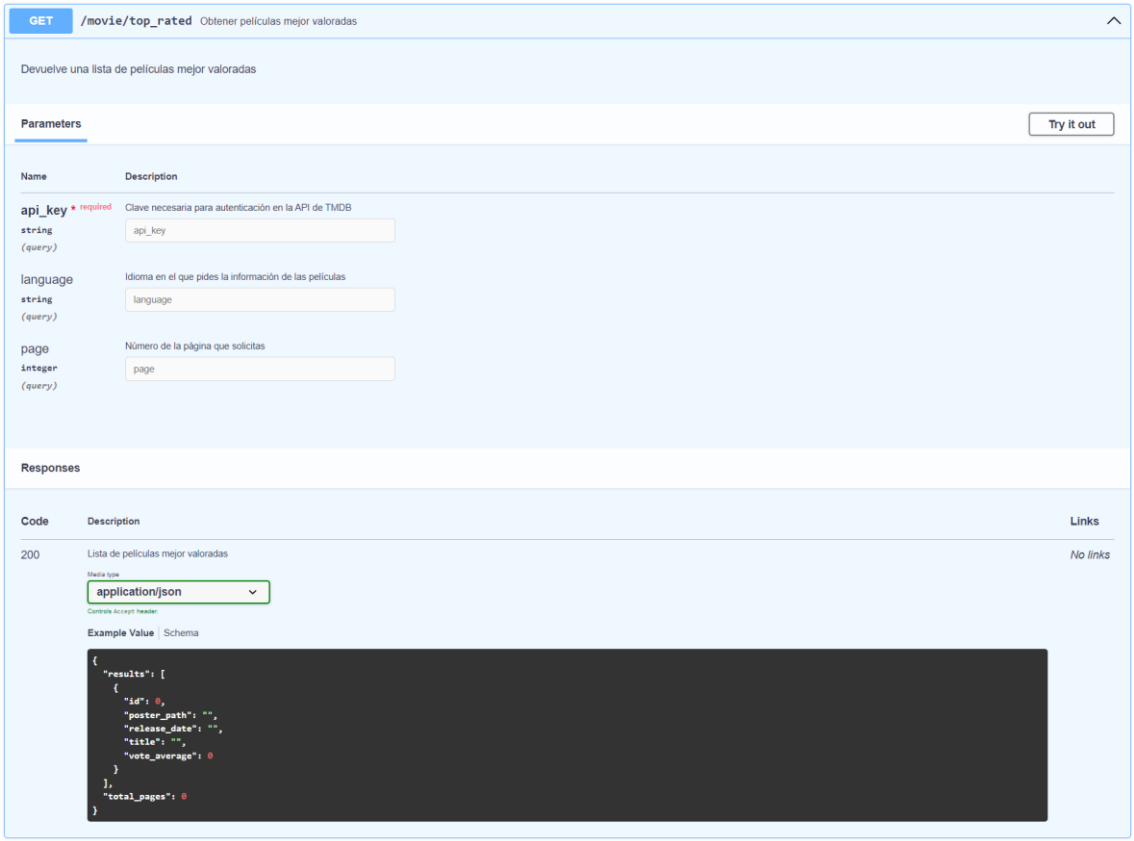

Ilustración 22: Llamada a TMDB API de Mejor Valoradas

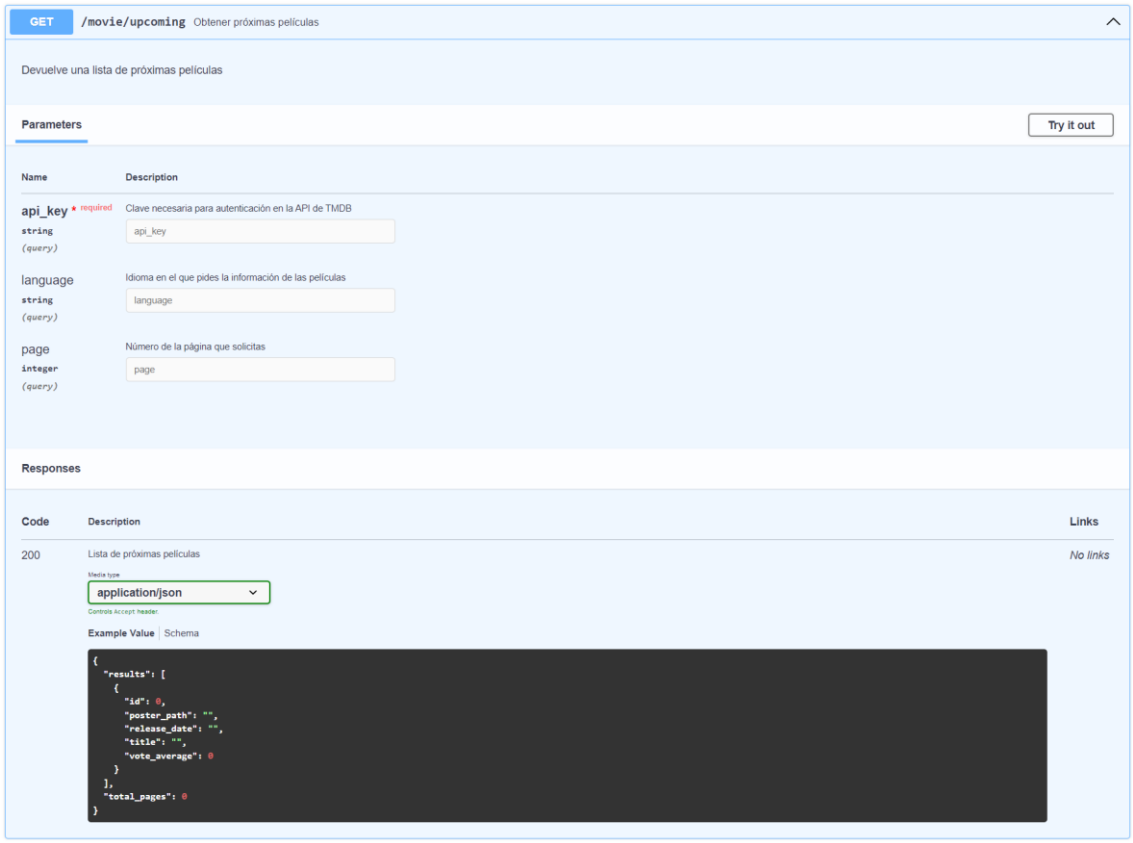

Ilustración 23: Llamada a TMDB API de Próximas Películas

Las llamadas para obtener el detalle de una película para la vista de InfoFilmActivity, son las siguientes:

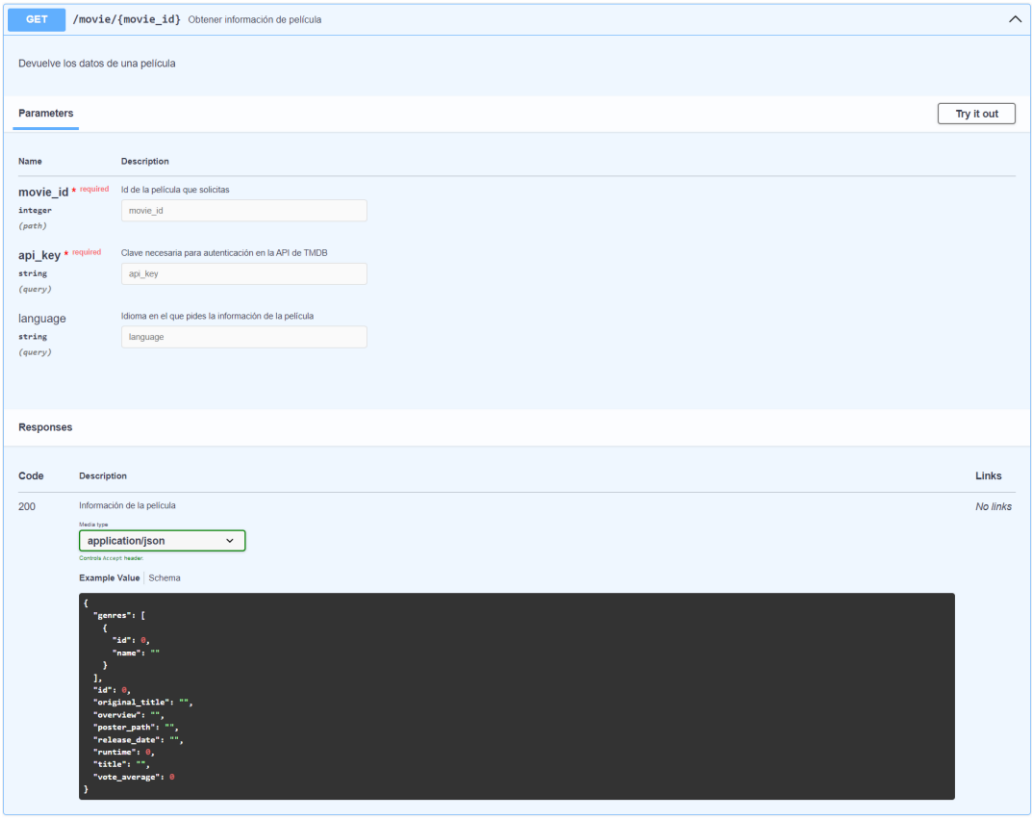

Ilustración 24: Llamada a TMDB API de Información de Película 1

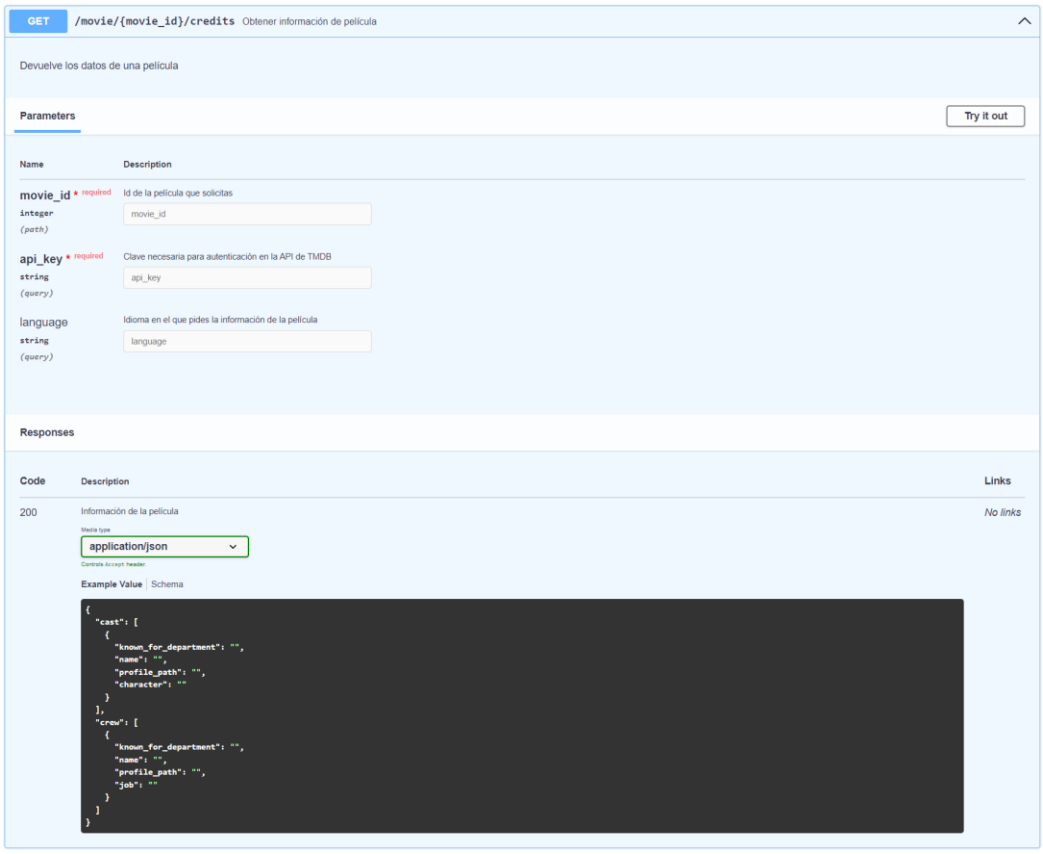

Ilustración 25: Llamada a TMDB API de Información de Película 2

La llamada para obtener la lista de películas al realizar una búsqueda en la pantalla de SearchActivity, es la siguiente:

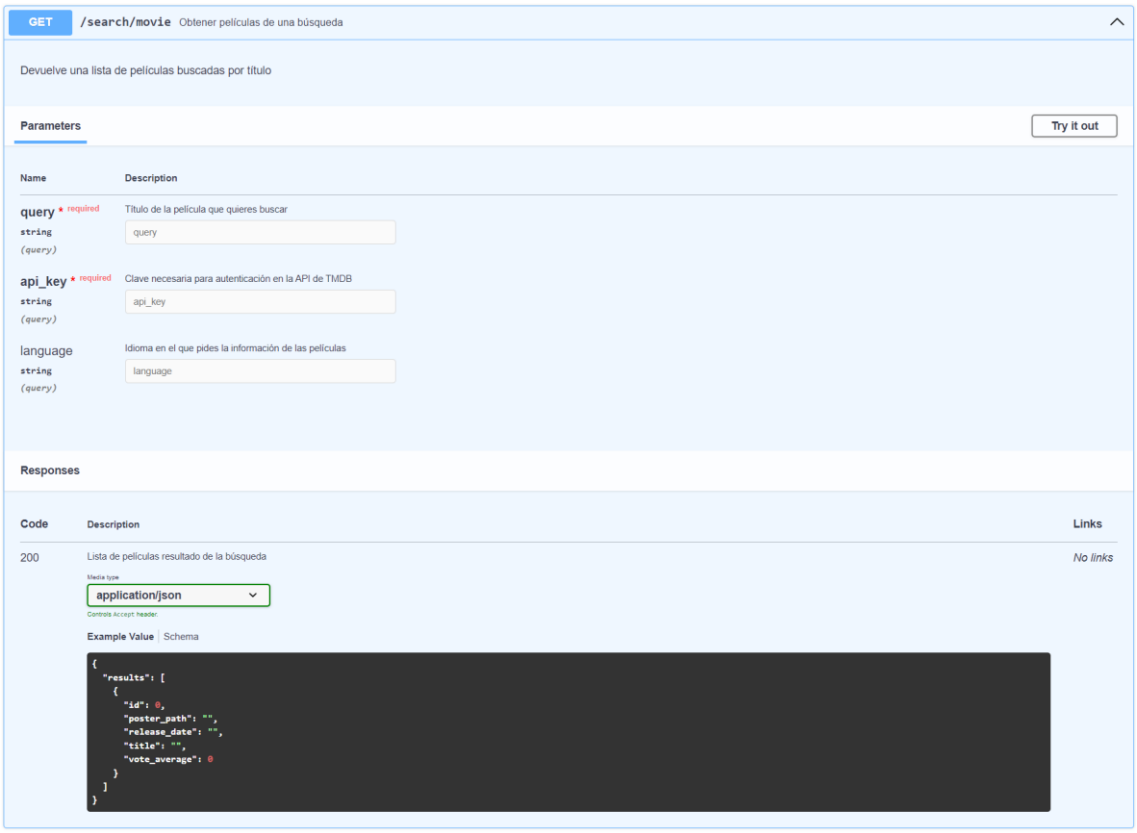

Ilustración 26: Llamada a TMDB API de Búsqueda de una Película

4.3.3.2 Diseños de las Bases de datos

En este apartado veremos el diseño de las dos bases de datos, utilizadas para el almacenamiento de los datos necesarios para algunas partes de la aplicación.

Las dos bases de datos son las ofrecidas por los servicios de Firebase Authentication y por Firebase Firestore Database.

- **Firebase Authentication**

Primero veremos el servicio de autenticación, el cual dispone de una base de datos, donde se almacenarán diferentes datos de un usuario cuando este se registre y algunos de ellos se modificarán a lo largo del tiempo.

Dichos datos almacenados son el identificador, el cual es diferente dependiendo de otro dato que es el proveedor, la contraseña, luego están la fecha de creación de la cuenta, la fecha del último acceso del usuario a la aplicación y finalmente tenemos un UID de usuario.

El identificador como he mencionado antes depende del proveedor, para esta aplicación habilité el método de acceso de "Anónimo" y el de "Correo electrónico/contraseña", por lo que el identificador del usuario al acceder con dichos métodos será "(anónimo)" o el correo electrónico introducido por el usuario respectivamente. El último mencionado, es decir, el correo electrónico,

será el identificador que usaré para la base de datos de Firestore Database, a la hora de almacenar las películas guardadas en las listas de cada uno de los usuarios. La contraseña, por seguridad, no será accesible.

Ilustración 27: Diseño base de datos Firebase Authentication

- **Firebase Firestore Database**

Ahora veremos el diseño de la base de datos del servicio de Firebase Firestore Database, en la cual se almacenarán las diferentes películas que el usuario guarde dentro de cada una de las tres listas: Favoritas, Vistas y Ver más tarde.

Como he mencionado antes, la base de datos de este servicio es NoSQL, por lo que en ella encontraremos colecciones y documentos para almacenar los datos necesarios.

Las colecciones guardan un grupo de documentos y cada documento puede almacenar varios tipos de datos, como string, boolean y más. También dentro de cada documento puedes crear más colecciones, que a su vez cada una contenga su propio grupo de documentos, consiguiendo una anidación. Las colecciones que se encuentran dentro de un documento se llaman subcolecciones. También destacar que cada colección y cada documento dentro de una colección deben tener un nombre único.

Para esta aplicación he creado una colección de usuarios llamada “users”, en la que almaceno documentos cuyo nombre es el correo electrónico de cada usuario. Dentro de cada uno de los documentos de un usuario creo 4 colecciones llamadas “fav_list”, “view_list”, “watch_later_list” y “saves”. Las tres primeras serán las encargadas de guardar los documentos de las películas guardadas en cada una de las listas Favoritas, Vistas y Ver más tarde respectivamente, en dichos documentos se almacenarán los datos necesarios de las películas que serán pedidos más tarde por la vista ListActivity. Estos documentos se guardarán con un nombre único que será el id de la película. La cuarta colección, “saves” se encargará de almacenar documentos también guardados con nombre el id de la película y dentro de cada documento habrá tres variables “saveF”, “saveV” y “saveWL”, las cuales servirán para saber que película está guardada en que lista.

