

TRABAJO DE FIN DE GRADO

APLICACIÓN PARA SAMSUNG SMART TV DE SEGUIMIENTO AUDIOVISUAL EN PLATAFORMAS DE STREAMING

Andrea Morey Sánchez

Grado de: Grado en Ingeniería Informática

Facultad de: Escuela Politécnica Superior

Año académico 2021-22

APLICACIÓN PARA SAMSUNG SMART TV DE SEGUIMIENTO AUDIOVISUAL EN PLATAFORMAS DE STREAMING

Andrea	Morey	Sánchez
---------------	-------	---------

Trabajo de Fin de Grado

Facultad de: Escuela Politécnica Superior

Universitat de les Illes Balears

Año académico 2021-22

Palabras clave del trabajo:

Samsung Smart TV, plataformas streaming, API, Tizen Studio

Tutores: Antoni Oliver Tomàs y Antoni Bibiloni Coll

	Au	tor/a	Tut	or/a
Autorizo a la Universitat a incluir este trabajo en el repositorio institucional para consultarlo en acceso abierto y difundirlo en línea, con finalidades exclusivamente académicas y de investigación	Sí	No	Sí	No
	\boxtimes			



Tabla de contenido

Tabla de contenido	i
Índice de figuras	v
Índice de tablas	ix
Nomenclatura	xi
Resumen	xiii
Capítulo 1. Introducción	1
1.1. Contextualización	1
1.2. Motivación	2
1.3. Objetivos	3
1.4. Alcance del proyecto	3
1.5. Estructura del documento	4
Capítulo 2. Estado del arte	5
2.1. Soluciones existentes a problemas parecidos	5
2.1.1. TV Time	5
2.1.2. Next Episode	5
2.1.3. JustWatch	6
2.2. Herramientas que pueden utilizarse para resolver este problema	6
2.2.1. The Movie Database	6
2.2.2. Tizen Studio	7
2.2.3. HTML, CSS y JavaScript	7
2.2.4. Node.js	7
2.2.5. SQL	8
2.2.6. MySQL WorkBench	8
Capítulo 3. Desarrollo del proyecto	9
3.1. Gestión del proyecto	9
3.1.1. Metodología de desarrollo	9
3.1.2. Planificación temporal	9
3.2. Análisis	10
3.2.1. Usuarios	10

3.2.2. Requisitos funcionales	
3.2.2.1. Perfil	
3.2.2.2. Amigos	
3.2.2.3. Tendencias	
3.2.2.4. Búsqueda	
3.2.2.5. Mis series	
3.2.2.6. Mis películas	12
3.2.2.7. Favoritos	
3.2.2.8. Recomendaciones	12
3.2.3. Requisitos de sistema	13
3.2.3.1. Seguridad	13
3.2.3.2. Usabilidad	
3.3. Diseño	13
3.3.1. Casos de uso	
3.3.2. Modelo de datos	15
3.3.2.1. Modelo conceptual	
3.3.2.1.1. Clases y atributos	15
3.3.2.1.2. Relaciones	18
3.3.2.2. Modelo relacional	
3.3.2.3. Creación de la base de datos	
3.3.3. Diagramas de estado	22
3.3.3.1. Diagrama de estados de la relación entre una serie y un usuario re	_
3.3.3.2. Diagrama de estados de la relación entre una película y un registrado	
3.3.4. Diagramas de secuencia	26
3.3.4.1. Perfil	26
3.3.4.2. Amigos	26
3.3.4.3. Tendencias	26
3.3.4.4. Búsqueda	26
3.3.4.5. Mis series	26
3.3.4.6. Mis películas	26
3.3.4.7. Favoritos	26
3 3 4 8 Recomendaciones	27

2.2 F. Disago annuitanténias	24
3.3.5. Diseño arquitectónico	
3.3.6. Diseño modular	
3.4. Implementación	
3.4.1. API RESTful	
3.4.1.1. Uso de la API de The Movie Database	
3.4.1.2. Estructura del proyecto	
3.4.1.3. Implementación de los módulos	
3.4.1.3.1. Perfil	
3.4.1.3.2. Amigos	
3.4.1.3.3. Tendencias	
3.4.1.3.4. Búsqueda	
3.4.1.3.5. Mis series	
3.4.1.3.6. Mis películas	
3.4.1.3.7. Favoritos	
3.4.1.3.8. Recomendaciones	
3.4.2. Aplicación GoWatch	
3.4.2.1. Estructura del proyecto	
3.4.2.2. Interfaz de usuario	
3.4.2.3. Implementación de los módulos	
3.4.2.3.1. Perfil	
3.4.2.3.2. Amigos	46
3.4.2.3.3. Tendencias	
3.4.2.3.4. Búsqueda	47
3.4.2.3.5. Mis series	48
3.4.2.3.6. Mis películas	48
3.4.2.3.7. Favoritos	48
3.4.2.3.8. Recomendaciones	48
3.5. Instalación	49
Capítulo 4. Resultados	51
Capítulo 5. Conclusiones	61
Capítulo 6. Referencias bibliográficas	63
Apéndice A. Creación base de datos	65
Apéndice B. Correos	69

