



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

WhatsNew: una aplicación móvil de gestión de series y películas

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Tejo Catala, Daniel Noel del

Tutor/a: Andrés Martínez, David de

CURSO ACADÉMICO: 2022/2023

Resum

La nostra societat ha experimentat un canvi completament en la seua manera de viure i de passar el temps lliure. Hui en dia som més selectius en la forma que gastem el nostre temps lliure, per això, necessitem aplicacions que ens faciliten la decisió i que millor manera de fer-ho, que amb una aplicació que estigui als nostres mòbils. Per això, aquesta aplicació et permet estalviar molt de temps a l'hora de decidir quina sèrie o pel·lícula veure. WhatsNew permet buscar i filtrar sèries i pel·lícules pels teus gustos, així com guardar les que vulguis veure en un futur per millorar l'algorisme de recomanació. Tota informació quant a les pel·lícules i sèries és obtinguda mitjançant l'API de TheMovieDB, així com gran part de les recomanacions.

Paraules clau: Android, Java, Firebase, Mòbil, Pel·lícules, Sèries, TheMovieDB

Resumen

Nuestra sociedad ha experimentado un cambio por completo en su forma de vivir y de pasar el tiempo libre. Hoy en día somos más selectivos en la forma que gastamos nuestro tiempo libre, por ello, necesitamos aplicaciones que nos faciliten la decisión y que mejor forma de hacerlo, que con una aplicación que esté en nuestros móviles. Por ello, esta aplicación te permite ahorrar mucho tiempo a la hora de decidir qué serie o película ver. WhatsNew permite buscar y filtrar series y películas por tus gustos, así como, guardar las que quieras ver en un futuro para mejorar el algoritmo de recomendación. Toda información en cuanto a las películas y series es obtenida mediante la API de TheMovieDB, así como gran parte de las recomendaciones.

Palabras clave: Android, Java, Firebase, Móvil, Películas, Series, TheMovieDB

Abstract

Our society has completely changed its way of living and spending free time. Nowadays, we are more selective in how we spend our free time. Therefore, we need applications that make the decision easier for us and what better way to do it than with mobile applications? Therefore, this application saves the user much time deciding which series or movie to watch. WhatsNew allows the user to search and filter series and movies by the user's tastes and save the ones the user wants to see in the future to improve the recommendation algorithm. All information regarding movies and series is obtained through TheMovieDB API and a large part of the recommendations.

Key words: Android, Java, Firebase, Mobile, Movies, Series, TheMovieDB

Índice general

Índice general	V
Índice de figuras	VII
Índice de tablas	X
<hr/>	
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	1
1.3 Metodología	2
1.4 Estructura de la memoria	3
2 Estado del arte	5
2.1 JustWatch	5
2.1.1 Características principales	5
2.2 AGoodMovieToWatch	6
2.2.1 Características principales	7
2.3 Tv Time	8
2.3.1 Características principales	8
2.4 Google TV	9
2.4.1 Características principales	9
2.5 Análisis del estado del arte	11
2.6 Funcionalidad del proyecto	12
3 Especificación de requisitos	15
3.1 Requisitos funcionales	15
3.2 Requisitos no funcionales	24
4 Diseño de la solución	27
4.1 Arquitectura MVVM	27
4.2 Arquitectura del sistema	27
4.3 Arquitectura del directorio	28
4.3.1 Conceptos previos	29
4.3.2 Carpeta java	31
4.3.3 Carpeta res	33
4.4 Diagrama UML	34
4.5 Bocetos	36
4.5.1 Inicio	36
4.5.2 Búsqueda	37
4.5.3 Archivo	39
4.5.4 Ajustes	39
4.5.5 Recomendación aleatoria	40
4.5.6 Película o series	41
4.6 Diagrama de flujo entre bocetos	42
5 Tecnologías utilizadas	45
5.1 Java	45
5.2 Google Play Console	45

5.3	Firebase	46
5.4	Github	46
5.5	Lucidchart	47
5.6	Figma	47
5.7	Android Studio	47
6	Implementación de la solución	49
6.1	Bibliotecas utilizadas	49
6.1.1	OkHTTP	49
6.1.2	Glide	50
6.1.3	Firestore	51
6.2	TheMovieDB API	54
6.3	Utilización del cache	54
6.4	Recomendaciones	55
6.4.1	Recomendaciones en el inicio	56
6.4.2	Recomendaciones aleatorias	57
7	Pruebas	61
7.1	UAT	61
7.2	Pruebas de calidad	68
7.2.1	Experiencia visual	68
7.2.2	Funcionalidad	69
7.2.3	Rendimiento y estabilidad	70
7.2.4	Privacidad y seguridad	71
7.2.5	Google Play	73
7.3	Pruebas de funcionalidad	73
8	Aplicación final	75
8.1	Interfaces	75
8.1.1	Inicio	75
8.1.2	Inicio de sesión	77
8.1.3	Registro	78
8.1.4	Búsqueda	79
8.1.5	Selector de perfiles	80
8.1.6	Visor de películas o series	81
8.1.7	Recomendaciones aleatorias	83
8.1.8	Biblioteca de películas o series marcadas	84
8.1.9	Ajustes	85
9	Conclusiones	87
9.1	Relación con los estudios cursados	87
9.2	Desarrollos futuros	88
	Bibliografía	89
<hr/>		
	Apéndice	
A	Anexo ODS	93

Índice de figuras

1.1	Metodología ágil implementada en Trello	3
2.1	Logo de JustWatch	5
2.2	Captura de pantalla del inicio de JustWatch	6
2.3	Logo de AGoodMovieToWatch	7
2.4	Captura de pantalla del inicio de AGoodMovieToWatch	7
2.5	Logo de Tv Time	8
2.6	Captura de pantalla del inicio de Tv Time	9
2.7	Logo de Google TV	9
2.8	Captura de pantalla del inicio de Google TV	10
3.1	Diagrama de casos de uso de un usuario	15
3.2	Diagrama de casos de uso de un usuario autenticado	19
4.1	Diagrama con la arquitectura del sistema	28
4.2	Directorio de la aplicación	29
4.3	Ciclo de vida de una actividad	30
4.4	Paquete de la aplicación	31
4.5	Paquete de actividades de la aplicación	32
4.6	Paquete de fragmentos de la aplicación	32
4.7	Paquete de datos de la aplicación	33
4.8	Paquete de vistas personalizadas de la aplicación	33
4.9	Directorio de recursos de la aplicación	34
4.10	Diagrama UML de la aplicación	35
4.11	Boceto inicio	37
4.12	Boceto inicio con vista desplegada	37
4.13	Boceto de la interfaz de búsqueda	38
4.14	Boceto de la interfaz de los resultados de la búsqueda	38
4.15	Boceto de la interfaz de las listas de marcados	39
4.16	Boceto de la interfaz de los ajustes	40
4.17	Boceto de la interfaz de las recomendaciones aleatorias	41
4.18	Boceto de la interfaz de las películas y series	42
4.19	Resto de la interfaz de las películas y series	42
4.20	Boceto de la interfaz de las series	42
4.21	Diagrama de flujo entre los bocetos	43
5.1	Logo de Java	45
5.2	Logo de Google Play Console	46
5.3	Logo de Firebase	46
5.4	Logo de GitHub	46
5.5	Logo de Lucidchart	47
5.6	Logo de Figma	47
5.7	Logo de Android Studio	48

6.1	Fragmento de código con la llamada asíncrona a un servicio externo	50
6.2	Código donde se obtiene una imagen para un póster	51
6.3	Página web de Firebase en el apartado Crashlytics	51
6.4	Página web de Firebase en el apartado Performance	52
6.5	Fragmento del código de descarga de información de los usuarios	53
6.6	Fragmento del código que inicia sesión en Firebase Auth	53
6.7	Fragmento del código que devuelve la URL a la que hacer la petición	54
6.8	Fragmento del código que comprueba si se encuentra en el cache rápido . .	55
6.9	Fragmento del código que comprueba si se encuentra en el segundo cache .	55
6.10	Fragmento del código que crea los registros en el cache rápido y en el secundario	55
6.11	URL de la API para las recomendaciones	56
6.12	Fragmento del código que guarda las categorías y URL de películas similares a las guardadas	56
6.13	Iniciación de la clase de recomendaciones	57
6.14	Método que devuelve una lista de recomendaciones	58
6.15	Método que ordena una lista de recomendados	58
7.1	Resultados de “La aplicación es fácil de usar”	62
7.2	Resultados de “La aplicación es fácil de aprender”	62
7.3	Resultados de “La aplicación cumple con mis expectativas”	63
7.4	Resultados de “Los textos de la aplicación son claros y entendibles”	63
7.5	Resultados de “La aplicación responde de manera rápida y eficiente”	63
7.6	Resultados de “La tipografía escogida es legible y clara”	64
7.7	Resultados de “Encuentro fácilmente las funciones que necesito”	64
7.8	Resultados de “Los botones y los elementos interactivos responden bien al tacto”	64
7.9	Resultados de “Las animaciones y transiciones entre pantallas me parecen fluidas y agradables”	65
7.10	Resultados de “La aplicación es visualmente atractiva”	65
7.11	Resultados de “La navegación dentro de la aplicación es intuitiva”	66
7.12	Resultados de “Recomendaría la aplicación a un amigo o familiar”	66
7.13	Resultados de “Han ocurrido errores en la prueba”	67
7.14	Resultados de “¿Has encontrado algún error?”	67
7.15	Resultados de “¿Algún aspecto a mejorar?”	68
8.1	Captura del inicio	76
8.2	Captura del inicio con desplegable	76
8.3	Captura del inicio con plataforma habiendo hecho el clic	77
8.4	Captura del inicio de sesión o registro	78
8.5	Captura de la pantalla de inicio o registro	78
8.6	Captura de la pantalla de inicio de sesión	78
8.7	Captura de la pantalla de registro	79
8.8	Captura de la pantalla de búsqueda	80
8.9	Captura de los resultados de la búsqueda	80
8.10	Captura de la pantalla de selección de perfil	81
8.11	Captura de pantalla de la información de la película Alias	82
8.12	Captura de pantalla de la parte de abajo de la interfaz	82
8.13	Captura de la pantalla de la información de una serie	83
8.14	Captura de la pantalla de la recomendación de “Juego de Tronos”	84
8.15	Captura de la pantalla para filtrar las recomendaciones aleatorias	84
8.16	Captura de la pantalla de la biblioteca la sección de películas	85
8.17	Captura de la pantalla de la biblioteca la sección de series	85

8.18 Captura de la pantalla de ajustes	86
--	----

Índice de tablas

2.1	Tabla de análisis del estado del arte	11
3.1	RF01 - Mostrar valoraciones de episodios.	16
3.2	RF02 - Mostrar el reparto de una película o serie.	16
3.3	RF03 - Mostrar películas o series en cada plataforma	16
3.4	RF04 - Mostrar información de películas o series	17
3.5	RF05 - Mostrar plataformas donde ver películas o series	17
3.6	RF06 - Mostrar tiempo restante a que se estrene una película	17
3.7	RF07 - Ver películas o series similares	18
3.8	RF08 - Buscar películas o series	18
3.9	RF09 - Iniciar sesión	18
3.10	RF10 - Registrarse	19
3.11	RF11 - Marcar o desmarcar como visto	20
3.12	RF12 - Recomendaciones en el inicio	20
3.13	RF13 - Recomendaciones aleatorias	20
3.14	RF14 - Seguir serie	21
3.15	RF15 - Filtrar el contenido por suscripciones	21
3.16	RF16 - Marcar o desmarcar como pendiente	22
3.17	RF17 - Crear perfiles para cada usuario	22
3.18	RF18 - Ocultar películas o series vistas	23
3.19	RF19 - Filtrar películas seguidas	23
3.20	RF20 - Filtrar películas vistas	24
3.21	RF21 - Cerrar sesión	24
3.22	Requisitos no funcionales	25
7.1	Resultados de las pruebas de “Experiencia visual”	68
7.1	Resultados de las pruebas de “Experiencia visual”	69
7.2	Resultados de las pruebas de “Funcionalidad”	70
7.3	Resultados de las pruebas de “Rendimiento y estabilidad”	71
7.4	Resultados de las pruebas de “Privacidad y seguridad”	71
7.4	Resultados de las pruebas de “Privacidad y seguridad”	72
7.5	Resultados de las pruebas de “Google Play”	73
7.6	Resultados de las pruebas de funcionalidad	74
A.1	Tabla de ODS	93

CAPÍTULO 1

Introducción

En el siguiente apartado se hará una breve explicación de cuales van a ser las motivaciones, objetivos, metodología aplicada y estructura del proyecto a realizar.

1.1 Motivación

En los años 60 y 70, el cine y la televisión eran las principales formas de entretenimiento audiovisual en la mayoría de los países. La televisión ofrecía una programación limitada y, por lo general, enfocada en la transmisión de noticias, deportes y programas de entretenimiento en un momento concreto. El cine, por su parte, era un lujo al alcance de pocos, y las salas de cine eran el lugar donde la mayoría de la gente iba a ver las últimas novedades del séptimo arte.

Con la llegada del siglo XXI y la popularización de internet, surgieron nuevas formas de consumo audiovisual [1]. Las plataformas de *streaming*, como Netflix, Amazon Prime o HBO, que ofrecen una amplia oferta de series y películas, disponibles para su visualización en cualquier momento y lugar.

Por otra parte, en la sociedad actual, el tiempo libre es un bien escaso y valioso. Muchas veces, después de una larga jornada de trabajo o de estudio, lo que más apetece es relajarse en el sofá y disfrutar de una buena serie o película. Sin embargo, la gran cantidad de plataformas de *streaming* disponibles en la actualidad puede hacer que la tarea de encontrar el contenido adecuado sea abrumadora y consuma mucho tiempo [2]. Además, el alto coste de algunas de estas plataformas puede suponer un obstáculo para muchas personas que quieren acceder al amplio catálogo de contenido que ofrecen. De esta manera, con esta aplicación se pretende maximizar el tiempo de ocio del usuario y facilitar el acceso a una amplia oferta de contenido de calidad sin necesidad de pagar precios excesivos.

Es en este contexto donde una aplicación que permita gestionar el consumo de series y películas se convierte en una herramienta valiosa para los usuarios, ya que ofrece soluciones prácticas y efectivas a estos desafíos y mejora la experiencia de entretenimiento.

1.2 Objetivos

Uno de los objetivos principales del trabajo es diseñar una aplicación en Android nativo que facilite al usuario la búsqueda de contenido, permitiendo al usuario filtrar y buscar series y películas por género, año, idioma, plataforma de *streaming*, entre otros

criterios. De esta forma, el usuario tendría acceso a una amplia variedad de contenido que se adapte a sus gustos y necesidades, ahorrando tiempo y esfuerzo en la búsqueda.

Por otra parte, tendrá como objetivos específicos:

- Reducir el costo de acceso al contenido deseado: Una de las barreras más importantes que existen para el acceso al contenido audiovisual es el costo de las suscripciones a plataformas de *streaming*, debido a la gran cantidad de plataformas que existen actualmente. Una aplicación de gestión de series y películas podría abordar este problema ofreciendo una opción de visualización de las series y películas que ofrecen los planes de pago de diferentes plataformas, para que el usuario pueda decidir cuál es la más rentable de pagar.
- Organizar el tiempo de visualización: Otra de las funcionalidades que podría ofrecer la aplicación sería la de organizar y gestionar el tiempo de visualización del usuario, permitiéndole crear listas de series y películas pendientes y recibir recomendaciones personalizadas en función de su historial de visualización. De esta forma, el usuario tendría una mejor gestión del tiempo de ocio y estaría más satisfecho con su experiencia de consumo.

1.3 Metodología

La metodología para la realización de este trabajo que se ha elegido es la metodología ágil [3] que se define como una aproximación iterativa e incremental para el desarrollo de software, que se enfoca en la entrega continua de valor al cliente.

Al aplicar la metodología ágil al desarrollo de una aplicación de gestión de series y películas, se priorizaría la entrega rápida de características funcionales a los usuarios, con el objetivo de ofrecer un valor tangible desde el inicio del proyecto. En este sentido, se trabajaría en ciclos cortos de desarrollo, denominados *sprints* [4].

Concretamente se ha seguido la metodología ágil de Kanban [5]. Es un enfoque visual y flexible de gestión de proyectos que permite a los equipos de desarrollo supervisar y optimizar el flujo de trabajo, utilizando la herramienta de Trello [6] con una columna para las tareas pendientes, las que se están haciendo y las hechas.

En una aplicación de gestión de series y películas, el uso de Kanban podría permitir al equipo de desarrollo seguir fácilmente el estado de los trabajos de desarrollo, desde la identificación de las necesidades del usuario hasta la implementación y lanzamiento de nuevas funcionalidades. Además, ayudaría a evitar los problemas más comunes como los cuellos de botella y los tiempos muertos. Con el uso de tarjetas visuales, el equipo podría seguir el progreso de cada tarea, desde su inicio hasta su finalización, y priorizarlas en función de su importancia y urgencia.

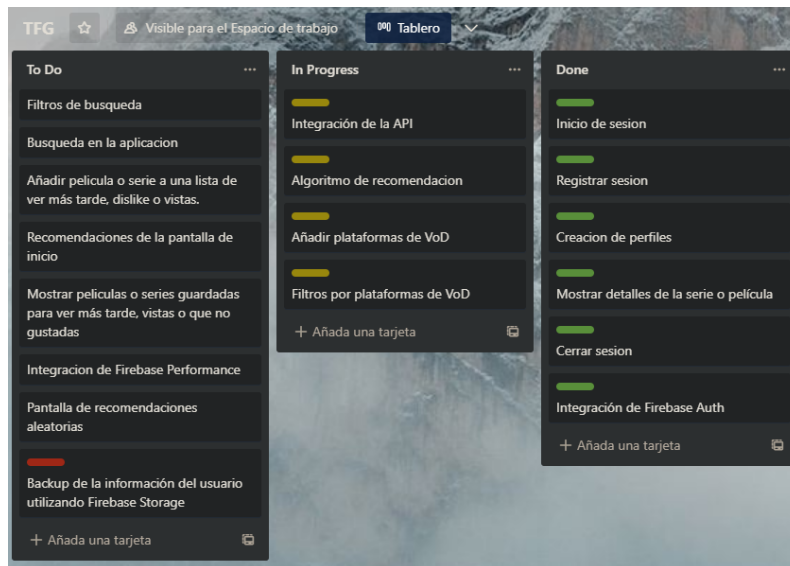


Figura 1.1: Metodología ágil implementada en Trello

En la figura 1.1, podemos observar que existen varios estados en los que puede estar una tarea, como son los estados de "To Do" (Pendiente), "In Progress" (En Desarrollo) o "Done" (Hechas). Estos estados permiten gestionar el flujo de avance y entrega del proyecto.

1.4 Estructura de la memoria

Seguidamente, vamos a explicar cuál va a ser la estructura de la memoria y los puntos que vamos a abordar posteriormente.

- Estado del arte: Se trata de una sección en la que se realiza una investigación exhaustiva sobre otras aplicaciones que compiten en el mismo mercado que la aplicación planteada. Esta sección implica un análisis detallado de las diferentes características y funcionalidades que ofrecen estas aplicaciones, así como su impacto en el mercado y las debilidades que el trabajo puede aprovechar. La finalidad de esta sección es obtener una comprensión completa del mercado y de la competencia existente.
- Análisis del estado del arte: En esta sección, se profundiza en el análisis de las aplicaciones analizadas en el estado del arte, centrándose en las funcionalidades ofrecidas. Se evalúa cómo estas aplicaciones abordan los problemas existentes y las oportunidades de mejora. El objetivo es obtener una comprensión más profunda del problema que se está intentando solucionar y, a su vez, identificar oportunidades de mejora que puedan ser implementadas en la aplicación propuesta.
- Especificación de requisitos: En esta sección se especifican los requisitos funcionales y no funcionales del proyecto. La finalidad de esta sección es establecer una guía clara sobre los objetivos y requerimientos del proyecto.
- Diseño de la solución: En esta sección se describe la arquitectura que se utilizará para la aplicación propuesta, el diagrama de clases y los bocetos de todas las pantallas de la aplicación. La finalidad de esta sección es establecer un marco detallado de cómo se llevará a cabo la implementación de la solución.

- Tecnologías utilizadas: se hacen referencia las tecnologías y herramientas utilizadas en la aplicación. Se trata de una sección importante ya que la elección de las tecnologías y herramientas puede tener un impacto significativo en el rendimiento y la funcionalidad de la aplicación.
- Implementación de la solución: en esta se detallan el cómo se ha pasado de la propuesta inicial a la solución final, los problemas y dificultades encontradas, las decisiones que se han tenido que tomar, particularidades de la solución final. Además de mostrar los aspectos complejos y novedosos del sistema desarrollado
- Pruebas: se detallan las pruebas realizadas y los resultados obtenidos.
- Aplicación final: En esta sección se exponen todas las interfaces de la aplicación desarrollada. La finalidad de esta sección es presentar la aplicación final al lector.
- Conclusiones: La finalidad de esta sección es resumir los hallazgos más importantes y proporcionar una reflexión sobre los resultados obtenidos.
- Bibliografía: aquí se detallan todas las fuentes de información utilizadas para la realización de este trabajo.

CAPÍTULO 2

Estado del arte

En esta sección se analizan y documentan las aplicaciones del mismo ámbito que la aplicación planteada, con el objetivo de conocer las funcionalidades que ofrecen, las tecnologías utilizadas y las limitaciones que presentan. Se ha hecho el análisis con aplicaciones con las que va a competir la aplicación presentada como: JustWatch, AGoodMovieToWatch, TvTime Y Google TV.

2.1 JustWatch

JustWatch, cuyo logo aparece en la figura 2.1, es una aplicación de gestión de películas o series que busca en plataformas de *streaming* para ayudarte a encontrar dónde verlas [7]. Esta aplicación se encuentra disponible para los dispositivos Android y IOS, además de estar disponible en varios sistemas operativos de televisión [8]. La aplicación apareció en la plataforma de Google Play en 2015. La principal desventaja es que la aplicación no ofrece la posibilidad de filtrar las recomendaciones para que estén disponibles en unas plataformas determinadas y que la aplicación tiene algunas funciones bloqueadas bajo una suscripción. Podemos ver la pantalla principal en la figura 2.2.



Figura 2.1: Logo de JustWatch

2.1.1. Características principales

- Búsqueda y exploración de contenido: los usuarios pueden buscar y explorar películas y programas de televisión de varios servicios de *streaming*.
- Filtros de búsqueda avanzados: los usuarios pueden aplicar filtros de búsqueda avanzados para encontrar contenido basado en diferentes criterios, como el género, la clasificación por edades y la fecha de lanzamiento.

- Listas de seguimiento personalizadas: los usuarios pueden crear y mantener listas de seguimiento personalizadas de sus programas y películas favoritas.
- Alertas de precio y disponibilidad: los usuarios pueden recibir alertas de precios y disponibilidad de programas y películas que estén interesados en ver.
- Calificaciones y reseñas de usuarios: los usuarios pueden leer y dejar calificaciones y reseñas de películas y programas de televisión para ayudar a otros usuarios a decidir qué ver.
- Integración con servicios de *streaming*: los usuarios pueden conectarse directamente a los servicios de *streaming* y ver el contenido que han encontrado en JustWatch.
- Información detallada del contenido: los usuarios pueden ver información detallada sobre el contenido, como la trama, el elenco y la duración, para ayudarles a decidir qué ver.

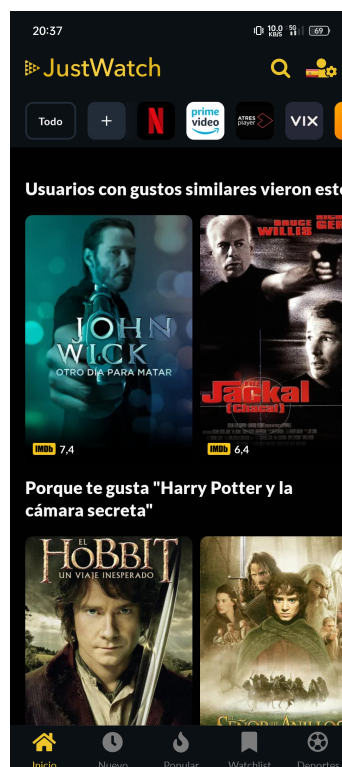


Figura 2.2: Captura de pantalla del inicio de JustWatch

2.2 AGoodMovieToWatch

A Good Movie To Watch, cuyo logo aparece en la figura 2.3, [9] es un sitio web que se dedica a recomendar películas y series poco conocidas, pero críticamente aclamadas. Todas las películas y series que aparecen aquí tendrán como mínimo un 70 % en Rotten Tomatoes y al menos un 7 en IMDB. La página principal aparece en la figura 2.4.



Figura 2.3: Logo de AGoodMovieToWatch

2.2.1. Características principales

- Recomendaciones instantáneas: la aplicación te recomienda películas y series sin tener que determinar cuáles son tus gustos.
- Búsqueda avanzada: puedes buscar películas y series por género, año, país, idioma...
- Listas de favoritos: puedes crear listas de películas y series que te gusten para verlas más tarde y descartar cuales ya has visto o no te interesan.
- Calificaciones: puedes calificar películas y series que hayas visto.