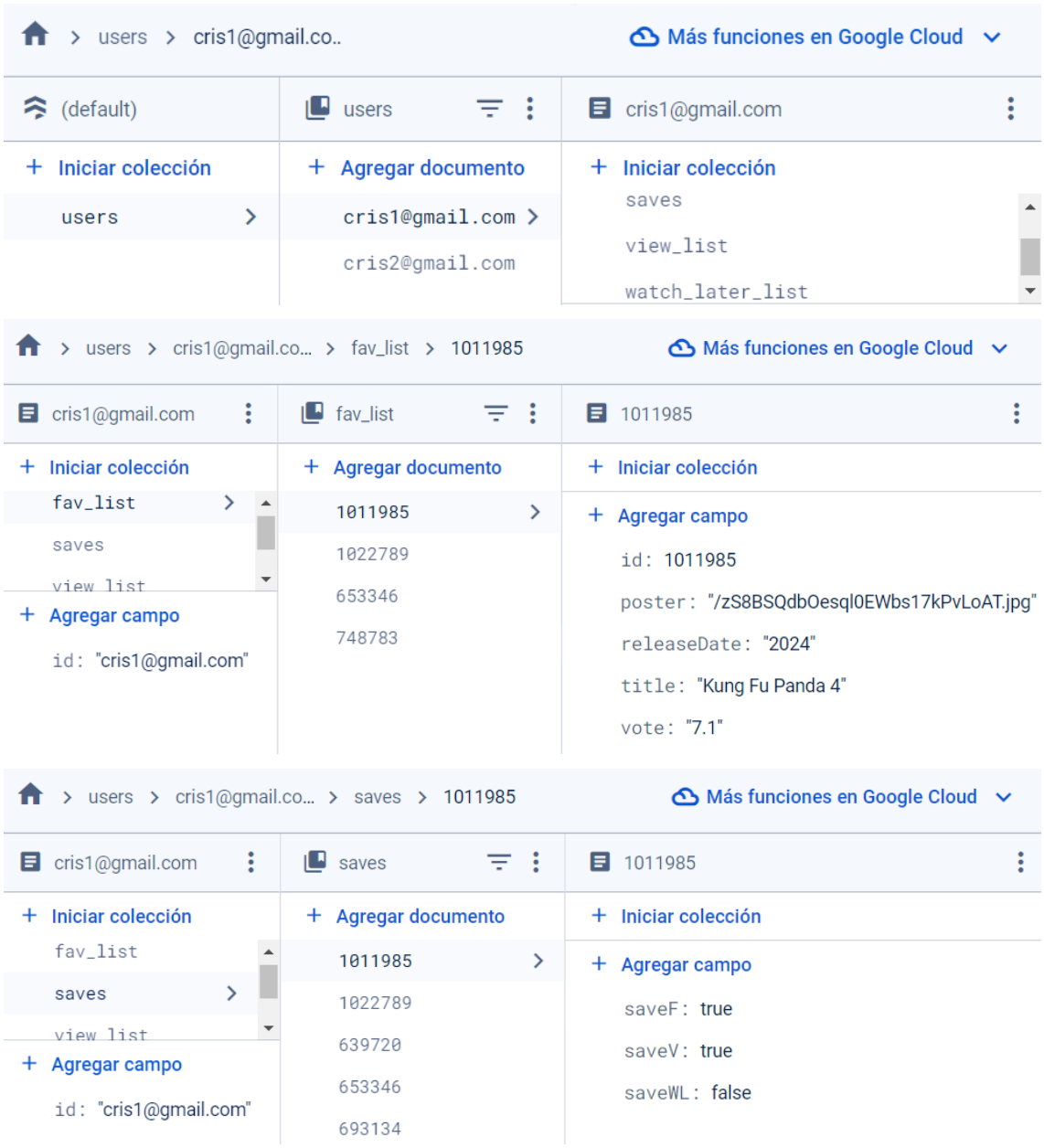

Ilustración 28: Diseño base de datos Firebase Firestore Database

4.4 Diseño de la aplicación

En estos apartados veremos el diseño y como se ven las diferentes pantallas que componen la aplicación, tal como las observarán los usuarios al acceder a la aplicación.

Primero se determinó que la paleta de colores que se iba a utilizar para el diseño de toda la aplicación, estaría compuesta por tres colores principales distintos y luego se utilizarían variaciones de esos colores. Dichos colores serían el rosa, el blanco y el negro, y como he mencionado antes se utilizarían variaciones de estos.

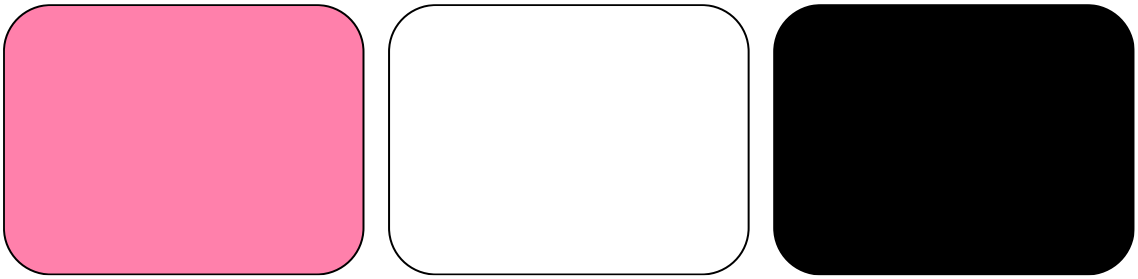

Ilustración 29: Paleta de colores principales aplicación

Por otro lado, antes de empezar a desarrollar la aplicación se hizo un boceto, en Figma, para determinar cómo iban a ser las diferentes pantallas y así tener una idea clara de lo que se quería hacer.

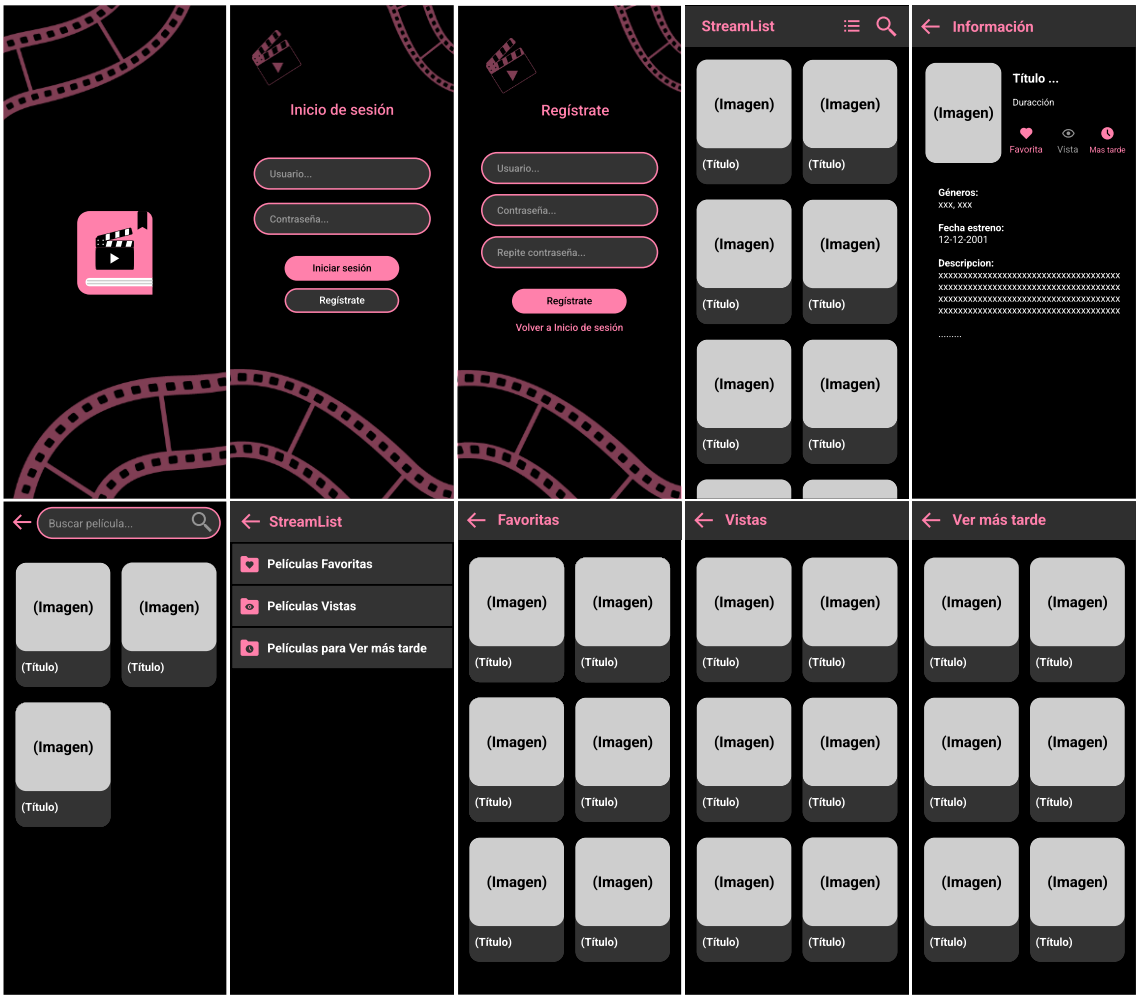

Ilustración 30: Prototipo diseño pantallas aplicación

También se crearon diferentes diseños para el logo e icono de la aplicación y se eligió el nombre de la aplicación, el cual sería “StreamList”. A continuación, se pueden ver los diferentes logos:

Ilustración 31: Logo diseño 1

Ilustración 32: Logo diseño 2

Finalmente se escogió el segundo logo para utilizar tanto en el icono, como en el splash. Aquí podemos ver el icono y el nombre de la aplicación:

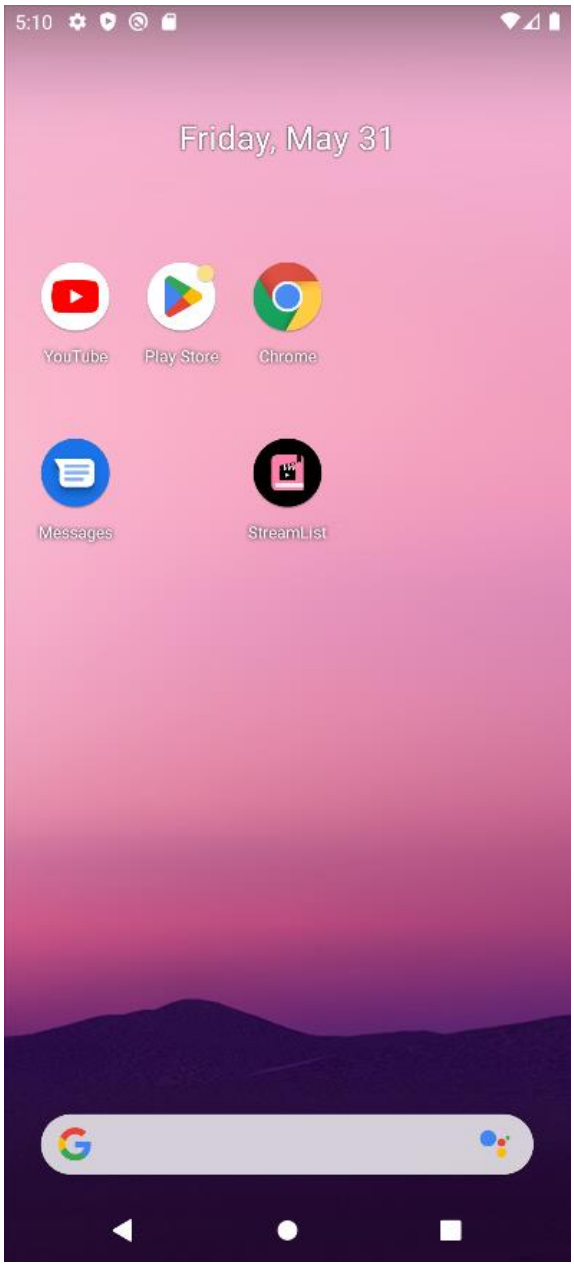

Ilustración 33: Pantalla icono y nombre de la aplicación

4.4.1 Vista splash

Lo primero que se lanzará al ejecutar la aplicación será el splash, ya que es la pantalla de carga de la aplicación, por lo tanto, esto es lo primero que verá el usuario.

Dicho splash contendrá el logo de la aplicación elegido como he mencionado antes y después de que salga dicho splash el usuario accederá a la vista AuthLogInActivity donde podrá iniciar sesión.

Ilustración 34: Vista splash

4.4.2 Vista inicio de sesión - AuthLogInActivity

Cuando la aplicación haya terminado de cargar, el usuario verá esta vista, la vista de inicio de sesión. En dicha vista el usuario podrá realizar varias acciones, por un lado, si ya se registró previamente el usuario podrá iniciar sesión rellenando los campos correctamente y presionando el botón de "Iniciar sesión", tras eso pasará a ver la pantalla de HomeActivity donde podrá ver las películas por categorías.

También podrá el usuario pasar directamente a la pantalla HomeActivity si presiona el botón que dice "Entrar sin registrarse", ya que de esta forma entrará de manera anónima a la aplicación.

Y la última acción que podrá realizar desde esta pantalla será ir a la vista de AuthSignUpActivity, presionando el botón que dice "Registrarse", donde el usuario se podrá registrar, si todavía no lo ha hecho.

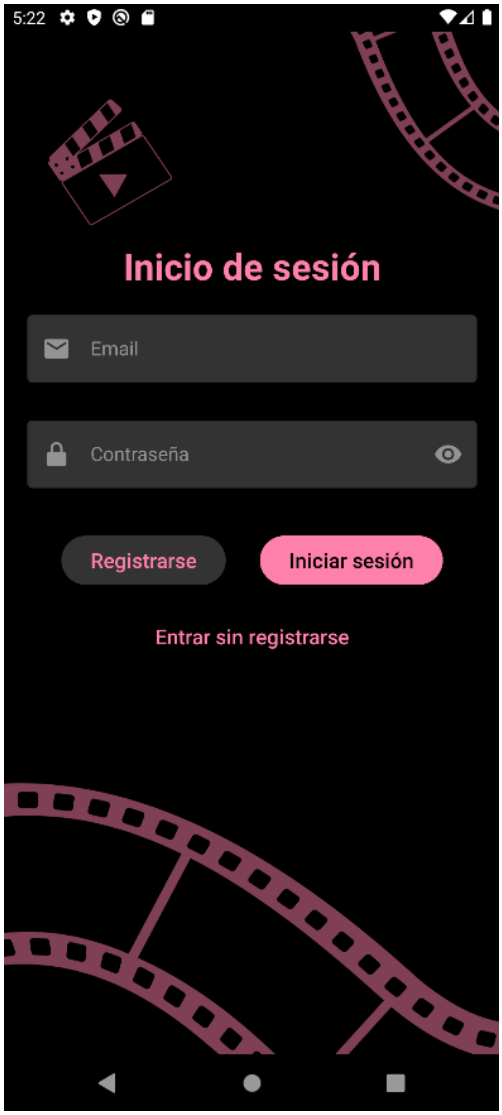

Ilustración 35: Vista inicio de sesión

4.4.3 Vista de registro - AuthSignUpActivity

En esta vista el usuario podrá registrarse si introduce los campos correctamente y presiona el botón de “Crear cuenta”, de ahí pasaría luego a la pantalla inicial de la aplicación HomeActivity.

También puede presionar en “Entrar sin registrarse” si desea entrar de forma anónima y accederá a la pantalla HomeActivity.

Por último, si el usuario lo desea, podrá regresar a la pantalla de inicio de sesión, presionando el botón que dice “Volver a Inicio de sesión”.

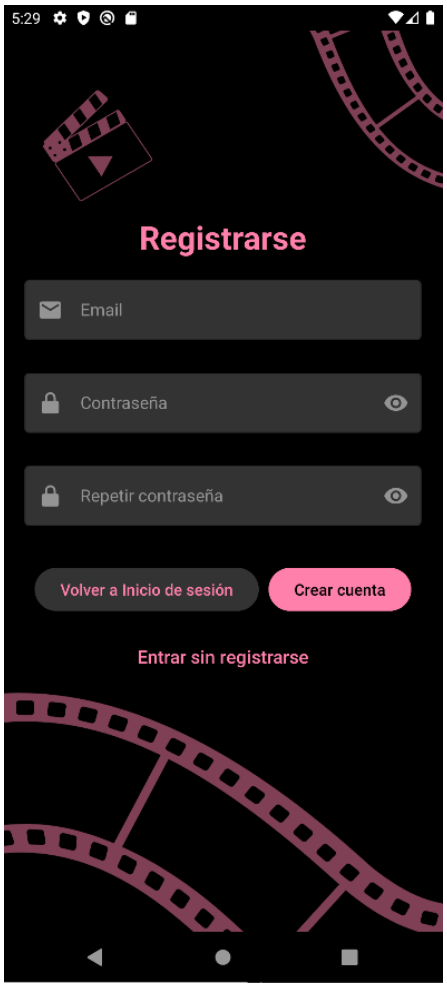

Ilustración 36: Vista de registro

4.4.4 Vista inicio - HomeActivity

Esta es la vista inicial de la aplicación, a la cual el usuario accederá después de haber estado en AuthLogInActivity o en AuthSignUpActivity, desde ella el usuario podrá acceder a prácticamente toda la aplicación. Como podemos ver en la ilustración 37, en esta pantalla los usuarios podrán ver grupos de películas agrupadas en cuatro categorías: Películas Actuales, Populares, Mejor Valoradas y Próximas Películas. También podemos ver que cada una de las categorías tiene un botón que dice “Ver más”, si el usuario lo presiona podrá acceder a la pantalla de FilmsActivity donde el usuario podrá navegar por las diferentes páginas para ver todas las películas de la categoría que haya seleccionado.

Si el usuario quiere ver la información de cualquiera de las películas de esta pantalla, solo tendrá que seleccionarla e irá directamente a la pantalla InfoFilmActivity donde podrá ver toda la información de la película.

Por otro lado, en esta pantalla hay una Toolbar arriba, donde se encuentra en primer lugar el nombre de la aplicación “StreamList”, y a la derecha hay tres botones. Empezando por el botón de la izquierda, si el usuario lo selecciona ocurrirán dos cosas, si el usuario está registrado irá a la pantalla de ListsActivity, pero si el usuario no está registrado no podrá acceder a dicha pantalla y saldrá un mensaje de alerta indicándole al usuario que debe iniciar sesión o registrarse para acceder a las listas.

El botón de en medio llevará al usuario a la pantalla de SearchActivity, donde podrá buscar las películas que quiera por su título.

Y el último botón, el de la derecha, será el perfil del usuario, donde este podrá ver el correo con el que se ha registrado y cerrar sesión, o si el usuario no estaba registrado podrá ver desde el perfil que está en anónimo y podrá registrarse o iniciar sesión.

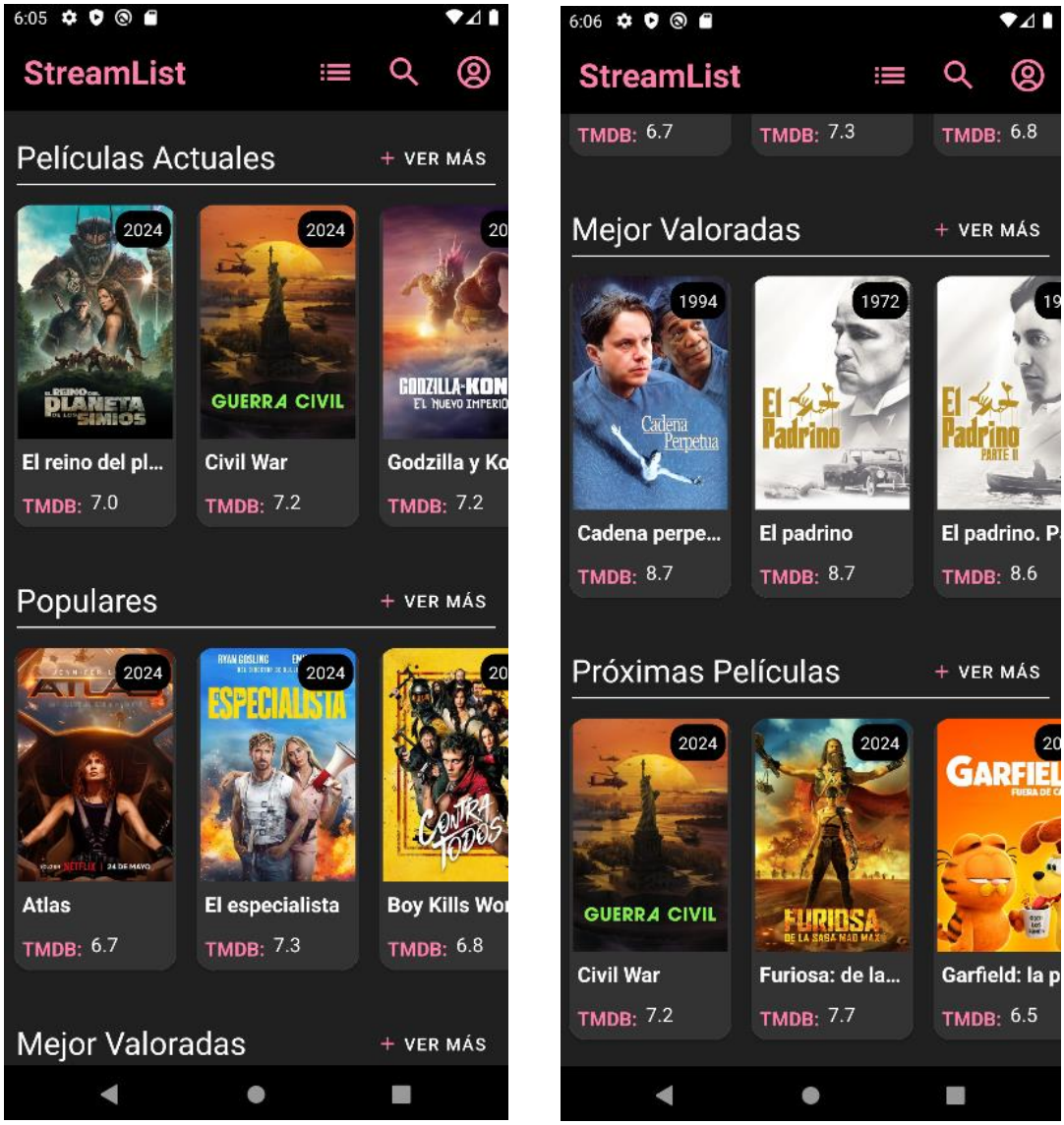

Ilustración 37: Vista inicio

4.4.5 Vista películas - FilmsActivity

A esta vista se accede cuando el usuario selecciona el botón de “Ver más” de una de las categorías de la vista anterior. En esta vista el usuario podrá visualizar todas las películas de una categoría, navegando por las diferentes páginas de esta, gracias a los botones de “Página anterior” y “Siguiente página”, y sabrá en que página se encuentra gracias al número situado en medio de los dos botones, como vemos en la ilustración 38.