Ial	าไว	പെ	COL	ኅt۵	nido
ı aı	JIO	uc	COL	ILC	muu

Apéndice C. Creación de una aplicación web con Tizen Studio73

Índice de figuras

Figura 1. Evolución del porcentaje de hogares con televisores inteligentes en España entre 2015 y 2021
Figura 2. Cuota del mercado global de televisores (2021)
Figura 3. Diagrama de Gantt. Calendario del proyecto9
Figura 4. Diagrama de casos de uso
Figura 5. Modelo conceptual
Figura 6. Representación visual de la base de datos
Figura 7. Diagrama de estados de la relación entre una serie y un usuario registrado 22
Figura 8. Diagrama de estados de la relación entre una película y un usuario registrado.
Figura 9. Diagrama de secuencia "RF1. Un usuario no registrado que disponga de una Samsung <i>Smart TV</i> podrá registrarse en la aplicación"
Figura 10. Diagrama de secuencia "RF10. Un usuario logueado que ha recibido una solicitud de amistad puede aceptarla o rechazarla. En cualquier caso, se notificará al usuario emisor de la solicitud"
Figura 11. Diagrama de secuencia "RF12. Cualquier usuario podrá consultar las tendencias diarias. Es decir, podrá ver las series, películas e integrantes más populares de ese día".
Figura 12. Diagrama de secuencia "RF15. Cualquier usuario podrá acceder directamente desde la aplicación a las plataformas <i>streaming</i> en las que se encuentran disponibles las series y películas"
Figura 13. Diagrama de secuencia "RF18. Un usuario logueado podrá marcar (o desmarcar) que ha visto un capítulo"
Figura 14. Diagrama de secuencia "RF28. Un usuario logueado podrá marcar (o desmarcar) que está viendo una película. Opcionalmente podrá indicar en qué plataforma streaming la está visualizando"
Figura 15. Diagrama de secuencia "RF36. Un usuario logueado podrá consultar las series, películas e integrantes que tiene marcadas como favoritos"
Figura 16. Diagrama de secuencia "RF37. Un usuario logueado podrá enviar recomendaciones de series o películas a sus amigos"
Figura 17. Arquitectura del sistema31
Figura 18. Logo de la aplicación GoWatch
Figura 19. Fragmento de código de estructura básica de gestión del foco con CAPH44

Índice de figuras

Figura 20.	Formulario de registro en la aplicación GoWatch	45
Figura 21.	Formulario de cambio de dirección de correo electrónico	46
Figura 22.	Formulario de cambio de contraseña.	46
Figura 23.	Pantalla inicial de la aplicación GoWatch (sin ningún usuario logueado)	51
Figura 24.	Pantalla inicial de la aplicación GoWatch (con un usuario logueado)	52
Figura 25.	Pantalla de registro en GoWatch o de acceso sin cuenta	52
Figura 26.	Pantalla de "Tendencias" (inicial) de la aplicación GoWatch	53
Figura 27.	Pantalla de "Tendencias" (series) de la aplicación GoWatch	53
Figura 28.	Pantalla de "Perfil" de la aplicación GoWatch.	54
Figura 29.	Pantalla de cambio de avatar de la aplicación GoWatch	54
Figura 30.	Pantalla de "Búsqueda" (con teclado virtual) de GoWatch	55
Figura 31.	Pantalla de "Búsqueda" (con resultados) de la aplicación GoWatch	55
_	Pantalla de "Búsqueda" (con resultados filtrados por tipo de contenido) de GoWatch.	
Figura 33.	Pantalla de "Mis series" (Siguiendo) de la aplicación GoWatch	56
Figura 34.	Pantalla de la serie Stranger Things de la aplicación GoWatch	57
Figura 35.	Pantalla de la temporada 1 de Stranger Things de la aplicación GoWatch	57
Figura 36.	Pantalla de "Mis películas" (Vistas) de la aplicación GoWatch	58
Figura 37.	Pantalla de la película The Batman de la aplicación GoWatch	58
Figura 38.	Pantalla de "Favoritos" (Integrantes) de la aplicación GoWatch	59
Figura 39.	Pantalla de la integrante Zendaya de la aplicación GoWatch	59
Figura 40.	Pantalla de "Amigos" de la aplicación GoWatch	60
Figura 41.	Pantalla de "Recomendaciones" (nuevas) de la aplicación GoWatch	60
Figura 42.	Correo de notificación de nuevos capítulos disponibles	69
Figura 43.	Correo de solicitud de cambio de dirección de correo electrónico	69
Figura 44.	Correo de solicitud de cambio de contraseña	70
Figura 45.	Correo de nueva solicitud de amistad recibida	70
Figura 46.	Correo de aceptación de solicitud de amistad	71
Figura 47.	Correo de denegación de solicitud de amistad	71
Figura 48.	Correo de pérdida de un amigo.	71
Figura 49.	Correo de recomendación recibida	72
Figura 50.	Fragmento de código con privilegios Tizen.	73
Figura 51	Proceso convencional de instalación vs WITs	74

ura 52. Parámetros de configuración de WITs7	4
ara 32. Farametros de comigaración de Wiris.	•

Índice de tablas

Tabla 1. Aclaraciones diagrama casos de uso	14
Tabla 2. Aclaraciones tablas base de datos	20
Tabla 3. Descripción de estados del diagrama de estados de la relación entre una se un usuario registrado.	-
Tabla 4. Descripción de transiciones del diagrama de estados de la relación entre una s y un usuario registrado	
Tabla 5. Descripción de estados del diagrama de estados de la relación entre una peli y un usuario registrado	
Tabla 6. Descripción de transiciones del diagrama de estados de la relación entre película y un usuario registrado	

Nomenclatura

TV Television.