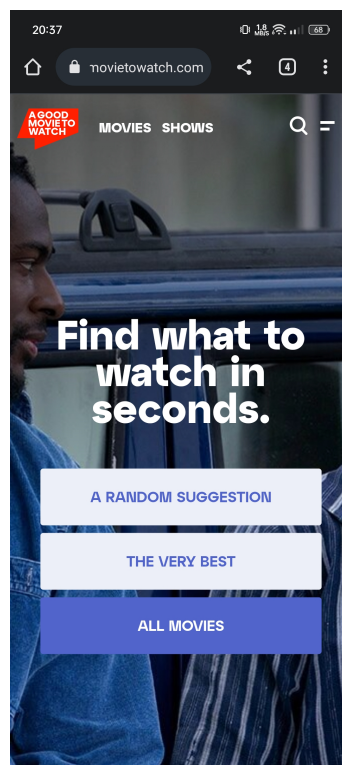


Figura 2.4: Captura de pantalla del inicio de AGoodMovieToWatch

2.3 Tv Time

TV Time, cuyo logo es este 2.5, es una aplicación móvil diseñada para ayudar a los usuarios a hacer seguimiento de los programas de televisión que están viendo y los que han visto en el pasado. Esta aplicación se encuentra en Android e IOS y fue lanzada el año 2011. La aplicación muestra inicialmente la pantalla de la figura 2.6.



Figura 2.5: Logo de Tv Time

2.3.1. Características principales

- Calendario de programas de televisión: puedes ver cuándo se emiten tus programas de televisión favoritos y recibir notificaciones cuando se emiten nuevos episodios.
- Seguimiento de programas de televisión y películas: puedes hacer un seguimiento de los programas de televisión y películas que has visto y los que quieres ver.
- Comentarios y reseñas: puedes leer comentarios y reseñas de otros usuarios sobre programas de televisión y películas.
- Estadísticas: puedes ver estadísticas sobre tus programas de televisión y películas favoritas, como el número de episodios que has visto y el tiempo total que has pasado viendo programas de televisión y películas. Así como la valoración de cada episodio por separado en forma de gráfica.

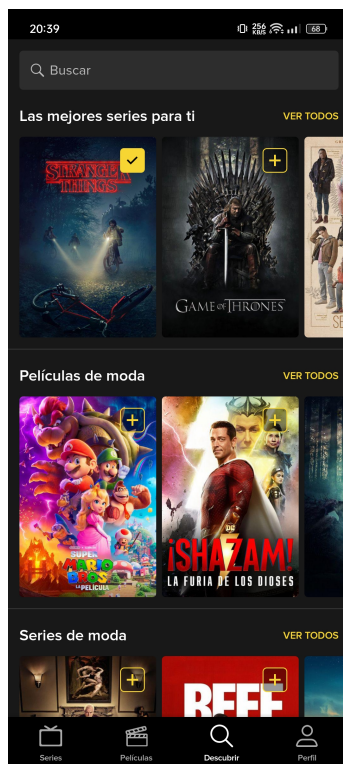


Figura 2.6: Captura de pantalla del inicio de Tv Time

2.4 Google TV

Google TV, cuyo logo aparece en la figura 2.7, [10] es una plataforma de *streaming* que te permite ver contenido de diferentes plataformas en un solo lugar. Esta aplicación busca ser el punto de unión de todas las plataformas de *streaming*, así como funcionar como un sistema operativo para los dispositivos de Smart TV de Google. Este está disponible en los dispositivos móviles tanto IOS como Android y en las Smart TV. La pantalla principal aparece en la figura 2.8.



Figura 2.7: Logo de Google TV

2.4.1. Características principales

- **Búsqueda de contenido:** puedes buscar contenido de diferentes servicios de *streaming*, como Amazon Prime Video, HBO Max, Viki, Discovery+ y Disney+.
- **Recomendaciones personalizadas:** la aplicación te recomienda contenido basado en tus gustos y preferencias.
- **Control de voz:** puedes controlar la aplicación con tu voz utilizando el Asistente de Google.
- **Integración con otros dispositivos:** puedes compartir contenido desde tus dispositivos móviles a la pantalla de la televisión en un solo clic.
- **Interfaz de usuario intuitiva, elegante y fácil de usar.**

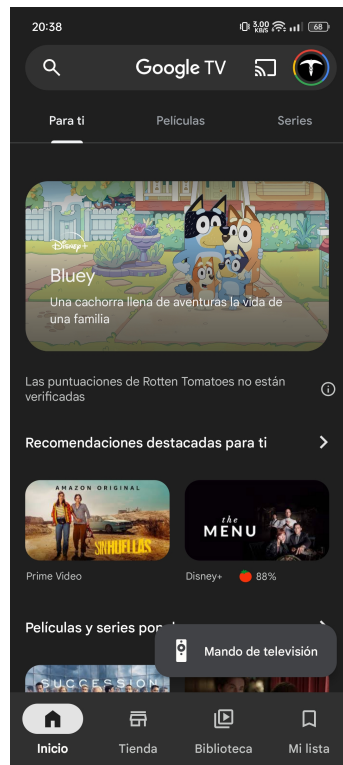


Figura 2.8: Captura de pantalla del inicio de Google TV

2.5 Análisis del estado del arte

A continuación, en esta sección se ha elaborado una tabla que registra una amplia gama de características compartidas y únicas de las aplicaciones mencionadas en el estado del arte.

Tabla 2.1: Tabla de análisis del estado del arte

Funcionalidad	JustWatch	TvTime	AGoodMovie ToWatch	Google TV
Iniciar sesión	X	X	X	X
Registrarse	X	X	X	X
Cerrar sesión	X	X	X	X
Crear listas de reproducción		X		X
Filtrar la búsqueda	X			
Mostrar contenido popular	X	X	X	X
Mostrar contenido recomendado	X	X	X	X
Recibir notificaciones de las novedades	X	X		X
Marcar como visto	X	X	X	X
Seguir series	X	X		X
Ordenar la búsqueda (popularidad, valoración)	X			
Mostrar detalles de una película o serie	X	X	X	X
Mostrar detalles de actores	X	X		
Mostrar películas ofertadas en una plataforma	X		X	X
Mostrar próximos lanzamientos	X	X		
Mostrar la valoración del contenido	X	X	X	X
Mostrar la valoración de los episodios	X	X		
Marcar como "ver más tarde"	X	X	X	X
Muestra estadísticas de visualización del usuario		X		
Comparte las visualizaciones con otros usuarios		X		
Muestra detalles de eventos deportivos	X			
Muestra el contenido que va a dejar de incluir la plataforma	X			
Muestra promociones de contenido de pago	X			
Filtrar las novedades	X			

Aunque todas las aplicaciones ofrecen funcionalidades parecidas, JustWatch es la más completa. Dispone de la mayoría las funcionalidades descritas y principales que debería tener una aplicación de estas características, aunque hay algunas que son de pago.

Por otra parte, la aplicación de AGoodMovieToWatch es la que menos variedad ofrece ya que, al ser una aplicación web, esta se ve muy limitado a la hora de ofrecer servicio a sus usuarios. Por ejemplo, a la hora de notificar a sus usuarios de las novedades existentes, ya que, al utilizar un navegador, esta no tiene una forma sencilla de enviar notificaciones.

Mientras que Google Tv como aplicación ofrece una gran cantidad de herramientas para el usuario, tiene como desventaja que es una aplicación con un catálogo muy reducido de películas y series ya que no incluye varias plataformas populares de *streaming*.

Todas las aplicaciones comparadas son capaces de manejar el registro del usuario en su sistema, por ello, esta funcionalidad es básica para la aplicación a desarrollar en este proyecto. Además, este registro debe proporcionar que la información guardada sobre este usuario pueda utilizarse en diferentes dispositivos, por lo tanto, esta información debe quedar guardada en la nube [11].

Para finalizar cabe recalcar que, JustWatch y TvTime, utilizan la misma API para la información de las películas y la series, y por ello, obtienen resultados similares a la hora de comparar sus funcionalidades.

2.6 Funcionalidad del proyecto

Realizado el análisis de las funcionalidades que ofertan los productos ya en el mercado, se determinará las funcionalidades que deben estar dentro del proyecto de WhatsNew.

Las funcionalidades que debe tener la aplicación son:

- Registrarse.
- Iniciar sesión.
- Cerrar sesión
- Mostrar las películas y las series popular
- Mostrar contenido recomendado al usuario
- Marcar las películas y las series como visto
- Poder seguir series
- Poder ordenar la búsqueda
- Mostrar detalles de las películas y las series
- Mostrar detalles de los actores
- Mostrar películas ofertadas en una determinada plataforma
- Mostrar próximos lanzamientos
- Mostrar la valoración de las películas y las series
- Marcar como “ver más tarde”

Por otra parte, también podrá tener características únicas como poder filtrar por contenido no visto, para obtener películas o series que el usuario no haya marcado o visto antes. Además de tener la capacidad de utilizar varios perfiles para organizar las películas o series que le gustan a diferentes usuarios, que vayan a utilizar la aplicación en el mismo dispositivo.

CAPÍTULO 3

Especificación de requisitos

Una especificación de requisitos de un *software* [12] es un documento que describe las necesidades que debe cumplir este sistema. Esta etapa define y documenta los requisitos funcionales y no funcionales del sistema, proporcionando una base sólida para el diseño, desarrollo, implementación y prueba del mismo.

La especificación de requisitos del *software* es fundamental para asegurar que el resultado final cumpla con los objetivos establecidos y satisfaga las necesidades del usuario final.

3.1 Requisitos funcionales

Para describir los requisitos funcionales de la aplicación debemos primero presentar el caso de uso que va a presentar un usuario y un usuario autenticado, teniendo este último, el acceso a toda la funcionalidad que puede ofrecer la aplicación.

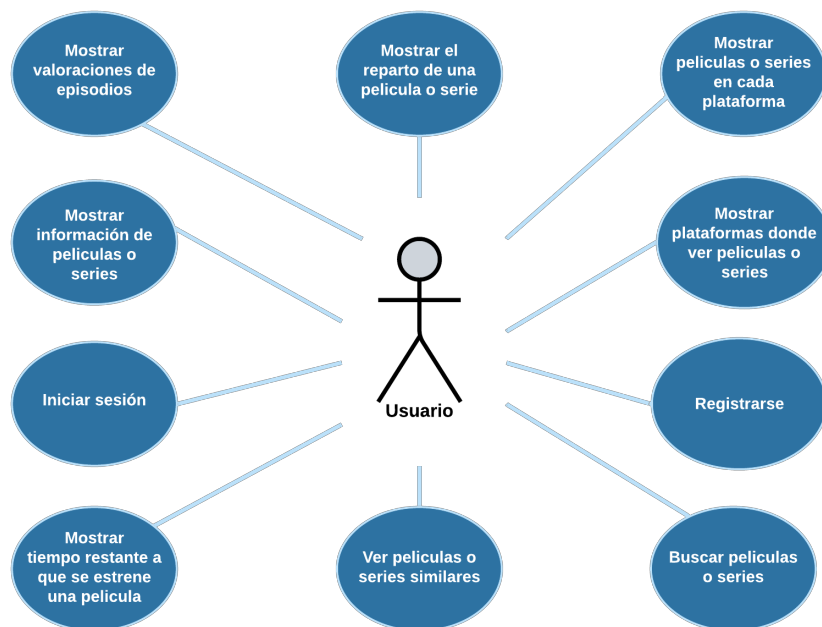


Figura 3.1: Diagrama de casos de uso de un usuario

Aquí se mostrarán los requisitos funcionales para un usuario que no se encuentra autenticado en la aplicación.

Tabla 3.1: RF01 - Mostrar valoraciones de episodios.

Id	RF-01
Nombre	Mostrar valoraciones de episodios.
Descripción	Se mostrarán las valoraciones de los episodios en una gráfica, mostrando por temporadas.
Proceso	1. El usuario presiona una serie. 2. El sistema muestra las valoraciones de la serie en forma de gráfica.
Precondición	Tiene que ser una serie.
Postcondición	Las valoraciones tienen que mostrarse en una gráfica.
Errores	Puede haber episodios sin valoración ya que depende de la API de TheMovieDB.

Tabla 3.2: RF02 - Mostrar el reparto de una película o serie.

Id	RF-02
Nombre	Mostrar el reparto de una película o serie.
Descripción	Se podrá ver todo el reparto de una película o serie en una lista de fotos del actor o actriz.
Proceso	1. El usuario presiona en una película o serie. 2. El sistema devuelve una lista de actores y actrices que han participado en la película o serie.
Precondición	Ninguna.
Postcondición	Las fotos se mostrarán en una lista de imágenes.
Errores	Puede no haber información sobre el reparto, en ese caso no se mostrará la lista.

Tabla 3.3: RF03 - Mostrar películas o series en cada plataforma

Id	RF-03
Nombre	Mostrar películas o series en cada plataforma.
Descripción	Se mostrará en una lista, las películas o series contenidas en una suscripción de una plataforma.
Proceso	1. En el inicio presiona en una plataforma. 2. El sistema muestra una lista de películas o series ofertadas en la plataforma seleccionada.
Precondición	Ninguna.
Postcondición	No se pueden mostrar películas o series de varias plataformas distintas a la vez.
Errores	Ninguno.

Tabla 3.4: RF04 - Mostrar información de películas o series

Id	RF-04
Nombre	Mostrar información de película o series.
Descripción	Aquí se mostrará toda la información contenida sobre la película o serie que se haya seleccionado.
Proceso	1. El usuario presiona sobre una película o serie en el inicio.
	2. El sistema devuelve toda la información sobre la película o serie.
Precondición	Ninguna.
Postcondición	Algunos datos pueden no aparecer debido a que la serie o película no ha sido estrenada.
Errores	Ninguno.

Tabla 3.5: RF05 - Mostrar plataformas donde ver películas o series

Id	RF-05
Nombre	Mostrar plataformas donde ver películas o series.
Descripción	El usuario podrá ver en que plataforma se puede ver las películas o series que ha seleccionado ver.
Proceso	1. El usuario presiona una serie o una película.
	2. El sistema le muestra la plataforma donde está ofertada esa serie o película.
Precondición	Ninguna.
Postcondición	Tiene que estar ofertada en alguna plataforma.
Errores	Ninguno.

Tabla 3.6: RF06 - Mostrar tiempo restante a que se estrene una película

Id	RF-06
Nombre	Mostrar tiempo restante a que se estrene una película.
Descripción	El usuario podrá ver el tiempo que falta para que se estrene una película.
Proceso	1. El usuario presiona en una película que no ha sido estrenada.
	2. El sistema devuelve un contador de tiempo que falta para que se estrene.
Precondición	Tiene que ser una película.
Postcondición	Ninguna.
Errores	Ninguno.

Tabla 3.7: RF07 - Ver películas o series similares

Id	RF-07
Nombre	Ver películas o series similares.
Descripción	Se muestra una lista de películas o series similares.
Proceso	1. El usuario presiona una película o serie que quiera ver.
	2. El sistema le devuelve un conjunto de películas o series que sean similares a la que ha escogido.
Precondición	Ninguna.
Postcondición	Ninguna.
Errores	Ninguno.

Tabla 3.8: RF08 - Buscar películas o series

Id	RF-08
Nombre	Buscar películas o series.
Descripción	El usuario puede buscar películas, series o actores que quiera buscar.
Proceso	1. El usuario presiona en el botón de búsqueda.
	2. El usuario escribe un texto que quiera buscar.
	3. El sistema le devuelve películas, series o actores que coincidan con el texto escrito en el buscador.
Precondición	Tiene que haber escrito algo en el campo de búsqueda.
Postcondición	Ninguna.
Errores	Puede no haber ningún resultado de la búsqueda.

Tabla 3.9: RF09 - Iniciar sesión

Id	RF-09
Nombre	Iniciar sesión.
Descripción	El usuario puede iniciar sesión en el sistema.
Proceso	1. El usuario presiona el botón de ajustes.
	2. El usuario presiona en el botón de iniciar sesión en el sistema.
	3. El usuario introduce los datos de inicio de sesión en las cajas de texto con su correo y contraseña.
	4. El usuario presiona en el botón de iniciar sesión.
	5. El sistema comprueba si el correo existe y la contraseña corresponde con esa cuenta.
	6. El sistema descarga del sistema de archivos en la nube el fichero con la información del usuario y sus perfiles en el dispositivo.
Precondición	Tiene que haberse registrado el usuario con su correo y contraseña adecuada.
Postcondición	Ninguna.
Errores	La contraseña escrita puede ser incorrecta.

Tabla 3.10: RF10 - Registrarse

Id	RF-10
Nombre	Registrarse.
Descripción	El usuario puede registrarse en el sistema.
Proceso	<ol style="list-style-type: none"> 1. El usuario presiona el botón de ajustes. 2. El usuario presiona en el botón de registrarse en el sistema. 3. El usuario introduce los datos de inicio de sesión en las cajas de texto con su correo y contraseña. 4. El usuario presiona en el botón de registrarse. 5. El sistema comprueba si el correo ya existe en el sistema. 6. El sistema crea un nuevo archivo de usuario en la nube solamente con el perfil de Familia que está por defecto en todos los usuarios.
Precondición	Tiene que ser un nuevo correo y no existir en el sistema. Además, tiene que aceptar las condiciones de uso del sistema.
Postcondición	Ninguna.
Errores	Ninguno.

Por otra parte, con el usuario autenticado:

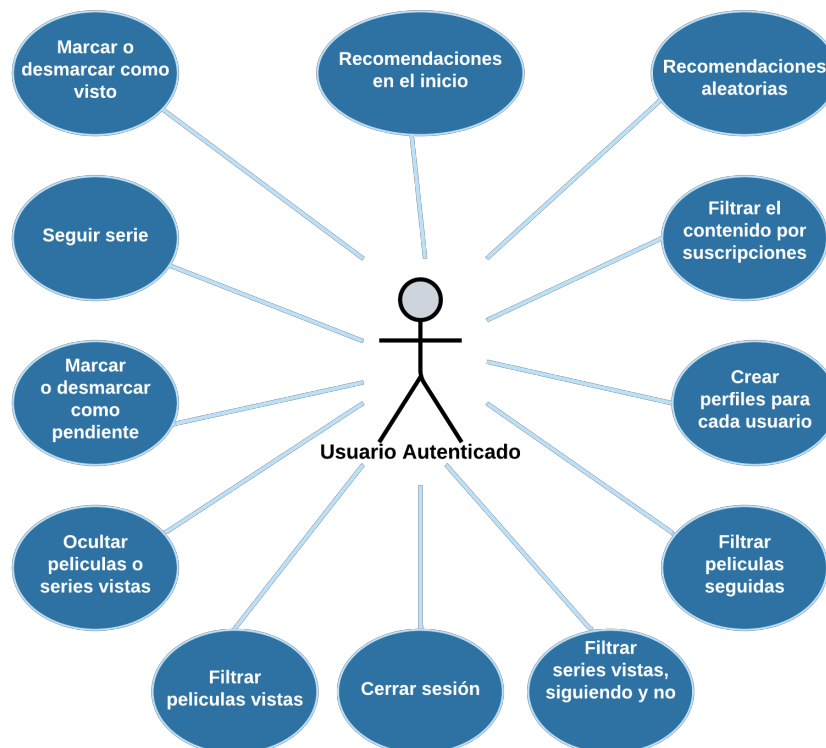


Figura 3.2: Diagrama de casos de uso de un usuario autenticado

Este usuario se encuentra autenticado en la aplicación y además de tener la misma funcionalidad que un usuario no autenticado este tiene sus propios casos de uso.

Tabla 3.11: RF11 - Marcar o desmarcar como visto

Id	RF-11
Nombre	Marcar o desmarcar como visto.
Descripción	El usuario puede marcar una película o series como vista.
Proceso	1. El usuario presiona sobre una película o serie.
	2. El usuario presiona sobre el elemento con forma de ojo.
	3. El sistema guarda que la película o serie ha sido vista.
	4. El sistema actualiza la vista para mostrar el botón con forma de ojo relleno.
Precondición	El usuario tiene que estar autenticado. El usuario no debe de haber marcado ya la serie o película como vista.
Postcondición	Ninguna.
Errores	Ninguno.
Alternativas	El usuario puede desmarcar una serie o película utilizando el mismo botón.

Tabla 3.12: RF12 - Recomendaciones en el inicio

Id	RF-12
Nombre	Recomendaciones en el inicio.
Descripción	El usuario puede recibir recomendaciones a la hora de abrir el inicio.
Proceso	1. Abre la aplicación.
	2. El sistema le ofrecerá un conjunto de series y películas recomendadas por los géneros de las películas o series ya marcadas como vistas.
Precondición	El usuario tiene que estar autenticado. Tiene que tener alguna serie o película marcada como vista.
Postcondición	Ninguna.
Errores	Ninguno.

Tabla 3.13: RF13 - Recomendaciones aleatorias

Id	RF-13
Nombre	Recomendaciones aleatorias.
Descripción	El usuario puede recibir un conjunto de recomendaciones aleatorias.
Proceso	1. El usuario presiona sobre el botón arriba a la derecha con forma de dos flechas entrelazadas.
	2. El sistema le devuelve un conjunto de películas y series ordenadas según los gustos del usuario.
Precondición	El usuario tiene que estar autenticado. El usuario debe de haber guardado como vista o como pendiente una serie o película.
Postcondición	Ninguna.
Errores	Ninguno.

Tabla 3.14: RF14 - Seguir serie

Id	RF-14
Nombre	Seguir serie.
Descripción	El usuario puede marcar para seguir la serie y guardársela como pendiente.
Proceso	1. El usuario presiona sobre una serie para abrir toda la información sobre ella.
	2. El usuario presiona sobre el botón con forma de marcador para seguir la serie.
	3. El sistema guarda la información de la serie marcada.
Precondición	El usuario tiene que estar autenticado. Tiene que ser una serie que no haya sido marcada antes.
Postcondición	Ninguna.
Errores	Ninguno.
Alternativas	El usuario puede desmarcar la opción de seguir una serie utilizando de la misma forma el botón.

Tabla 3.15: RF15 - Filtrar el contenido por suscripciones

Id	RF-15
Nombre	Filtrar el contenido por suscripciones.
Descripción	El usuario puede filtrar las películas o series que le aparezcan en toda la aplicación por las plataformas en las que tiene una suscripción.
Proceso	1. El usuario presiona el botón de ajustes de la barra de navegación.
	2. El usuario marca como activo la opción de "Filtrar películas o series por las suscripciones".
	3. El sistema cada vez que muestre una película o serie comprobará si pertenece a las plataformas en las que el usuario está suscrito.
Precondición	El usuario tiene que estar autenticado.
Postcondición	Ninguna.
Errores	Puede haber categorías en la pantalla de inicio en las que no se muestre nada porque no habrá ninguna serie o película en esa categoría que cumpla la condición.

Tabla 3.16: RF16 - Marcar o desmarcar como pendiente

Id	RF-16
Nombre	Marcar o desmarcar como pendiente.
Descripción	El usuario puede marcar como pendiente una película para verla más tarde.
Proceso	1. El usuario presiona sobre una película.
	2. El usuario presiona sobre el botón con forma de marcador.
	3. El sistema guarda la información sobre la película marcada.
Precondición	El usuario tiene que estar registrado. El contenido clicado tiene que ser una película. El usuario no debe de haber marcado como pendiente esa misma película antes.
Postcondición	Ninguna. El usuario tiene que estar autenticado.
Errores	Ninguno.
Alternativa	El usuario si ya ha marcado como pendiente una película puede desmarcarla utilizando de la misma forma el botón.

Tabla 3.17: RF17 - Crear perfiles para cada usuario

Id	RF-17
Nombre	Crear perfiles para cada usuario.
Descripción	El usuario puede crear perfiles para un mismo usuario para dividir las recomendaciones por personas dentro de una misma familia.
Proceso	1. El usuario presiona en el botón de ajustes en la barra de navegación.
	2. El usuario presiona en su perfil seleccionado.
	3. El usuario presiona en el botón con forma de un "+" y accede a la opción de crear un perfil.
	4. Introduce el nombre que quiere dar a ese perfil y le da al botón de "OK".
	5. El sistema guarda en ese usuario un nuevo perfil con datos por defecto.
Precondición	El usuario tiene que estar autenticado. El nombre del perfil tiene que ser uno distinto a los ya creados.
Postcondición	Ninguna.
Errores	Ninguno.

Tabla 3.18: RF18 - Ocultar películas o series vistas

Id	RF-18
Nombre	Ocultar películas o series vistas.
Descripción	El usuario puede ocultar las series o películas que ya ha marcado como vistas para que no le aparezca en ninguna parte de la aplicación, excepto en la vista de la biblioteca de películas y series marcadas como pendientes o vistas.
Proceso	1. El usuario presiona sobre el botón de ajustes en la barra de navegación.
	2. El usuario presiona sobre la opción de "Ocultar películas o series vistas".
	3. El sistema guarda el ajuste en el archivo del usuario.
Precondición	El usuario tiene que estar autenticado.
Postcondición	Ninguna.
Errores	Ninguno.

Tabla 3.19: RF19 - Filtrar películas seguidas

Id	RF-19
Nombre	Filtrar películas seguidas.
Descripción	El usuario puede escoger entre diferentes filtros que se aplican sobre la lista de películas seguidas.
Proceso	1. El usuario presiona sobre el botón de marcadores que se encuentra en la barra de navegación.
	2. El usuario presiona en el botón de filtros.
	3. El usuario elige los filtros que quiere aplicar a esta lista, además de aplicar el filtro de mostrar solo películas seguidas, y presiona el botón de "OK".
	4. El sistema les devuelve una lista de películas que cumplen los requisitos que se han especificado en los filtros.
Precondición	El usuario tiene que estar autenticado en la aplicación.
Postcondición	Ninguna.
Errores	Puede no aparecer ninguna película que cumpla los filtros.

Tabla 3.20: RF20 - Filtrar películas vistas

Id	RF-20
Nombre	Filtrar películas vistas.
Descripción	El usuario puede aplicar filtros a la lista de películas vistas.
Proceso	1. El usuario presiona sobre el botón de marcadores que se encuentra en la barra de navegación.
	2. El usuario presiona en el botón de filtros.
	3. El usuario elige los filtros que quiere aplicar a esta lista, además de aplicar el filtro de mostrar solo películas seguidas, y presiona el botón de "OK".
Precondición	El usuario tiene que estar autenticado en la aplicación.
Postcondición	Ninguna.
Errores	Puede no aparecer ninguna película que cumpla los filtros.