Desde esta vista el usuario podrá realizar varias acciones, como ir a la vista de InfoFilmActivity tras seleccionar una película, para ver su información. En la parte de arriba de la vista vemos una ToolBar, en ella, empezando desde la izquierda tenemos un botón (la flecha) y si el usuario selecciona dicho botón volverá a la pantalla de HomeActivity, lo siguiente que vemos en la ToolBar será el nombre de la categoría en la que nos encontramos, que se modificará dependiendo de la categoría en la que estemos, y por último tenemos dos botones, el que llevaría al usuario a las listas si está registrado o si no lo está saldría un mensaje de error y el botón que nos lleva a la búsqueda de una película, estos dos botones realizan lo mismo que he explicado en la pantalla anterior.

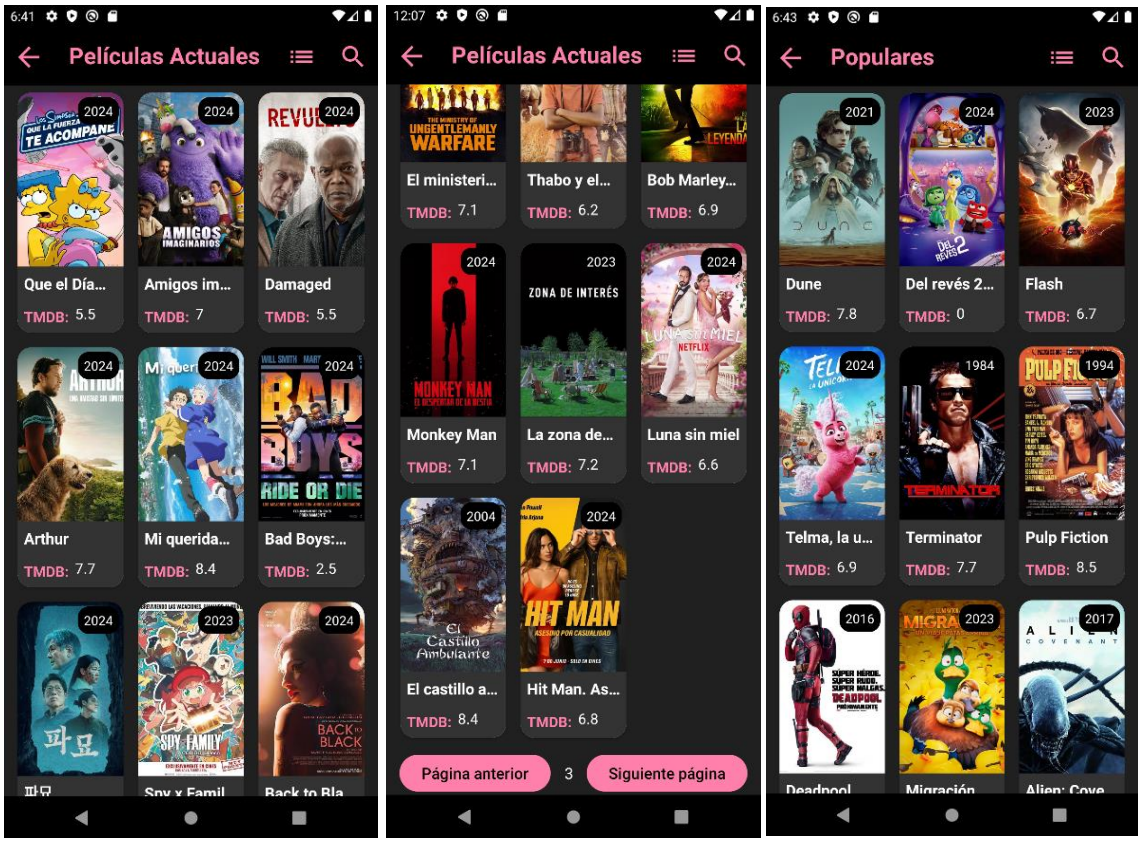

Ilustración 38: Vista películas

4.4.6 Vista información de una película - InfoFilmActivity

A esta vista se accede desde varias otras vistas, las cuales son HomeActivity, FilmsActivity, ListActivity y SearchActivity. En esta vista el usuario podrá ver toda la información de la película seleccionada. Dicha información será el poster de la película, el título, la fecha de estreno, la duración, la puntuación que le han dado los usuarios de TMDB, los géneros, el título original, la descripción, el reparto y el equipo de dirección y desarrollo. También en la parte de arriba de la vista hay una ToolBar, en la cual se encuentra el botón de volver a la vista desde la que se accedió a esta y el nombre de la película.

Por otro lado, hay tres botones que ponen “Favorita”, “Vista” y “Ver más tarde”, con ellos si el usuario está registrado podrá guardar la película en la lista correspondiente y el botón se seleccionará indicando que se ha guardado, si el usuario ya había guardado la película en alguna de las listas, podrá deseleccionar el botón para eliminar la película de dicha lista.

Por el contrario, si el usuario no está registrado y trata de seleccionar uno de estos botones, saldrá un mensaje de error diciendo que no puede guardar la película en la lista correspondiente y que para ello deberá registrarse o iniciar sesión. En dicho mensaje también habrá dos botones con los cuales el usuario podrá ir a las vistas de AuthLogInActivity y de AuthSignUpActivity y realizar la acción correspondiente.

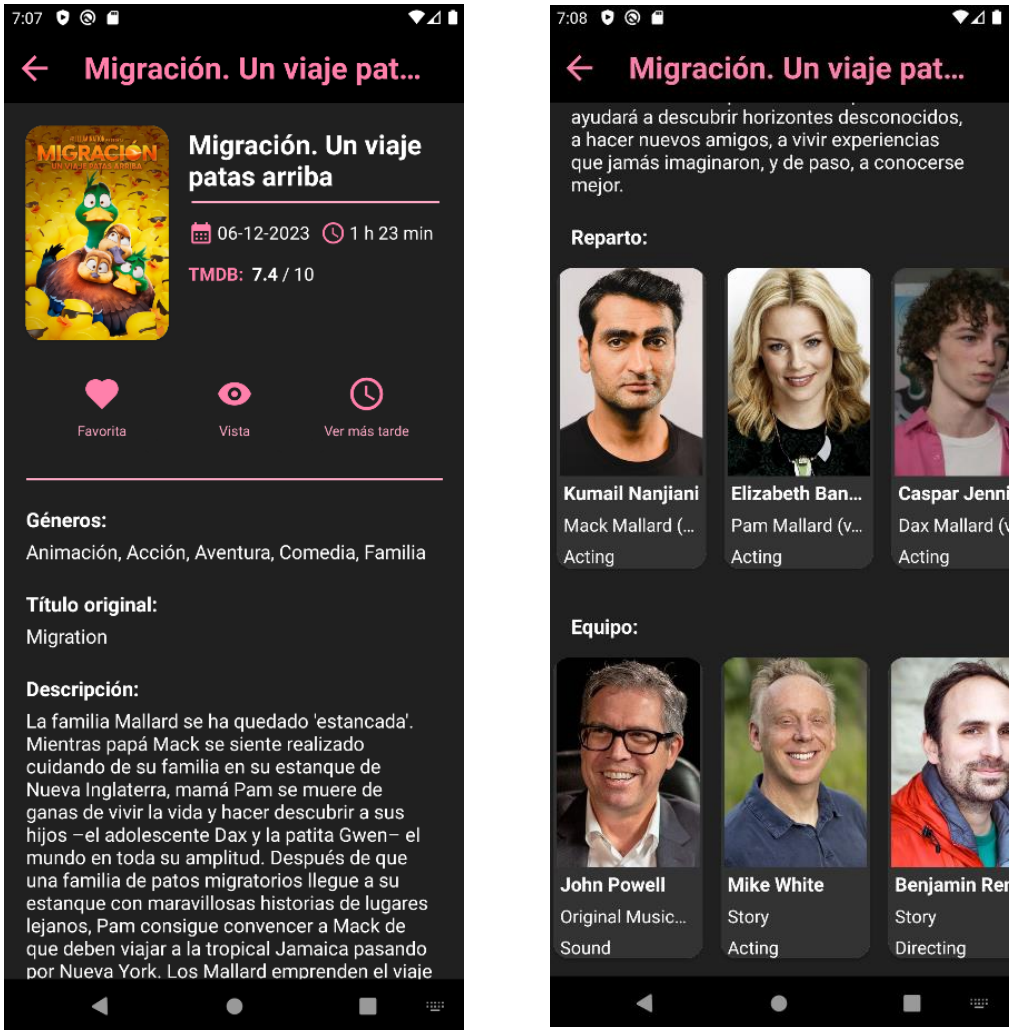

Ilustración 39: Vista información de una película

4.4.7 Vista listas - ListsActivity

A esta pantalla se puede acceder desde HomeActivity y desde FilmsActivity, pero solo pueden acceder a ella los usuarios que estén registrados. En esta pantalla podemos ver tres carpetas, las cuales son las tres listas: Favoritas, Vistas y Ver más tarde, en las que el usuario puede guardar las películas que desee.

En la parte de arriba tenemos la Toolbar, que contiene el botón para volver a la vista desde la que el usuario accedió a esta y también podemos ver el título de esta pantalla que es “Listas”.

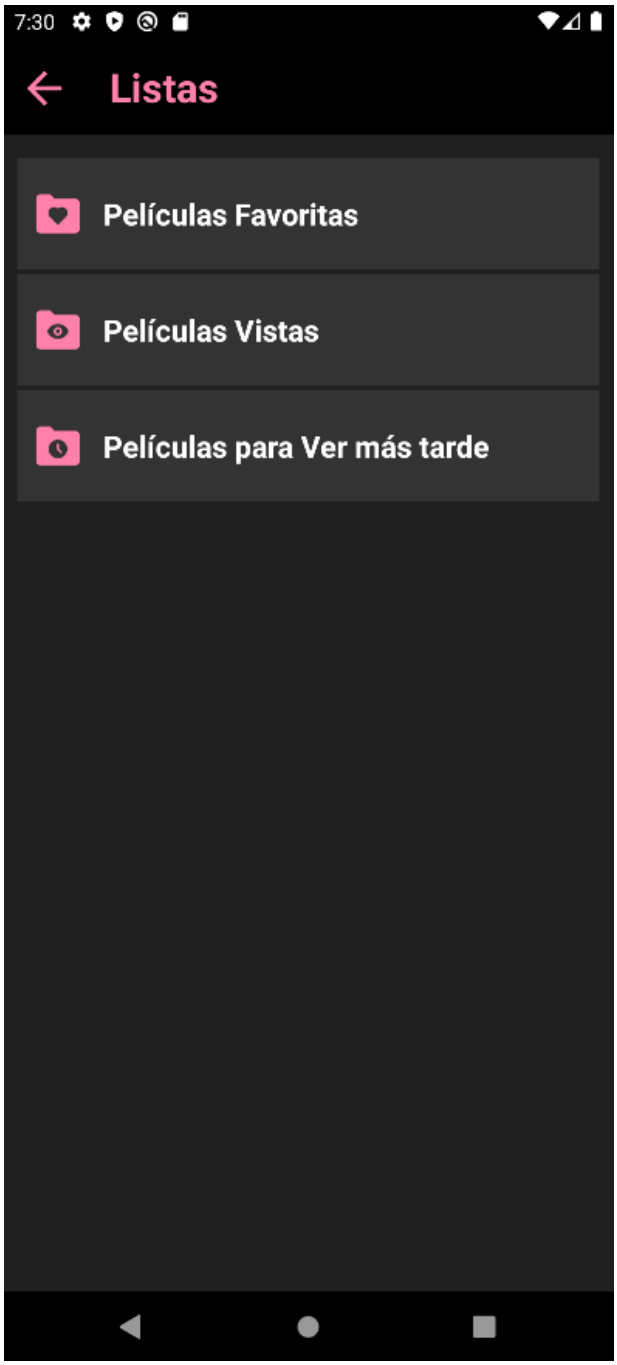

Ilustración 40: Vista listas

4.4.8 Vista una lista - ListActivity

El usuario puede acceder a esta pantalla solo desde la vista de ListsActivity y si está registrado. En esta pantalla el usuario podrá ver las películas que haya guardado en cada una de las listas, también podemos ver que justo debajo de la Toolbar aparece el número de películas guardadas que hay en la lista correspondiente. Por otro lado, arriba en la Toolbar el usuario podrá volver a la vista anterior gracias al botón de la izquierda y también en la Toolbar podemos ver el nombre de la lista en la que te encuentras.

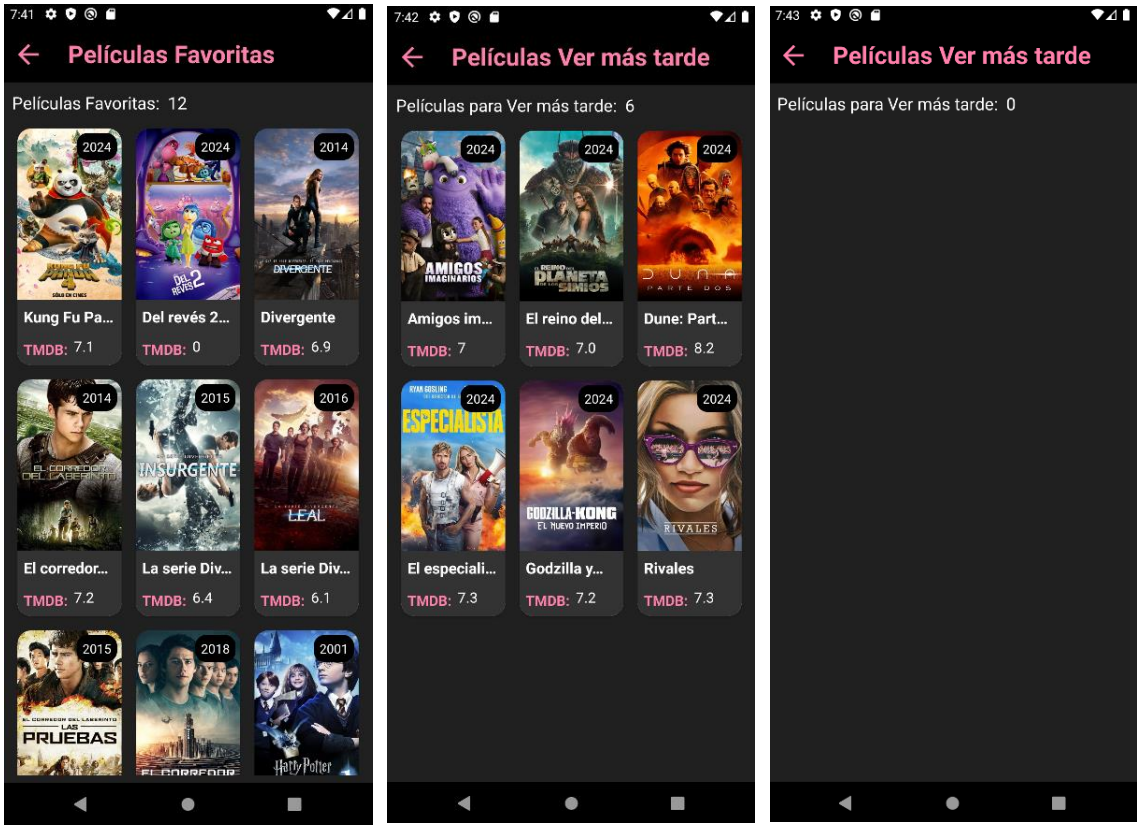

Ilustración 41: Vista una lista

4.4.9 Vista de búsqueda de películas - SearchActivity

A esta vista se puede acceder desde HomeActivity y desde FilmsActivity. En ella al principio podremos ver en la Toolbar un botón para volver a la pantalla anterior y al lado un buscador, con el que el usuario podrá buscar la película que desee por su título. En cuanto el usuario empiece a escribir, empezaran a salir películas relacionadas con lo que el usuario está escribiendo y se irá modificando hasta que el usuario pare de escribir. Al final, el usuario podrá observar la lista de las películas que coincidan con lo que ha buscado y si lo desea podrá seleccionar el botón con forma de x que hay ahora en el buscador, para limpiar dicho buscador, por si quiere realizar otra búsqueda nuevamente.

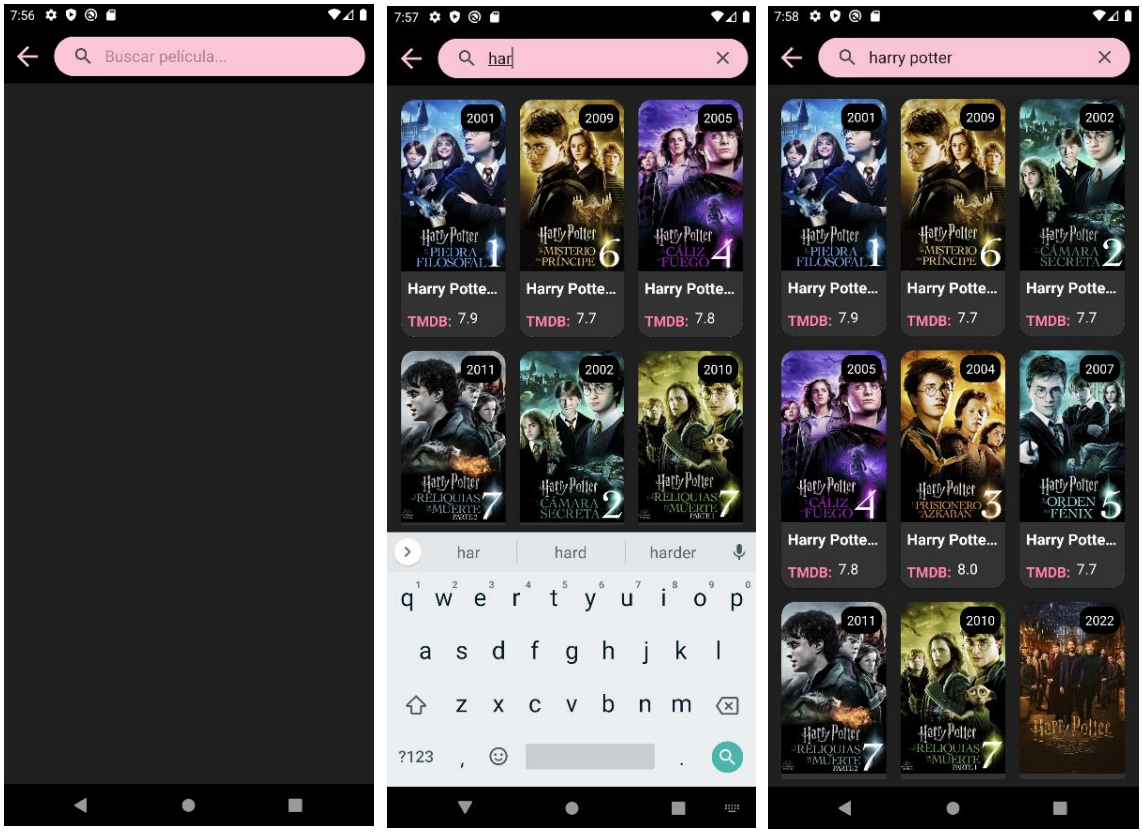

Ilustración 42: Vista de búsqueda de películas

5 Pruebas

En este capítulo se realizarán una serie de pruebas de las distintas funcionalidades, para comprobar el correcto funcionamiento de la aplicación. En dichas pruebas se tendrán en cuenta todos los casos posibles, a la hora de que un usuario realice cualquiera de las acciones del sistema.