API Application Programming Interface.

HTML Hypertext Markup Language.

iOS *iPhone Operating System.*

TMDB The Movie Database.

REST Representational State Transfer.

JSON JavaScript Object Notation.

IDE Integrated Development Environment.

CSS Cascade Style Sheet.

NPM Node Project Manager.

PHP Hypertext Preprocessor.

SQL Structured Query Language.

SSH Secure Shell.

RF Requisito Funcional.

RNF Requisito No Funcional.

ID Identificador.

LTIM Laboratorio de Tecnologías de la Información Multimedia.

SDK Software Development Kit.

UI User Interface.

HAXM Intel Hardware Accelerated Execution Manager.

DDoS Distributed Denial of Service.

Resumen

Debido al cambio de tendencia que se está produciendo en el consumo televisivo, las plataformas *streaming* han cobrado un papel muy importante en nuestro tiempo de ocio. La mayoría de las personas tienen contratados los servicios de varias plataformas, lo cual puede complicar llevar un control de todo el contenido de cada una de ellas. Por esto mismo, en este proyecto se llevará a cabo el desarrollo de una aplicación, concretamente para las Samsung *Smart TV*, que proponga una solución a este problema. Con dicha aplicación, los usuarios podrán saber en qué plataformas *streaming* se encuentra disponible un contenido que deseen ver y si se registran, podrán tener un seguimiento personalizado de los diferentes contenidos que están viendo en todas sus plataformas *streaming*.

El backend de la aplicación estará formado por una API creada con Node.js que se encargará de interactuar con la propia aplicación instalada en el televisor, con la base de datos y con servicios de Internet que proporcionarán la información de los contenidos que se mostrarán en la aplicación. Por otra parte, el frontend será una aplicación web creada con Tizen Studio, el IDE oficial de Samsung para el desarrollo de aplicaciones.



Capítulo 1. Introducción

En este primer capítulo se expondrá una contextualización inicial del proyecto, la motivación personal para la elección de este tema, los objetivos que se pretenden conseguir y el alcance que abarcará el proyecto. Por último, se esquematizará la estructura que seguirá el documento tras este capítulo introductorio.

1.1. Contextualización

A principios de la década de 2010, los televisores inteligentes o *Smart TV* llegaron al mercado de los televisores para revolucionarlo por completo. Tal y como se refleja en la gráfica representada en la Figura 1 [1], en 2021 aproximadamente el 60% de la población española ya disponía de un televisor inteligente en sus hogares.

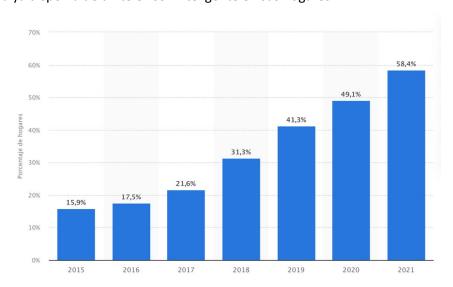


Figura 1. Evolución del porcentaje de hogares con televisores inteligentes en España entre 2015 y 2021.

La clave del éxito de esta nueva generación de televisores es que además de incorporar las funcionalidades básicas de un televisor tradicional, tienen integrados Internet y características de la web 2.0.

Gracias a estas nuevas incorporaciones, los televisores inteligentes son capaces de proporcionar muchas más formas de entretenimiento. Un claro ejemplo de ello son las plataformas *streaming*, que permiten la transmisión de contenido multimedia a través de Internet sin necesidad de los proveedores tradicionales.

En 2015, aterrizó en España Netflix, una de las plataformas *streaming* más populares a nivel mundial y desde aquel entonces la seguirían muchas más. A partir de este momento, empezó a producirse un cambio inevitable en la manera de consumir televisión, llegando

1. Introducción

al punto en el que, tal y como se constata en un artículo del diario digital El Independiente [2], "el consumo televisivo había sufrido en 2021 una bajada histórica (...). No se registraban datos tan bajos desde 1993". Esto se traduce en que los telespectadores prefieren cada vez más las plataformas *streaming* por encima de la televisión convencional. Esta diferencia de consumo es mucho más significativa entre la población joven.

Cambiando el foco de la utilización a la producción, tenemos que uno de los principales fabricantes de *Smart TV* es Samsung, consiguiendo en 2021 su decimosexto año consecutivo liderando el mercado de los televisores. En la gráfica de la Figura 2 [3], se refleja como Samsung superó a todos sus competidores directos acaparando el 29,5% del mercado global de televisores.

Samsung es el propietario del sistema operativo Tizen, que está basado en Linux y es de código abierto. Además, ofrecen a los desarrolladores la posibilidad de instalar en sus dispositivos con sistema operativo Tizen aplicaciones nativas o basadas en HTML5.