Tabla 3.21: RF21 - Cerrar sesión

Id	RF-21
Nombre	Cerrar sesión.
Descripción	El usuario puede cerrar su sesión.
Proceso	1. El usuario presiona en el botón de ajustes para abrir la pestaña de ajustes de la aplicación.
	2. Este pulsa el botón para controlar su usuario autenticado
	3. Este pulsa el botón de cerrar sesión.
	4. El sistema le envía a la pantalla para que se vuelva a iniciar sesión o se registre con otra cuenta.
Precondición	El usuario tiene que estar autenticado en la aplicación.
Postcondición	Ninguna.
Errores	Ninguno.

3.2 Requisitos no funcionales

Los requisitos no funcionales [13] son las necesidades que debe cumplir el sistema para satisfacer las necesidades del usuario. En estos destacamos la seguridad, portabilidad, la interfaz y usabilidad, el rendimiento y la autenticación.

Tabla 3.22: Requisitos no funcionales

Seguridad	La aplicación deberá encriptar todos los datos que se comunican a servidores externos.
Portabilidad	El sistema tiene que estar disponible en cualquier dispositivo Android, tanto móviles, tabletas y Android TV. No será necesario disponer de conexión siempre ya que se guardarán cuando sea posible las respuestas en el dispositivo. La versión mínima del SDK de Android debería ser la versión 26 (disponible en el 90,7 % de los dispositivos).
Interfaz y usabilidad	La aplicación debe constar de una interfaz intuitiva y atractiva, que motive el aprendizaje de las principales funcionalidades que puede utilizar el usuario. Esta deberá ser fácil de descargar e instalar.
Rendimiento	La aplicación deberá tener tiempos de respuesta como máximo de dos segundos, a no ser que se muestre algún tipo de indicador de progreso.
Autenticación	La aplicación debe garantizar que el usuario pueda registrarse con un correo electrónico único en la base de datos, que será identificado por un identificador único generado por los servicios de autenticación de Firebase.

CAPÍTULO 4

Diseño de la solución

En esta sección se explicará detalladamente como se ha desarrollado la aplicación y se mostrarán los primeros bocetos de la aplicación.

4.1 Arquitectura MVVM

Se ha decidido seguir el patrón de diseño de Model-View -ViewModel (MVVM). Esta arquitectura es un patrón de diseño utilizado comúnmente en el desarrollo de aplicaciones Android. Proporciona una separación clara de las funciones de los componentes de la aplicación, lo que facilita el mantenimiento, la prueba y la escalabilidad del código.

En la arquitectura MVVM, los componentes principales son:

- **Modelo (Model):** Representa los datos de la aplicación. Esto puede incluir el acceso a la base de datos, llamadas a API, manipulación de datos y reglas de negocio. El modelo no tiene conocimiento de la interfaz de usuario y no depende de ella. En esta capa se encontrarán los datos del usuario, los datos obtenidos de la API... Esta capa es la responsable de exponer los datos al resto de la aplicación. En Android, las bibliotecas que brindan soporte para esta capa son OkHttp (para realizar llamadas a API REST) y Firebase (para el almacenamiento en la nube y la autenticación).
- **Vista (View):** Es la capa de presentación de la aplicación y se encarga de mostrar los datos al usuario y de capturar las interacciones del usuario. En estas se encuentran las clases actividades (Activity) y los fragmentos (Fragment). La vista se mantiene lo más libre posible de la lógica de negocio, y su principal objetivo es la presentación visual.
- **VistaModelo (ViewModel):** Actúa como un intermediario entre el modelo y la vista. El ViewModel se encarga de obtener los datos necesarios del modelo y prepararlos para su presentación en la vista, por ejemplo, en las listas RecyclerView. También maneja las interacciones del usuario y actualiza los datos en el modelo en función de esas interacciones.

4.2 Arquitectura del sistema

En esta sección, exploraremos la arquitectura del sistema de nuestra aplicación y cómo los diversos componentes se conectan para proporcionar funcionalidades clave. Durante el desarrollo, hemos utilizado una combinación de componentes nativos de Andro-

id y bibliotecas de terceros que nos brindan funcionalidades adicionales y simplifican la implementación de características complejas.

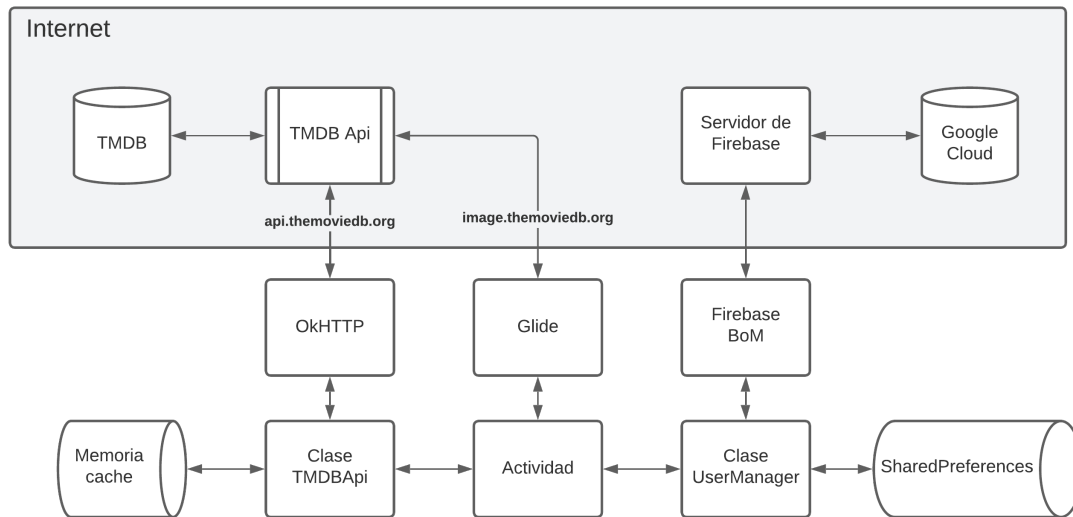


Figura 4.1: Diagrama con la arquitectura del sistema

En la figura 4.1, se muestra que las clases que van a utilizar las bibliotecas externas son la clase de `TMDBApi` y `UserManager`. La clase de `TMDBApi` utiliza la biblioteca de `OkHttp` para realizar las llamadas a la API de `TMDB` guardando la respuesta en la memoria cache de la aplicación. Sin embargo, a esta API también se le harán solicitudes desde los componentes `Glide` para descargar los pósteres de las películas y las series de la aplicación.

Por otra parte, la clase de `UserManager` se dedicará a realizar las comunicaciones con los servidores de `Firebase` con las herramientas de `Firebase` importadas en la aplicación. Estos son `Firebase Auth`, `Firebase Performance`, `Firebase Crashlytics` y `Firebase Storage`. Mientras que los datos descargados del usuario, cuando se descarguen desde `Firebase`, se guardarán en local utilizando las `SharedPreferences`.

Las `SharedPreferences` son un mecanismo de almacenamiento ligero y persistente que se utiliza para almacenar pequeñas cantidades de datos en forma de pares clave-valor. Estas preferencias compartidas se almacenan en un archivo XML en el almacenamiento interno del dispositivo y se pueden acceder y modificar fácilmente en cualquier momento durante la ejecución de la aplicación.

4.3 Arquitectura del directorio

Esta aplicación se ha desarrollado en `Android Studio` utilizando la SDK de `Android` nativa. La estructura del directorio es la que ofrece `Android Studio` para sus proyectos:

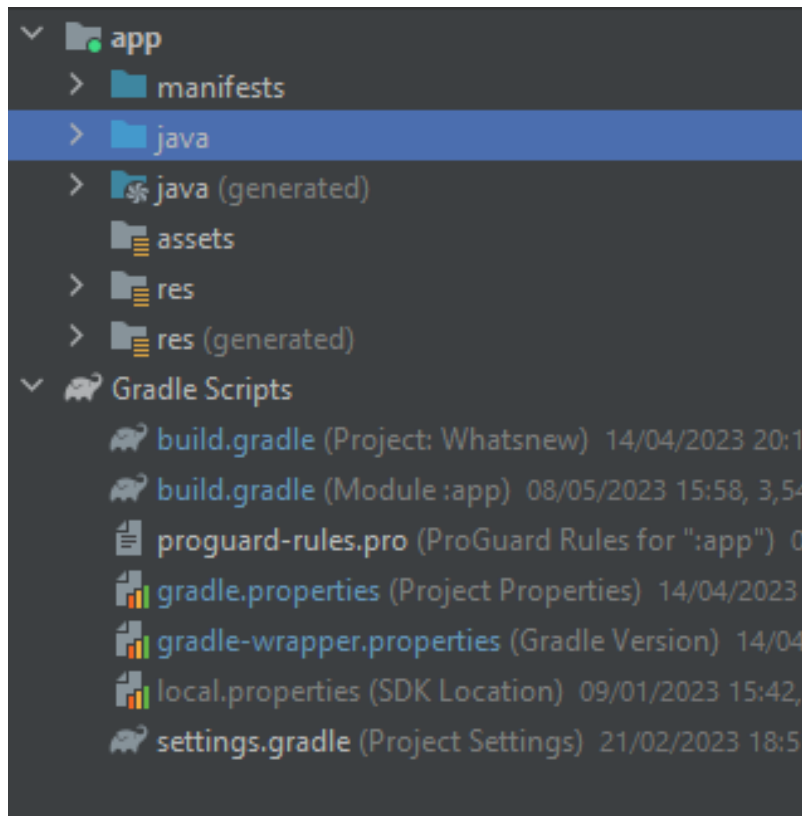


Figura 4.2: Directorio de la aplicación

En la figura 4.2 podemos ver los directorios del código fuente, en la que se encuentra toda la funcionalidad de la aplicación, en la carpeta “java” y los directorios de los recursos, en los que se encuentran las imágenes, textos y vistas que se muestran en la aplicación, que se encuentra en la carpeta “res”. Por otra parte, podemos observar que existe una sección que contiene todos los ejecutables y ficheros de configuración de Gradle que se utilizan para la compilación de la aplicación.

Antes de ver el desglose de las carpetas en el directorio primero tenemos que explicar tres elementos básicos en una aplicación realizada en Android nativo. Estos elementos son las actividades, los fragmentos y las vistas como RecyclerView, que posteriormente se mencionará.

4.3.1. Conceptos previos

Actividades

Las actividades son fundamentales en una aplicación de Android y todas estas se implementan como una subclase de Activity. En Android las actividades se inician mediante específicos que corresponden a su ciclo de vida [34]. En esta imagen 4.3, se muestra el ciclo de vida de una actividad. Se observa que cuando se inicia una actividad pasa por un conjunto de etapas: “onCreate”, “onStart” y “onResume”. Además, si se fuerza la detención de la actividad esta pasará por el estado “onDestroy”.

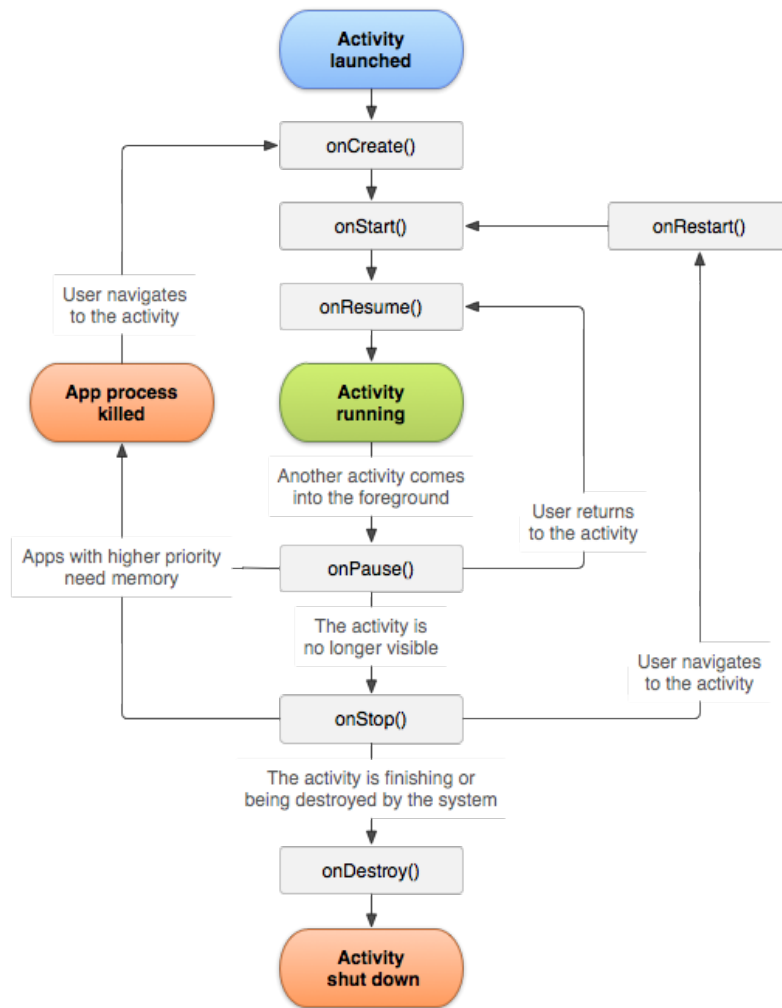


Figura 4.3: Ciclo de vida de una actividad

Una actividad en Android proporciona la ventana en la que se muestra la interfaz de usuario de la aplicación, normalmente, en la clase “onCreate”. Por lo general, una actividad representa una pantalla de la aplicación, como una pantalla de preferencias o una pantalla de selección de fotos. Las aplicaciones suelen tener múltiples actividades, y generalmente se especifica una actividad principal que aparece al iniciar la aplicación. Las actividades que se inician por la aplicación tienen que estar descritas en el archivo AndroidManifest en el que, también, se determina cual va a ser la actividad principal. A partir de la actividad principal, se pueden iniciar otras actividades para llevar a cabo diferentes acciones.

Fragmentos

Un Fragmento [35] en Android es una parte modular de la interfaz de usuario que se puede combinar con otros fragmentos en una actividad para crear una interfaz con varias vistas y reutilizarse en diferentes actividades. Cada fragmento tiene su propio ciclo de vida, eventos de entrada y puede ser agregado o eliminado mientras la actividad está en ejecución. El ciclo de vida del fragmento está directamente vinculado al ciclo de vida de la actividad anfitriona: cuando la actividad se pausa o destruye, los fragmentos también lo hacen. Sin embargo, mientras la actividad está en estado reanudado, los fragmentos pueden manipularse de forma independiente.

Las transacciones de fragmentos permiten agregar, quitar y reemplazar fragmentos, y se registran en una pila de actividades administrada por la actividad. Esto permite al usuario deshacer una transacción de fragmentos al presionar el botón Atrás. Un fragmento se inserta en la jerarquía de vistas de la actividad, dentro de un ViewGroup, y define su propio diseño de vistas.

RecyclerView

Un RecyclerView [36] en Android es un componente de interfaz de usuario que muestra una lista de elementos en forma de vista desplazable. Es una versión mejorada y más flexible de ListView, que permite mostrar grandes conjuntos de datos de manera eficiente. El RecyclerView utiliza un adaptador para proporcionar los datos y una clase de diseño para definir cómo se muestra cada elemento en la lista.

A diferencia de ListView, el RecyclerView no tiene una implementación predeterminada para la disposición de los elementos en la lista. En su lugar, utiliza un administrador de diseño (LayoutManager) que controla la posición y el diseño de los elementos. Esto permite diferentes estilos de diseño, como una lista vertical, en una lista horizontal o en una cuadrícula.

El RecyclerView también es altamente optimizado en términos de rendimiento y eficiencia. Utiliza el concepto de vista reutilizable, lo que significa que solo se crean y se mantienen en memoria las vistas visibles actualmente en la pantalla. A medida que el usuario desplaza la lista, las vistas que salen de la pantalla se reciclan y se utilizan para mostrar nuevos elementos que entran en la vista.

Otra característica importante del RecyclerView es la capacidad de realizar animaciones de manera fluida al agregar, eliminar o actualizar elementos en la lista. Esto proporciona una experiencia de usuario más atractiva y dinámica.

4.3.2. Carpeta java

En esta figura 4.4 se observa que se ha categorizado las clases de Java en paquetes como: actividades, eventos, fragmentos, modelos, herramientas (“utils”) y vistas (“views”). Vamos a detallar los paquetes de Java más relevantes.

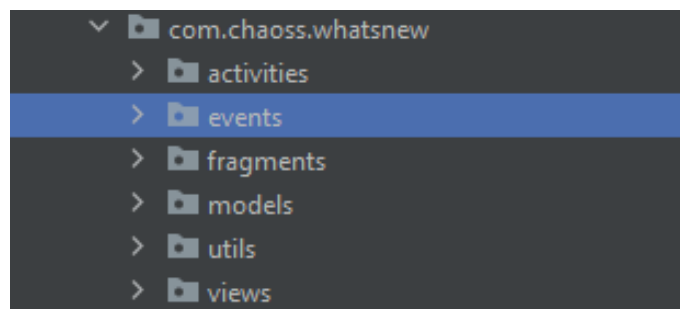


Figura 4.4: Paquete de la aplicación

Paquete “activities” (figura 4.5): en este se encuentran todas las actividades de la aplicación. En esta encontramos las más importantes como: las de inicio de sesión (“SignInActivity”, “SignUpActivity”, “WelcomeActivity”, “ForgotPasswordActivity”), la actividad principal que muestra los fragmentos y maneja el inicio de todas las actividades de esta carpeta, la actividad de “MultimediaViewActivity” en la que se muestra la información de las películas y series, la de PersonActivity y SeasonActivity que son clones a la

anterior pero mostrando información de los actores y de las temporadas respectivamente, y la de “RandomActivity” en la que se muestran recomendaciones a pantalla completa para que el usuario pueda recibir recomendaciones mucho más rapido.

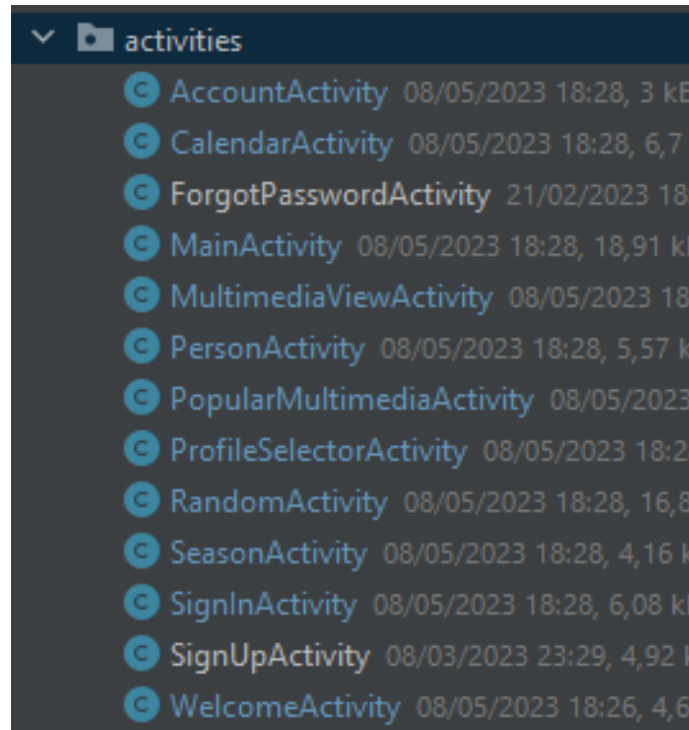


Figura 4.5: Paquete de actividades de la aplicación

Paquete “fragments” (figura 4.6): que se encuentran todos los fragmentos que se inician desde la actividad de “MainActivity”. En este encontramos los fragmentos de: el “ArchiveFragment”, donde se encuentra todas las películas y series guardadas por el usuario, el “HomeFragment”, que es la primera vista que tiene el usuario de la aplicación con sus recomendaciones, el “SearchFragment”, donde se realiza la búsqueda de películas o series y “SettingsFragment”, donde se muestran todos los ajustes del usuario.

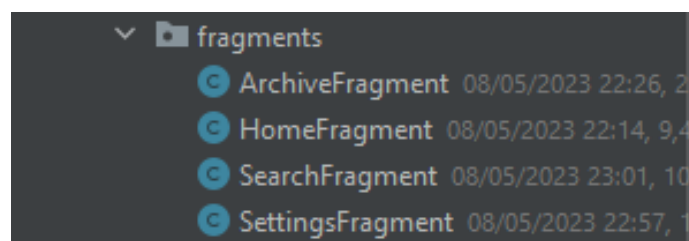


Figura 4.6: Paquete de fragmentos de la aplicación

El paquete “models” (figura 4.7): que tiene todo el conjunto de objetos que ayudan al manejo de información sobre películas, series, actores, perfiles y usuarios de la aplicación.

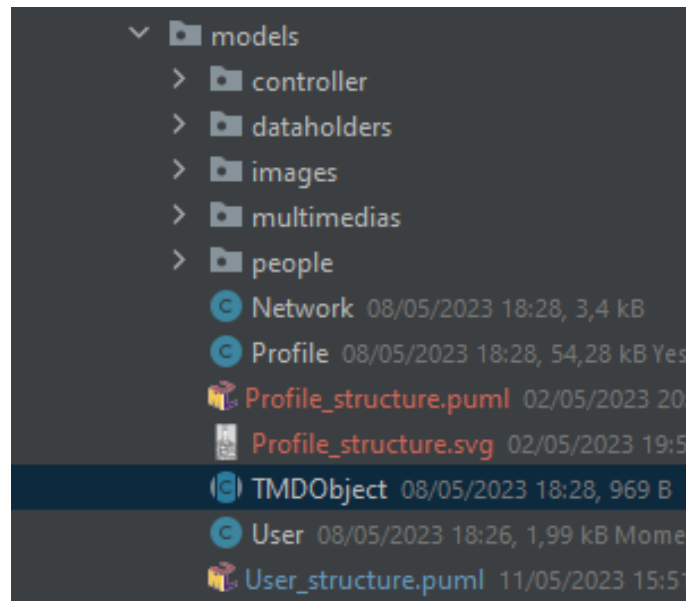


Figura 4.7: Paquete de datos de la aplicación

El paquete “views” (figura 4.8): que contiene todas las vistas personalizadas, como botones personalizados, y todos los controladores del objeto “RecyclerView” que se utiliza para mostrar listas de actores, películas, series o temporadas tanto en vertical como en horizontal.

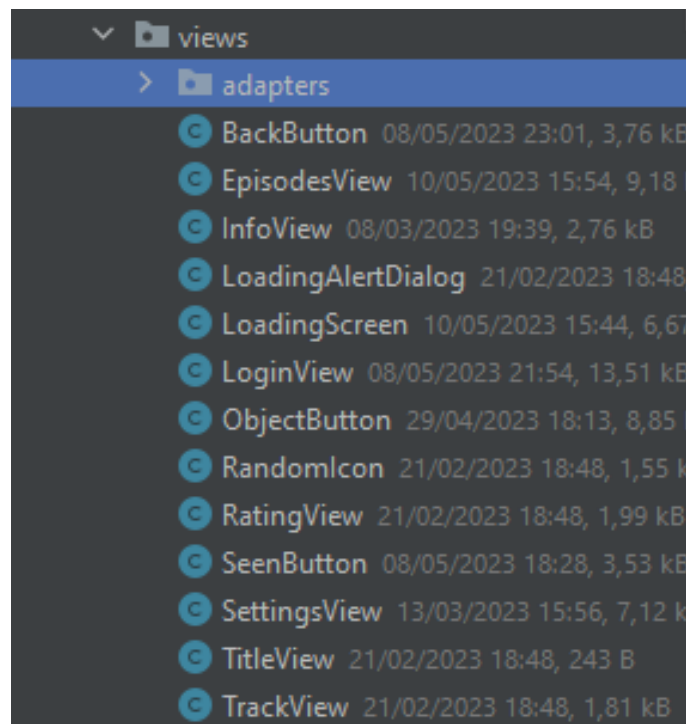


Figura 4.8: Paquete de vistas personalizadas de la aplicación

4.3.3. Carpeta res

En esta se encuentran todos los recursos utilizados por la aplicación. Estos se encuentran categorizados en carpetas dependiendo de donde van a ser utilizados esos recursos:

anim”, para las animaciones, “drawable”, para las imágenes o fondos, “Font”, para las fuentes del texto, “layout”, que contiene todas las vistas y pestañas de la aplicación, “mipmap”, para el logo de la aplicación, “values”, para definir valores estándar para utilizarlo en varias partes de la aplicación. Esta carpeta es la mostrada en la figura 4.9.

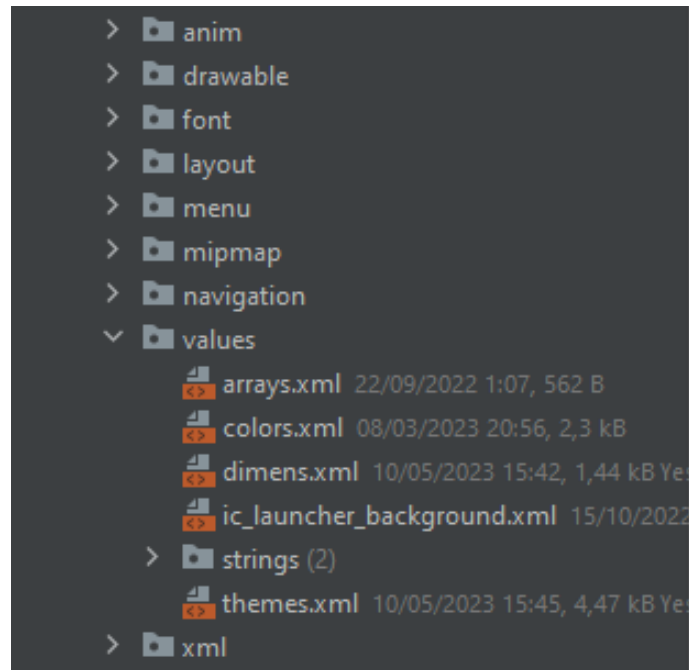


Figura 4.9: Directorio de recursos de la aplicación

4.4 Diagrama UML

En esta sección, presentaremos el diagrama de clases definido mediante el lenguaje UML (Unified Modeling Language) [14] de aplicación móvil presentada. Este diagrama es una herramienta útil para representar visualmente la arquitectura del código y la implementación propuesta de sistemas de software complejos.