También se tendrá en cuenta cuando un usuario esté registrado y cuando no, ya que como he explicado antes, un usuario no registrado se debe registrar para hacer algunas acciones y veremos como el sistema se lo indica.

5.1 Pruebas de Inicio de sesión

A continuación, realizaré varias pruebas relacionadas con el inicio de sesión, unas en las que el usuario no podrá iniciar sesión debido al incumplimiento de las validaciones que ahora explicaré y otra en la que el usuario podrá iniciar sesión:

- **Inicio de sesión incorrecto**

Cada campo del inicio de sesión tiene varias validaciones, por un lado, el campo del email no puede quedarse vacío y debe cumplir con el formato correcto de un correo, si alguna de estas dos validaciones no se cumple se mostrarán sus respectivos mensajes de error como vemos a continuación:

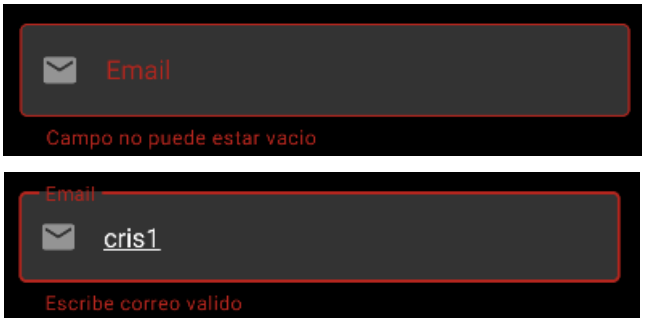

Ilustración 43: Prueba Inicio de sesión incorrecto campo email

Por otro lado, para el campo de la contraseña también hay varias validaciones, la cuales son que el campo no puede quedarse vacio y que la contraseña debe ser de mínimo 6 caracteres:

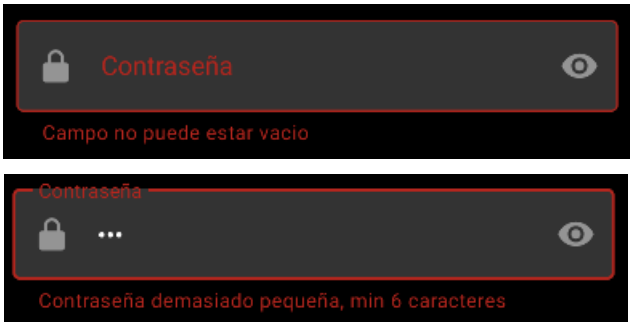

Ilustración 44: Prueba Inicio de sesión incorrecto campo contraseña

Por último, también puede ocurrir que el usuario cumpla con las validaciones que acabo de explicar, pero que se equivoque o escriba mal cualquiera de los dos campos, el del email o el de la contraseña, o también puede ocurrir que esa cuenta no exista, estos dos casos se mostraran con el siguiente mensaje:

Ilustración 45: Prueba Inicio de sesión incorrecto

En el primer caso, de la ilustración 45, sale el error porque la cuenta "cris3@gmail.com" no existe y el segundo caso da error porque se ha escrito mal el email, pasaría lo mismo si se escribe mal la contraseña y para ello el usuario puede presionar el botón, con forma de ojo, en el campo de la contraseña para comprobar lo que ha escrito y escribirlo bien si fuese necesario.

- **Inicio de sesión correcto**

El caso contrario a lo explicado justo antes, sería que el usuario cumpla con todas las validaciones y por lo tanto pueda iniciar sesión y pasar a la pantalla de HomeActivity y ver las películas por categorías.

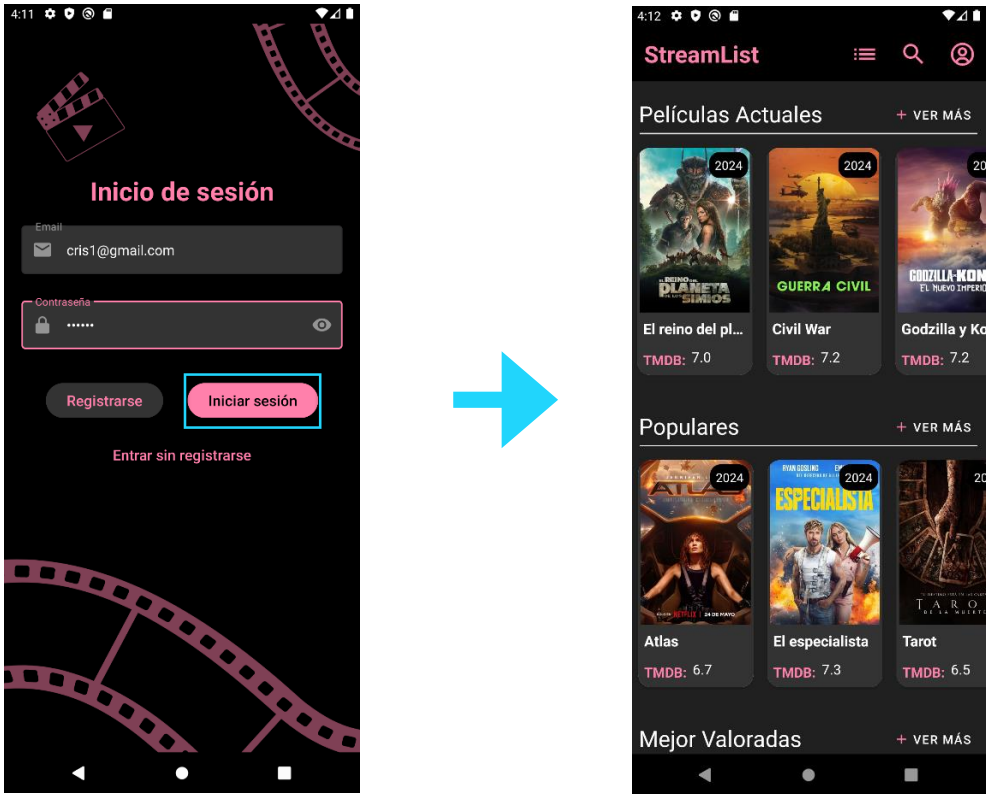

Ilustración 46: Prueba Inicio de sesión correcto

5.2 Pruebas de Registro

En este caso realizaré pruebas relacionas con el registro de un nuevo usuario, en unas el usuario no podrá registrarse ya que incumplirá las validaciones que ahora explicaré y en la otra el usuario se podrá registrar correctamente:

- **Registro incorrecto**

En este caso como en el anterior, cada campo tiene sus propias validaciones, por lo que los veremos de uno en uno, primero el campo del email no puede quedarse vacío y debe cumplir con el formato de un correo electrónico, como vemos a continuación:

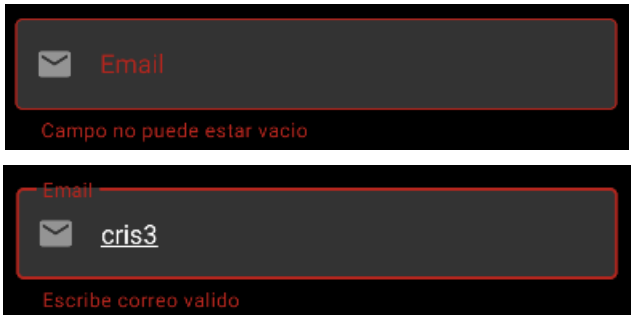

Ilustración 47: Prueba de Registro incorrecto campo email

Luego el campo de la contraseña y el campo de repetir contraseña tienen las mismas validaciones cada uno, así que los explicaré juntos, deben cumplir que el campo no se quede vacio, que la contraseña sea de mínimo 6 caracteres y que en ambos campos debe estar escrita la misma contraseña, lo vemos a continuación:

Ilustración 48: Prueba de Registro incorrecto campo contraseña

Finalmente, se puede dar el caso de que se cumplan las validaciones anteriores, pero que la cuenta que se está intentando registrar ya exista, por lo que saldría el siguiente mensaje de error:

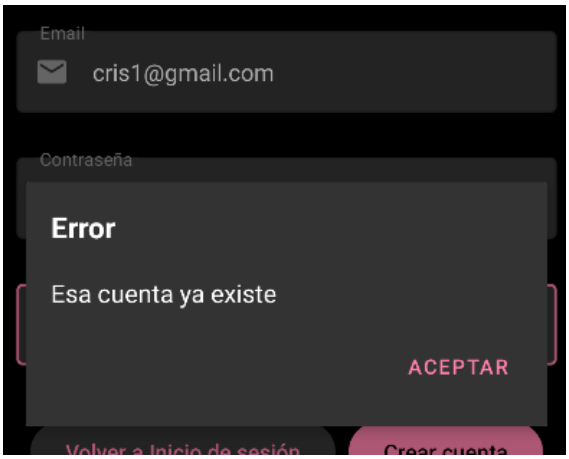

Ilustración 49: Prueba de Registro incorrecto

- **Registro correcto**

Por otro lado, si el usuario se quiere registrar y cumple con las validaciones anteriormente explicadas, podrá registrarse y acceder a la pantalla inicial HomeActivity para ver las películas en categorías:

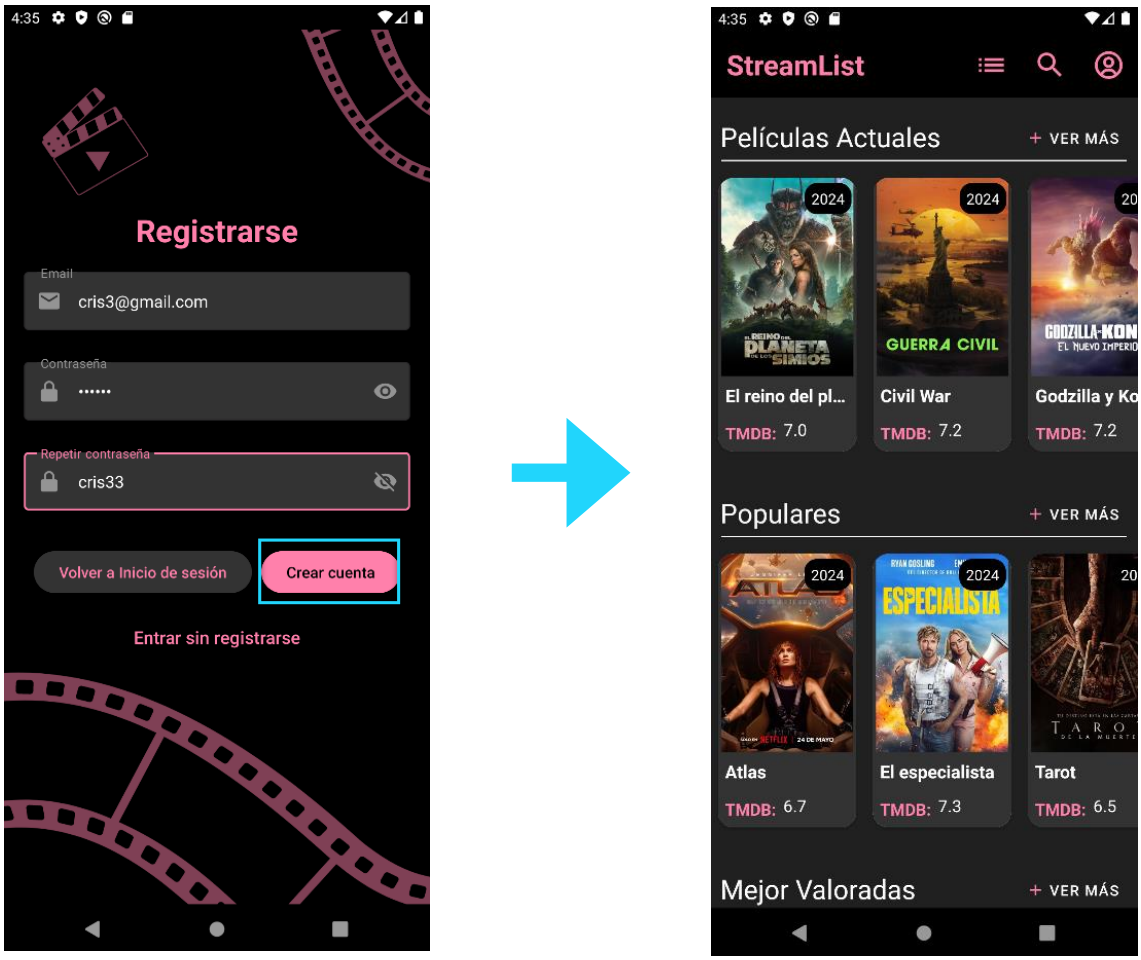

Ilustración 50: Prueba de Registro correcto

5.3 Prueba de Entrar sin registrarse

Esta prueba consistirá en que el usuario accederá a la aplicación sin registrarse, presionando el botón que pone “Entrar sin registrarse”, tanto desde la pantalla de inicio de sesión como desde la de registro y se verá que puede acceder directamente a la vista inicial HomeActivity:

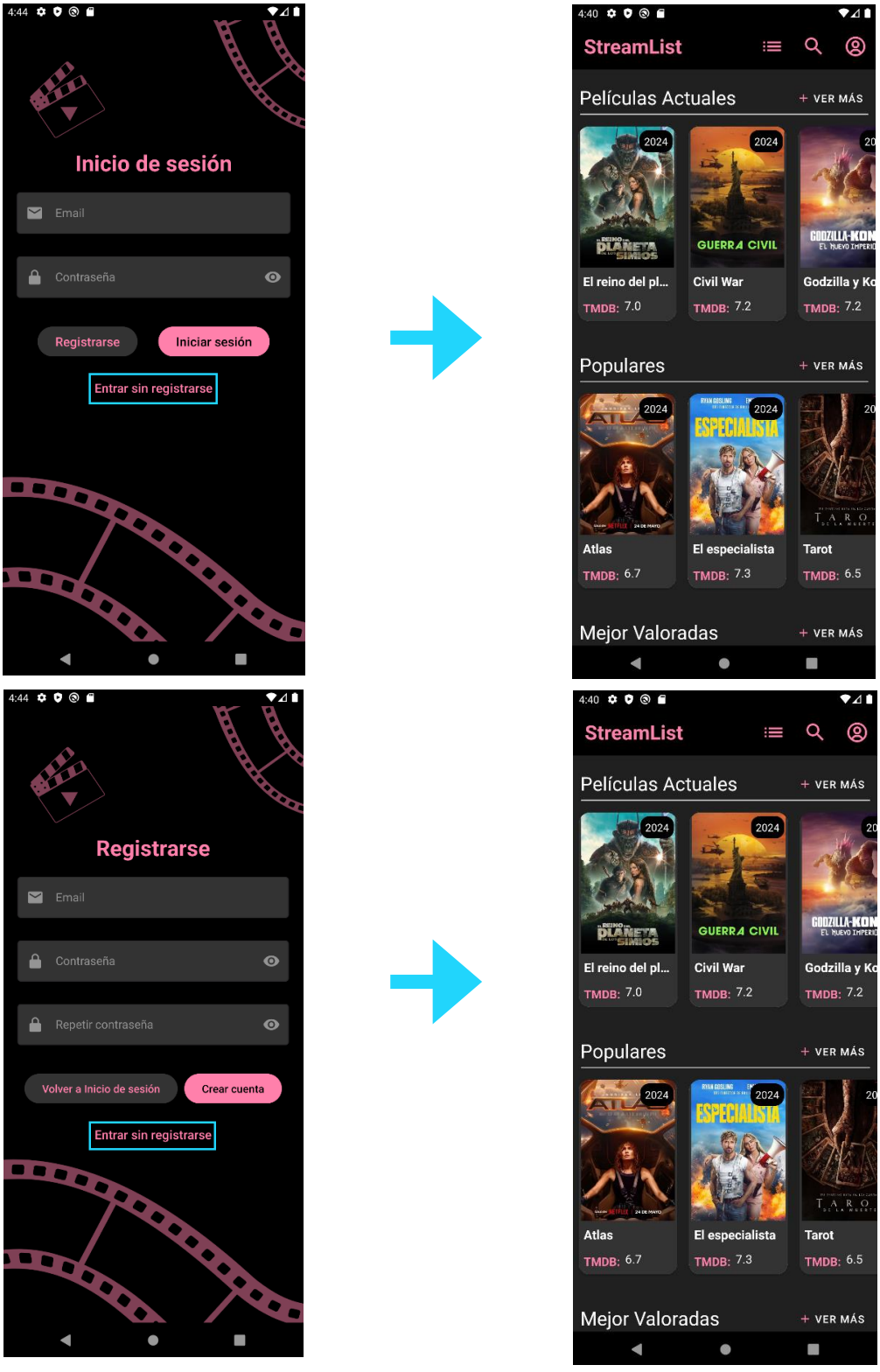

Ilustración 51: Prueba de Entrar sin registrarse

5.4 Prueba de visualizar películas por categorías

En esta prueba comprobaremos que se ven correctamente las películas en la vista del inicio, justo después de que el usuario acceda a la aplicación como él desee:

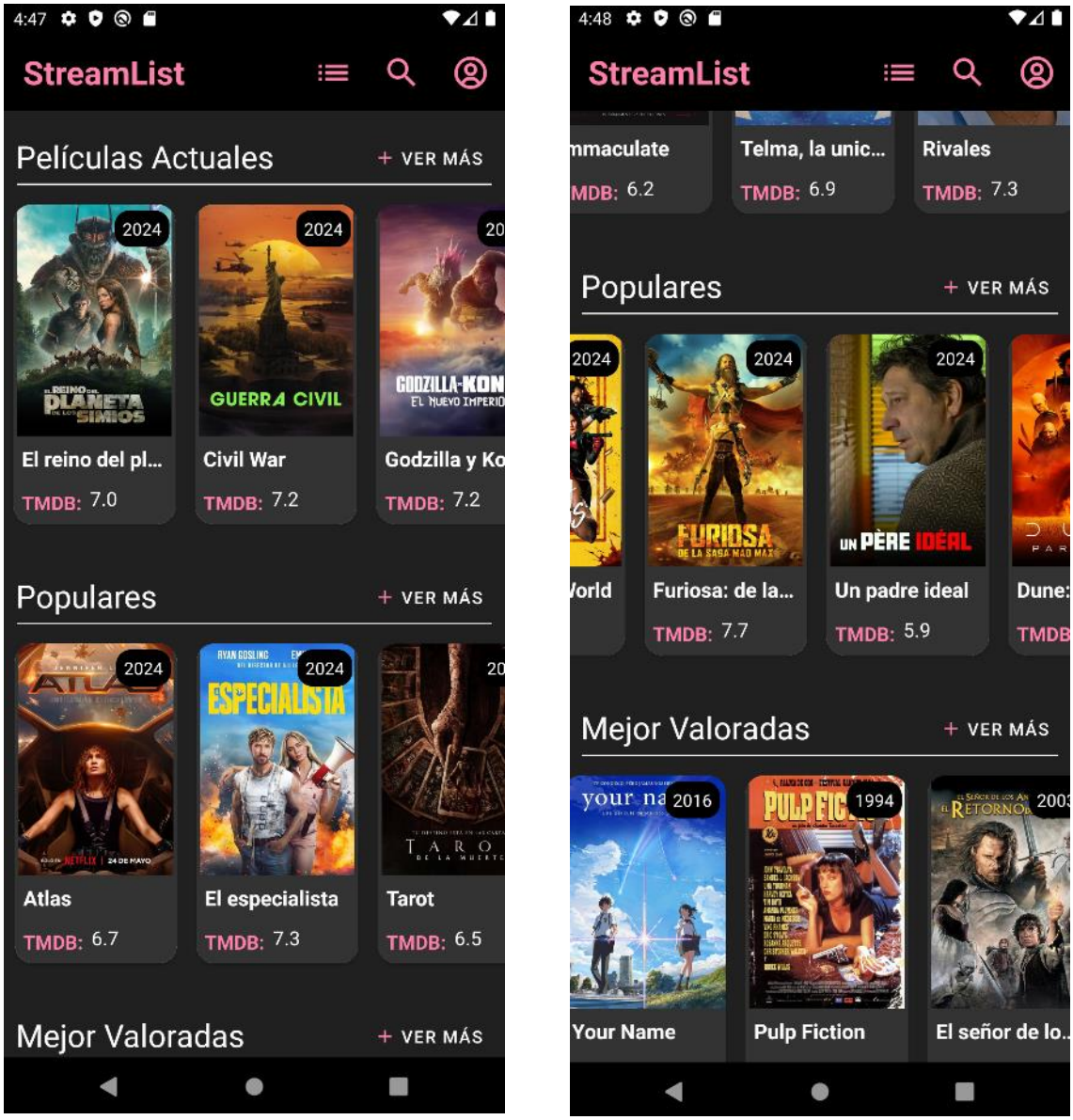

Ilustración 52: Prueba visualizar películas por categorías

5.5 Pruebas de todas las películas de una categoría

En este caso realizaremos varias pruebas relacionadas con la pantalla de todas las películas de una categoría:

- **Visualizarlas**

Primero comprobaremos que se pueden visualizar correctamente las películas al acceder a esta vista, tras presionar el botón de “Ver más” de una categoría:

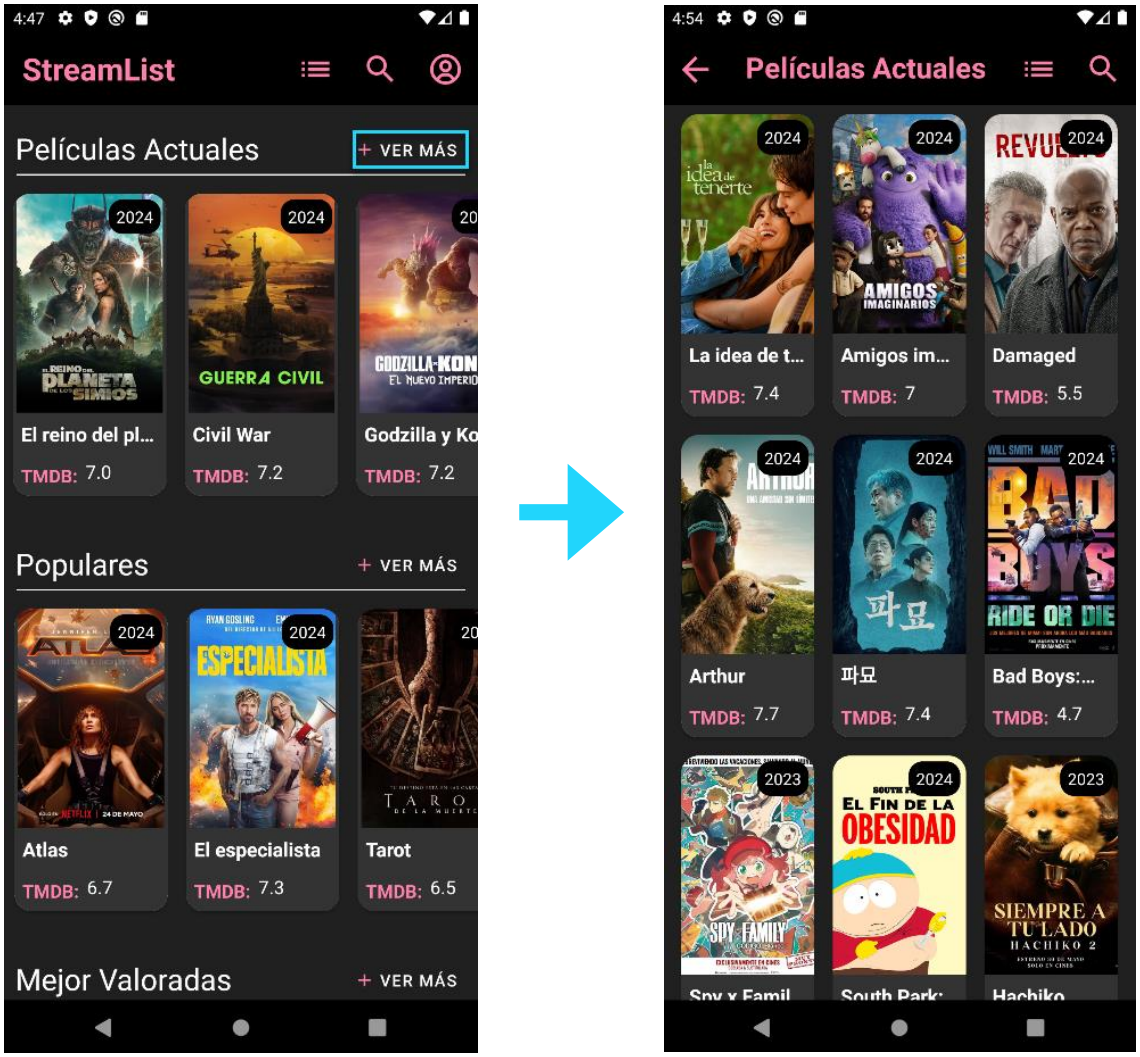

Ilustración 53: Prueba de todas las películas de una categoría (visualizarlas)

Cabe destacar que las películas que se ven en la pantalla de HomeActivity son las películas de la primera página de dicha categoría y al presionar en el botón de “Ver más” de esa categoría accedes a la pantalla de FilmsActivity en la segunda página de la categoría, por eso las películas que se muestran son diferentes, pero el usuario puede volver a la página anterior gracias a la navegación de la cual realizaremos la prueba a continuación.

- **Navegar entre páginas**

Como he explicado en el apartado anterior, en este caso realizaremos la prueba de navegar por las diferentes páginas de una categoría para así poder visualizar todas las películas de dicha categoría:

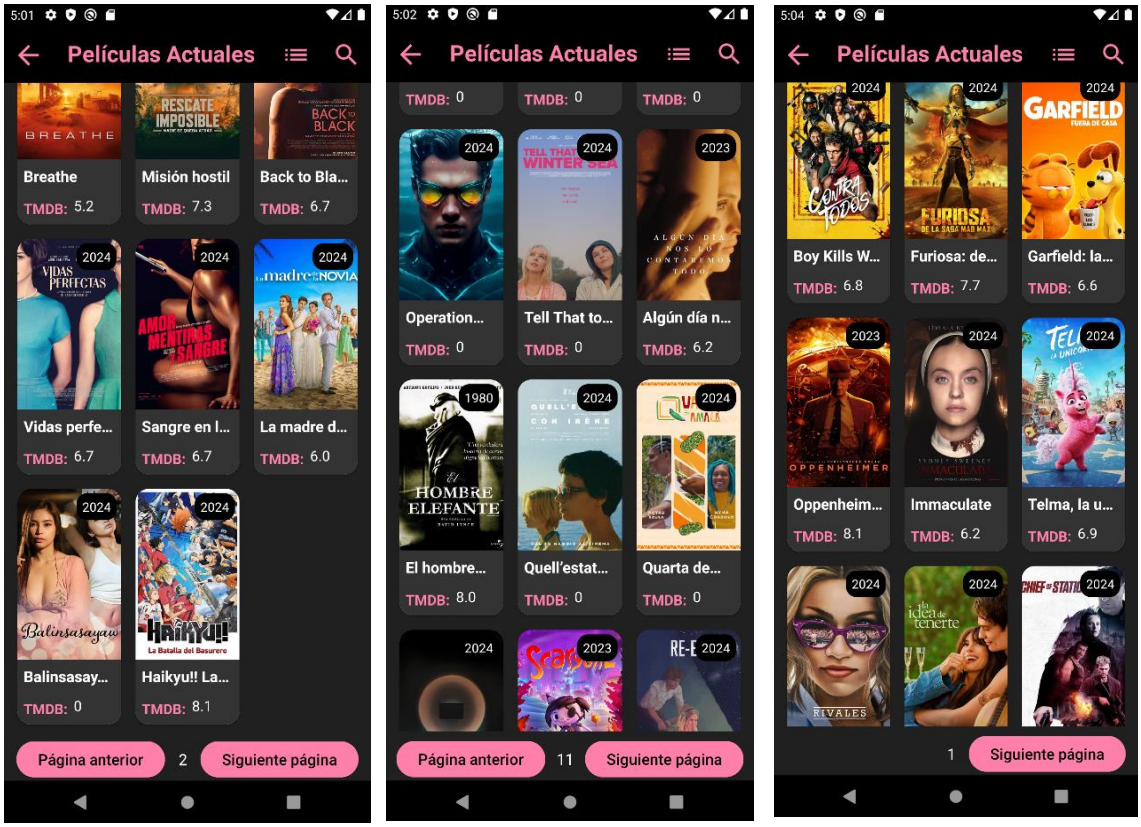

Ilustración 54: Prueba de todas las películas de una categoría (navegar)

Destacar que al llegar a la primera y a la última página el botón de “Página anterior” y “Siguiente página” desaparecerá respectivamente.

5.6 Prueba visualizar información de una película

En esta prueba se comprobará que se visualiza correctamente toda la información disponible de una película cuando el usuario quiera ver dicha información:

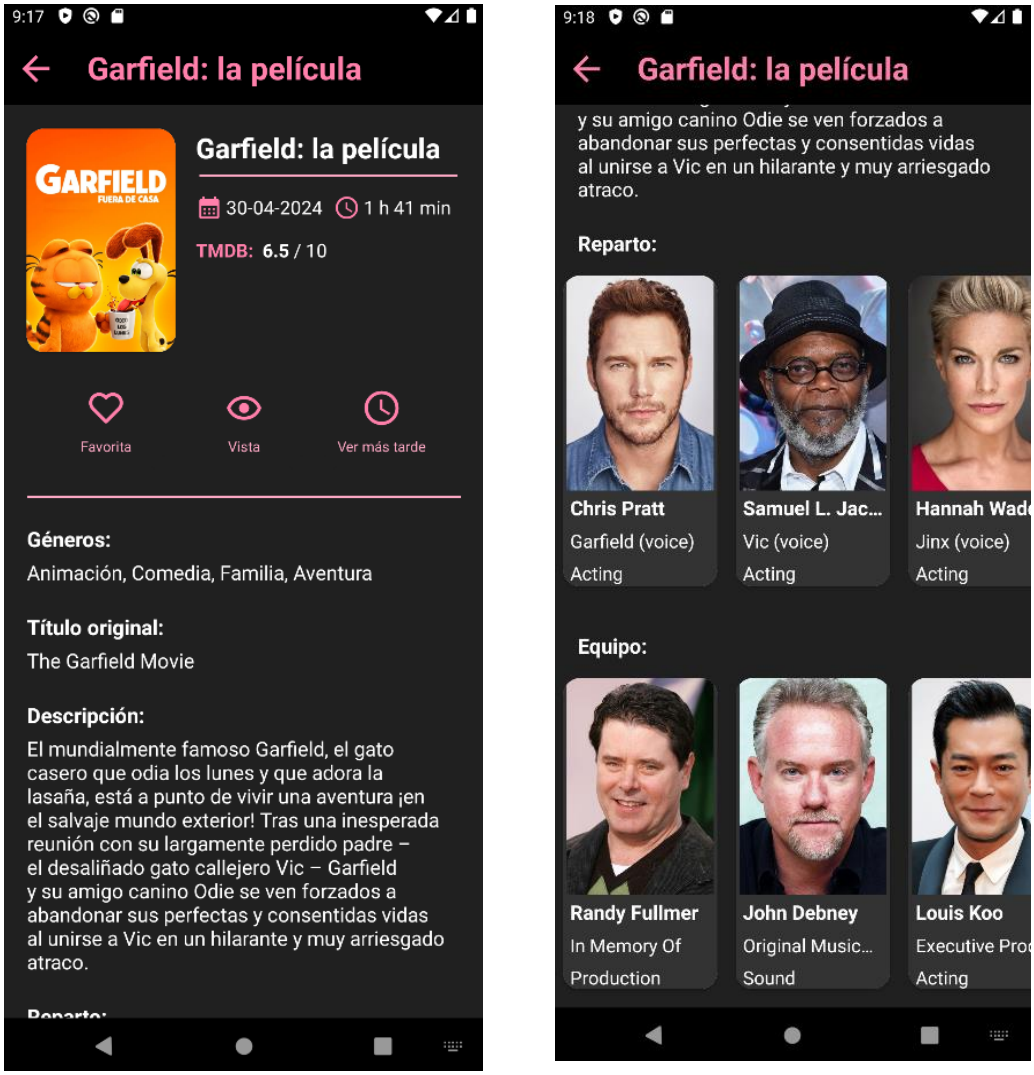

Ilustración 55: Prueba visualizar información de una película

5.7 Prueba de búsqueda de una película

En esta prueba se comprobará que se realiza correctamente la búsqueda deseada y que se podrán visualizar las películas buscadas correctamente:

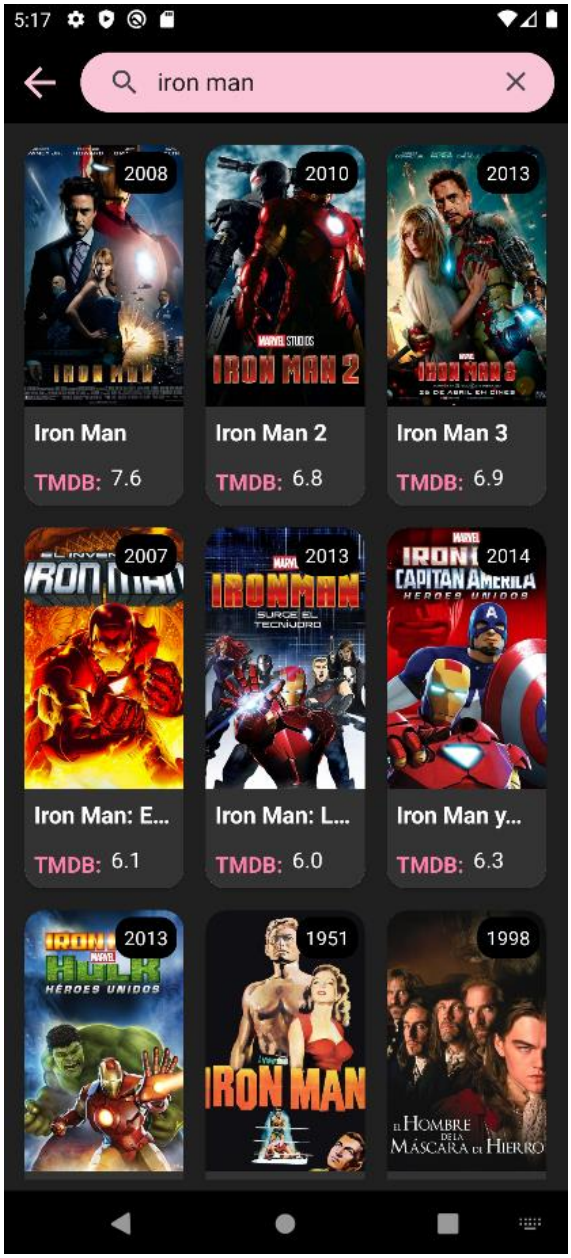

Ilustración 56: Prueba de búsqueda de una película

5.8 Pruebas de acceder a las listas

Ahora se comprobará dos casos diferentes a la hora de acceder a la vista de las listas, por un lado, se intentará realizar dicha acción con un usuario no registrado, es decir, que ha entrado a la aplicación de forma anónima y por otro lado se hará con un usuario registrado:

- **Usuario no registrado**

En este primer caso se intentará acceder a las listas con un usuario no registrado y se verá como se muestra un mensaje de error al hacerlo desde las dos vistas desde donde puedes acceder:

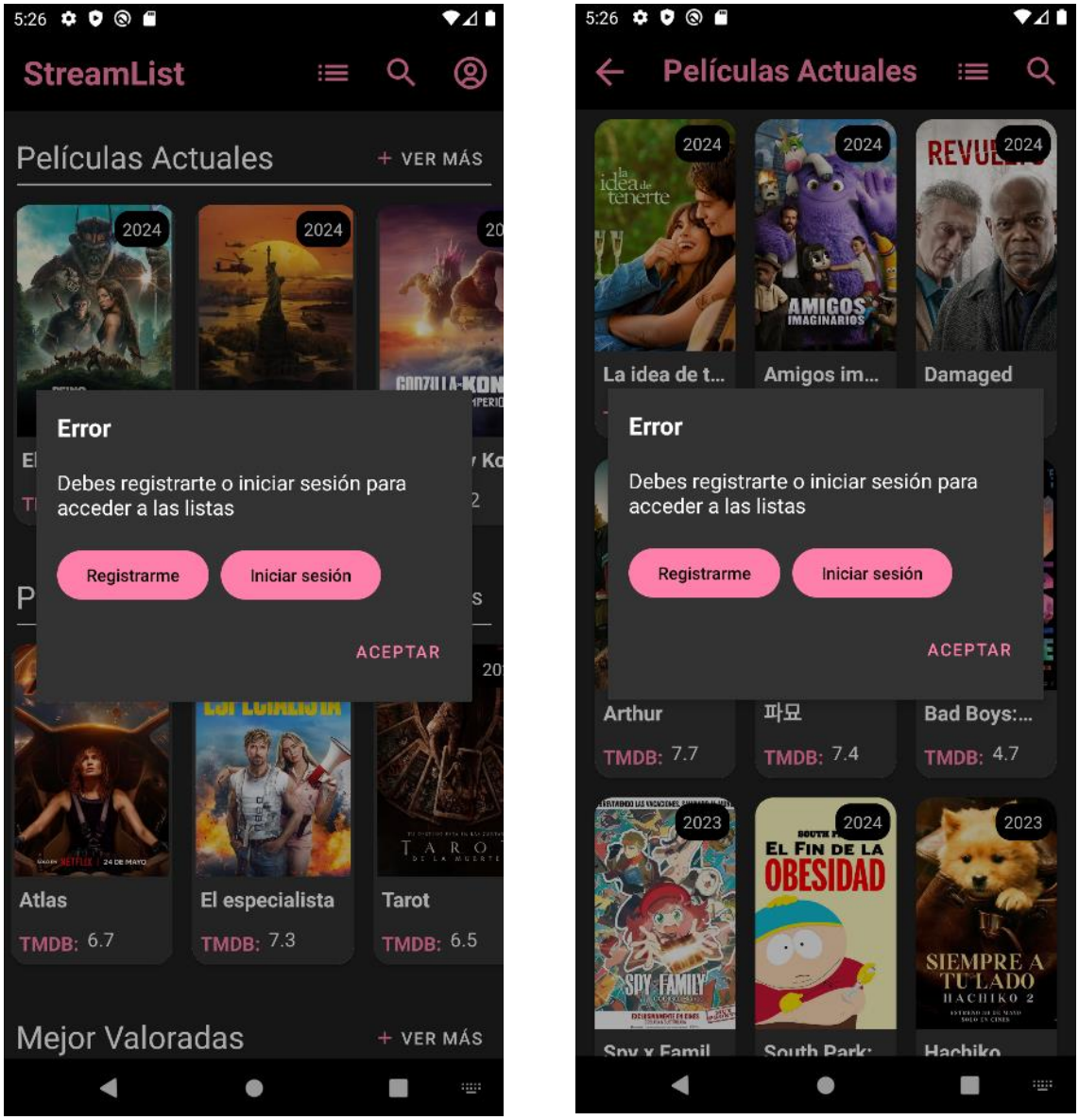

Ilustración 57: Prueba acceder a listas usuario no registrado

Como se puede ver en el mensaje, el usuario podrá ir a la pantalla de iniciar sesión o a la de registro si así lo desea o por el contrario puede dar al botón de aceptar para que se cierre el mensaje.