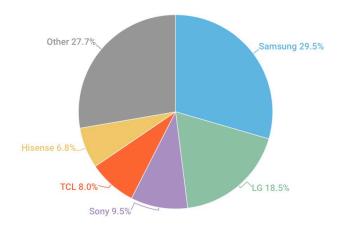


Figura 2. Cuota del mercado global de televisores (2021).

1.2. Motivación

En consonancia con el cambio de tendencia en el consumo televisivo, la gran mayoría de los usuarios de plataformas *streaming* tienen contratados los servicios de más de una de estas plataformas. Cada plataforma ofrece un catálogo de contenido multimedia diferente que se va actualizando periódicamente, lo cual dificulta saber dónde se encuentra disponible una película o una serie en un momento concreto.

Además, como gran aficionada a las series y usuaria de múltiples plataformas *streaming*, considero que se genera un cierto descontrol cuando estás siguiendo simultáneamente diversas series que se encuentran distribuidas entre las diferentes plataformas que tienes contratadas. En muchas ocasiones, se llega a una situación en la que no recuerdas qué series estabas siguiendo o no sabes en cuál de todas tus plataformas estaba la serie que te apetecía seguir viendo, por lo que tienes que ir entrando en la aplicación de cada una de tus plataformas *streaming* para buscarla hasta que por fin das con esa serie.

1.3. Objetivos

El principal objetivo que persigue este proyecto es el desarrollo de GoWatch, una aplicación interactiva basada en HTML5 para Samsung *Smart TV*. Esta aplicación permitirá a los usuarios de televisores Samsung *Smart TV* y de plataformas *streaming* establecer un control centralizado de todas las series y películas que están viendo en el momento, que querrán ver en un futuro y además, les permitirá mantener un historial de todas las que ya han visto. También podrán guardar las series, películas o integrantes (actores, directores, guionistas, etc.) que el usuario considere que son sus favoritos para poder recurrir a estos de una manera más directa.

Para poder realizar este seguimiento personalizado de cada usuario, será imprescindible que este se registre en la aplicación. Sin embargo, habrá una serie de funcionalidades abiertas a todo usuario que disponga de la aplicación y de un televisor Samsung *Smart TV*, sin la necesidad de crearse una cuenta. De entre estas, se pueden destacar la posibilidad de visualizar las tendencias diarias de series, películas e integrantes o la búsqueda de cualquiera de estos contenidos que se acaban de enumerar y la posterior visualización de sus detalles.

Será posible la obtención de todos estos datos actualizados gracias a servicios de Internet que proporcionan esta información de manera gratuita.

1.4. Alcance del proyecto

En este apartado se va a delimitar el alcance del proyecto indicando una serie de restricciones y las funcionalidades que no se van a cubrir.

La aplicación estará disponible únicamente en español y además, la disponibilidad del contenido de las diferentes plataformas *streaming* será el que estas ofrezcan en su servicio en España.

No será posible la sincronización de las plataformas *streaming* del usuario con la aplicación GoWatch. Esto significa que será el usuario quien deba indicar qué capítulos ha visto o si ha finalizado de ver una película en la aplicación. Tampoco se guardará el minuto hasta el que un usuario ha visto un capítulo o una película, ya que este requisito recae en las propias plataformas *streaming*.

No estarán disponibles recomendaciones basadas en contenido visto previamente o en los gustos del usuario, tal y como hacen actualmente la mayoría de las plataformas *streaming*, sino que solo se podrá visualizar contenido similar a una serie o película concreta.

Los usuarios podrán acceder desde GoWatch a las aplicaciones de las plataformas streaming, sin embargo no se podrá acceder directamente a un contenido específico sino que solo será posible acceder a la página de inicio.

1. Introducción

desplazar el cursor por todo el teclado para llegar hasta el siguiente carácter, haciendo la escritura en estos teclados una tarea lenta y tediosa.

Por último, cabe mencionar que la aplicación no se publicará en la tienda de aplicaciones de las Samsung *Smart TV*.

1.5. Estructura del documento

Después de este primer capítulo introductorio, el documento consta de los siguientes capítulos:

- Capítulo 2. Estado del arte Este es un capítulo de investigación en el que se reflejarán las soluciones actuales a problemas similares al que se ha planteado.
- Capítulo 3. Desarrollo del proyecto Inicialmente se comentará la gestión del proyecto. A continuación, se detallarán las fases de análisis, diseño e implementación. Los últimos apartados de este capítulo los formarán las pruebas y la instalación de la aplicación.
- Capítulo 4. Resultados Aquí se mostrarán pruebas gráficas de los resultados conseguidos tras finalizar la fase de desarrollo.
- Capítulo 5. Conclusiones En este capítulo se recogerán las reflexiones y posibles mejoras una vez acabado el proyecto.
- Capítulo 6. Referencias bibliográficas Contendrá un listado de todas las referencias utilizadas a lo largo del documento.

Capítulo 2. Estado del arte

En este capítulo se discutirán las diferentes soluciones a problemas similares que se pueden encontrar a día de hoy, así como sus similitudes y diferencias con la aplicación a desarrollar. Por último, se hará un análisis y selección de las herramientas que se emplearan para desarrollar GoWatch.