A través de este diagrama, podemos identificar las clases, interfaces y relaciones que conforman nuestra aplicación móvil basada en el lenguaje de programación de Java. Además, nos permite entender cómo los distintos componentes de la aplicación interactúan entre sí.

Este es el diagrama UML de la aplicación:

En este podemos ver el “UserManager” es el objeto principal de la base de datos. Este es el que maneja el guardado de los datos en la base de datos, tanto local como en la nube, del usuario que está registrado en la aplicación. Este debe tener un objeto “User” del que guardar los datos.

Por otro lado, el objeto “User” deberá tener al menos un perfil, el cual será el perfil “Familia” que será un perfil por defecto para cada usuario. El usuario podrá tener un conjunto de perfiles, donde el usuario guarda toda la información por separado sobre las películas o series que le gustan, que tiene pendientes de ver, como su historial de búsquedas, la lista de plataformas de las que tiene una suscripción, el idioma por defecto del perfil, la región, y los diferentes ajustes de la aplicación.

Este objeto de Perfil (“Profile”) tiene objetos como:

- Historial, o “History”: que tiene la fecha de la búsqueda, la miniatura del objeto seleccionado y la búsqueda.
- Multimedia, que es un objeto del que extiende las películas y las serie, que contiene toda la información de la serie o la película obtenida mediante la API de TheMovieDB.

4.5 Bocetos

En este apartado introduciremos los bocetos de las interfaces que se encuentran a lo largo de la aplicación planteada.

4.5.1. Inicio

En la figura 4.11 se puede ver la primera pantalla que aparecerá al iniciar la aplicación, ya que el usuario debe tener al principio acceso a todas las funcionalidades de la aplicación para un uso ágil de esta. Desde esta vista el usuario podrá acceder a la vista que mostrará más información de la aplicación haciendo clic en la película o serie que quiera ver, como podemos observar en la figura 4.12.

Por otra parte se podrá acceder a la vista donde se le ofrece una recomendación rápida, únicamente si el usuario se encuentra registrado en la aplicación. Desde esta vista también se podrá acceder a las funciones de búsqueda, ajustes y a la vista donde se encuentran las películas o series marcadas para ver más tarde.

Además, en esta se puede observar que hay una sección para mostrar las películas o series que se encuentran disponibles en la plataforma que elijas.

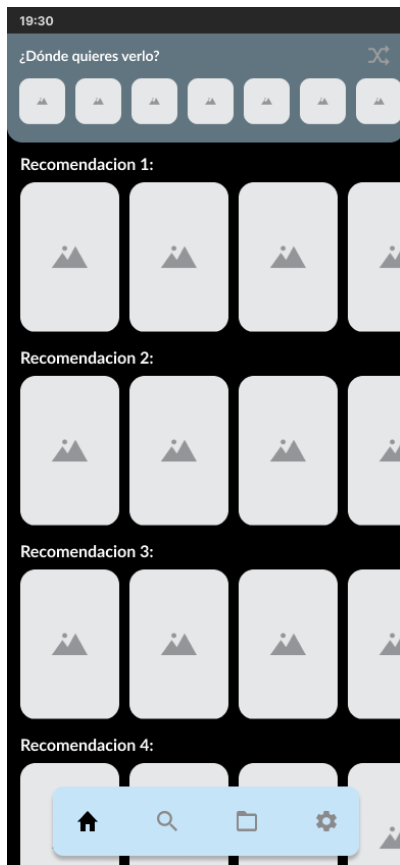


Figura 4.11: Boceto inicio



Figura 4.12: Boceto inicio con vista desplegada

En esta última figura, podemos observar que muestra más información de la película o serie escogida por el usuario, además de darnos la opción de marcarla para ver más tarde. No obstante, tenemos la opción de obtener más información sobre esta.

4.5.2. Búsqueda

En la pestaña de búsqueda, lo principal será el historial de las búsquedas ya realizadas, como podemos observar en la figura 4.13. Además, una vez que el usuario introduzca caracteres en el buscador, se irá mostrando por pantalla en la misma lista que el historial, palabras que suelen buscar otros usuarios con los mismos caracteres que se han escrito en el buscador.

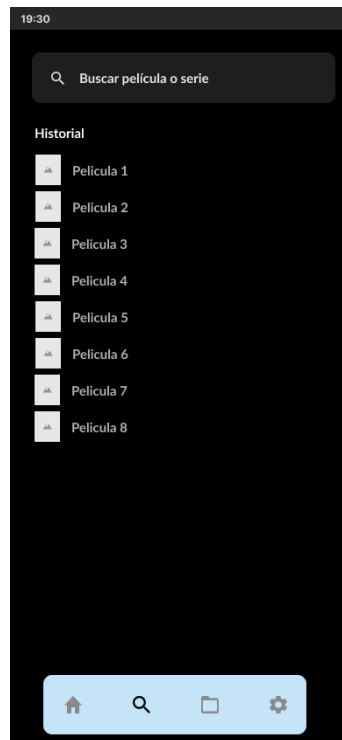


Figura 4.13: Boceto de la interfaz de búsqueda

Finalmente, los resultados de la búsqueda se mostrarán de la forma mostrada en la figura 4.14.

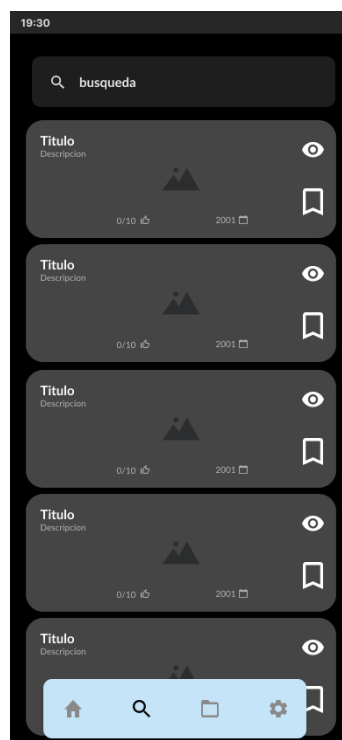


Figura 4.14: Boceto de la interfaz de los resultados de la búsqueda

4.5.3. Archivo

La pestaña Archivo es donde se guardan todas las series o películas que han sido marcadas para ver más tarde por el usuario. Esta solamente es accesible si el usuario se encuentra registrado. En la figura 4.15 se muestra la vista, utilizando un menú que categoriza este contenido en películas, series y películas que se estrenarán próximamente. Este, además se puede filtrar y ordenar según las preferencias del usuario.

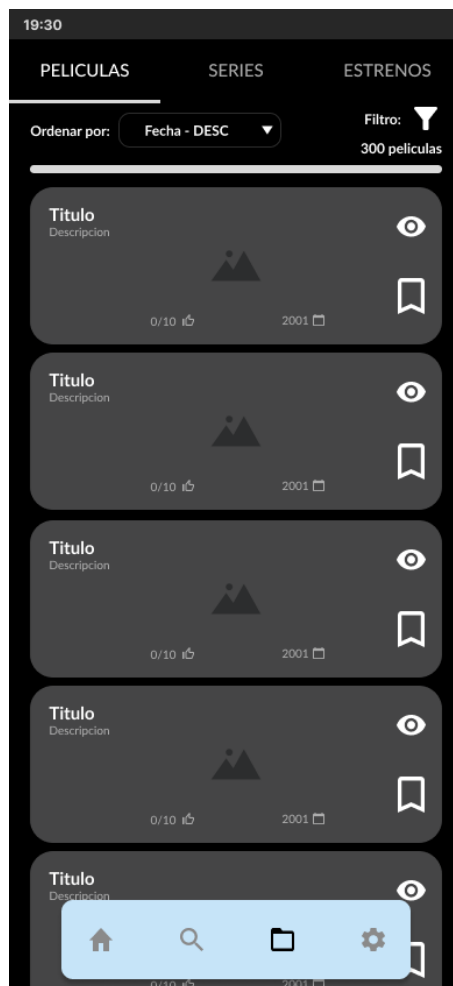


Figura 4.15: Boceto de la interfaz de las listas de marcados

4.5.4. Ajustes

La pestaña de los ajustes de la aplicación, que se muestra en la figura 4.16, permite al usuario además de personalizar su experiencia con la aplicación, cambiar de perfil en un mismo usuario.

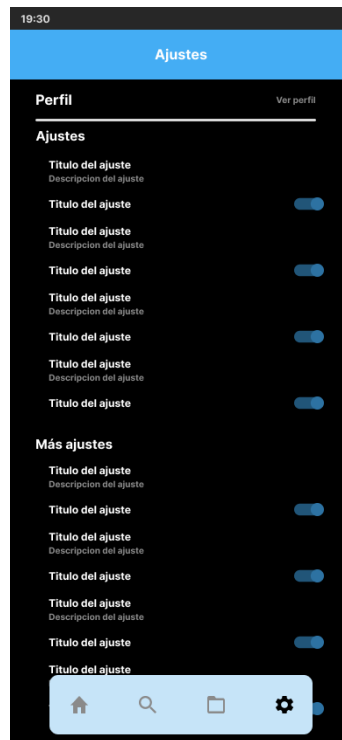


Figura 4.16: Boceto de la interfaz de los ajustes

4.5.5. Recomendación aleatoria

En la vista de las recomendaciones aleatorias, figura 4.17, encontramos varios botones repartidos por la pantalla. Esta vista deberá ocupar toda la pantalla y mostrará el póster de una película o serie que sea recomendada por el sistema.

Está además mostrará la plataforma en la que se puede ver esta película o serie en la parte inferior de la pantalla. Esa vista tendrá controles básicos para hacer que el usuario pueda observar y entrenar al sistema de recomendaciones lo más rápido posible.

En esta vista se podrá, marcar como me gusta, marcar como no me gusta, ver más información y marcar como vista una película o serie utilizando los botones que se encuentran en el centro de la pantalla. Por otro lado, en la parte superior, encontramos botones para filtrar las recomendaciones, salir de la pestaña y la probabilidad de que le guste al usuario la película o serie recomendada por el sistema. Este último se obtiene mediante un algoritmo que se explicará posteriormente.



Figura 4.17: Boceto de la interfaz de las recomendaciones aleatorias

4.5.6. Película o series

En esta vista, figura 4.18 y figura 4.19, podemos observar que se muestra toda la información de una película o serie. Está muestra la descripción, el género, la fecha de estreno, la duración, la valoración, así como, la lista del reparto, una lista de películas o series similares.

También se muestran los botones en la parte derecha del título de la película o serie que sirven para marcar como vista y como pendiente una película o marcar una serie para seguirla. En la figura 4.18, podemos ver la vista que se muestra cuando es la información de una película y, en la figura 4.20, cuando la información es de una serie. En la vista de las series se puede observar que existe una gráfica que indica la valoración de cada episodio en una temporada que elija el usuario ver.

Por otro lado, en la figura 4.19, se muestra la parte de debajo de la figura 4.18, con las películas o series similares y el reparto de esta.

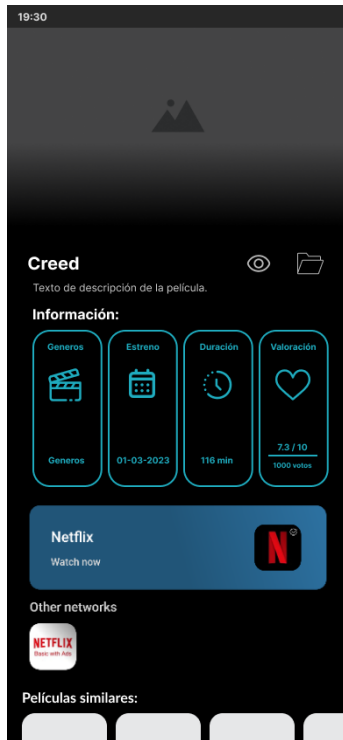


Figura 4.18: Boceto de la interfaz de las películas y series

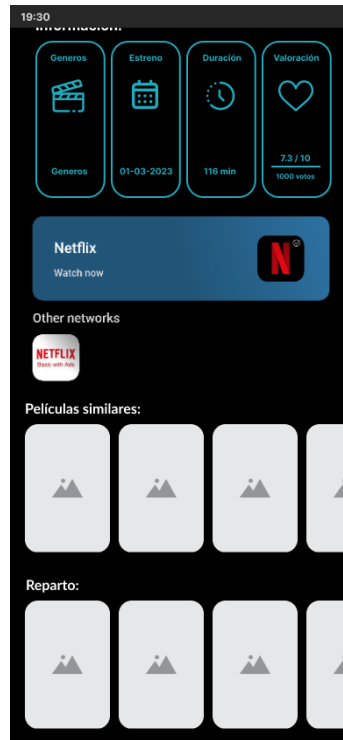


Figura 4.19: Resto de la interfaz de las películas y series



Figura 4.20: Boceto de la interfaz de las series

4.6 Diagrama de flujo entre bocetos

En la figura 4.21 se muestra el diagrama de flujo entre interfaces para mostrar más en detalle la interacción del usuario con la interfaz propuesta.

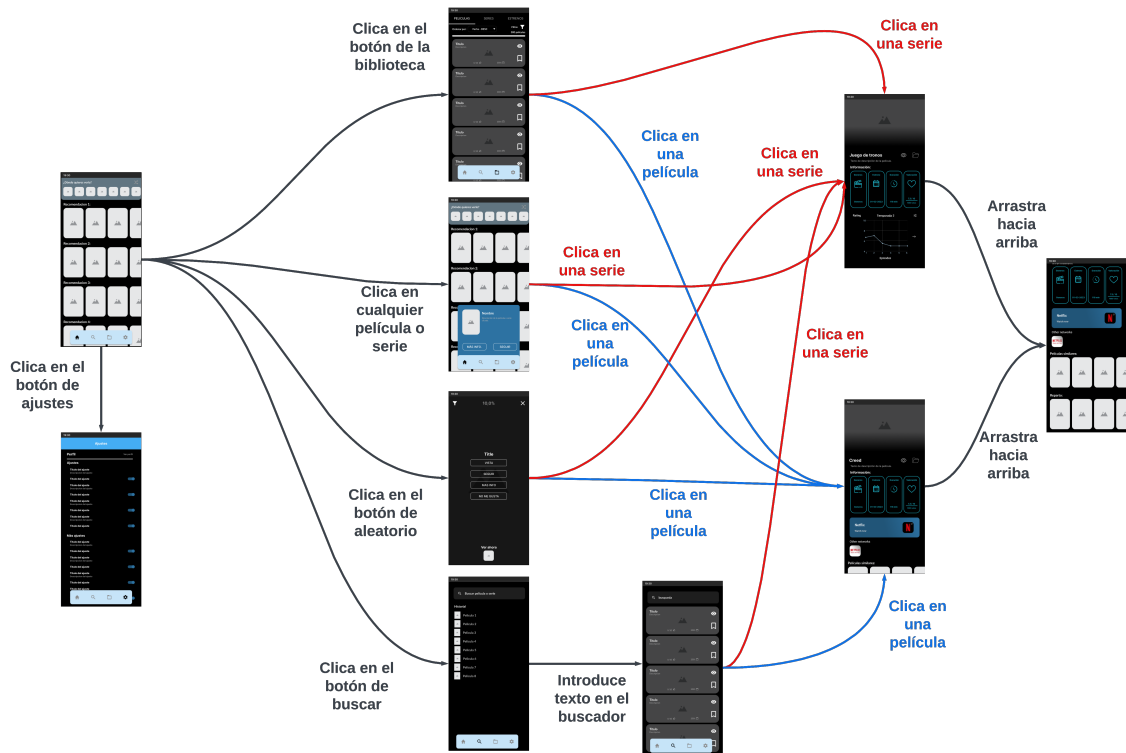


Figura 4.21: Diagrama de flujo entre los bocetos

CAPÍTULO 5

Tecnologías utilizadas

5.1 Java

El lenguaje utilizado para la programación de la aplicación es Java [15], cuyo logo es el de la figura 5.1. Es un lenguaje de programación de alto nivel que se utiliza para desarrollar una amplia variedad de aplicaciones. Fue creado en 1991 por James Gosling, Patrick Naughton en Sun Microsystems (adquirida posteriormente por Oracle Corporation). La primera versión de Java, llamada Oak, fue lanzada en 1995 como parte de la estrategia de Sun para tener una plataforma de software que pudiera ejecutarse en cualquier dispositivo, independientemente del sistema operativo y la arquitectura de hardware.

Java se utiliza principalmente para desarrollar aplicaciones de escritorio, aplicaciones móviles, aplicaciones web, servidores de aplicaciones, aplicaciones empresariales, juegos, herramientas de desarrollo, y mucho más. Una de las principales características de Java es su portabilidad, lo que significa que el código escrito en Java se puede ejecutar en cualquier plataforma que tenga una máquina virtual Java (JVM) instalada, lo que permite que el mismo programa se ejecute en diferentes sistemas operativos y hardware sin cambios como, por ejemplo, Android.



Figura 5.1: Logo de Java

5.2 Google Play Console

Google Play Console, el cual su logo aparece en la figura 5.2, es una plataforma proporcionada por Google para que los desarrolladores puedan publicar y distribuir sus aplicaciones móviles a través de la tienda de aplicaciones de Google Play. Permite a los desarrolladores gestionar sus aplicaciones, publicar actualizaciones, ver estadísticas de descargas y ganancias, y recibir comentarios y calificaciones de los usuarios. En la plata-

forma también podrás obtener información sobre cómo se está utilizando tu aplicación, cómo está siendo valorada por los usuarios y cómo se está vendiendo.



Figura 5.2: Logo de Google Play Console

5.3 Firebase

Firebase [16] es una plataforma de procesamiento en la nube proporcionada por Google. Firebase, cuyo logo aparece en 5.3, ofrece una amplia gama de herramientas y servicios para ayudar a los desarrolladores a crear aplicaciones móviles de alta calidad, desde autenticación de usuarios y almacenamiento en la nube hasta análisis de aplicaciones y mensajería en tiempo real. Firebase también ofrece una amplia documentación y soporte para ayudar a los desarrolladores a implementar sus aplicaciones de manera efectiva.



Figura 5.3: Logo de Firebase

5.4 Github

GitHub, cuyo logo es 5.4, [17] es una plataforma de alojamiento de código fuente basada en la nube que permite a los desarrolladores colaborar en proyectos y mantener un control de versiones del código fuente. GitHub es muy popular entre los desarrolladores debido a su facilidad de uso y amplia gama de herramientas para la colaboración, como solicitudes de extracción, problemas y discusiones. GitHub también se integra con una amplia gama de herramientas de desarrollo de software.



Figura 5.4: Logo de GitHub

5.5 Lucidchart

Lucidchart [18] es una herramienta en línea de diagramación y visualización de datos. Permite a los usuarios crear diagramas de flujo, diagramas de red, diagramas de UML, diagramas de base de datos y muchos otros tipos de diagramas. Lucidchart es muy popular entre los desarrolladores de software debido a su facilidad de uso y la capacidad de colaborar en tiempo real con otros usuarios. Su logo es el siguiente 5.5:



Figura 5.5: Logo de Lucidchart

5.6 Figma

Figma [19], cuyo logo es el de la figura 5.6, es una herramienta de prototipado de interfaz de usuario en línea que permite a los diseñadores crear diseños de alta calidad para aplicaciones móviles y web. También ofrece una amplia gama de herramientas para el diseño de interfaces de usuario, como la creación de componentes reutilizables y la colaboración en tiempo real. Además, se integra con otras herramientas de diseño y desarrollo de software.



Figura 5.6: Logo de Figma

5.7 Android Studio

Android Studio [20], con logo que aparece en la figura 5.7, es un entorno de desarrollo integrado (IDE) para el desarrollo de aplicaciones móviles de Android. Android Studio ofrece una amplia gama de herramientas y servicios para ayudar a los desarrolladores a crear aplicaciones de alta calidad, desde la edición de código y la depuración hasta la creación de interfaces de usuario y la compilación de aplicaciones. Podrás integrar fácilmente otras herramientas, como Firebase y GitHub, en Android Studio para mejorar tu flujo de trabajo.



Figura 5.7: Logo de Android Studio

CAPÍTULO 6

Implementación de la solución

Ahora se describirá la solución a la que se ha alcanzado, desde el diseño de la solución hasta la solución terminada. Describiendo los problemas que han ocurrido durante el desarrollo y las decisiones que se han tomado para resolverlos.

6.1 Bibliotecas utilizadas

En este apartado recogeremos las bibliotecas utilizadas para facilitar la creación y desarrollo de esta aplicación. El manejo de estas dependencias se hace a través de la herramienta de Gradle [21] que permite la compilación de aplicaciones en Android, así como sus bibliotecas.

6.1.1. OkHTTP

OkHTTP [22] es una herramienta de Java que permite la llamada de peticiones HTTP. Concretamente, en la aplicación, se ha decidido usar la versión "5.0.0" debido a que es la versión más reciente.

Todas las llamadas HTTP que realiza la aplicación utilizando esta biblioteca son peticiones GET, es decir, solo obtienen datos de un servidor externo. Estas llamadas se encuentran únicamente en una misma clase "TMDApi" que maneja la comunicación con la API de TMDB. Por otro lado, la mayoría de comunicaciones se hacen de forma asíncrona, es decir, que las llamadas no tienen que esperar a que se obtenga el dato que piden a la API y así, permitiendo que la aplicación vaya cargando otras cosas esenciales mientras se reciben los datos.

```

549 public void getResultsAsync(@NonNull String url, @NonNull NetworkCacheCallback callback) {
550     if(responsesCached ==null) responsesCached = new ConcurrentHashMap<>();
551
552     if(url.contains("@") && url.contains("com.chaoss.lanflixnews")) throw new RuntimeException(url);
553
554     String filmNameCache = CommonMethods.getCacheFileName(url,API_KEY);
555     try {
556         if (ac != null && profile.isWantsToCache()) {
557             if (responsesCached.containsKey(url)) {
558                 callback.onResponse(responsesCached.get(url), url);
559                 return;
560             } else {
561                 File cacheDir = new File(ac.getCacheDir(), "requests");
562                 if (!cacheDir.exists()) {
563                     boolean mkdir = cacheDir.mkdir();
564                     CommonMethods.showLog( tag: "FileManager", message: "File " + cacheDir + " created: " + mkdir);
565                 }
566                 File filecache = new File(cacheDir, filmNameCache);
567                 if (filecache.exists()) {
568                     //si el archivo es demasiado viejo se actualiza
569                     if(!(cacheExpired(filecache) && filecache.delete())) {
570                         String res = CommonMethods.readCacheFile(filecache);
571                         if (!res.equals("")) {
572                             callback.onResponse(res, url);
573                             return;
574                         }
575                     }
576                 }
577             }
578         }
579     } catch (IOException ignore) {}
580
581     request(url, callback);
582 }
583
584

```

Figura 6.1: Fragmento de código con la llamada asíncrona a un servicio externo

En el método que muestra la figura 6.1, se observa donde se realizan las llamadas asíncronas (línea 581). Estas llamadas requieren el *link* de la API y el “callback”, que determina las operaciones que va a hacer al dato que se reciba de la llamada. Este “NetworkCacheCallback” es una ligera modificación al “callback” que ofrece la biblioteca, ya que es una clase que extiende de ella, pero ofrece la posibilidad de guardar en el sistema de archivos del dispositivo el dato de respuesta.

En el caso de que la respuesta ya se encuentre en la memoria, se ejecutará el método del “callback” donde se reciben los resultados de las peticiones, “onResponse()”, que se encuentra en la línea 572. En este método se realizarán todos los guardados en memoria para posteriores llamadas al mismo *link*.

6.1.2. Glide

Glide [23] es una biblioteca de código libre para el manejo y la carga de imágenes en Android. Incluye la posibilidad de descargar imágenes de internet, guardarlas en cache y cargarlas en las vistas de Android.

Por otra parte, como podemos ver en la figura 6.2 también se puede realizar una transformación a la imagen. En este caso, se ha querido dar a todos los elementos unas esquinas redondeadas para que el diseño llame más la atención.


```
Glide.with( activity, this).load(multimedia.getPoster_path(PosterQuality.HIGH))
    .transform(new RoundedCorners( roundingRadius: 20))
    .into(postersFilmOverview);
```

Figura 6.2: Código donde se obtiene una imagen para un póster

6.1.3. Firebase

Firebase [24] es un conjunto de herramientas desarrolladas por Google que ofrecen al desarrollador de una aplicación servicios que hagan la tarea de desarrollo más rápida.

Firestore Crashlytics

Esta herramienta de Google permite al desarrollador ver cuándo y dónde se han producido errores que han interrumpido la ejecución de la aplicación. También permite hacer un análisis de qué versiones y que móviles en particular han sufrido un error.