- **Usuario registrado**

En caso contrario, si esta acción la realiza un usuario que si está registrado podrá acceder correctamente a sus listas de Favoritas, Vistas y Ver más tarde:

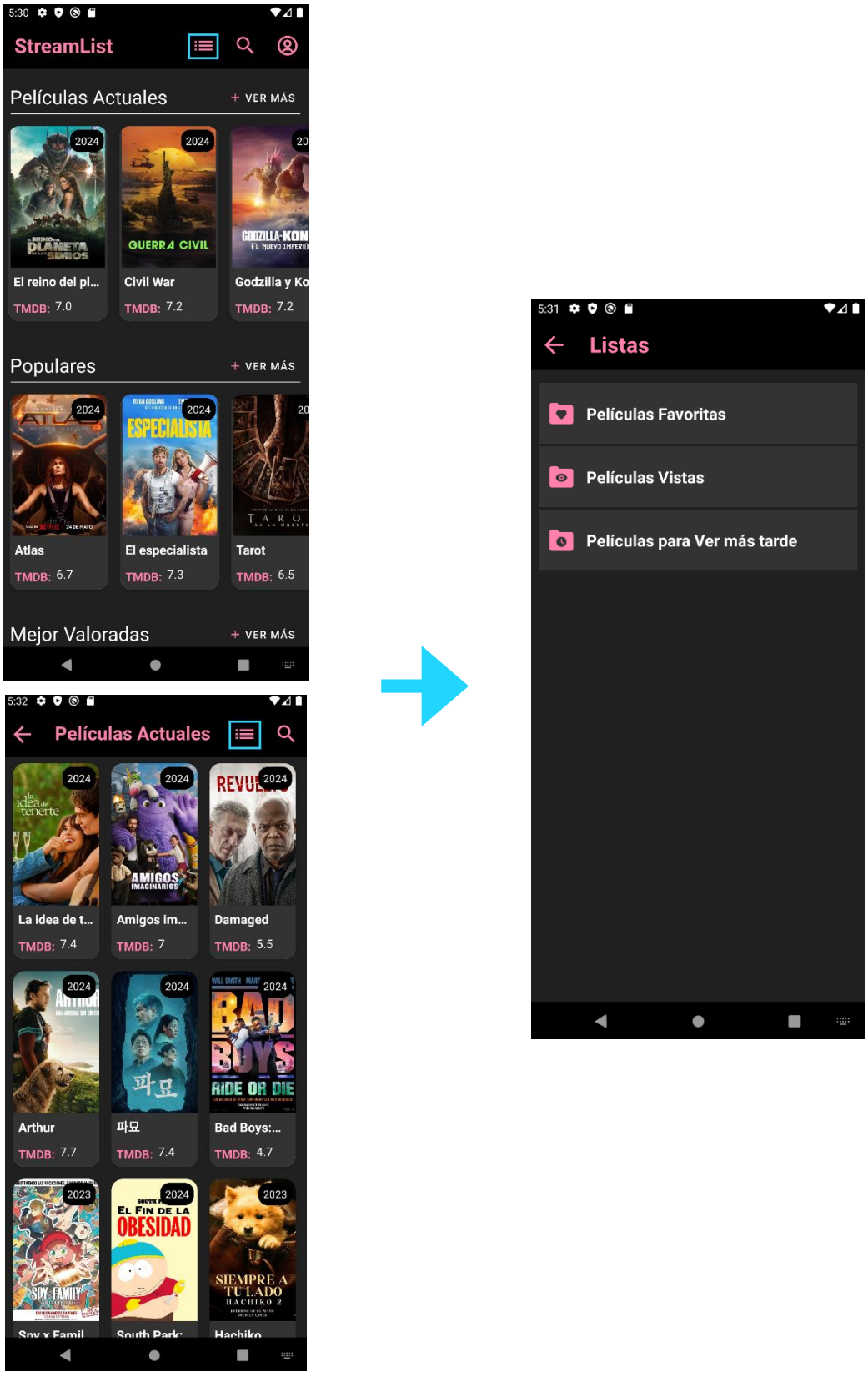

Ilustración 58: Prueba acceder a listas usuario registrado

5.9 Pruebas de guardar película en Favoritas

A continuación, realizaré pruebas a la hora de guardar una película en la lista de Favoritas, donde tenemos dos casos esta acción la quiere realizar un usuario no registrado y por otro lado la quiere realizar un usuario registrado:

- **Usuario no registrado**

Si esta acción la realiza un usuario no registrado, saltará un mensaje de error diciendo que se debe registrar o iniciar sesión para poder guardar la película en Favoritas:

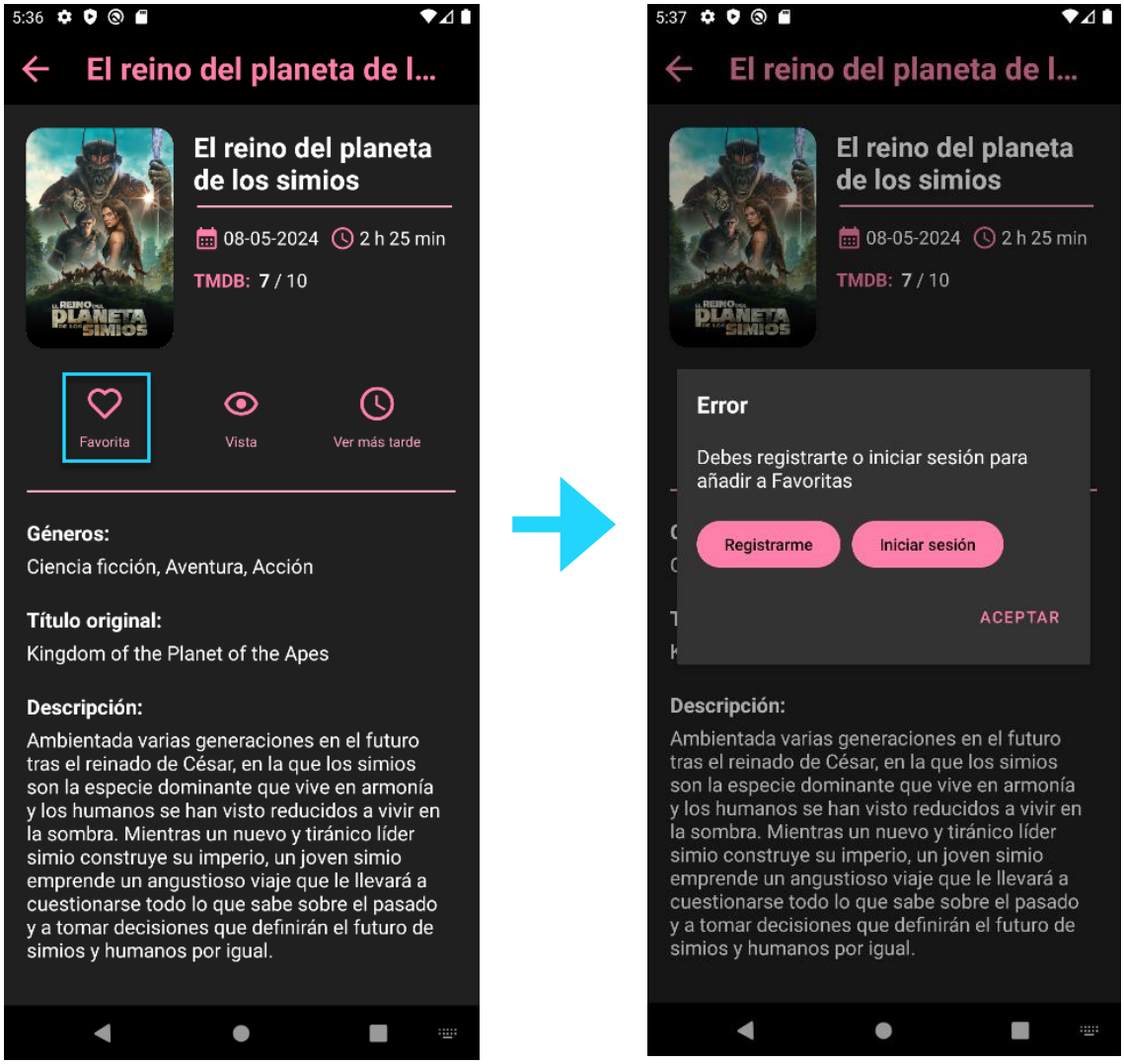

Ilustración 59: Prueba guardar película Favoritas usuario no registrado

En dicho mensaje el usuario podrá ir a las pantallas de inicio de sesión o de registro para realizar la acción correspondiente si lo desea o darle a aceptar para cerrar el mensaje.

- **Usuario registrado**

Si esta acción la realiza un usuario registrado, la película se guardará en la lista de Favoritas y se seleccionará el botón de Favorita. También el usuario podrá ir a ver dicha película guardada en la lista correspondiente.

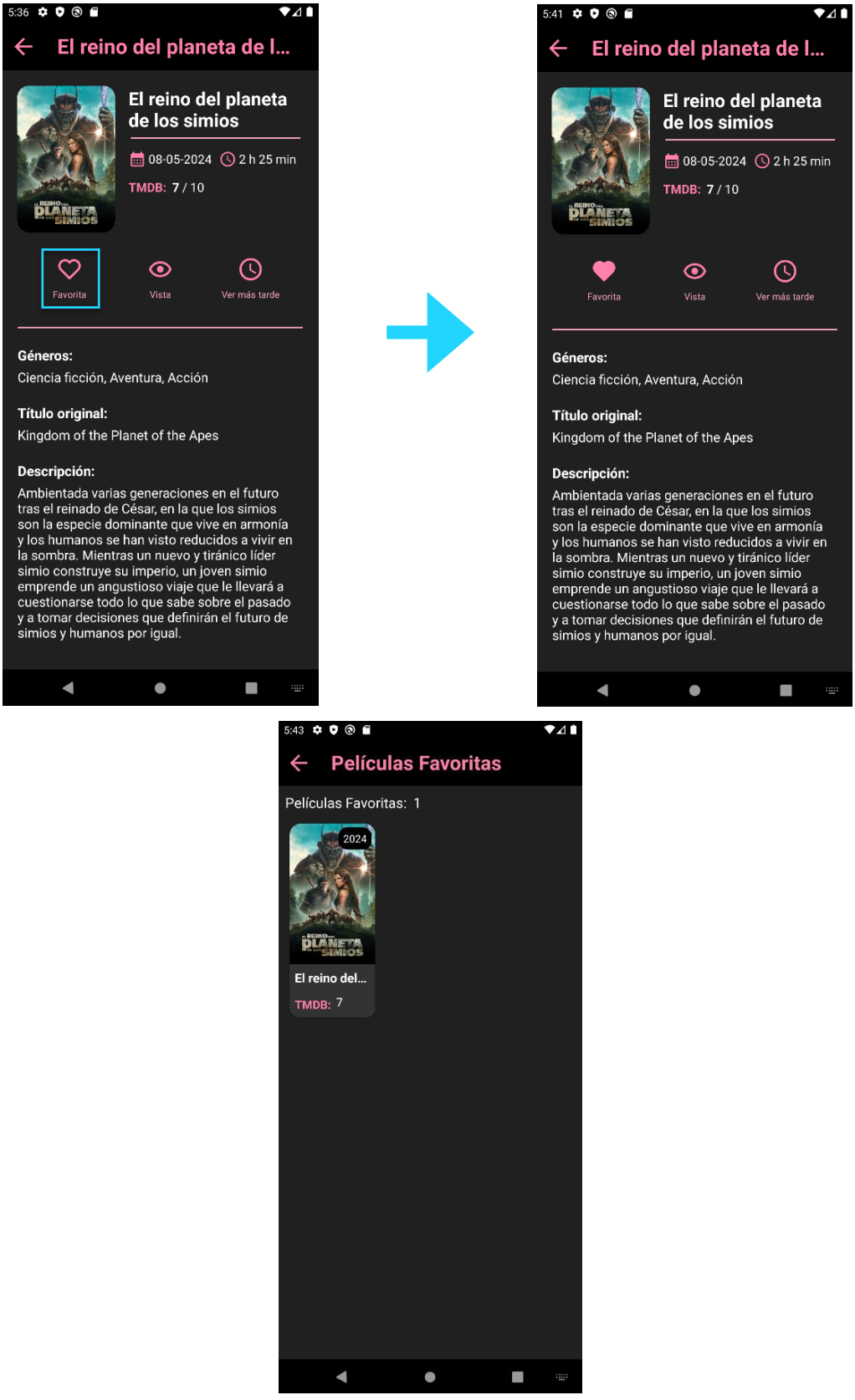

Ilustración 60: Prueba guardar película Favoritas usuario registrado

5.10 Prueba de eliminar película de Favoritas

Esta prueba solo se puede hacer con un usuario registrado y una vez se haya guardado una película, para así poder eliminarla. Los usuarios no registrados no pueden realizar esta acción.

El usuario registrado dará al botón que estará seleccionado para eliminar la película, se deseleccionará el botón y la película desaparecerá de la lista correspondiente.

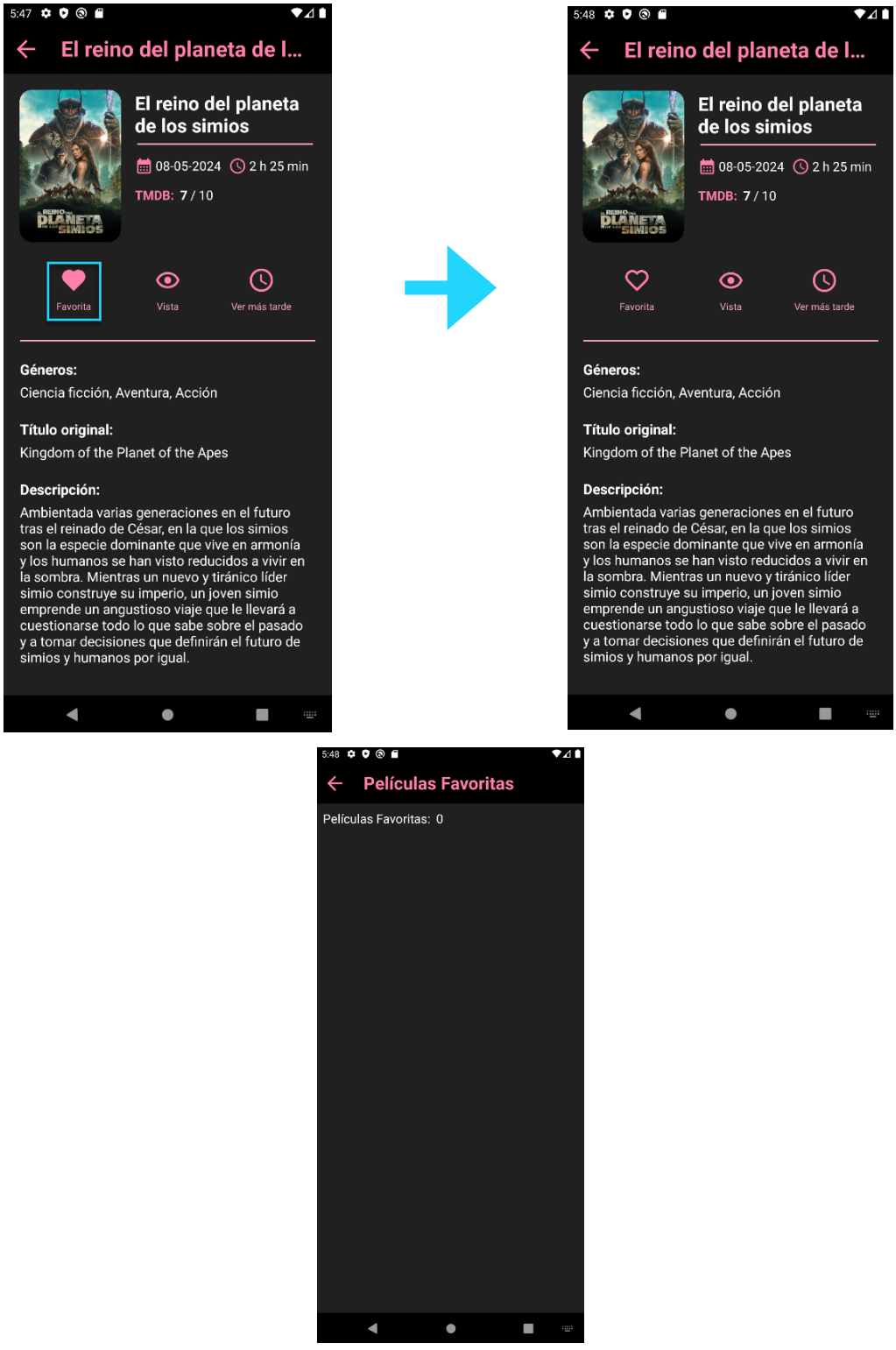

Ilustración 61: Prueba eliminar película Favoritas

5.11 Pruebas de guardar película en Vistas

Esta prueba es exactamente igual que la de guardar película Favoritas, pero para la lista de Vistas, la veremos con un usuario no registrado y con uno registrado:

- **Usuario no registrado**

Saldrá un mensaje de error al intentar guardar en Vistas:

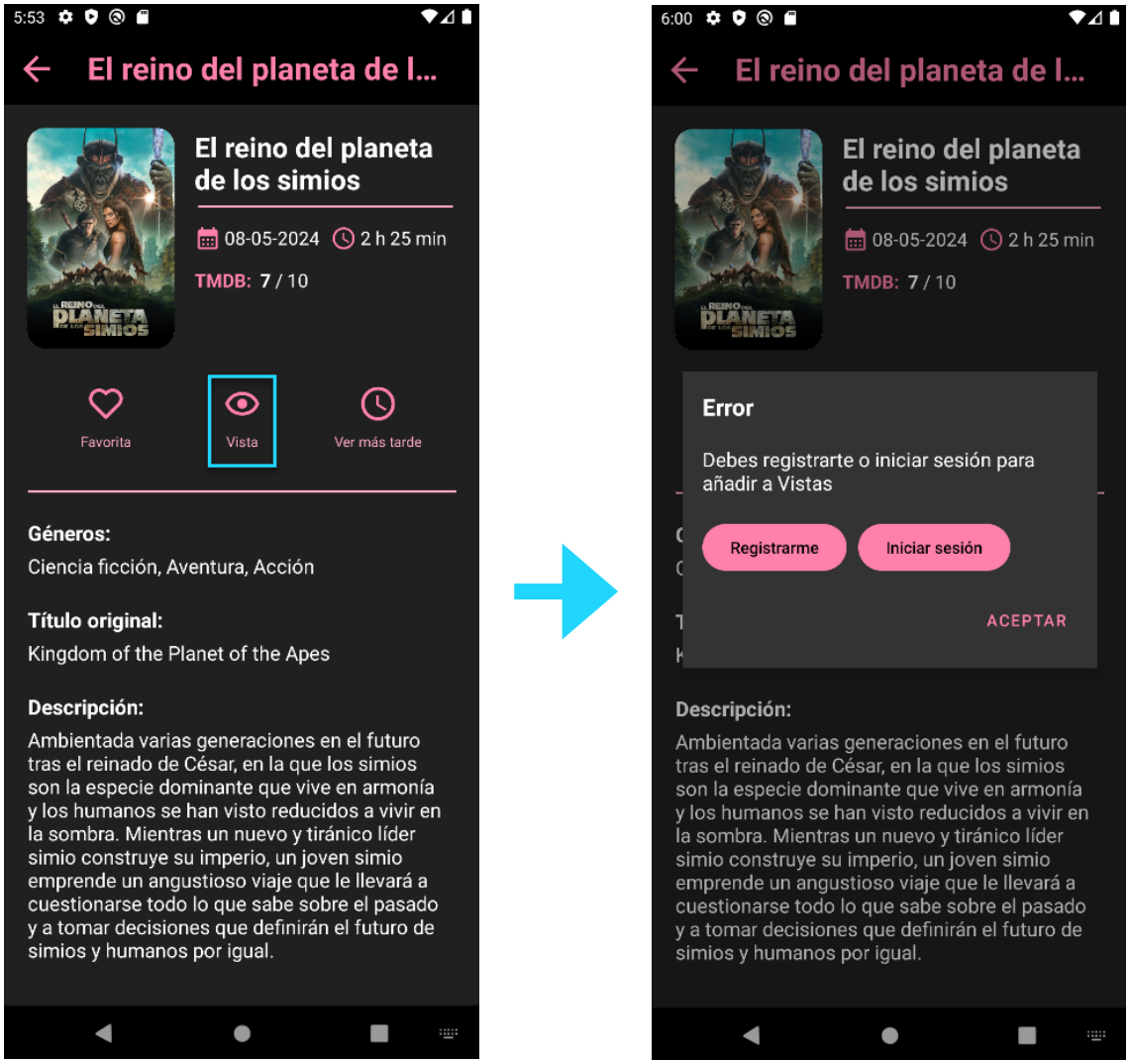

Ilustración 62: Prueba guardar película Vistas usuario no registrado

- **Usuario registrado**

La película se guarda en Vistas, se selecciona el botón de Vista y se podrá ver la película guardada:

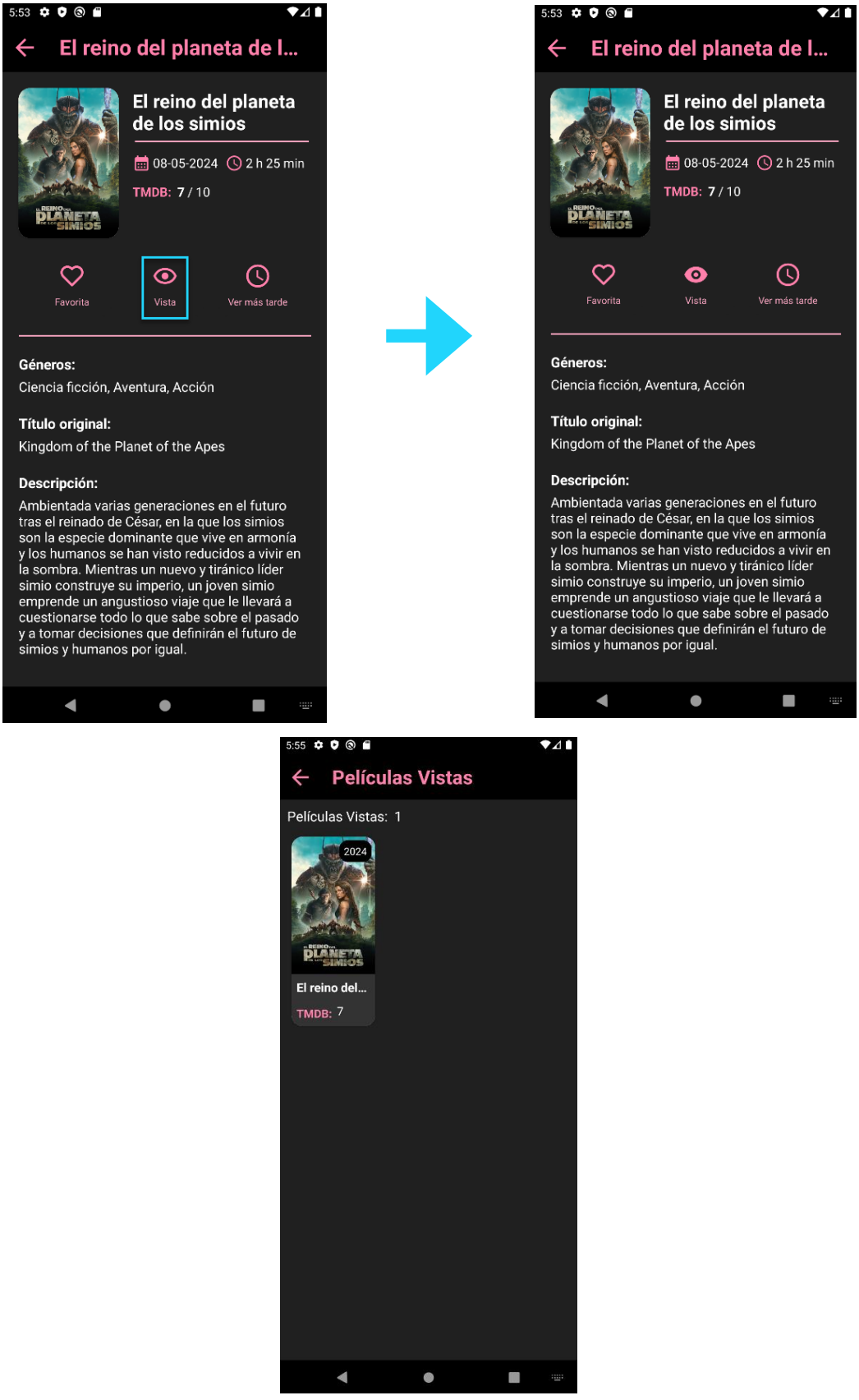

Ilustración 63: Prueba guardar película Vistas usuario registrado

5.12 Prueba de eliminar película de Vistas

Solo un usuario registrado puede hacer esta acción, se eliminará la película de Vistas, se deseleccionará el botón Vista y desaparecerá de la lista.

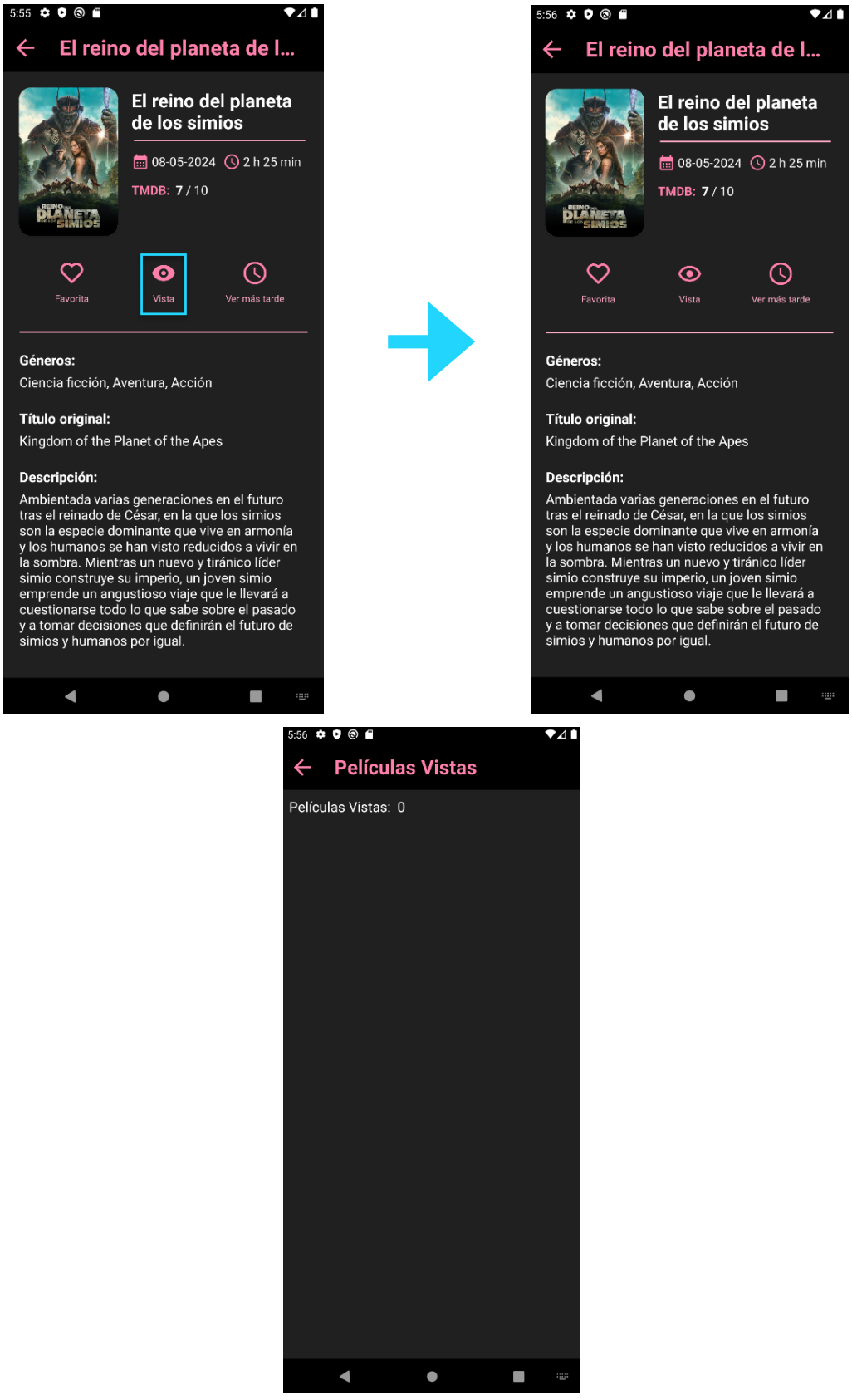

Ilustración 64: Prueba eliminar película Vistas

5.13 Pruebas de guardar película en Ver más tarde

Como las dos anteriores, lo probaremos con un usuario no registrado y luego con uno registrado:

- **Usuario no registrado**

Saldrá un mensaje de error:

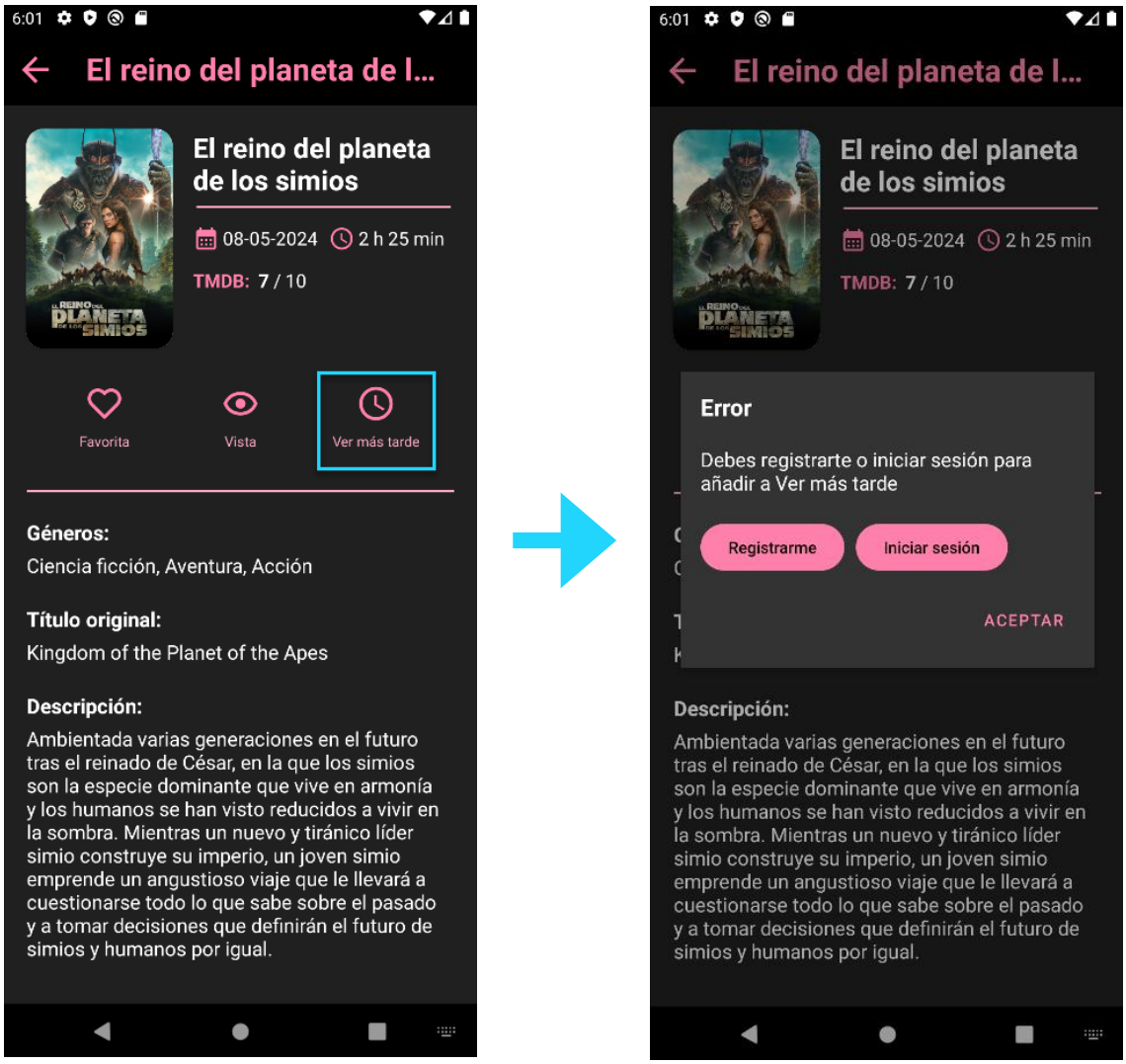

Ilustración 65: Prueba guardar película Ver más tarde usuario no registrado

- **Usuario registrado**

La película se guarda en Ver más tarde, se selecciona el botón de Ver más tarde y se podrá ver en la lista guardada:

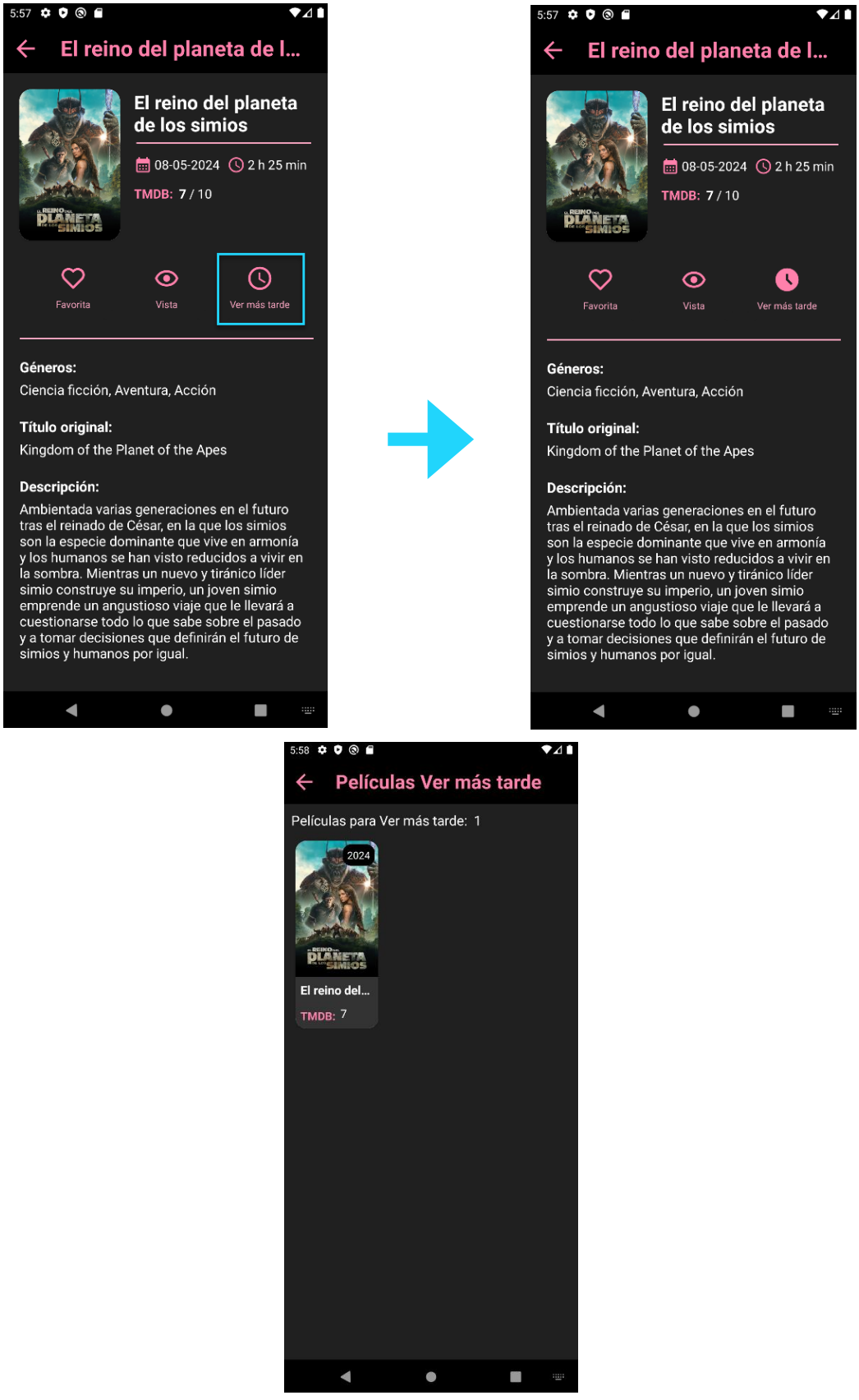

Ilustración 66: Prueba guardar película Ver más tarde usuario registrado

5.14 Prueba de eliminar película de Ver más tarde

Un usuario registrado podrá realizar esta acción, se eliminará la película de Ver más tarde, se deseleccionará el botón Ver más tarde y desaparecerá de la lista.