2.1. Soluciones existentes a problemas parecidos

Otras personas ya se habían percatado del problema plasmado en el capítulo anterior, llevar un control de todas las series y películas de tus plataformas *streaming* no es una tarea nada sencilla. Por este motivo, actualmente se pueden encontrar una inmensa variedad de herramientas que facilitan esta tarea a los usuarios.

Una característica que tienen en común todas estas soluciones ya existentes es que están disponibles solo en aplicaciones web o aplicaciones móviles. Por este motivo, la principal diferencia con la aplicación que se pretende desarrollar es que esta estará diseñada específicamente para las Samsung *Smart TV*.

A continuación, se expondrán 3 ejemplos de aplicaciones disponibles para llevar un control de las series y películas que un usuario sigue en sus diferentes plataformas streaming.

2.1.1. TV Time

TV Time [4] es una plataforma, propiedad de Whip Media, de seguimiento de series y películas que cuenta con más de 22 millones de usuarios. Obtienen los datos de su comunidad global Content Value Management™ (CVM) Advanced Insights [5]. Está disponible como aplicación web y como aplicación móvil para iOS y Android.

Una de sus características más destacables es que además de ser una plataforma de seguimiento, es una plataforma social, ya que sus usuarios pueden interactuar entre ellos e incluso establecer relaciones de amistad. Siguiendo su ejemplo, GoWatch también tendrá un componente social y será posible tener amigos en la aplicación y compartir recomendaciones de series y películas entre usuarios que sean amigos.

Tienen una restricción importante de que solo los usuarios registrados pueden usar su aplicación. Sin embargo, tal y como se ha mencionado en el capítulo anterior, GoWatch tendrá una serie de funcionalidades generales abiertas a cualquier usuario y para poder realizar las funciones de seguimiento personalizado ya será necesario que el usuario disponga de una cuenta.

2.1.2. Next Episode

Next Episode [6] es una aplicación móvil disponible para iOS y Android, que originalmente se creó como una página web para ayudar a organizar y a mantener un seguimiento de

2. Estado del arte

las películas y series favoritas de los usuarios. Obtiene los datos a través de las APIs de TheTVDB [7] y TVmaze [8] para las series y de The Movie Database [9] para las películas.

A diferencia de TV Time, Next Episode ofrece algunas funcionalidades abiertas a cualquier usuario sin necesidad de registrarse. Una de estas es la posibilidad de visualizar un calendario con los nuevos capítulos de cada día. Sin embargo, no ofrecen la posibilidad de mostrar un calendario personalizado con solo las series que sigue cada usuario. Tomando como referencia el mostrar los estrenos de capítulos, en GoWatch se dispondrá de notificaciones vía correo electrónico para avisar de la disponibilidad de nuevos capítulos, pero exclusivamente de las series que el usuario esté siguiendo.

2.1.3. JustWatch

JustWatch [10] es un motor de búsqueda para contenido en *streaming*. Disponen de una aplicación web y una aplicación móvil, disponible para iOS y Android, que se caracteriza por permitir a sus usuarios saber en qué plataformas están disponibles las series y películas según el país del usuario. Como las anteriores aplicaciones que se han nombrado en esta sección, también permite a los usuarios mantener un registro de sus series y películas. La aplicación se puede usar en su totalidad tanto sin estar registrado como si se dispone de una cuenta, pero si el usuario desea que la aplicación se sincronice en sus diferentes dispositivos debe registrarse. El usuario puede indicar las plataformas *streaming* que tiene contratadas y así tendrá la opción de filtrar el contenido por sus plataformas. Precisamente siguiendo este ejemplo, en GoWatch se podrán filtrar las series y películas que un usuario esté viendo por las plataformas que el usuario tenga registradas.

Otro punto a destacar de esta aplicación es que recolectan datos de todos sus usuarios para ofrecer recomendaciones personalizadas basadas en usuarios con gustos similares. Esto quedará totalmente fuera del alcance de GoWatch, ya que no se desarrollará ningún tipo de inteligencia artificial capaz de proporcionar esta información.

2.2. Herramientas que pueden utilizarse para resolver este problema.

A continuación, se explicarán las herramientas que se utilizarán para resolver el problema que se ha planteado en el primer capítulo del documento. En el caso de que existan herramientas similares a las que se han escogido, se justificará el motivo de su elección por delante de las otras.

2.2.1. The Movie Database

Se pueden identificar 2 facetas de TMDB. La primera de estas, es como cualquiera de las aplicaciones que se han explicado en la sección anterior, ofrecer al usuario general la posibilidad de realizar un seguimiento de sus series y películas, conocer cuáles son las tendencias y encontrar nuevo contenido multimedia. Por otro lado, ofrecen una API RESTful [11] gratuita que proporciona todo tipo de datos (en formato JSON) relacionados con películas, series e integrantes. Su API también contempla la gestión de usuarios, pero para ello estos deben estar registrados en TMDB.

De entre todas las APIs que ofrecen metadatos de películas, series e integrantes, he encontrado que esta era la más adecuada ya que proporciona desde un mismo sitio todos los datos que se necesitarán para GoWatch, mientras que con las otras APIs disponibles se tendría que utilizar una combinación de ellas para obtener la información de manera completa. Por ejemplo, la API de TMDB también proporciona la disponibilidad de las series y películas en las diferentes plataformas *streaming* ya que internamente se encarga de obtener esta información a través de JustWatch.