Se ejecuta internamente el envío del error a los servidores de Firebase cuando este detiene por completo la aplicación. Se pueden hacer envíos forzados de estos errores, pero esta funcionalidad no se ha utilizado en la aplicación. En la página web de Firebase podemos ver la siguiente interfaz, figura 6.3, que nos informa de los errores que se producen y en que dispositivos:

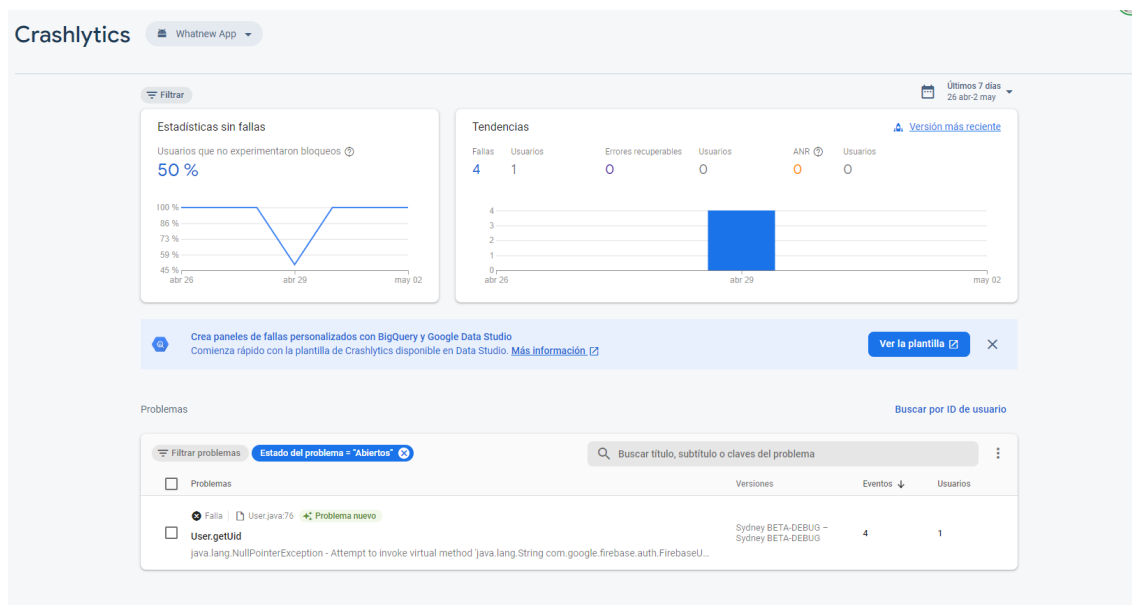


Figura 6.3: Página web de Firebase en el apartado Crashlytics

Firestore Performance

Esta herramienta permite realizar seguimiento de métricas ya predefinidas como el tiempo de inicio de la aplicación o personalizadas como el tiempo de descarga de un archivo con la información del usuario.

También se puede comparar con versiones que haya tenido la aplicación a lo largo del tiempo. Además, de monitorizar el tiempo de respuestas de las llamadas a servicios externos que se hacen dentro de la aplicación, como, por ejemplo, las llamadas a la API.

En la figura 6.4, podemos ver que la aplicación en la última versión se inicia un 73 % más lento que los 30 días de antes.



Figura 6.4: Página web de Firebase en el apartado Performance

Firestore Storage

Es una herramienta que permite guardar archivos determinados en la nube de Google. Toda la información del usuario se guarda en este servicio para que pueda ser transferido a cualquier dispositivo.

Esta funcionalidad está descrita en la clase "userManager" que mediante los métodos "downloadAllFromCloud", figura 6.5, y "updateOnline" se actualiza un JSON [31] formado por los datos, sin la información de registro, del usuario.

En la figura 6.5, en la línea 190, se puede observar que se asigna la referencia de la carpeta de la base de datos a donde se encuentra la información del usuario. Esta es una carpeta ya que dentro de esta carpeta se encuentran los ficheros JSON con la información separada de cada perfil. En la línea 193 se obtienen todos los ficheros guardados en el directorio de referencia y se convierten de JSON al objeto "Profile" (línea 206), guardando el resultado en la base de datos local (línea 207). Finalmente, este ejecuta el método que se ha pasado por parámetro como una función para ejecutar cuando la descarga finalice.

```

185 public void downloadAllFromCloud(@NonNull UserManagerCallback userManagerCallback){
186     long in = System.currentTimeMillis();
187     long MAX_BYTES = 5 * 1024*1024;
188     long start= System.currentTimeMillis();
189     FirebaseStorage firebaseStorage = FirebaseStorage.getInstance();
190     StorageReference userRef = firebaseStorage.getReference( location: "users/" + Objects.requireNonNull(user).getUid());
191     FirebasePerformance perf = new FirebasePerformance();
192     perf.init( tag: "download_user_files");
193     userRef.listAll().addOnSuccessListener(result -> {
194         CommonMethods.perf(start, tag: "Firebase Storage: GET LIST USERS");
195         AtomicInteger completed = new AtomicInteger( initialValue: 0);
196         if (result.getItems().size() > 0
197             && getCountAccountsuser(user, includeHOME: true) != result.getItems().size()
198         ) {
199             for (StorageReference fileRef : result.getItems()) {
200                 long downloadTime = System.currentTimeMillis();
201                 fileRef.getBytes(MAX_BYTES).addOnCompleteListener(task -> {
202                     boolean islast = completed.get() == result.getItems().size() - 1;
203                     String result1 = new String(task.getResult());
204                     if (result1.startsWith("{") && result1.endsWith("}") && result1.length() > 2) {
205                         try {
206                             Profile profile = Profile.convertJSONToUser(new JSONObject(result1));
207                             storeUserData(profile);
208                             //IF ITS LAST
209                             if (islast) {
210                                 CommonMethods.perf(in, tag: "Firebase Storage: Download Content");
211                                 userManagerCallback.onPostExecute();
212                                 perf.send();
213                             }
214                         } catch (JSONException | ServiceException e) {
215                             CommonMethods.showError(e, ac);
216                         }
217                     }
218                     int progress = completed.incrementAndGet();
219                     userManagerCallback.onStatusUpdate( progress: progress*100/result.getItems().size());
220                     CommonMethods.perf(downloadTime, tag: "Firebase Storage: Download time each");
221                 });
222             }
223         } else {...}
224     }).addOnFailureListener(e -> {...});

```

Figura 6.5: Fragmento del código de descarga de información de los usuarios

Mientras que la subida de los datos se realiza de una forma parecida a la descarga, creando una referencia del directorio donde se va a guardar los datos del usuario y subiendo los datos del usuario, pasando los datos del usuario a bytes.

Firestore Auth

Es una herramienta que facilita la creación de usuarios, así como, el inicio de sesión dentro de la aplicación. En la figura 6.6 se puede ver el inicio de sesión utilizando esta librería, usando el correo electrónico y contraseña que ha introducido el cliente.

```

mAuth.signInWithEmailAndPassword(Objects.requireNonNull(emailTextInput.getText()).toString(),
    Objects.requireNonNull(passwordTextInput.getText()).toString())
    .addOnCompleteListener( activity: SignInActivity.this,
        onCompleteListener: SignInActivity.this);

```

Figura 6.6: Fragmento del código que inicia sesión en Firestore Auth

Cabe destacar que esta herramienta guarda internamente el usuario cuando se inicia sesión o se registra en la aplicación. Este usuario de Firestore, únicamente sirve de autenticación para la descarga de información del usuario, guardado en el almacenamiento en la nube de Firestore Storage.

6.2 TheMovieDB API

Una API es una herramienta que permite que los servicios se comuniquen sin saber como están implementados. En este proyecto se ha utilizado la API, de la empresa de TheMovieDB [32], con la que se accede a la base de datos, con la información almacenada gracias a la colaboración de sus usuarios.

Ofrece su API mediante un registro y para utilizarla se requiere de una clave que debe ir integrada en cada una de sus llamadas. Para esta aplicación se utiliza la tercera versión de esta API. Las llamadas se realizan como antes se ha mencionado, aunque para obtener la URL a la que hacer la llamada HTTP se llaman a varios métodos que personalizan la URL con datos del usuario.

En la figura 6.7 podemos ver una llamada que obtiene una URL de las películas o series de los 90 según los datos de usuario: como su idioma predefinido y, en caso de que tenga alguno guardado, sus plataformas preferidas.

```

@NonNull
public String getTop90s(boolean isFilm){
    if(isFilm)
        return SERVER + "discover/movie?api_key=" + API_KEY+ "&language=" + profile.getLang()+
            "&sort_by=vote_average.desc&primary_release_date.lte=2000-12-31&vote_count.gte=1000&primary_release_date.gte=1989-12-31"+getWithWatchProvidersSaved();
    else
        return SERVER + "discover/tv?api_key=" + API_KEY+ "&language=" + profile.getLang()+
            "&sort_by=vote_average.desc&first_air_date.lte=2000-12-31&vote_count.gte=1000&first_air_date.gte=1989-12-31"+getWithWatchProvidersSaved();
}

```

Figura 6.7: Fragmento del código que devuelve la URL a la que hacer la petición

Todas las llamadas se realizan desde la misma clase, “TMDApi.java”, la cual también contiene las URL a las que se realizarán las peticiones. Estás vienen descritas en la documentación de la API [25] y mediante el uso de los perfiles del usuario registrado en la aplicación se pueden adquirir datos que se añaden a esa URL. Por ejemplo, en la figura 6.7, se puede observar que se utiliza el perfil del usuario para adquirir el lenguaje del sistema, pero también las plataformas de *streaming* a las que está suscrito ese perfil.

6.3 Utilización del cache

En el caso de esta aplicación, Whatsnew, se ha elegido no utilizar un servidor que haga de intermediario entre la API y el cliente. Debido a esta decisión la aplicación hace varias llamadas a la API, saturando la conexión del dispositivo móvil y excediendo el límite de solicitudes a la API asignado por TheMovieDB, cuyo limite son 50 peticiones por segundo [33].

Por ello, se ha optado por realizar varias funciones que guarden los datos de estas solicitudes a la API de una forma personalizada debido a que la información de una serie o película no suele cambiar durante un día.

En la función asíncrona, donde se realiza la llamada HTTP hemos podido observar dos tipos de caches utilizados. Un cache rápido (figura 6.8), que se almacena los datos en un diccionario de datos en los que la dirección URL es la clave para los datos, y un cache secundario (figura 6.9), que este se almacena en el sistema de archivos del dispositivo Android. La desventaja del cache más rápido es que al ser un cache que se guarda en la memoria estática de la aplicación, esta una vez que se cierre se borrará, mientras que la que se guarda en el sistema de archivos no se borrará hasta que pase un día como mínimo.

```
if (responsesCached.containsKey(url)) {
    callback.onResponse(responsesCached.get(url), url);
    return;
} else {
```

Figura 6.8: Fragmento del código que comprueba si se encuentra en el cache rápido

```
File cacheDir = new File(ac.getCacheDir(), child: "requests");
if (!cacheDir.exists()) {
    cacheDir.mkdir();
}
File filecache = new File(cacheDir, fileNameCache);
if (filecache.exists()) {
    //si el archivo es demasiado viejo se actualiza
    if (cacheExpired(filecache) && filecache.delete());
    else {
        String res = CommonMethods.readCacheFile(filecache);
        if (!res.equals("")) {
            callback.onResponse(res, url);
            return;
        }
    }
}
```

Figura 6.9: Fragmento del código que comprueba si se encuentra en el segundo cache

Por otra parte, la creación de estos registros del cache se realiza desde el método “callback” de la petición HTTP como podemos observar en la figura 6.10.

```
if(!TMDApi.responsesCached.containsKey(url)) TMDApi.responsesCached.put(url, res);

String filenameCache = CommonMethods.getCacheFileName(url, User.profileLogged.getAPI_KEY());
File cacheDir = new File(ac.getCacheDir(), child: "requests");
if (!cacheDir.exists()) {
    cacheDir.mkdir();
}
File filecache = new File(cacheDir, filenameCache);
if(!filecache.exists() || TMDApi.cacheExpired(filecache))
    CommonMethods.writeCacheFile(res, filecache, ac);
```

Figura 6.10: Fragmento del código que crea los registros en el cache rápido y en el secundario

Cabe destacar que la función “getCacheFileName”, devuelve un nombre de archivo normalizado según la URL de la petición.

6.4 Recomendaciones

Las recomendaciones de películas o series se basan en la utilización de la API de TheMovieDB, con la que se manda el identificador de una película o serie y la API devuelve una lista de películas o series recomendadas por su sistema. La URL con la que hacemos la petición HTTP(GET) la podemos obtener de la documentación de su API [26].

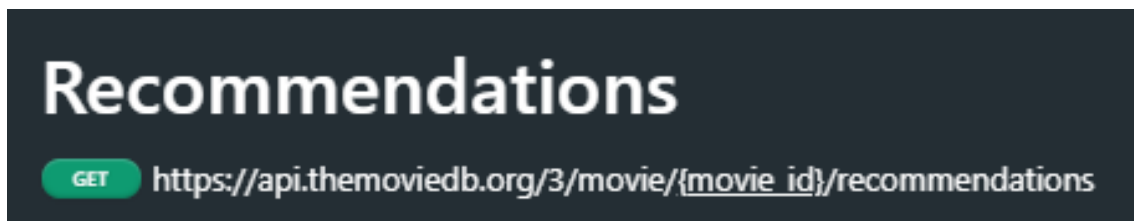


Figura 6.11: URL de la API para las recomendaciones

6.4.1. Recomendaciones en el inicio

Estas recomendaciones se realizan utilizando la clase “HomeRecomendaciones” en la cual, las listas de películas o series que se muestran en el inicio de la aplicación son mostradas mediante un RecyclerView, el cual muestra vistas progresivamente, en forma de lista, a medida que el usuario va arrastrando para mostrar más listas.

Se ha elegido mostrar este tipo de listas utilizando un RecyclerView y no un ScrollView debido a que este último carga de golpe todas las vistas que tiene como hijas, siendo estas imágenes. Esto provocaría una alta carga de procesador del dispositivo ya que tendría que cargar un alto número de imágenes a la vez. Por eso se ha preferido el uso del “lazy-loading” o carga progresiva [27], para cargar la alta cantidad de imágenes que debe cargar al inicio de la aplicación.

Esto se ve reflejado en la figura 6.12 de la implementación del adaptador personalizado del RecyclerView. En este adaptador se crean dos listas, uno con los nombres de las categorías que se van a mostrar y otro con las URL a las que se van a solicitar esa información sobre la categoría.

```
//recomendaciones
notRepeat = new HashMap<>();
List<Multimedia> listSaved = User.profileLogged.getSeenMultimedia();
listSaved.addAll(User.profileLogged.getFollowMultimedia());
Collections.shuffle(listSaved);
long timeoutstart = System.currentTimeMillis();
for(int i = 0; i < listSaved.size() && i < 20 && !timeout(System.currentTimeMillis(), i++){
    Multimedia random = listSaved.get(i);
    if(!notRepeat.containsKey(random.getId())){
        if(random.isFilm()){
            urlS.add(api.getSimilarMovies(random.getId()));
            names.add("Similar movies to" + " " + random.getTitle());
        }else{
            urlS.add(api.getSimilarShows(random.getId()));
            names.add("Similar shows to" + " " + random.getTitle());
        }
        notRepeat.put(random.getId(), null);
    }
}
```

Figura 6.12: Fragmento del código que guarda las categorías y URL de películas similares a las guardadas

En este adaptador se pasan como argumentos dos listas de textos que representan los datos que se van a mostrar en el inicio, el título de la sección (como “Películas Populares”) y la URL con la que se va a obtener ese contenido en la API. Este adaptador creará una vista por cada URL y título de la lista que se pasen por argumentos una vez que el usuario se desplace por la lista y vaya mostrando más resultados. Esta vista será un texto y una lista de pósters de películas o series obtenidas por la llamada a la API.

La llamada a la API se realizará en la función “onBindViewHolder” que dependiendo de la cantidad de resultados que obtenga en esa petición variará la vista que se obtenga. En el caso de que la llamada devuelva solo un resultado se mostrará de la misma forma que se muestran en la biblioteca de películas o series marcadas. Esta decisión se ha tomado para que una petición que, debido a los filtros que se han aplicado en la aplicación, no muestre más de un resultado, no de la sensación al usuario que la aplicación está “vacía”.

Sin embargo, las modificaciones que se hagan a alguna de estas dos listas, nombres y URL, se deberán hacer simétricamente ya que la categoría de una solicitará la información a la URL de la misma posición, solo que en la otra lista.

Por otra parte, es importante mencionar que estas listas se mostrarán de forma que cuando se llegue al final de la lista de pósteres, este adaptador realizará una nueva llamada a la API para mostrar más resultados para la misma petición. Esto se puede hacer utilizando el argumento de la API que determina la página en que se encuentra y cuando esta página aumenta va mostrando más resultados para una misma petición. Es importante mencionar que esta llamada se realizará cuando llegue a los 15 últimos pósteres ya que, así, una vez que se arrastre sobre la lista, mostrará los nuevos resultados de una forma más fluida.

6.4.2. Recomendaciones aleatorias

Estas recomendaciones se pueden encontrar en la actividad de “RandomActivity”. Se obtienen de una forma más compleja que la anterior y son calculadas en la clase de “Recommendations.java”.

En la figura 6.13, se muestra como se inicializa esta clase, pasando como parámetros la “ProgressBar”, un objeto filtros y una lista de porcentajes. El objeto de filtros se utilizará para filtrar las recomendaciones que salgan posteriormente. La lista de porcentajes ayudará a determinar cómo de importantes son en la recomendación la valoración, lo nuevo que es y la popularidad de una película o serie, además de determinar cómo de importante es que esa película o serie haya salido varias veces recomendada antes.

```
3 usages  Chaos +1
public Recommendations(@NonNull Activity ac, RopeProgressBar progress_random, Filter filter, @NonNull int[] preset) throws ServiceException {
    if(preset.length<3 || preset.length > 4) throw new ServiceException("Preset error. Can't be less than 3.",new Exception());
    else if(preset.length ==3){
        PERC_REPS = preset[0];
        PERC_VAL = preset[1];
        PERC_POPULARITY = preset[2];
        PERC_TIME = 0;
    }else {
        PERC_REPS = preset[0];
        PERC_VAL = preset[1];
        PERC_POPULARITY = preset[2];
        PERC_TIME = preset[3];
    }
}

this.filter = filter;
this.progress_random = progress_random;
this.ac = ac;
cachedMultimedia=new ConcurrentHashMap<>();
counterMultimedia = new ConcurrentHashMap<>();
mult_ratio = new ConcurrentHashMap<>();
recom_stats = new ConcurrentHashMap<>();
this.popularities_list= new LinkedList<>();
api = new TMDBApi(Objects.requireNonNull(User.profileLogged),ac);
}
```

Figura 6.13: Iniciación de la clase de recomendaciones

Las recomendaciones se realizan utilizando las películas o series que el usuario ha marcado para seguir, como pendiente o visto.

```

@NonNull
public List<Multimedia> getRecommendations(){
    List<Multimedia> listFollowedAndSeen = Objects.requireNonNull(User.profileLogged).getFollowMultimedia();
    for(Multimedia seenMult : User.profileLogged.getSeenMultimedia()){
        if(!User.profileLogged.isFollowed(seenMult))
            listFollowedAndSeen.add(seenMult);
    }
    initialize(listFollowedAndSeen);
    //Max repeated film
    ac.runOnUiThread()->{if(this.progress_random!=null)this.progress_random.animateProgress(PERC_INITIALIZE);});
    List<Pair<Multimedia, Integer>> maxFilmList = getOrderedMultimedia();
    List<Multimedia> res = new LinkedList<>();
    for(int i = 0; i < maxFilmList.size(); i++){
        res.add(maxFilmList.get(i).component1());
    }
    ac.runOnUiThread()->{if(this.progress_random!=null)this.progress_random.animateProgress(100);});
    return res;
}

```

Figura 6.14: Método que devuelve una lista de recomendaciones

En el método de “initalize” se realizan todas las llamadas a la API y le devuelve una lista de películas o series recomendadas. Estas se añaden a una lista de contenido recomendado y además se calcula las veces que ha salido repetida una película o serie recomendada para posteriormente calcular el porcentaje de recomendación. Estas recomendaciones se tienen que ordenar de mayor a menor porcentaje por lo que la lista de contenido recomendado se filtrará utilizando un cálculo del porcentaje, como podemos ver en la figura 6.15.

```

Map<String,Double> cacheAlgorithmValue = new HashMap<>();
maxFilmList.sort((o1, o2) -> {
    double value1;
    double value2;
    if(cacheAlgorithmValue.containsKey(o1.getFirst().getId())) {
        value1 = Objects.requireNonNull(cacheAlgorithmValue.get(o1.getFirst().getId()));
    }else{
        value1 = getAlgorithmValue(o1);
        cacheAlgorithmValue.put(o1.getFirst().getId(),value1);
    }
    if(cacheAlgorithmValue.containsKey(o2.getFirst().getId())) {
        value2 = Objects.requireNonNull(cacheAlgorithmValue.get(o2.getFirst().getId()));
    }else{
        value2 = getAlgorithmValue(o2);
        cacheAlgorithmValue.put(o2.getFirst().getId(),value2);
    }

    if (value1 != value2) {
        if (value2 > value1) return 1;
        else return -1;
    } else {
        if (rand.nextBoolean()) {
            return 1;
        } else {
            return 0;
        }
    }
});

```

Figura 6.15: Método que ordena una lista de recomendados

Este cálculo del valor del porcentaje se realiza en el método de “getAlgorithmValue” de la figura 6.15. En este podemos observar que se pasa por parámetro la película y las repeticiones de estas.

Generalmente, el porcentaje se calculará tomando en cuenta un 45 % de las repeticiones, un 5 % de lo nuevas que son, un 25 % de su valoración media y un 25 % de lo popular que es. Las repeticiones se calcularán teniendo como referencia el máximo de referencias que ha tenido una película o serie en el método explicado anteriormente.

CAPÍTULO 7

Pruebas

En este apartado introduciremos las pruebas que se han realizado a la aplicación para asegurar una calidad mínima del producto.

7.1 UAT

En las pruebas *User Acceptance Testing* o UAT [28] se mide el grado de aceptación del usuario sobre un software desarrollado. Es una fase del ciclo de vida de desarrollo de software en la que el cliente debe probar si el software funciona y se comporta según sus requisitos.

Esta prueba ha sido realizada por una representación del usuario final, gracias a la ayuda de 10 personas, los cuales han probado la aplicación siguiendo el guion de la prueba. Posteriormente, se le ha pedido al usuario que rellene un cuestionario con preguntas para medir lo intuitivo, rápido y útil que es la aplicación. El entorno en el que se han hecho estas comprobaciones han sido en dispositivos Android privados de cada persona, utilizando la última versión de la aplicación descargada desde Google Play.