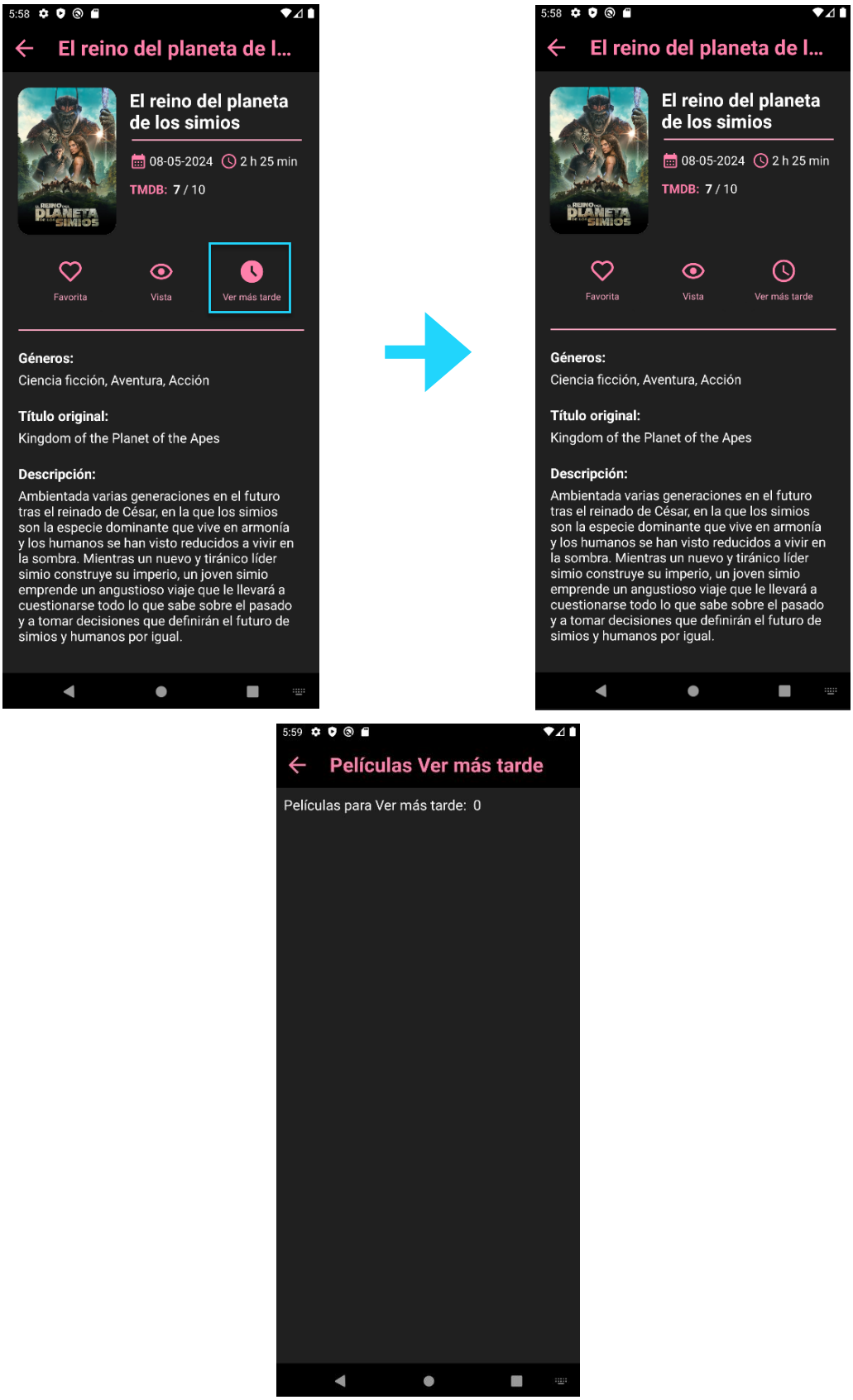

Ilustración 67: Prueba eliminar película Ver más tarde

5.15 Pruebas acceder al perfil

A continuación, se realizarán pruebas de acceder al perfil, primero con un usuario no registrado y luego con uno registrado:

- **Usuario no registrado**

En el perfil saldrá que el usuario está de forma anónima, y podrá iniciar sesión o registrarse.

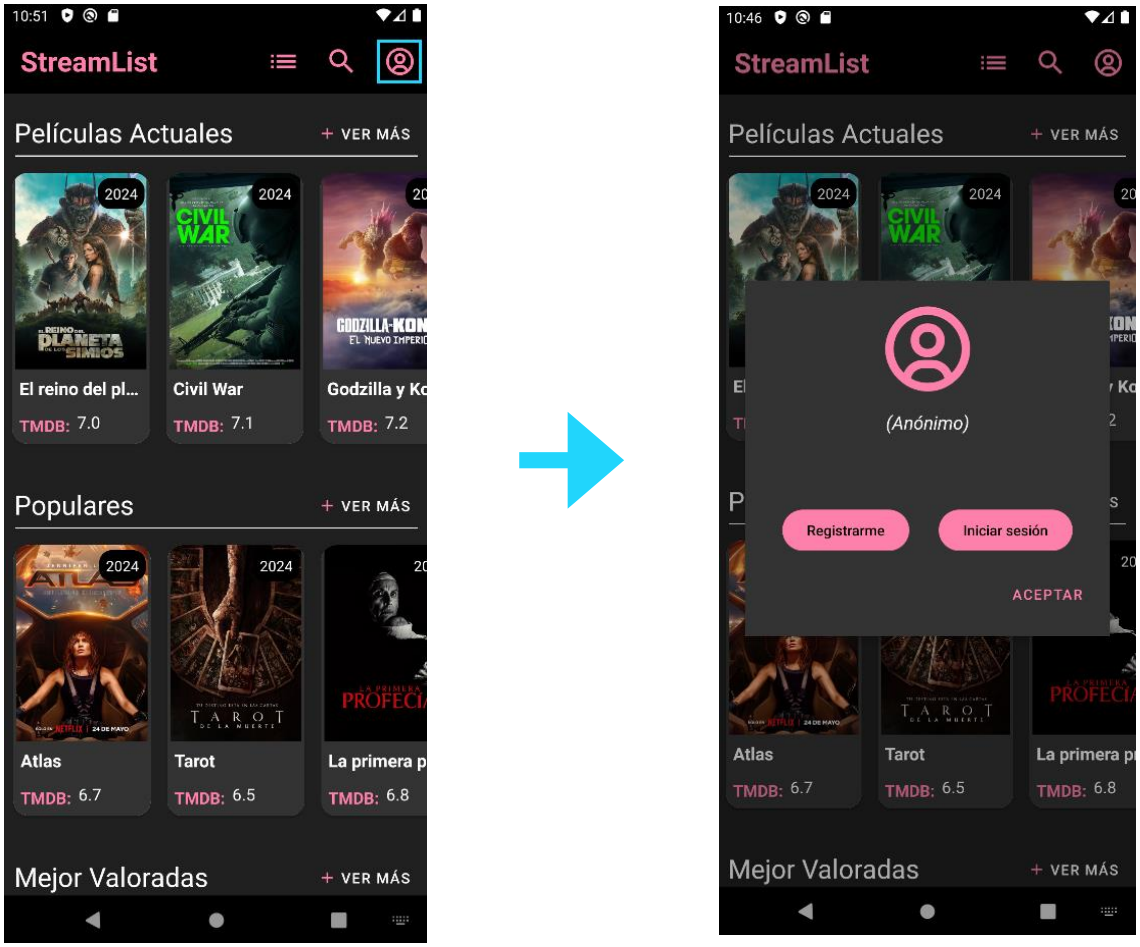

Ilustración 68: Prueba acceder al perfil usuario no registrado

- **Usuario registrado**

En este caso, en el perfil saldrá el correo electrónico con el que el usuario se ha registrado y también podrá cerrar sesión si lo desea.

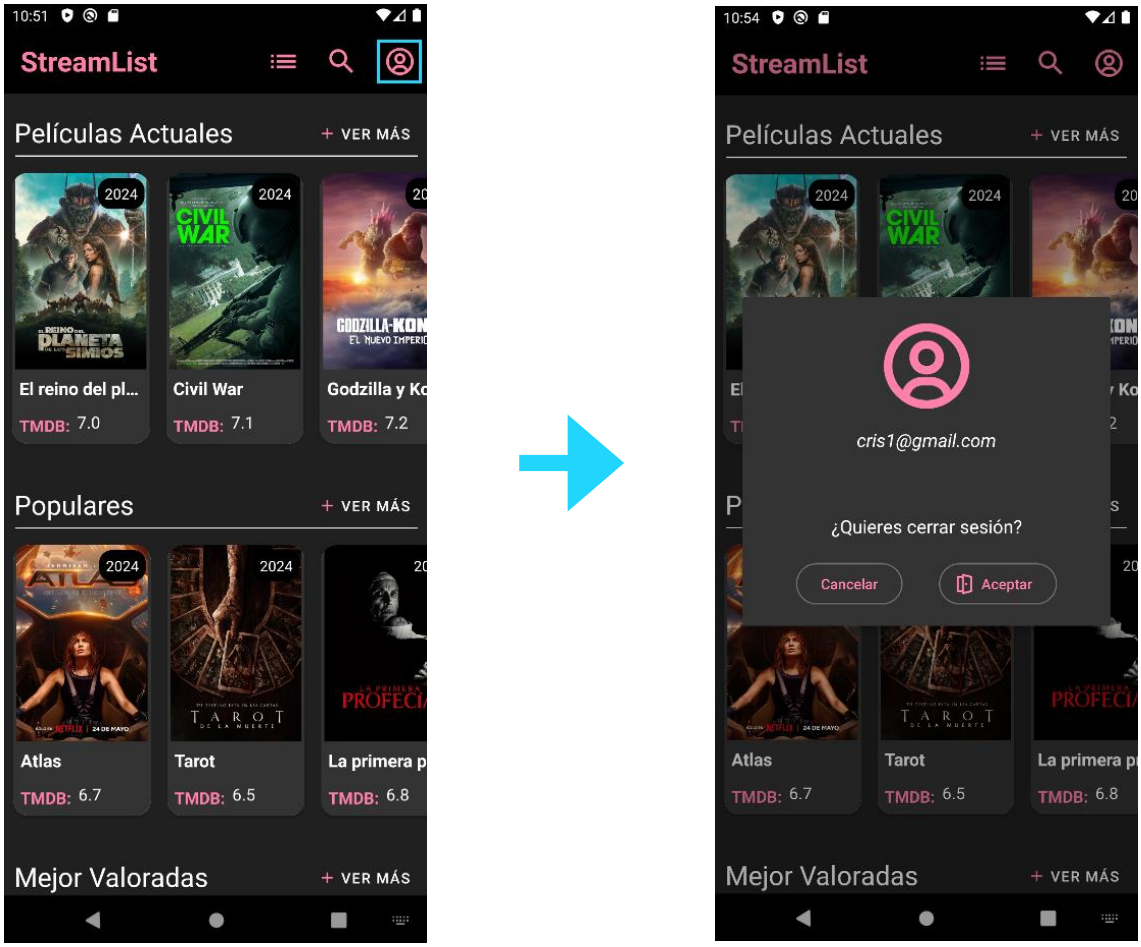

Ilustración 69: Prueba acceder al perfil usuario registrado

5.16 Prueba del cierre de sesión del usuario

Esta acción solo la puede realizar un usuario registrado, desde su perfil el usuario podrá cerrar sesión y se mostrará un mensaje de que ha cerrado la sesión.

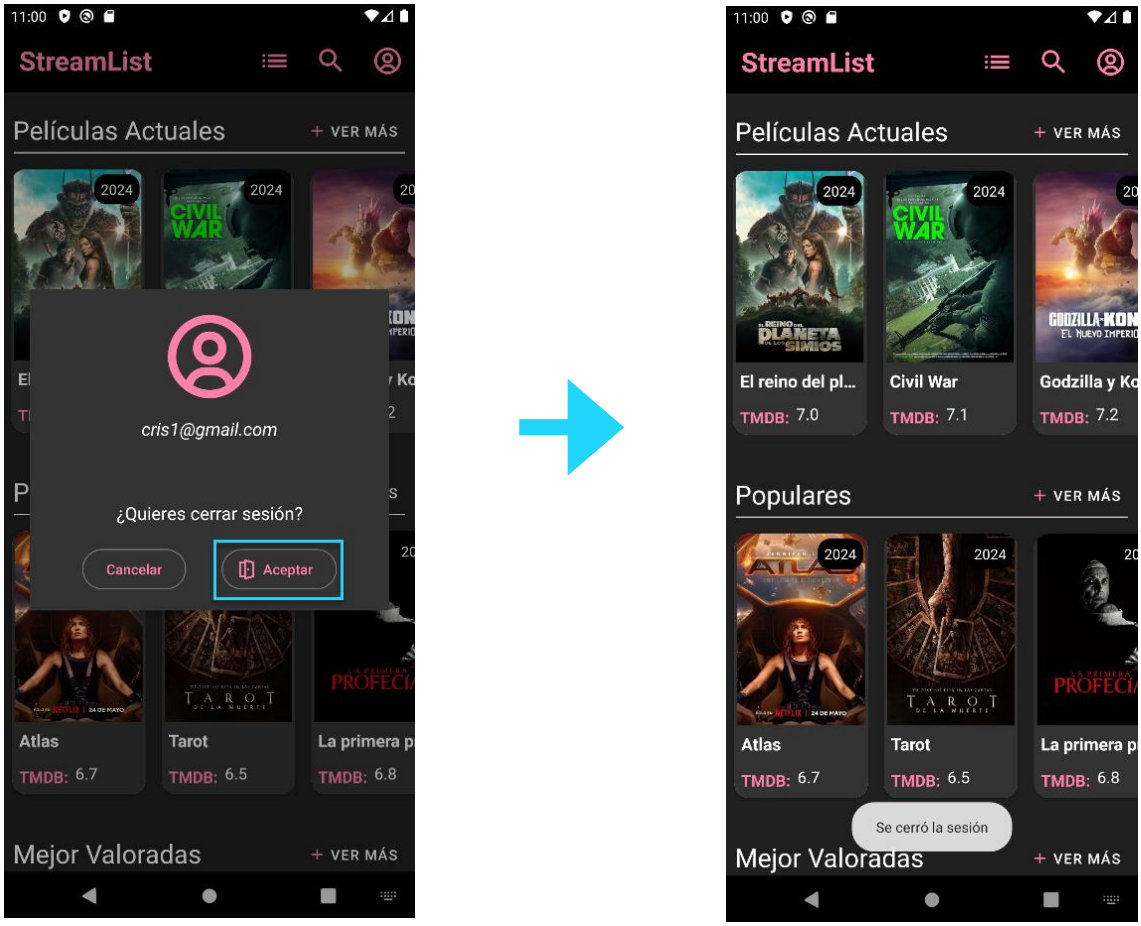

Ilustración 70: Prueba del cierre de sesión del usuario

6 Resultados y conclusiones

Como conclusiones decir que estoy muy satisfecha con el resultado de esta aplicación que he desarrollado, ya que cumplí con todas las tareas y objetivos propuestos desde un principio. Todas las funcionalidades, tales como disponer de un catálogo completo con todas las películas, tener un buscador para poder realizar búsquedas de la película que quieras por su título, poder guardar todas las películas que quieras en las tres listas disponibles, que son tus películas Vistas, tus Favoritas y las que quieres Ver más tarde y poder visualizar toda la información deseada de una película en concreto, fueron cumplidas he implementadas correctamente.

Destacar que me he encontrado con diversas dificultades al principio con respecto al entorno de desarrollo, Android Studio, y con el lenguaje de programación, Kotlin, ya que esta era la primera vez que los utilizaba para desarrollar un proyecto. Debido a esto me llevó mucho tiempo el estudio de estas dos tecnologías, en vez de haber desarrollado otras funcionalidades adicionales que se me fueron ocurriendo mientras desarrollaba la aplicación, las cuales al final no he podido implementar por falta de tiempo. Estas funcionalidades adicionales las explicaré en un apartado a continuación.

También decir que no había utilizado antes Firebase, ni una base de datos NoSQL, pero el estudio de esta herramienta no fue tan complicado ya que me pareció muy sencilla de usar y muy intuitiva. También veo muy útil todos sus servicios, aunque realmente solo utilicé para esta práctica dos, el de autenticación y el de la base de datos, seguramente los demás sean igual de interesantes.

A continuación, explicaré lo que comenté antes sobre las funcionalidades adicionales que quería implementar en mi aplicación, pero no me dio tiempo de hacerlo, en el apartado de líneas futuras.

- **Líneas futuras**

En este apartado explicaré un poco algunas funcionalidades adicionales que quería desarrollar en mi aplicación, pero no pudo ser al final por falta de tiempo.

Primero añadiría más información sobre una película como por ejemplo, pondría las plataformas donde está disponible para ver dicha película, ya sea comprada, alquilada o pagando una tarifa mensual, también añadiría que se pudiera ver el trailer de la película, para que así el usuario pueda verlo cómodamente y decidir si quiere ver la película.

Por otro lado, en la pantalla de registro pediría más datos del usuario para así poder desarrollar una pantalla más completa del perfil del usuario, o simplemente desarrollaría una pantalla de perfil donde el usuario pueda rellenar o modificar los datos que quiera, como por ejemplo su nombre de usuario o que pudiera modificar su foto de perfil. Todos esos datos serían guardados en la base de datos actual, aunque para almacenar la foto de perfil utilizaría otro servicio de Firebase llamado Storage que me habría gustado utilizar y saber cómo funciona, pero como he mencionado por falta de tiempo no ha podido ocurrir.

También me habría gustado hacer una pantalla con toda la información de las personas, es decir, del reparto y del equipo de una película. Y que en las listas los usuarios pudieran crear las suyas propias, para así tener más de tres listas.

Por último, me gustaría realizar todo lo desarrollado con series también.

7 Análisis de Impacto

Para finalizar este proyecto, realizaré un análisis del impacto potencial de los resultados obtenidos durante la realización de este trabajo en diferentes contextos.

En el contexto personal, he adquirido nuevos conocimientos relacionados con el desarrollo de aplicaciones móviles, en concreto para Android. Como he mencionado antes no había usado antes ni Android Studio, ni Kotlin, por lo que gracias a esta práctica he podido aprender a utilizar un nuevo entorno de desarrollo y un nuevo lenguaje de programación. También, como ya mencioné antes, he aprendido a usar una base de datos NoSQL la cual funciona de distinta manera a las que había utilizado hasta ahora, que eran SQL.

Y en un contexto más general, la aplicación podría tener un impacto positivo para aquellas personas que quieran tener organizadas en un único lugar todas las películas que ha visto, que quiere ver más tarde o que son sus favoritas, en vez de tener guardadas las películas que ha visto o quiere ver en cada una de las plataformas disponibles de streaming, ya que de esta manera se complica cuando quieres ver una película nueva y no recuerdas si la has visto o la tenías guardada para ver más tarde en una plataforma, pero la ves en otra. También como impacto positivo sería que el usuario que utilice esta aplicación dispondrá de un catálogo completo con todas las películas estrenadas o futuras de cada una de las diferentes plataformas, lo que hace que sea más sencillo para el usuario elegir la próxima película que quiere ver.

8 Bibliografía

Publicaciones utilizadas en el estudio y desarrollo del trabajo.

[1] Google Play. CineTrak [Online]. Disponible: https://play.google.com/store/apps/details?id=com.cinetrak.mobile&hl=es&gl=US

[2] Google Play. MyMovies [Online]. Disponible: https://play.google.com/store/apps/details?id=dk.mymovies.mymovies4forandroidfree&hl=es_US

[3] Google Play. SeriesGuide [Online]. Disponible: https://play.google.com/store/apps/details?id=com.battlelancer.seriesguide&hl=es&gl=US

[4] Google Play. JustWatch [Online]. Disponible: https://play.google.com/store/apps/details?id=com.justwatch.justwatch&hl=es_419&gl=US

[5] Android Studio. [Online] Disponible: https://developer.android.com/studio/intro?hl=es-419

[6] Kotlin. [Online]. Disponible: https://kotlinlang.org/docs/android-overview.html

[7] TMDB. [Online]. Disponible: https://developer.themoviedb.org/docs/getting-started

[8] Firebase. [Online]. Disponible: https://firebase.google.com/docs/guides?hl=es-419

[9] Figma. [Online]. Disponible: https://www.figma.com/dev-mode/

[10] Requisitos funcionales y requisitos no funcionales. [Online]. Disponible: https://www.gluo.mx/blog/requisitos-no-funcionales-por-que-son-importantes

[11] Casos de uso. [Online]. Disponible: https://www.ibm.com/docs/es/product-master/12.0.0?topic=processes-defining-use-cases

[12] Arquitectura MVVM. [Online]. Disponible: https://cursokotlin.com/mvvm-en-android-con-kotlin-livedata-y-view-binding-android-architecture-components/

Este documento esta firmado por

	**Firmante**	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
	Fecha/Hora	Mon Jun 03 22:24:53 CEST 2024
	Emisor del Certificado	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
	Numero de Serie	561
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)