A fin de cuentas, esta es una herramienta muy similar a lo que se desarrollará en este proyecto, ya que por una parte tiene una API que proporciona datos y gestiona usuarios y por otra, está la aplicación que realmente utilizan los usuarios finales.

2.2.2. Tizen Studio

Tizen Studio es un conjunto de herramientas para desarrollar aplicaciones Tizen nativas o aplicaciones web. Este conjunto está compuesto por un IDE, un emulador, un simulador y una serie de herramientas más necesarias para poder crear una aplicación para un televisor Samsung *Smart TV*. Tizen Studio es compatible con Windows, Ubuntu y MacOS.

2.2.3. HTML, CSS y JavaScript

HTML, CSS y JavaScript son los lenguajes básicos para cualquier *frontend* de una aplicación web, por tanto estos serán los que se emplearán para realizar la aplicación GoWatch.

2.2.4. Node.js

Node.js es un *framework* de tiempo de ejecución dirigido por eventos y basado en JavaScript. Se puede emplear tanto en el lado del cliente como del servidor, aunque en este proyecto se utilizará exclusivamente para el *backend*. Gracias a que permite que se produzcan varias conexiones de manera simultánea, no se producen bloqueos entre procesos.

Una de sus principales características son sus módulos, un conjunto de funciones y objetos JavaScript que facilitan en gran medida el uso de este *framework*. Estos se pueden obtener gracias a su gestor de paquetes NPM.

La principal alternativa a Node.js es PHP, un lenguaje de programación muy frecuente en el desarrollo de páginas web. La mayor ventaja que presenta Node.js frente a PHP es la escalabilidad, pudiendo tener miles de conexiones concurrentes más. Esto se debe a que PHP genera un nuevo hilo para cada conexión y cada uno de estos ocupa un cierto espacio de memoria, pudiendo llegar a colapsarla cuando se alcanza un cierto número de conexiones simultáneas. Sin embargo, en Node.js cada conexión nueva dispara una ejecución de evento dentro de su motor. Por este motivo, Node.js es perfecto para APIs RESTful, donde se espera recibir un gran número de peticiones y se debe dar una respuesta lo más rápido posible.

2. Estado del arte

2.2.5. SQL

SQL es un lenguaje de programación para trabajar con bases de datos relacionales. Este permite crear la propia base de datos, obtener información de esta, insertar nuevos datos y la manipulación de los datos ya almacenados.

2.2.6. MySQL WorkBench

Es una herramienta visual que incorpora el sistema de gestión de base de datos relacionales MYSQL. Además, ofrece la posibilidad de establecer conexiones con bases de datos remotas a través de túneles SSH

Capítulo 3. Desarrollo del proyecto

Este es el capítulo principal del documento, ya que es en el que se refleja todo el trabajo llevado a cabo para realizar el proyecto propuesto. Inicialmente se hará una breve explicación de la gestión del proyecto, tanto en metodología de desarrollo como en planificación temporal, y después lo seguirán las fases de análisis, diseño e implementación. Por último, se hará una breve explicación de cómo realizar la instalación de la aplicación.

3.1. Gestión del proyecto

En este apartado se explicará la metodología seleccionada para el desarrollo del proyecto, así como la planificación de las diferentes tareas que se tendrán que llevar a cabo para poder culminar el proyecto.

3.1.1. Metodología de desarrollo

Para desarrollar el proyecto se va a seguir un modelo de cascada. Esto implica que las diferentes fases en las que se divide el proyecto se realizarán de secuencialmente, de manera que la siguiente fase no comenzará hasta que finalice la previa (con excepción de la documentación, que se irá realizando a lo largo de gran parte del proyecto).

3.1.2. Planificación temporal

En la Figura 3 se muestra un diagrama de Gantt donde se refleja la planificación que se ha realizado de las diferentes fases y tareas del proyecto previo a su comienzo.

CALENDARIO DEL PROYECTO

Previsión de tareas para los próximos meses

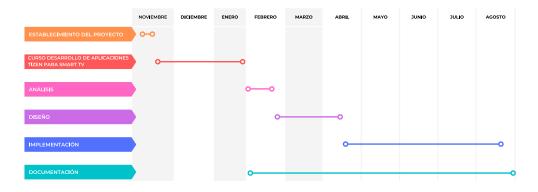


Figura 3. Diagrama de Gantt. Calendario del proyecto.

3.2. Análisis

El análisis es la fase inicial de todo desarrollo de proyecto. En esta se definen los usuarios finales de la aplicación y los requisitos funcionales y no funcionales que deberá cumplir. Los resultados de esta fase servirán de base para la fase de diseño.

3.2.1. Usuarios

La aplicación estará dirigida a las personas que en sus hogares disponen de un televisor Samsung *Smart TV* y que hacen uso frecuente de las plataformas *streaming*.

Se distinguirán 3 tipos de usuarios de la aplicación:

- 1. Usuario no registrado. Es aquel usuario que no dispone de una cuenta en la aplicación.
- 2. Usuario sin loguear. Es aquel usuario que dispone de una cuenta pero tiene su sesión cerrada.
- 3. Usuario logueado. Es aquel usuario que dispone de una cuenta y tiene la sesión abierta.