Este guion es únicamente para guiar y comprobar que la aplicación está siendo probada en profundidad. El guion para la prueba es el siguiente:

1. Abrir la aplicación
2. Ir a la pestaña de ajustes e iniciar sesión (te puedes iniciar sesión con correo electrónico: "tester@whatsnew.com" y contraseña: "prueba393837aaaa")
3. Selecciona el perfil: "Familia"
4. Seleccionar una plataforma con la que ver una película
5. Cambiar la vista a solo ver series de esa plataforma
6. Abrir una película o serie
7. Darle a seguir y a Más información
8. Marcar como visto
9. Ver un actor de la lista de reparto
10. Buscar una película en la vista de búsqueda
11. Ir a la página de ajustes y añadir plataformas a la lista de suscripciones

12. Ir al inicio y ver como se han filtrado las recomendaciones
13. Darle al icono de aleatorio para entrar en la vista de recomendaciones aleatorias
14. Darle a marcar como seguir a algunas películas o series

Los resultados se muestran en gráficos de barras que miden el grado de desacuerdo, siendo este un 1, y de acuerdo, siendo este un 5, del usuario, respecto a las afirmaciones planteadas:

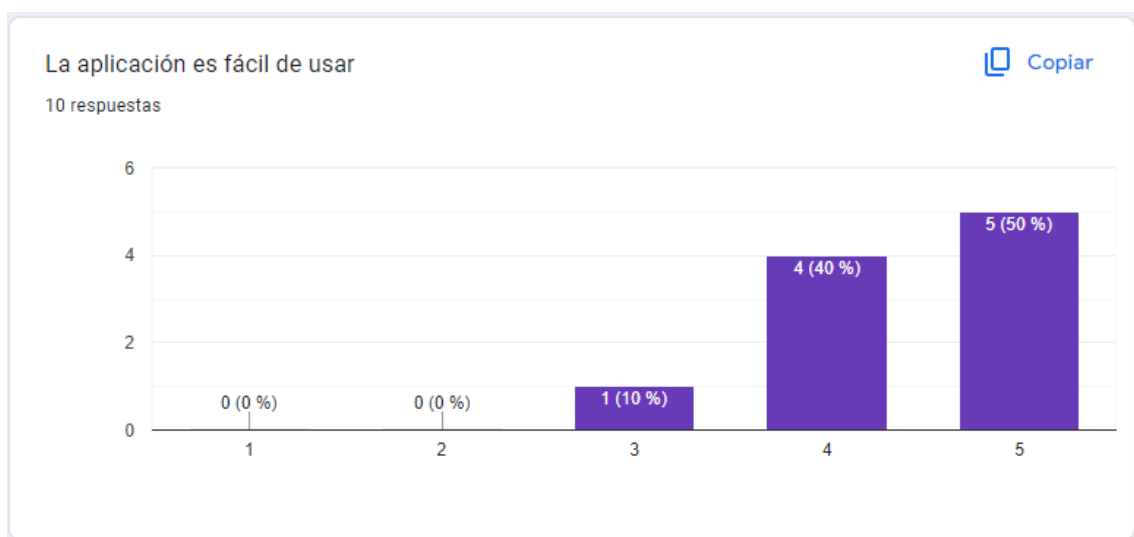


Figura 7.1: Resultados de “La aplicación es fácil de usar”

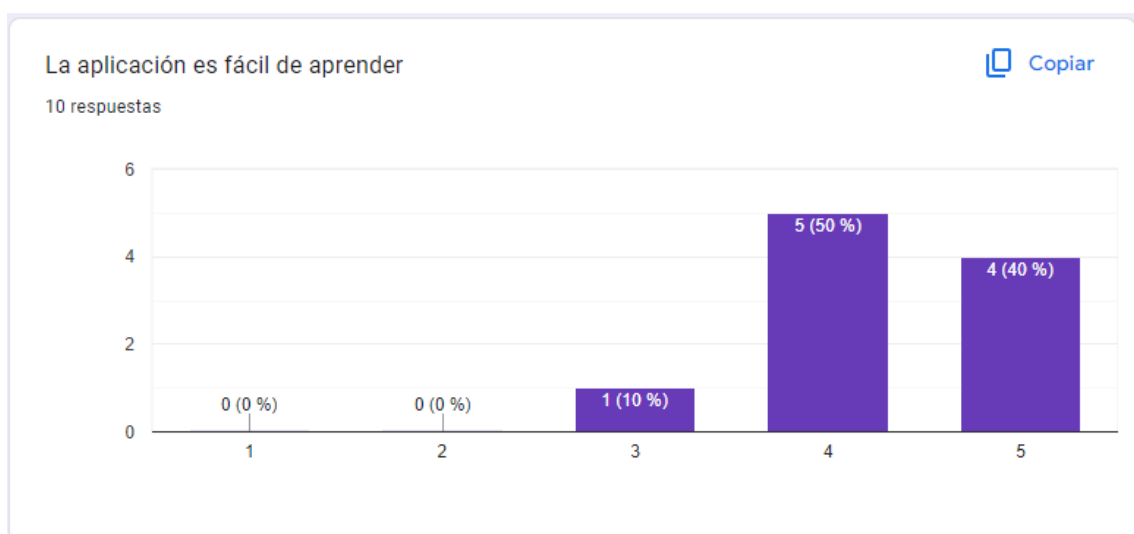


Figura 7.2: Resultados de “La aplicación es fácil de aprender”

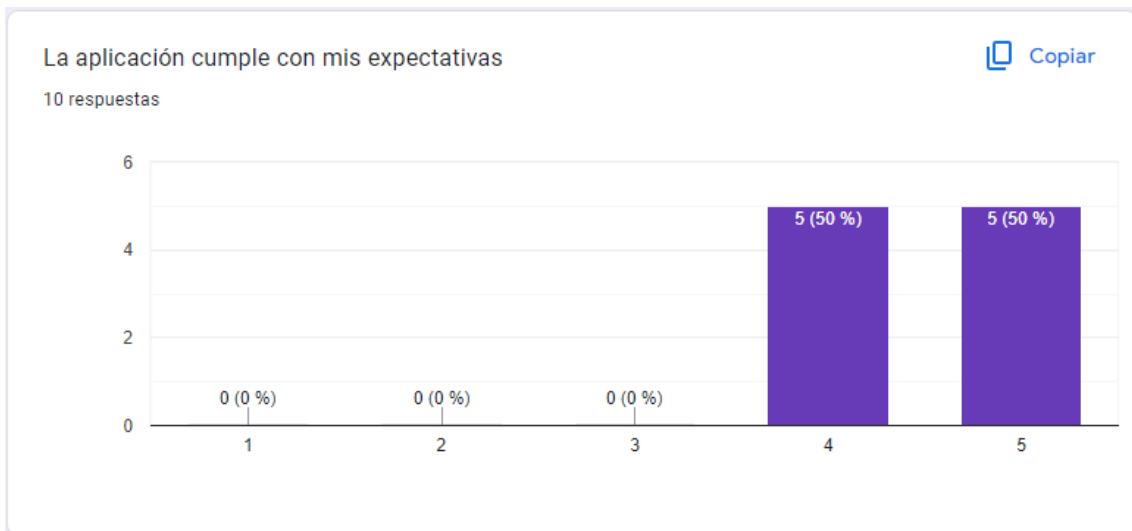


Figura 7.3: Resultados de “La aplicación cumple con mis expectativas”

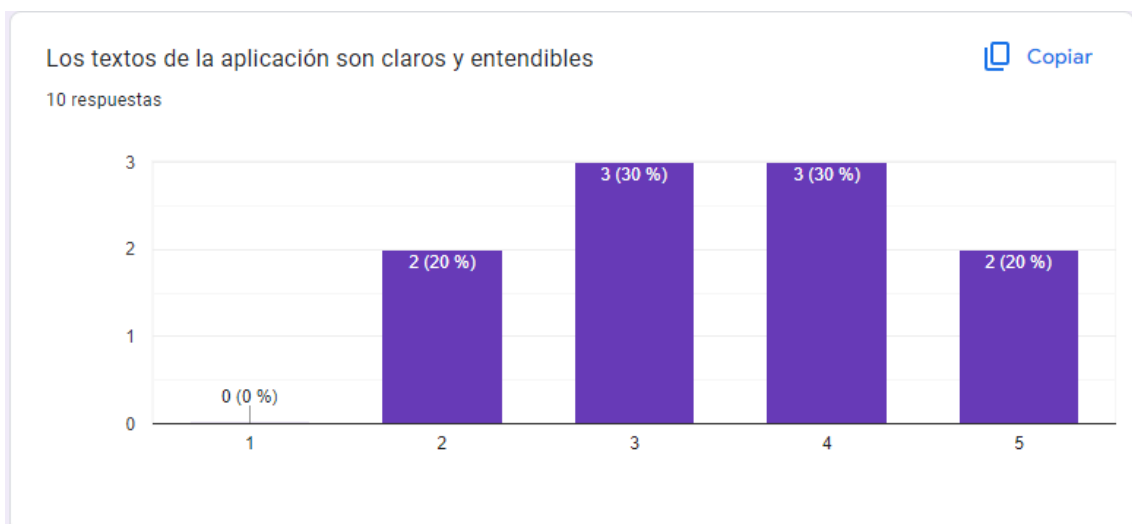


Figura 7.4: Resultados de “Los textos de la aplicación son claros y entendibles”



Figura 7.5: Resultados de “La aplicación responde de manera rápida y eficiente”

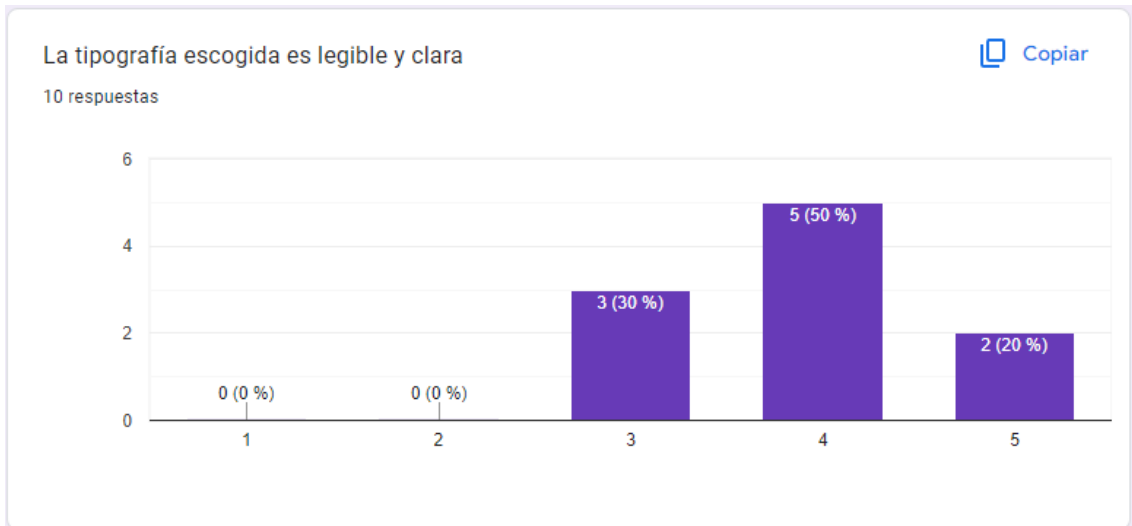


Figura 7.6: Resultados de “La tipografía escogida es legible y clara”

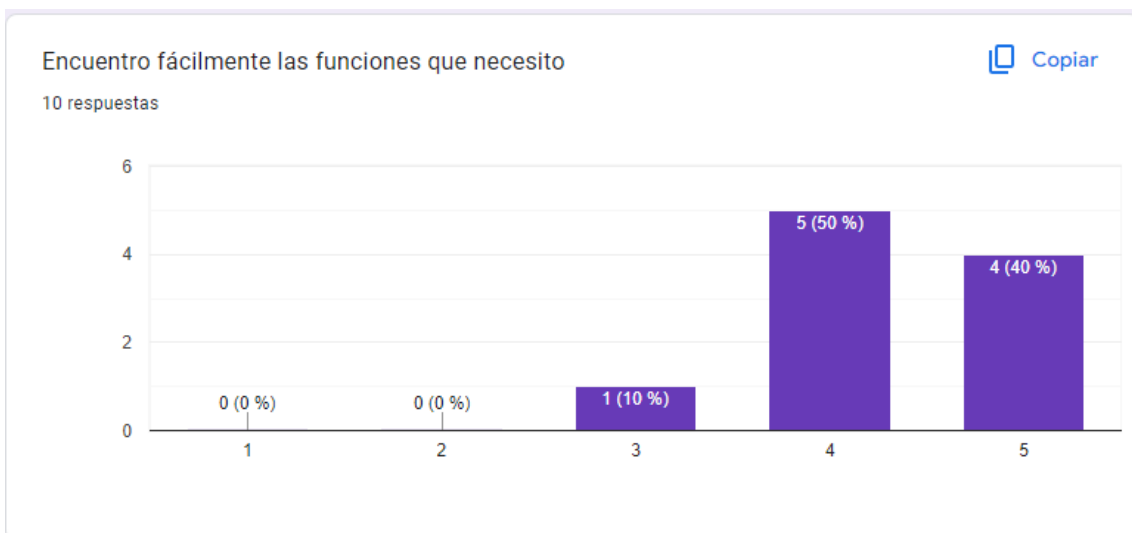


Figura 7.7: Resultados de “Encuentro fácilmente las funciones que necesito”

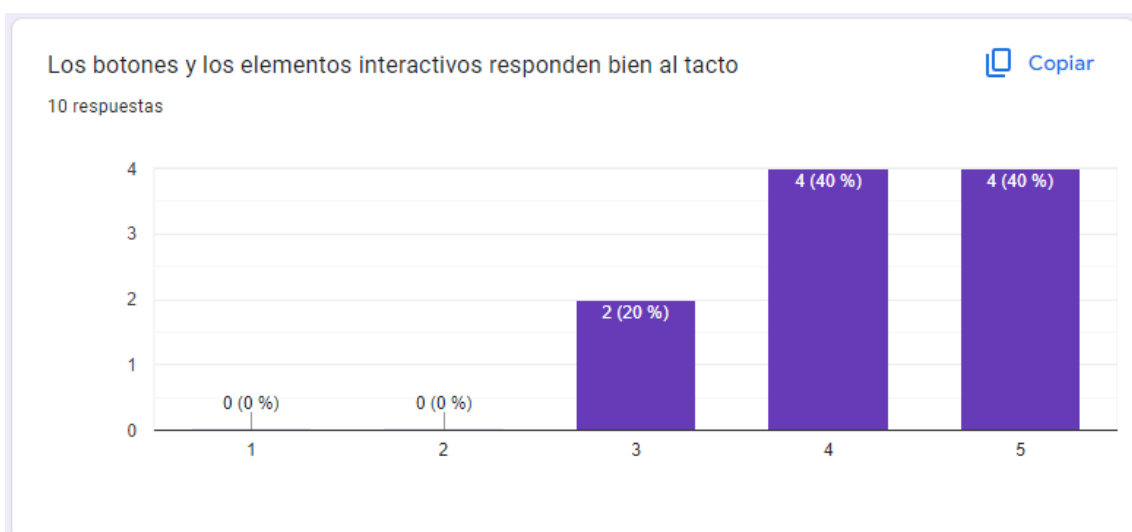


Figura 7.8: Resultados de “Los botones y los elementos interactivos responden bien al tacto”

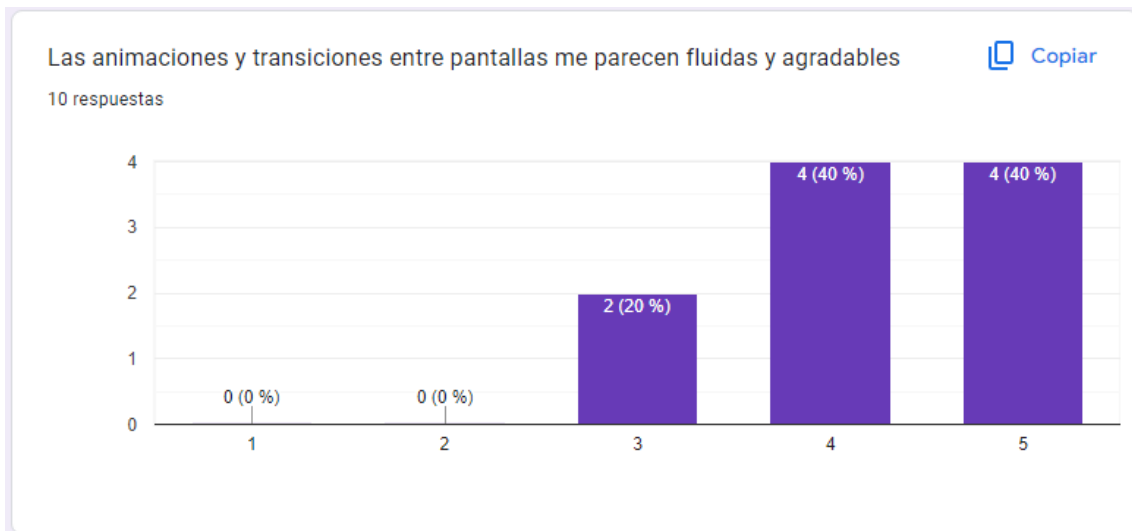


Figura 7.9: Resultados de “Las animaciones y transiciones entre pantallas me parecen fluidas y agradables”

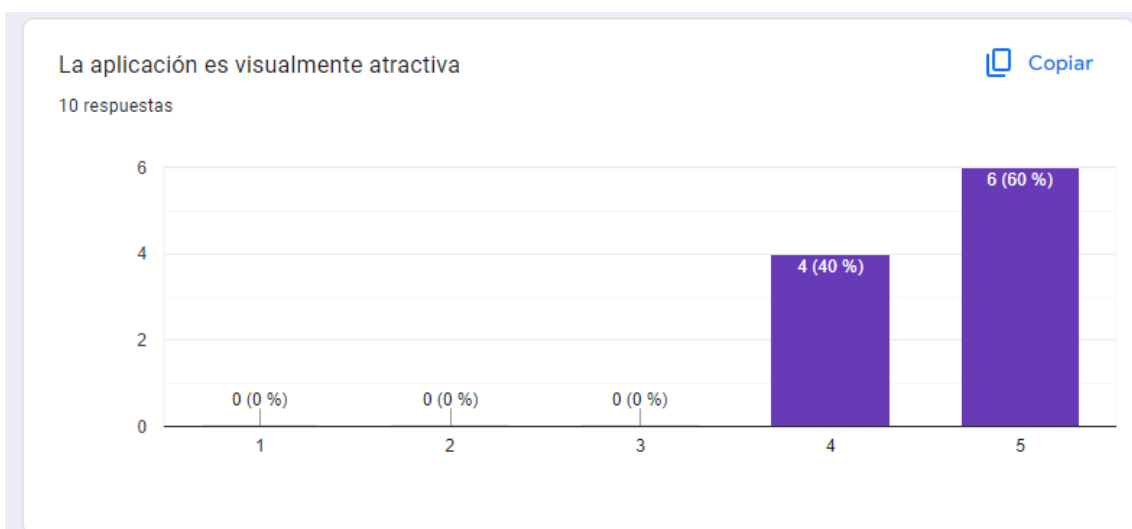


Figura 7.10: Resultados de “La aplicación es visualmente atractiva”

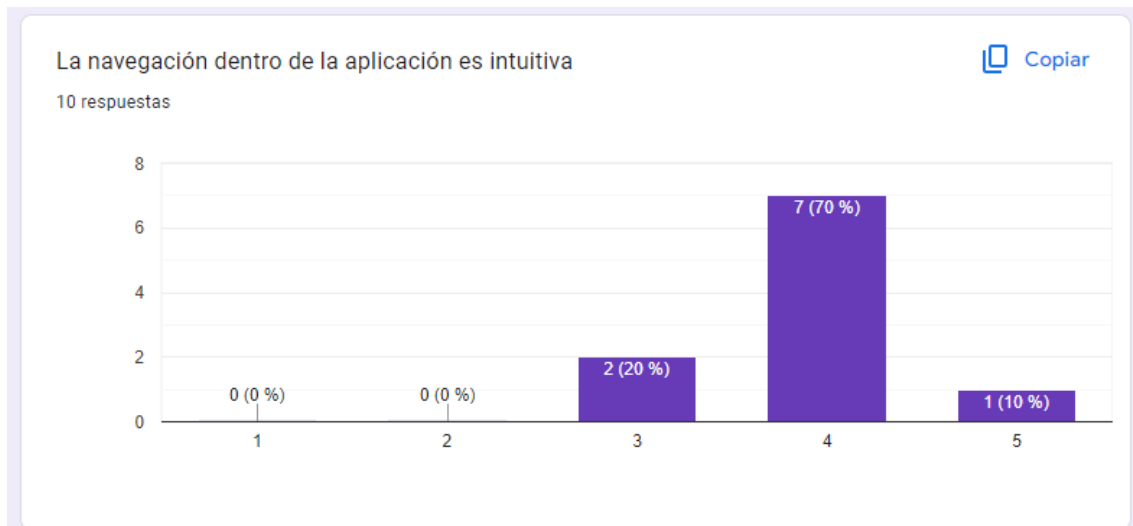


Figura 7.11: Resultados de “La navegación dentro de la aplicación es intuitiva”



Figura 7.12: Resultados de “Recomendaría la aplicación a un amigo o familiar”

Los participantes de las pruebas han expresado que la aplicación es fácil de usar, lo cual indica que su diseño y disposición de elementos facilitan la interacción y navegación intuitiva. Además, han destacado que la aplicación es fácil de aprender, lo que sugiere que la curva de aprendizaje es suave y que los usuarios pueden familiarizarse rápidamente con sus funciones.

Los usuarios también han manifestado que la aplicación cumple con sus expectativas, lo que indica que se han satisfecho sus necesidades y requisitos observados en las especificaciones. Han destacado que los textos de la aplicación son claros y entendibles, lo cual contribuye a una comunicación efectiva y a una comprensión adecuada de la información presentada.

En cuanto a la capacidad de respuesta de la aplicación, los participantes han expresado que la aplicación responde de manera rápida y eficiente a sus acciones, lo que les permite realizar tareas de manera fluida y sin retrasos molestos. Asimismo, han valorado positivamente la elección de una tipografía legible y clara, lo cual contribuye a una lectura cómoda y evita la fatiga ocular.

Los participantes han destacado que encuentran fácilmente las funciones que necesitan, lo que indica que la organización y la estructura de la aplicación son adecuadas y permiten una búsqueda eficiente de las herramientas y características que el usuario requiere. Además, han expresado que los botones y los elementos interactivos responden bien al tacto, lo que garantiza una interacción sin problemas y una retroalimentación táctil precisa.

Los usuarios han manifestado su disposición a recomendar la aplicación a un amigo o familiar. Este aspecto es un indicador clave de la satisfacción del usuario y la confianza en la calidad y utilidad de la aplicación.

Por último, los participantes han señalado que las animaciones y transiciones entre pantallas les parecen fluidas y agradables, lo que contribuye a una experiencia visualmente atractiva y coherente en toda la aplicación.

Los resultados de las pruebas han confirmado que la aplicación ha obtenido un desempeño satisfactorio en los aspectos evaluados.



Figura 7.13: Resultados de “Han ocurrido errores en la prueba”

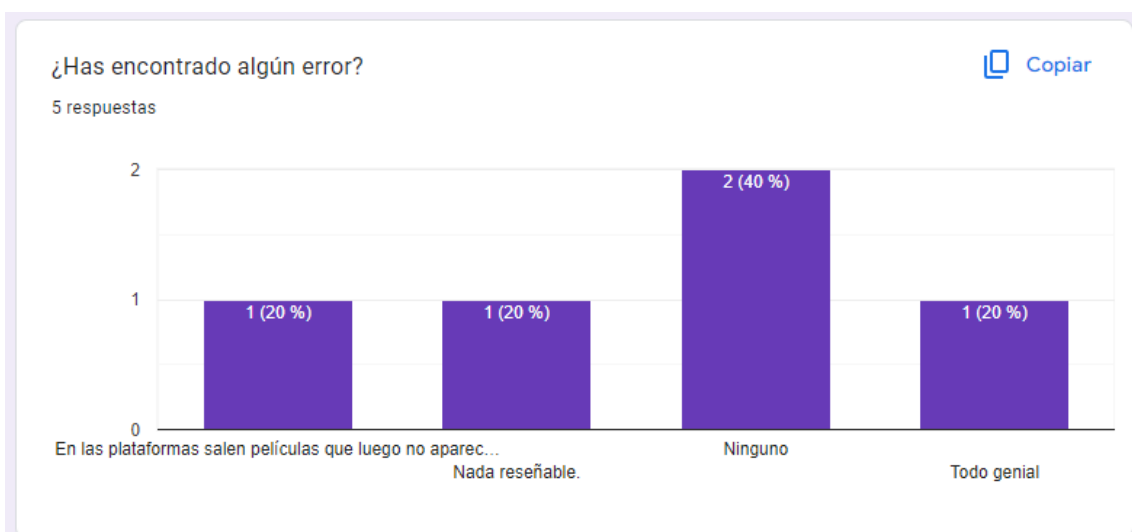


Figura 7.14: Resultados de “¿Has encontrado algún error?”

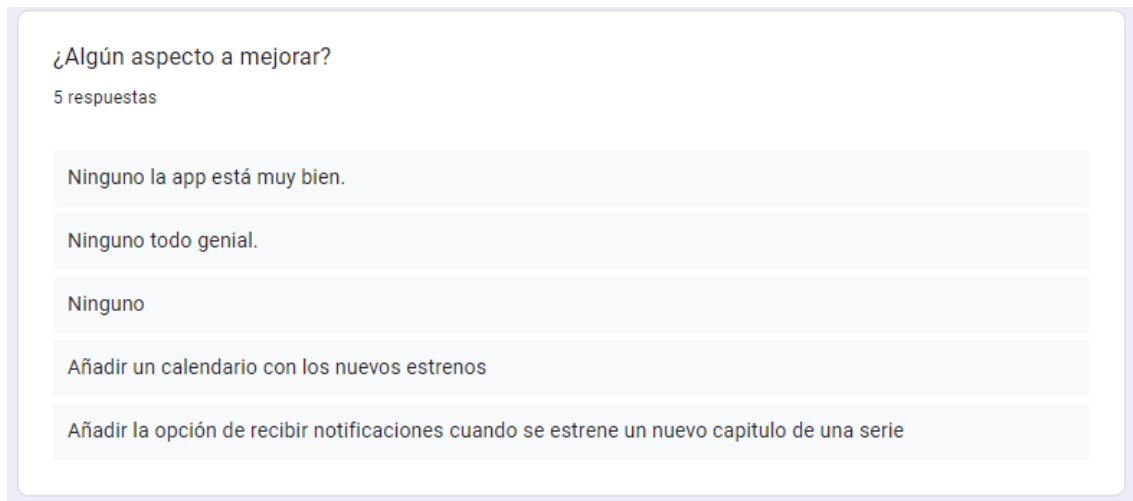


Figura 7.15: Resultados de “¿Algún aspecto a mejorar?”

En la figura 7.13 podemos observar que solo se ha producido un error, este es el que aparece en la figura 7.14, “En las plataformas salen películas que luego no aparecen como para ver en esas plataformas”, el cual está solucionado en la última versión de la aplicación.

En la figura 7.15, podemos ver que varios usuarios han sugerido unos cambios para la aplicación interesantes. Estos son “Añadir un calendario con los nuevos estrenos” y “Añadir la opción de recibir notificaciones cuando se estrene un nuevo capítulo de una serie” que son propuestas de cambio que se considerarán para futuras modificaciones de la aplicación.

7.2 Pruebas de calidad

Las pruebas de calidad son esenciales para evaluar la excelencia de una aplicación. Para esto, Google ha definido estos criterios principales de calidad y pruebas relacionadas que permitirán evaluar la calidad de la aplicación presentada [37]. La lista de tareas propuesta se enfoca en destacar la calidad mínima que todas las aplicaciones deben cumplir. Estos criterios son importantes, aunque a veces pueden pasar desapercibidos.