En los siguientes apartados, cuando se mencionen usuarios registrados, se estará haciendo referencia tanto a los usuarios sin loguear como a los usuarios logueados.

No se dispone de un usuario administrador ya que los datos de las series, películas e integrantes que se mostrarán en la aplicación se importarán directamente desde la API de TMDB. Por este motivo, no hay ningún requisito asociado a la actualización de estos datos ni se requiere un usuario con privilegios especiales para ello.

3.2.2. Requisitos funcionales

En este apartado se expondrán los requisitos funcionales organizados por el módulo de la aplicación al que pertenecen.

3.2.2.1. Perfil

- **RF1**. Un usuario no registrado que disponga de una Samsung *Smart TV* podrá registrarse en la aplicación.
- **RF2**. Un usuario sin loguear podrá iniciar sesión bien en su televisor habitual o en cualquier otro televisor Samsung *Smart TV*.
- **RF3**. Un usuario logueado podrá cerrar su sesión, por lo que su perfil ya no aparecerá al iniciar la aplicación.
- **RF4**. Un usuario logueado podrá cambiar su nombre de usuario, correo electrónico o contraseña.
- **RF5**. Un usuario logueado podrá editar su imagen de perfil seleccionando un avatar de los que se le proporcionarán desde la aplicación.
- **RF6**. Un usuario logueado podrá registrar las plataformas *streaming* que tiene contratadas.
- **RF7**. Un usuario logueado podrá eliminar plataformas *streaming* que tenía contratadas.

RF8. Un usuario logueado podrá eliminar su cuenta, lo cual conllevará la eliminación de todos sus datos de la base de datos.

3.2.2.2. Amigos

- **RF9**. Un usuario logueado podrá enviar solicitudes de amistad a otros usuarios registrados.
- **RF10**. Un usuario logueado que ha recibido una solicitud de amistad puede aceptarla o rechazarla. En cualquier caso, se notificará al usuario emisor de la solicitud.
- **RF11**. Un usuario logueado podrá eliminar su relación de amistad con otro usuario registrado. Se notificará al usuario registrado del que ha dejado de ser amigo.

3.2.2.3. Tendencias

RF12. Cualquier usuario podrá consultar las tendencias diarias. Es decir, podrá ver las series, películas e integrantes más populares de ese día.

3.2.2.4. Búsqueda

- **RF13**. Cualquier usuario podrá buscar una serie, película o integrante y se le mostrará un listado de resultados que coincidan con su búsqueda.
- RF14. Cualquier usuario podrá consultar los detalles de una serie, película o integrante.
- **RF15**. Cualquier usuario podrá acceder directamente desde la aplicación a las plataformas *streaming* en las que se encuentran disponibles las series y películas.

3.2.2.5. Mis series

- **RF16**. Un usuario logueado podrá marcar que está siguiendo una serie si esta ya se ha estrenado. Opcionalmente podrá indicar en qué plataforma *streaming* la está visualizando.
- **RF17**. Un usuario logueado podrá acceder directamente desde la aplicación a la plataforma *streaming* en la que está siguiendo una serie determinada en caso de haberlo indicado previamente.
- **RF18**. Un usuario logueado podrá indicar que quiere abandonar una serie que estaba siguiendo. De todos modos, se mantendrá un registro de los capítulos que había visualizado de dicha serie en caso de que quisiera retomarla en un futuro.
- RF19. Un usuario logueado podrá marcar (o desmarcar) que ha visto un capítulo.
- **RF20**. Un usuario logueado podrá marcar (o desmarcar) que ha visto una temporada.
- **RF21**. Un usuario logueado podrá consultar qué series está siguiendo actualmente y el siguiente capítulo que le toca visualizar de cada serie. Además, podrá filtrarlas por las plataformas *streaming* en las que las está siguiendo.
- **RF22**. Un usuario logueado podrá marcar (o desmarcar) una serie como pendiente de ver en un futuro.
- **RF23**. Un usuario logueado podrá consultar las series que tiene marcadas como pendientes de ver.

- **RF24**. Un usuario logueado podrá marcar (o desmarcar) una serie como vista cuando esta haya finalizado o haya sido cancelada.
- **RF25**. Un usuario logueado podrá consultar las series que ya ha visto.
- **RF26.** Un usuario logueado recibirá una notificación por correo cuando salgan nuevos capítulos de series que está siguiendo si ya había visualizado todos los capítulos disponibles y tiene las notificaciones de nuevos capítulos activadas.
- **RF27**. Un usuario logueado podrá desactivar las notificaciones de nuevos capítulos de una serie en caso de no querer recibir correos sobre esa serie en concreto.
- **RF28**. Un usuario logueado podrá desactivar las notificaciones de nuevos capítulos disponibles en caso de no querer recibir correos de este tipo.