7.2.1. Experiencia visual

Con el fin de garantizar que la aplicación es intuitiva y coherente, la aplicación deberá proporcionar un patrón de diseño estándar en Android. Por ello, se realizarán las siguientes pruebas:

Tabla 7.1: Resultados de las pruebas de “Experiencia visual”

Área	Criterio	Test
Navegación	La aplicación permite el uso del botón Atrás.	SI
Navegación	La aplicación permite la navegación por gestos.	SI
Navegación	La aplicación guarda el estado de esta misma cuando está en segundo plano	SI
Notificaciones	Las notificaciones dentro de la aplicación siguen los estándares de Material Design.	Omitido

Tabla 7.1: Resultados de las pruebas de "Experiencia visual"

Área	Criterio	Test
Notificaciones	Utiliza un estilo de notificaciones propio de una aplicación para la mensajería	Omitido
IU y gráficos	La aplicación se puede visualizar tanto en vertical como en horizontal.	SI
IU y gráficos	La aplicación utiliza toda la pantalla en ambas orientaciones.	SI
IU y gráficos	La aplicación realiza transiciones fluidas entre las orientaciones de pantalla.	SI
Calidad visual	La aplicación muestra, sin distorsión, el contenido.	SI
Calidad visual	La aplicación muestra texto en un formato legible en los dos idiomas que admite.	SI
Calidad visual	La aplicación admite el tema oscuro.	SI
Accesibilidad	El tamaño de los botones tiene que ser como mínimo de 48dp.	SI
Accesibilidad	El contenido e primero plano debe tener un color que haga contraste con el fondo.	SI
Accesibilidad	Cada elemento de la interfaz está descrito por un "contentDescription", excepto los textos.	SI

En esta, las pruebas de las notificaciones se han omitido para esta aplicación ya que el sistema no envía ningún tipo de notificación al usuario. Además, se puede destacar que, aunque la aplicación no está diseñada para ser usada de forma horizontal, esta escala bien las interfaces para mantener el diseño pensado, aunque no es recomendable usarse así.

7.2.2. Funcionalidad

Se realizan pruebas para comprobar el funcionamiento correcto de la aplicación.

Tabla 7.2: Resultados de las pruebas de "Funcionalidad"

Área	Criterio	Test
Audio	La reproducción de un audio continúa cuando la aplicación vuelve a estar en primer plano.	Omitido
Audio	Si la reproducción de audio es una función principal, esta debería permitir escuchar el audio en segundo plano.	Omitido
Audio	Cuando se inicie la reproducción, esta deberá indicar que el audio se está cargando con un indicador visual.	Omitido
Audio	La app deberá solicitar foco de audio cuando el audio comience a reproducirse y deberá abandonarlo cuando se detenga la reproducción.	Omitido
Audio	La aplicación deberá reducir el volumen de su reproducción cuando otra aplicación empiece a reproducir audio.	Omitido
Multimedia	Cuando la aplicación reproduzca audio deberá enviar una notificación.	Omitido
Multimedia	Si la aplicación reproduce vídeo deberá admitir Picture-In-Picture.	Omitido
Multimedia	Si la aplicación codifica vídeo deberá utilizar el codec HEVC.	Omitido
Se comparte	La aplicación utilizará Android Sharesheet cuando comparta contenido.	Omitido
Servicio en segundo plano	La aplicación deberá evitar que se ejecuten servicios muy largos en segundo plano.	Omitido

Todas estas pruebas se han omitido porque la aplicación en cuestión no reproduce ningún audio, ni vídeo, y esta no se ejecuta en segundo plano.

7.2.3. Rendimiento y estabilidad

En estas pruebas se comprueba si la aplicación ofrece una experiencia estable y sin tiempos de espera excesivos que empeoren la experiencia del usuario con esta.

Tabla 7.3: Resultados de las pruebas de "Rendimiento y estabilidad"

Área	Criterio	Test
Estabilidad	La aplicación no provoca errores de ANR "Android no responde".	SI
Rendimiento	La aplicación se carga rápidamente o mostrará un indicador de progreso.	SI
Rendimiento	La aplicación deberá renderizarse a 60 fotogramas por segundo.	SI
Rendimiento	Con el StrictMode activado la aplicación mostrará destellos en rojo, indicando que se están produciendo comportamientos inadecuados en la aplicación.	SI
SDK	La aplicación se orientará a la última versión sin que se produzca ningún error.	SI
SDK	La aplicación utilizará la versión más reciente del SDK de Android.	SI
SDK	La aplicación se compilará con la última versión del SDK de Android.	SI
SDK	Todos los SDK de Google o de terceros utilizados en la aplicación estarán actualizados.	SI
SDK	La app no usa interfaces que no pertenecen al SDK.	SI
SDK	La aplicación no incorpora bibliotecas de depuración.	SI
Batería	La aplicación admite las funciones de administración de energía desde el Android 6.0.	SI

Como podemos observar la aplicación cumple con todas las pruebas que se realizan de estabilidad y rendimiento. Sin embargo, para que la experiencia sea lo más fluida posible se recomienda que la aplicación se ejecute en un dispositivo que tenga más de medio *gigabyte* de memoria.

7.2.4. Privacidad y seguridad

Se ha probado que la aplicación administre de forma segura los datos del usuario, cumpliendo, además, las políticas de Datos del usuario [28] para la publicación de aplicaciones en Google Play.

Tabla 7.4: Resultados de las pruebas de "Privacidad y seguridad"

Área	Criterio	Test
Permisos	La aplicación solicita la mínima cantidad de permisos que necesita.	SI
Permisos	La aplicación no solicita permisos de acceso a datos sensibles.	SI
Permisos	La aplicación solicita los permisos cuando esta está en primer plano.	SI
Permisos	La aplicación transmite claramente el motivo por el que se solicitan los permisos.	SI
Permisos	La aplicación no deberá impedir que se ejecute la aplicación sin aceptar los permisos.	SI
Datos y archivos	Los datos sensibles se almacenan en el dispositivo, únicamente.	SI

Tabla 7.4: Resultados de las pruebas de "Privacidad y seguridad"

Área	Criterio	Test
Datos y archivos	No se registran datos personales de los usuarios.	SI
Datos y archivos	La aplicación no deberá utilizar IDs del <i>hardware</i> del dispositivo.	SI
Identidad	La aplicación utiliza el autocompletar de credenciales de la cuenta.	SI
Identidad	La aplicación utiliza el One Tap para iniciar sesión.	NO
Identidad	La aplicación admite autenticación biométrica para acceder a información sensible.	Omitido
Componentes de la app	La aplicación asigna el atributo "android:exported" a "true" en todas sus actividades.	SI
Componentes de la app	Los Intent comparten datos de manera segura.	SI
Componentes de la app	Todos los componentes que comparten contenido entre aplicaciones usan el "android:protectionLevel".	Omitido
Redes	Todas las peticiones se envían mediante SSL.	SI
Redes	La aplicación declara una configuración de seguridad para la red.	SI
Redes	La aplicación si utiliza los servicios de Google Play, iniciará el proveedor de seguridad antes.	Omitido
WebViews	No uses "setAllowUniversalAccessFromFileURLs()".	Omitido
WebViews	Las WebViews no deben usar "addJavaScriptInterface()".	Omitido
Ejecución	La aplicación no carga código dinámicamente desde fuera de la APK.	SI
Criptografía	La aplicación implementa algoritmos criptográficos fuertes.	SI

En cuanto a la privacidad y seguridad que ofrece la aplicación. Esta solamente solicita los permisos de internet para acceder a todos los datos externos sobre películas o series. Además, se encuentra declarada una configuración de red para que solo acepte comunicación de los servidores del dominio de "themoviedb.org".

Además, este no integra el sistema de Google de One Tap, ya que se ha decidido utilizar simplemente un correo electrónico y contraseña para el registro e inicio de sesión. Aunque, no se descarta utilizarlo para proporcionar un acceso rápido a la aplicación.

Por otra parte, la aplicación no solicita permisos para acceder a datos sensibles ni a los IDs del hardware del dispositivo y, por ello, no necesitará el uso de la autenticación biométrica para acceder a esta.

Por último, se han omitido las pruebas para los WebViews, que son vistas con las que se pueden mostrar páginas web, ya que estas vistas no se utilizan en ninguna interfaz de la aplicación presentada. Sin embargo, también se ha omitido la prueba de seguridad cuando se utilizan los servicios de Google Play en la aplicación, debido a que, la aplicación, no utiliza estos servicios.

7.2.5. Google Play

Se realizan las pruebas para que la aplicación pueda ser publicada en Google Play.

Tabla 7.5: Resultados de las pruebas de “Google Play”

Área	Criterio	Test
Políticas	La aplicación cumple con los términos de la Política de Contenido para desarrolladores de Google Play.	SI
Políticas	El nivel de madurez de la app se establece de forma correcta en función de los Lineamientos de Clasificación del Contenido.	SI
Página de detalles de la app	La página de la aplicación incluye un gráfico central, sin capturas, y con texto legible	SI
Página de detalles de la app	Las capturas no hacen referencias a dispositivos que no sean Android.	SI
Página de detalles de la app	Las capturas no representan la experiencia que ofrece la aplicación de forma confusa.	SI
Asistencia para el usuario	Los errores informados en la pestaña de reseñas se abordan si se produce en varios dispositivos diferentes.	SI

7.3 Pruebas de funcionalidad

En este caso se probará si se han cumplido los casos de uso presentados al inicio del diseño de la aplicación.

Tabla 7.6: Resultados de las pruebas de funcionalidad

Caso de uso	Funcionalidad	Prueba
RF-01	Mostrar valoraciones de episodios	Correcto
RF-02	Mostrar el reparto de una película o serie	Correcto
RF-03	Mostrar películas o series en cada plataforma.	Correcto
RF-04	Mostrar información de película o series.	Correcto
RF-05	Mostrar plataformas donde ver películas o series.	Correcto
RF-06	Mostrar tiempo restante a que se estrene una película.	Correcto
RF-07	Ver películas o series similares.	Correcto
RF-08	Buscar películas o series.	Correcto
RF-09	Iniciar sesión.	Correcto
RF-10	Registrarse.	Correcto
RF-11	Marcar o desmarcar como visto.	Correcto
RF-12	Recomendaciones en el inicio.	Correcto
RF-13	Recomendaciones aleatorias.	Correcto
RF-14	Seguir series.	Correcto
RF-15	Filtrar el contenido por suscripciones.	Correcto
RF-16	Marcar o desmarcar como pendiente.	Correcto
RF-17	Crear perfiles para cada usuario.	Correcto
RF-18	Ocultar películas o series vistas	Correcto
RF-19	Filtrar películas seguidas.	Correcto
RF-20	Filtrar películas vistas	Correcto
RF-21	Cerrar sesión	Correcto

Como podemos ver en la tabla 7.6 la aplicación cumple con todos los casos de uso tanto para el usuario como para el usuario autenticado.

CAPÍTULO 8

Aplicación final

A continuación, en este apartado, como se presenta la aplicación finalizada.

8.1 Interfaces

En este subapartado, se muestra el conjunto de interfaces de usuario que se puede observar a lo largo de la aplicación.

8.1.1. Inicio

En la figura 8.1, se representa la interfaz que se muestra al inicio de la aplicación en la que se muestra un conjunto de categorías en las que se ordenan un conjunto de películas o series, como, por ejemplo, la lista de películas y series más populares. En esta vista también se muestra una lista de plataformas en las que se pueden observar películas y series que se pueden encontrar con la suscripción a esa plataforma.

Desde esta interfaz se podrá recibir recomendaciones (RF12 – Recomendaciones en el inicio) y mostrar, desde la misma vista, información básica sobre la película o la serie que se clique como se puede observar en la figura 8.2.

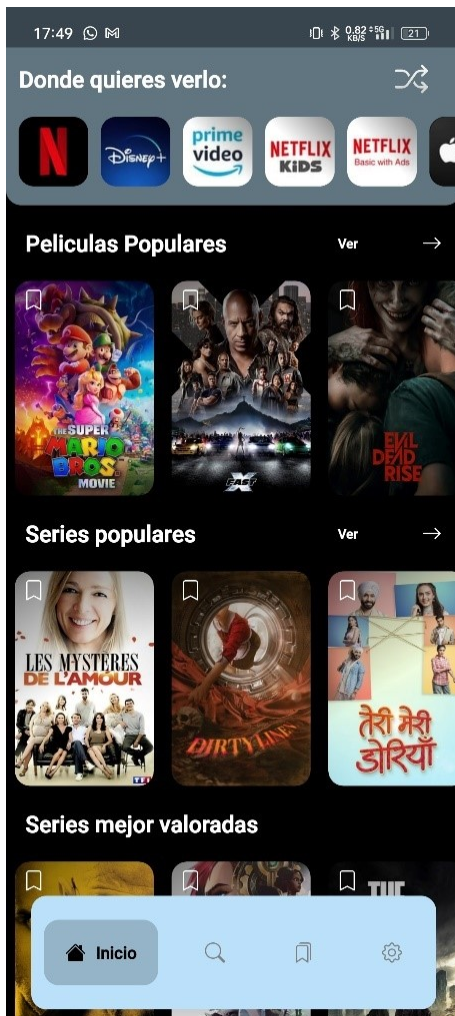


Figura 8.1: Captura del inicio

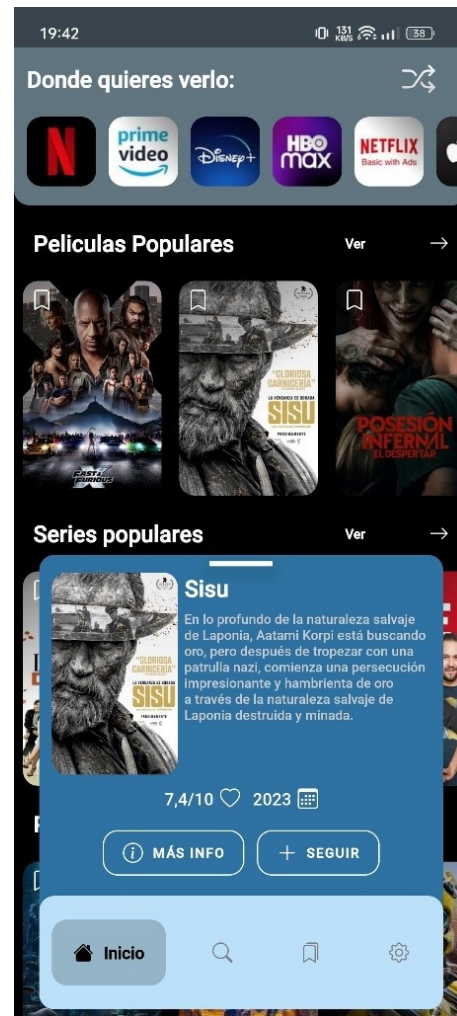


Figura 8.2: Captura del inicio con desplegable

Tanto en esta interfaz, como en la anterior vista (figura 8.1), aparece un botón de marcador con el que se podrá marcar para seguir una serie (RF14- Seguir serie) o marcar una película (RF16- Marcar como pendiente) como pendiente.

Por otra parte, en la figura 8.3, se observa que haciendo clic en las plataformas de contenido en *streaming* se puede obtener una lista de películas o series que se encuentran ofertadas en estas mismas.

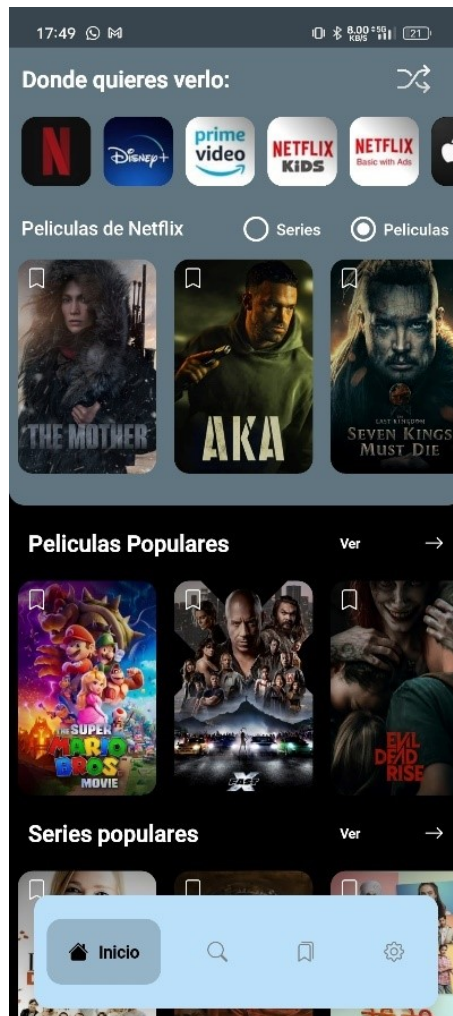


Figura 8.3: Captura del inicio con plataforma habiendo hecho el clic

8.1.2. Inicio de sesión

El inicio de sesión (RF09- Iniciar sesión) puede ocurrir de distintas formas, como el usuario puede acceder a la aplicación y a alguna funcionalidad, no se muestra el inicio de sesión al abrirse la aplicación, sino que se mostrará en la interfaz de la biblioteca o la de ajustes cuando intente acceder a funcionalidad que necesita un usuario autenticado. Por ello, como podemos ver en la figura 8.4, se muestra un mensaje de información para que se inicie sesión en la aplicación o se pueda registrar en esta.

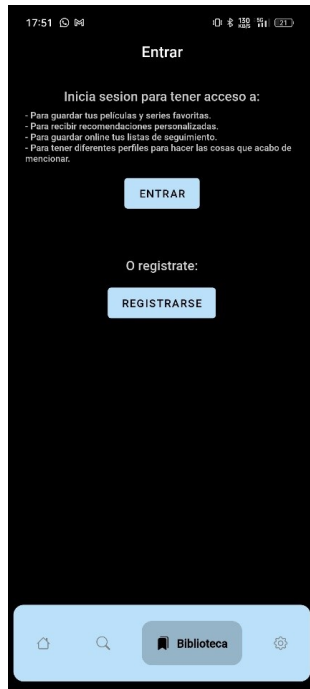


Figura 8.4: Captura del inicio de sesión o registro

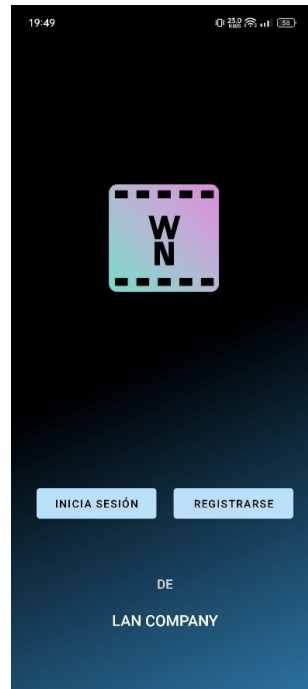


Figura 8.5: Captura de la pantalla de inicio o registro

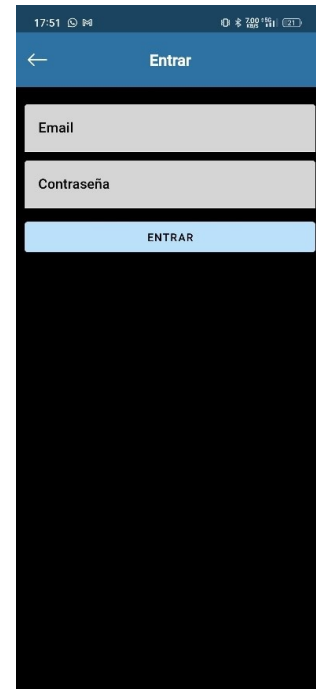


Figura 8.6: Captura de la pantalla de inicio de sesión

También se puede acceder a la interfaz de inicio de sesión cuando el usuario autenticado cierre sesión. En ese caso se mostrará la siguiente interfaz de la figura 8.5 para que pueda elegir si registrarse con un nuevo usuario o iniciar sesión con otro distinto.

Por tanto, si el usuario decide iniciar sesión le aparecerá la interfaz de la figura 8.6. Una vez que el usuario acceda con el usuario correcto y la contraseña correcta, se le llevará a la interfaz de selección de perfiles para acceder con uno a la aplicación.

8.1.3. Registro

Tras explicar la interfaz de inicio de sesión, se muestra en la figura 8.7 como es la interfaz de usuario para registrarse en la aplicación (RF10- Registrarse). En esta encontramos un acceso directo a los términos de uso de la aplicación.

Al registrarse en la aplicación, el sistema llevará al usuario a inicio.

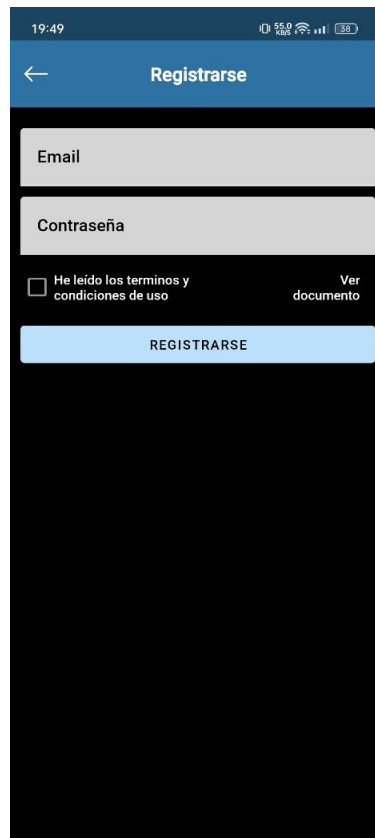


Figura 8.7: Captura de la pantalla de registro

8.1.4. Búsqueda

La figura 8.8 muestra la pestaña donde el usuario puede buscar películas o series (RF08- Buscar películas o series). Antes de que introduzca un texto que quiera buscar, se le mostrará el historial de búsquedas que se han realizado anteriormente.

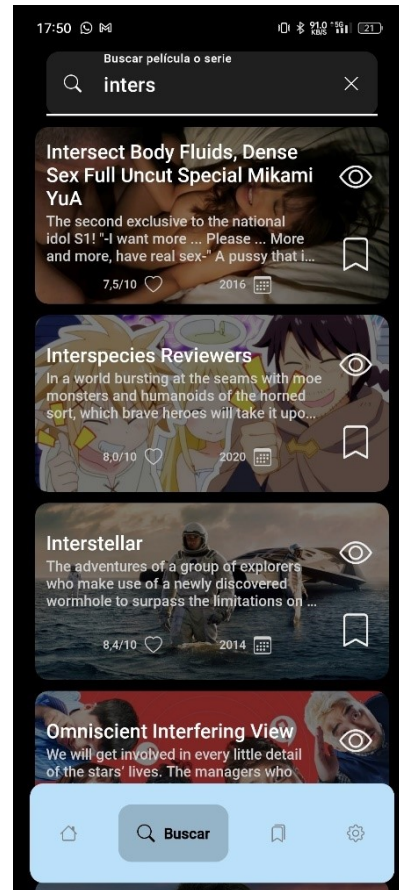
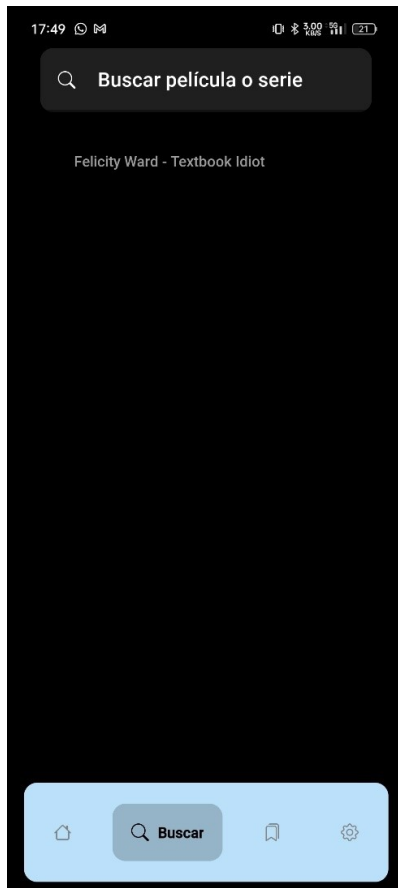


Figura 8.8: Captura de la pantalla de búsqueda

Figura 8.9: Captura de los resultados de la búsqueda

Cuando este introduzca texto en el cuadro de búsqueda obtendrá una lista de resultados que contienen el texto que se ha buscado, como podemos ver en la figura 8.9.

8.1.5. Selector de perfiles

En la siguiente interfaz de la figura 8.10 se muestran todos los perfiles con los que un usuario puede acceder a la aplicación. El único perfil que estará creado por defecto en todos los usuarios autenticados será el de "Familia". Como se observa, en el fondo del botón para seleccionar un perfil, se van mostrando películas o series que ha ido marcando el perfil como pendiente.

En esta interfaz también podremos acceder, con el botón en la parte superior-derecha, a la información del usuario autenticado y tendremos la opción de cerrar sesión.

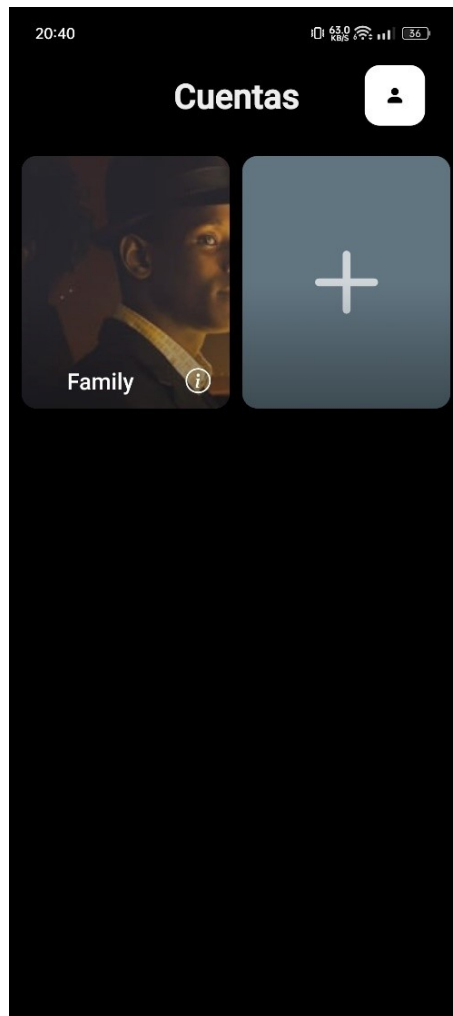


Figura 8.10: Captura de la pantalla de selección de perfil

8.1.6. Visor de películas o series

En cuanto a la información de las películas y las series, se muestra mediante la interfaz de la figura 8.11. En esta, el usuario tiene acceso a la funcionalidad de marcar o desmarcar las películas como vista (RF11- Marcar o desmarcar como visto), marcar las series para seguirlas (RF14- Seguir serie) y como pendiente (RF16- Marcar o desmarcar como pendiente). Además, se mostrará una visión más detallada de la información respectiva a la película.

Por otra parte, también se mostrarán las plataformas en las que se puede encontrar la película seleccionada (RF05- Mostrar plataformas donde ver películas o series), las películas o series similares a la seleccionada (RF07- Ver películas o series similares) y el reparto (RF02- Mostrar el reparto de una película o serie) como vemos en la figura 8.12.



Figura 8.11: Captura de pantalla de la información de la película Alias



Figura 8.12: Captura de pantalla de la parte de abajo de la interfaz

No obstante, si la información a mostrar es una serie en la interfaz se podrá observar la valoración de los episodios de una temporada de la serie (RF01- Mostrar valoraciones de episodios).



Figura 8.13: Captura de la pantalla de la información de una serie

Sin embargo, si la información que se obtiene de la película es una que no ha sido estrenada todavía, en la interfaz, se mostrará un contador con el tiempo que falta para que se estrene (RF06- Mostrar tiempo restante a que se estrene una película).

8.1.7. Recomendaciones aleatorias

Las recomendaciones aleatorias se han decidido mostrar en una interfaz aparte, debido a que se ha buscado maximizar el número de interacciones que tiene el usuario para entrenar las recomendaciones.

Esta pantalla, figura 8.14, incorpora cuatro botones centrales, con los que se entrenan las posteriores recomendaciones. En estos se encuentran los botones para marcar como visto (RF11- Marcar o desmarcar como visto), marcar como pendiente (RF16- Marcar o desmarcar como pendiente), ver más información y el botón de no me gusta.

También se puede obtener información sobre en qué plataforma se puede ver la película o series (RF05- Mostrar plataformas donde ver películas o series).

Por otra parte, existe la posibilidad de filtrar las recomendaciones con el botón en la parte superior de la pantalla, según el género y el tipo de contenido que se desea obtener, como vemos en la figura 8.15.

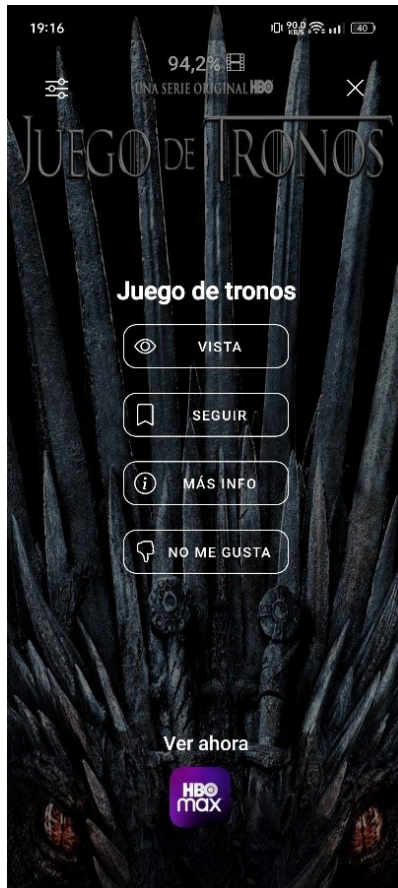


Figura 8.14: Captura de la pantalla de la recomendación de “Juego de Tronos”

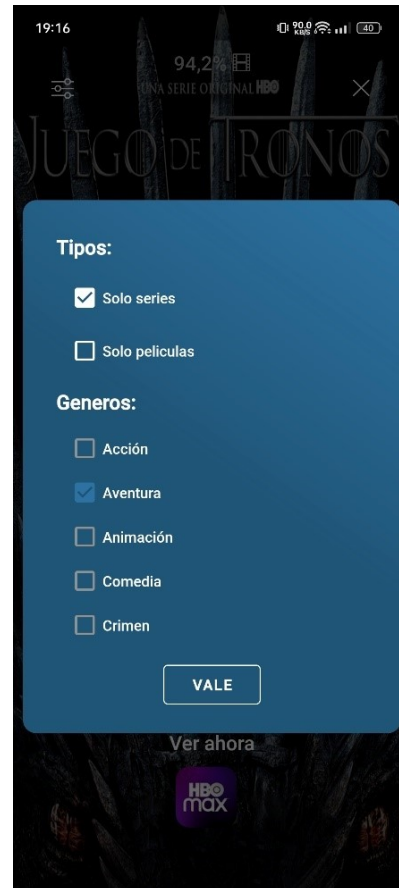


Figura 8.15: Captura de la pantalla para filtrar las recomendaciones aleatorias

8.1.8. Biblioteca de películas o series marcadas

Las películas o series que han sido marcadas como vistas, como seguidas o para ver más tarde, se pueden encontrar en la interfaz de la biblioteca, que vemos en la figura 8.16. En esta se puede observar que se divide el contenido guardado en el apartado de “PELÍCULAS”, “SERIES” o “ESTRENOS”. Cada uno de estos apartados tienen distintas opciones que se pueden aplicar a la vista.

En la vista de películas de la figura 8.16, observamos que se pueden ordenar por fecha, título, valoración y duración.

Mientras que en las series se muestran, como se observa en la figura 8.17, divididos por series no empezadas, series empezadas y series finalizadas.

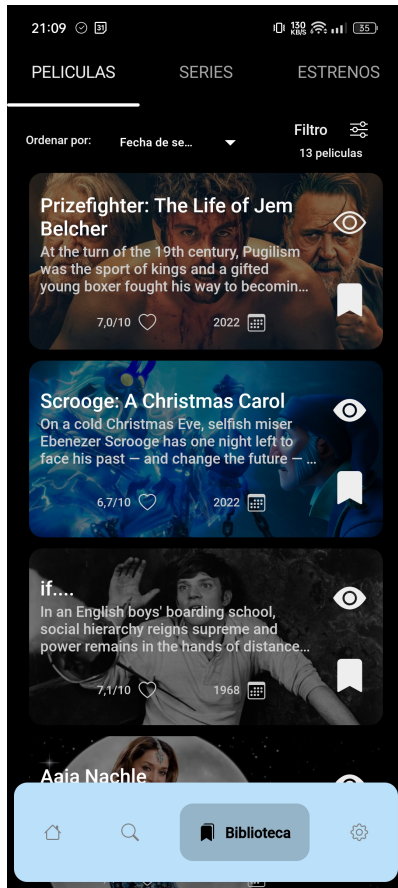


Figura 8.16: Captura de la pantalla de la biblioteca la sección de películas

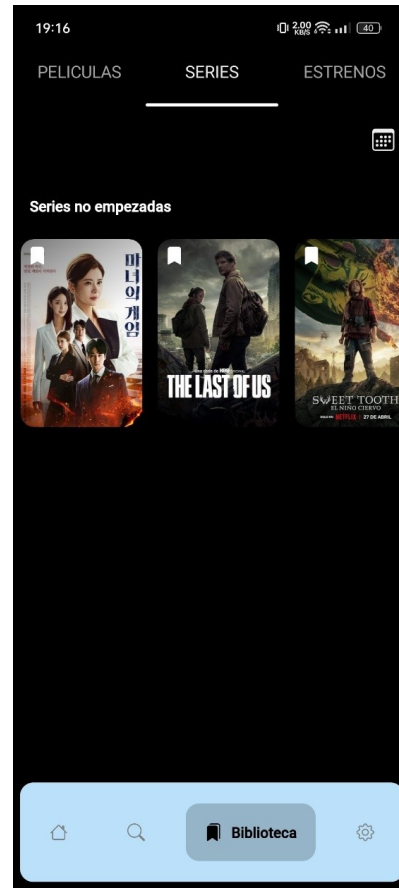


Figura 8.17: Captura de la pantalla de la biblioteca la sección de series

8.1.9. Ajustes

En esta interfaz se muestran todas las configuraciones que se pueden aplicar a la aplicación. En esta se muestra la opción de ocultar las series o películas ya vistas y marcadas o mostrar solo las que se muestran en las plataformas en las que el usuario tiene una suscripción.

Además, aquí se podrán añadir las plataformas a las que se está suscrito y controlar el usuario al que está registrado el perfil seleccionado.

Por otra parte, tenemos los botones para eliminar el perfil y un botón para ir a la interfaz de selección de perfiles.

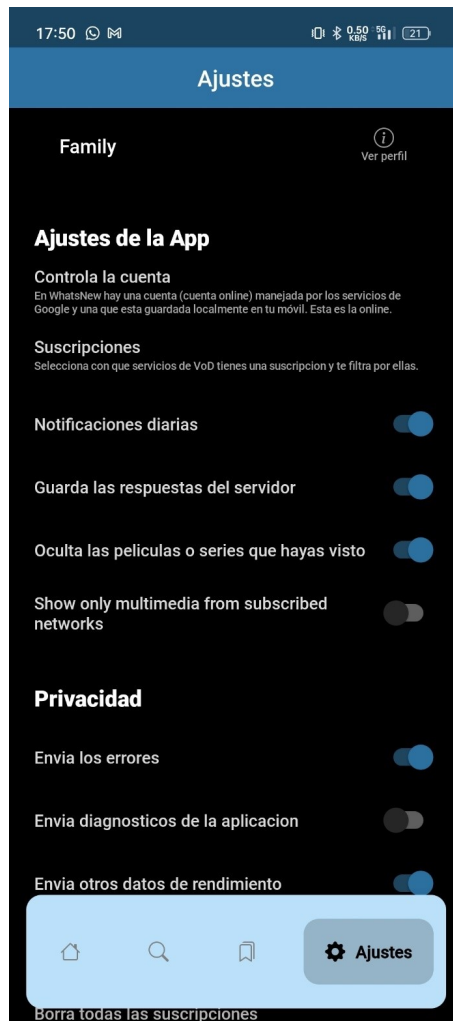


Figura 8.18: Captura de la pantalla de ajustes

CAPÍTULO 9

Conclusiones

Para concluir, este trabajo ha conseguido solucionar todos los problemas que han salido durante su fase de desarrollo y, además, llegando a cumplir todos los objetivos que se especificaron (ver apartado **Objetivos**).

En primer lugar, se ha ofrecido una gran variedad de herramientas para buscar y filtrar el contenido que el usuario quiere ver en sus plataformas de *streaming*. Haciendo uso de opciones tanto para filtrar únicamente por sus plataformas como filtrar el contenido ya marcado por genero y valoración. Además, de ofrecer una forma de optimizar sus recomendaciones de una forma rápida como con las ya mencionadas “Recomendaciones Aleatorias”.

En segundo lugar, se ha ofrecido al usuario la capacidad de marcar las series y películas que quiere seguir, marcar como pendiente o marcar como visto para tener una lista que le permita acceder a estos contenidos de una manera rápida.

Los problemas se detectaron en las fases iniciales del proyecto y se solucionaron gracias a una buena planificación y a la metodología elegida. Por ejemplo, uno de estos problemas, fue la dificultad para realizar una gran carga de peticiones tanto para las imágenes como para los datos a los servidores de TMDB. Por otra parte, también, resultó ser una complicación el guardado de la información del usuario, ya que, para que el coste de la aplicación para su desarrollo fuese nulo y ofrecer una disponibilidad razonable del servicio, se necesitaba utilizar unos servidores externos. Por suerte, los servicios de Firebase ofrecen varias herramientas gratuitas en la nube útiles para el proyecto a realizar, por lo que se optó por utilizarla.

Como experiencia personal este proyecto me ha servido para familiarizarme con las fases de desarrollo de un software. Utilizando herramientas como los casos de uso, para reconocer los requisitos del usuario, y siguiendo la metodología Kanban, para identificar las tareas que están pendientes y que están siendo realizadas. Además, este me ha permitido adquirir conocimientos sólidos sobre la utilización de las API y otros servicios externos, como Firebase. Por otra parte, debido a que se ha desarrollado en Android Nativo, este proyecto me ha permitido reforzar mis conocimientos sobre Java ya aprendidos durante la carrera.

9.1 Relación con los estudios cursados

Los estudios durante 4 años de Ingeniería Informática me han permitido tener un conocimiento amplio sobre Java el lenguaje utilizado en este proyecto. Además, me han permitido entender el uso de las API y del protocolo HTTP y sus variantes para aplicarlos a la comunicación que se realiza el dispositivo Android con los servidores de TMDB.

Por otra parte, gracias a los estudios cursados de desarrollo web y otros relacionados al desarrollo de interfaces para el usuario, me han ayudado al diseño de las interfaces mostradas en la aplicación, así como, los principios que deben seguir estas.

9.2 Desarrollos futuros

Aunque la aplicación final sea utilizable y haya cumplido todos los objetivos principales mencionados, en el futuro se desean hacer varias modificaciones para mejorar la funcionalidad de la aplicación y ofrecer un servicio más robusto al usuario. Se quieren mejorar los siguientes aspectos de la aplicación:

- Ofrecer un calendario donde se puedan ver los siguientes estrenos de películas o series marcadas.
- Mostrar al usuario notificaciones cuando las series que sigue tengan un capítulo nuevo no visto por el usuario.
- Implementar la función para que los usuarios puedan valorar dentro de la aplicación las series o películas.
- A largo plazo, se quiere que la aplicación ofrezca una mejora de velocidad al mostrar las recomendaciones dentro de la aplicación, implementando un servidor que actúe de intermediario entre la aplicación y la API.
- La disponibilidad de los filtros para las series tanto para las películas no estrenadas.

Bibliografía

- [1] Vida Económica, "Plataformas digitales: la televisión del siglo XXI", 20 de enero de 2020. [En línea] Disponible en: <https://www.vidaeconomica.com/2020/01/plataformas-digitales-televisión/>. [Último acceso: 15 de junio de 2023].
- [2] BCC, "Qué es la fatiga de decisión, el gran reto de las plataformas de *streaming* para no perder consumidores", 29 marzo 2021. [En línea] Disponible en: <https://www.bbc.com/mundo/noticias-56513675>. [Último acceso: 15 de junio de 2023].
- [3] Hubspot, "Metodología Agile: qué es y cómo aplicarla a tu proyecto", 1 de diciembre 2021. [En línea] Disponible en: <https://blog.hubspot.es/marketing/metodologia-agile>. [Último acceso: 15 de junio de 2023].
- [4] Atlassian, "Sprints de scrum". [En línea] Disponible en: <https://www.atlassian.com/es/agile/scrum/sprints>. [Último acceso: 15 de junio de 2023].
- [5] Ealde, "La metodología ágil Kanban: en qué consiste y para qué sirve", 04 de agosto de 2020. [En línea] Disponible en: <https://www.ealde.es/metodologia-agil-kanban/#:~:text=La%20metodolog%C3%ADa%20%C3%A1gil%20Kanban%20consiste,de%20los%20flujos%20%E2%80%9Cpull%E2%80%9D..> [Último acceso: 15 de junio de 2023].
- [6] Hubspot, "Qué es Trello, para qué sirve y cómo funciona", 10 de marzo de 2022. [En línea] Disponible en: <https://blog.hubspot.es/marketing/que-es-trello>. [Último acceso: 15 de junio de 2023].
- [7] Justwatch, "What is JustWatch?", 2021. [En línea] Disponible en: <https://support.justwatch.com/hc/en-us/articles/360020520097-What-is-JustWatch->. [Último acceso: 15 de junio de 2023].
- [8] Justwatch, "Aplicaciones JustWatch para teléfonos inteligentes y TVs". [En línea] Disponible en: <https://www.justwatch.com/es/aplicaciones>. [Último acceso: 15 de junio de 2023].
- [9] AGoodMovieToWatch, "Qué es AGoodMovieToWatch". [En línea] Disponible en: <https://agoodmovietowatch.com/about/>. [Último acceso: 15 de junio de 2023].
- [10] Google, "Qué es Google TV". [En línea] Disponible en: https://tv.google/intl/es_es/#hero. [Último acceso: 15 de junio de 2023].
- [11] Cloudflare, "¿Qué es la nube? | Definición de nube". [En línea] Disponible en: <https://www.cloudflare.com/es-es/learning/cloud/what-is-the-cloud/>. [Último acceso: 15 de junio de 2023].
- [12] Visuresolutions, "Qué es la especificación de requisitos: definición, mejores herramientas y técnicas". [En línea] Disponible en: <https://visuresolutions.com/>

- [es/blog/requirements-specification/#:~:text=Una%20especificaci%C3%B3n%20de%20requisitos%20es, trabajo%20futuro%20en%20el%20proyecto..](#) [Último acceso: 15 de junio de 2023].
- [13] Visuresolutions, "Qué son los requisitos no funcionales: ejemplos, definición, guía completa". [En línea] Disponible en: <https://visuresolutions.com/es/blog/non-functional-requirements/>. [Último acceso: 15 de junio de 2023].
- [14] Miro, "¿Para qué necesitas un diagrama UML?". [En línea] Disponible en: <https://miro.com/es/diagrama/que-es-diagrama-uml/>. [Último acceso: 15 de junio de 2023].
- [15] Cursosaula21, "¿Qué es Java y para qué sirve?". [En línea] Disponible en: <https://www.cursosaula21.com/que-es-java/>. [Último acceso: 15 de junio de 2023].
- [16] Seidor, "¿Qué es Firebase? ¿Qué ventajas ofrece en 2023 a nuestras apps?", 20 de febrero de 2023. [En línea] Disponible en: <https://www.seidor.com/blog/firebase-que-es>. [Último acceso: 15 de junio de 2023].
- [17] Xataka, "Qué es Github y qué es lo que le ofrece a los desarrolladores", 30 de octubre de 2019. [En línea] Disponible en: <https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores>. [Último acceso: 15 de junio de 2023].
- [18] GCFglobal, "¿Qué es y cómo funciona Lucidchart?". [En línea] Disponible en: <https://edu.gcfglobal.org/es/educacion-virtual/que-es-y-como-funciona-lucidchart/1/>. [Último acceso: 15 de junio de 2023].
- [19] CEI, "¿Qué es Figma y para qué sirve?". [En línea] Disponible en: <https://cei.es/que-es-figma/>. [Último acceso: 15 de junio de 2023].
- [20] Google, "Introducción a Android Studio". [En línea] Disponible en: <https://developer.android.com/studio/intro?hl=es-419>. [Último acceso: 15 de junio de 2023].
- [21] Openwebinars, "Qué es Gradle: La herramienta para ser más productivo desarrollando", 25 de febrero de 2020. [En línea] Disponible en: <https://openwebinars.net/blog/que-es-gradle/>. [Último acceso: 15 de junio de 2023].
- [22] Square, "OkHttp". [En línea] Disponible en: <https://square.github.io/okhttp/>. [Último acceso: 15 de junio de 2023].
- [23] Bumptech, "Glide". [En línea] Disponible en: <https://github.com/bumptech/glide>. [Último acceso: 15 de junio de 2023].
- [24] Digital55, "Firebase: qué es, para qué sirve, funcionalidades y ventajas", 17 de mayo de 2020. [En línea] Disponible en: <https://digital55.com/blog/que-es-firebase-funcionalidades-ventajas-conclusiones/>. [Último acceso: 15 de junio de 2023].
- [25] ThemovieDB, "API Reference". [En línea] Disponible en: <https://developer.themoviedb.org/reference/intro/getting-started>. [Último acceso: 15 de junio de 2023].
- [26] ThemovieDB, "API Reference - Recommendations". [En línea] Disponible en: <https://developer.themoviedb.org/reference/movie-recommendations>. [Último acceso: 15 de junio de 2023].

- [27] DigitalOcean, "Android RecyclerView Load More, Endless Scrolling", 3 de agosto de 2022. [En línea] Disponible en: <https://www.digitalocean.com/community/tutorials/android-recyclerview-load-more-endless-scrolling>. [Último acceso: 15 de junio de 2023].
- [28] Geekflare, "Prueba de aceptación del usuario (UAT): ¿Qué es?", 29 de noviembre de 2022. [En línea] Disponible en: <https://geekflare.com/es/user-acceptance-testing/>. [Último acceso: 15 de junio de 2023].
- [29] Google, "Políticas de Play". [En línea] Disponible en: <https://developer.android.com/distribute/play-policies?hl=es-419>. [Último acceso: 15 de junio de 2023].
- [30] UN, "Objetivos de desarrollo sostenible". [En línea] Disponible en: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>. [Último acceso: 15 de junio de 2023].
- [31] Hostinger, "¿Qué es JSON?", 10 de enero de 2023. [En línea] Disponible en: <https://www.hostinger.es/tutoriales/que-es-json>. [Último acceso: 15 de junio de 2023].
- [32] Themoviedb, "Página web de TheMovieDB". [En línea] Disponible en: <https://www.themoviedb.org/>. [Último acceso: 15 de junio de 2023].
- [33] Themoviedb, "Is there a hidden request rate limit?", 8 de julio de 2022. [En línea] Disponible en: <https://www.themoviedb.org/talk/62c7c1b258361b005fd2e747>. [Último acceso: 15 de junio de 2023].
- [34] Google, "Cómo interpretar el ciclo de vida de una actividad". [En línea] Disponible en: <https://developer.android.com/guide/components/activities/activity-lifecycle?hl=es-419>. [Último acceso: 15 de junio de 2023].
- [35] Google, "Fragmentos". [En línea] Disponible en: <https://developer.android.com/guide/components/fragments?hl=es-419>. [Último acceso: 15 de junio de 2023].
- [36] Google, "Cómo crear listas dinámicas con RecyclerView". [En línea] Disponible en: <https://developer.android.com/guide/topics/ui/layout/recyclerview?hl=es-419>. [Último acceso: 15 de junio de 2023].
- [37] Google, "Calidad básica de las apps". [En línea] Disponible en: <https://developer.android.com/docs/quality-guidelines/core-app-quality?hl=es-419>. [Último acceso: 15 de junio de 2023].

APÉNDICE A

Anexo ODS

Los Objetivos de Desarrollo Sostenible (ODS) [30] son los objetivos adoptados por las Naciones Unidas para poner fin a la pobreza, proteger el planeta y garantizar que para el 2030 todas las personas disfruten de paz y prosperidad.

Tabla A.1: Tabla de ODS

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.				X
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.				X
ODS 10. Reducción de las desigualdades.			X	
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.			X	
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.			X	
ODS 17. Alianzas para lograr objetivos.				X

Son 17 ODS, pero esta aplicación puede estar relacionada con tres de ellos: la acción por el clima (13), la reducción de las desigualdades (10) y la acción por la paz, justicia e instituciones sólidas (16). Aunque en un principio pueda parecer que una aplicación de entretenimiento no tiene relación directa con estos temas, es importante considerar el alcance y las posibles implicaciones de su uso en el contexto actual.

En primer lugar, la aplicación promueve la sensibilización y la educación sobre temas relacionados con el cambio climático y la desigualdad. A través de su funcionalidad de recomendación y búsqueda de películas y series, la aplicación destaca contenidos que abordan estas temáticas de manera directa o indirecta. Al proporcionar a los usuarios información relevante y accesible, la aplicación puede ayudar a difundir mensajes y perspectivas relacionadas con la acción por el clima y la reducción de las desigualdades. Esto puede contribuir a crear conciencia y fomentar la reflexión sobre estos temas entre un amplio público.

La aplicación también puede incentivar prácticas más sostenibles en el consumo de contenidos audiovisuales. Al proporcionar información sobre la disponibilidad y la forma de acceso a películas y series, la aplicación fomenta opciones de visualización más sostenibles, como la elección de formatos digitales en lugar de soportes físicos que distribuyan películas o series. Por otra parte, cabe destacar que, al utilizar un dispositivo móvil, estos suelen consumir menos energía que un ordenador personal, siendo esta la opción menos contaminante. Al usar herramientas centralizadas como el uso de servicios externos, como Firebase y la API de TheMovieDB, para proporcionar servicio a la aplicación se consigue reunir el gasto de energía en un solo sitio optimizando el gasto energético en contraposición a utilizar servidores privados para proporcionar el servicio a la aplicación y tener una base de datos únicamente para la aplicación.

En relación a la reducción de las desigualdades, la aplicación proporciona acceso a contenidos provenientes de diferentes culturas, países y realidades socio-económicas. Además de poder filtrar las películas o series según el país de donde es el usuario. Esto puede contribuir a ampliar la diversidad de narrativas y representaciones presentes en la industria audiovisual, desafiando estereotipos y promoviendo una mayor inclusión.

Además, la aplicación puede facilitar el acceso a contenido audiovisual a personas que enfrentan barreras económicas o geográficas. A través de su funcionalidad de búsqueda y filtrado por plataformas, la aplicación puede ayudar a los usuarios a descubrir opciones gratuitas o de bajo costo para acceder a películas y series. Esto es especialmente relevante para comunidades con menos recursos o que se encuentren en zonas rurales o con acceso limitado a salas de cine o servicios de *streaming*. Al brindar opciones accesibles, la aplicación puede contribuir a reducir la brecha de acceso al entretenimiento y a fomentar una mayor equidad en este ámbito.

Es importante destacar que, si bien la aplicación puede tener un impacto positivo, también existen desafíos y consideraciones importantes a tener en cuenta. Por ejemplo, es fundamental garantizar que los contenidos promovidos sean rigurosos, verídicos y no contribuyan a la desinformación o a la propagación de estereotipos perjudiciales. Es fundamental que se eviten estereotipos, prejuicios y representaciones negativas que puedan perpetuar la violencia o la discriminación. La aplicación aspira a ser un espacio inclusivo y respetuoso, promoviendo valores que contribuyan al fortalecimiento de las instituciones democráticas y al fomento de una sociedad justa y pacífica.