3.2.2.6. Mis películas

- **RF29.** Un usuario logueado podrá marcar (o desmarcar) que está viendo una película si esta se ha estrenado. Opcionalmente podrá indicar en qué plataforma *streaming* la está visualizando.
- **RF30**. Un usuario logueado podrá consultar qué películas está viendo actualmente. Además, podrá filtrarlas por las plataformas *streaming* en las que las está viendo.
- **RF31**. Un usuario logueado podrá acceder directamente desde la aplicación a la plataforma *streaming* en la que está viendo una película determinada en caso de haberlo indicado previamente.
- **RF32**. Un usuario logueado podrá marcar (o desmarcar) una película como pendiente de ver en un futuro.
- **RF33**. Un usuario logueado podrá consultar las películas que tiene marcadas como pendientes de ver.
- RF34. Un usuario logueado podrá marcar (o desmarcar) una película como vista.
- **RF35**. Un usuario logueado podrá consultar las películas que ya ha visto.

3.2.2.7. Favoritos

- **RF36**. Un usuario logueado podrá marcar (o desmarcar) una película, serie o integrante como favorito. En el caso de las películas y series, será necesario que el usuario logueado haya marcado que la está viendo/siguiendo, que la tiene como pendiente o que la ha visto.
- **RF37**. Un usuario logueado podrá consultar las series, películas e integrantes que tiene marcadas como favoritos.

3.2.2.8. Recomendaciones

- **RF38**. Un usuario logueado podrá enviar recomendaciones de series o películas a sus amigos.
- **RF39**. Un usuario logueado podrá consultar las recomendaciones que ha recibido de sus amigos.

RF40. Un usuario logueado podrá eliminar las recomendaciones que ha recibido de sus amigos.

RF41. Un usuario logueado recibirá una notificación por correo cuando tenga una recomendación de un amigo.

RF42. Un usuario logueado podrá desactivar las notificaciones de recomendaciones en caso de no querer recibir correos de este tipo.

3.2.3. Requisitos de sistema

En este apartado se van a detallar aspectos de seguridad y usabilidad que deberá cumplir la aplicación.

3.2.3.1. Seguridad

RNF1. Los datos que manejará la aplicación de los usuarios registrados se almacenarán en una base de datos privada. Si el usuario registrado se da de baja, todos sus datos serán borrados.

RNF2. Se exigirán que las contraseñas de los usuarios registrados tengan una longitud de entre 8-26 caracteres y que incluyan una letra mayúscula, una letra minúscula, un número y un símbolo.

RNF3. Para garantizar la máxima protección de las contraseñas de los usuarios registrados, se almacenarán aplicando un hash basado en el algoritmo de cifrado en bloque *Blowfish* y *salt*.

3.2.3.2. Usabilidad

RNF4. Al tratarse de una televisión, el método de interacción con la interfaz de la aplicación será un mando a distancia. Por este motivo, se deberá poder acceder al contenido que el usuario desea con los menores clics posibles y que el contenido se muestre en la pantalla sin que el usuario tenga que realizar una cantidad excesiva de desplazamientos por la pantalla.

RNF5. La aplicación seguirá los patrones de diseño de los televisores Samsung *Smart TV*, facilitando así su uso a los usuarios.

3.3. Diseño

El diseño es una de las fases del desarrollo de software más importantes, ya que en esta se va a establecer cómo debe ser el software a implementar para que se cumplan las especificaciones establecidas en la fase de análisis previa.

3.3.1. Casos de uso

En esta sección se mostrará un diagrama de casos de uso con todos los tipos de usuario y las funcionalidades de la aplicación, que representará de manera abstracta una visión completa de la aplicación a nivel funcional (ver Figura 4).

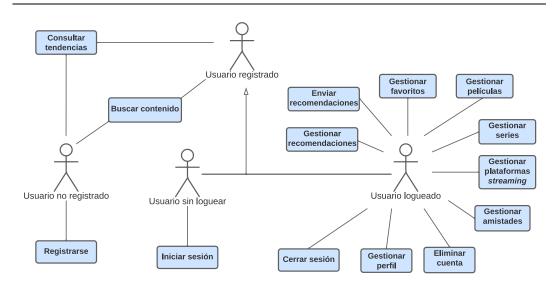


Figura 4. Diagrama de casos de uso.

En la Tabla 1 se muestran una se muestran una serie de aclaraciones sobre la Figura 4.

Buscar contenido	Engloba todos los requisitos funcionales del módulo "Búsqueda".
Gestionar perfil	Representa las funcionalidades de cambio de nombre de usuario, correo electrónico, contraseña e imagen de perfil.
Gestionar amistades	Engloba todos los requisitos funcionales del módulo "Amigos".
Gestionar plataformas streaming	Representa las funcionalidades de registrar y eliminar plataformas <i>streaming</i> .
Gestionar series	Engloba todos los requisitos funcionales del módulo "Mis series".
Gestionar películas	Engloba todos los requisitos funcionales del módulo "Mis películas".
Gestionar favoritos	Engloba todos los requisitos funcionales del módulo "Favoritos".
Gestionar recomendaciones	Representa las funcionalidades de consultar y eliminar recomendaciones y desactivar las notificaciones de recomendaciones.

Tabla 1. Aclaraciones diagrama casos de uso.

3.3.2. Modelo de datos

En este apartado se reflejarán todas las etapas necesarias para el diseño de la base de datos relacional que almacenará la información necesaria de la aplicación.

3.3.2.1. Modelo conceptual

Teniendo en cuenta todos los requisitos obtenidos en la fase de análisis, se ha diseñado el modelo conceptual completo MER-UML de la Figura 5.

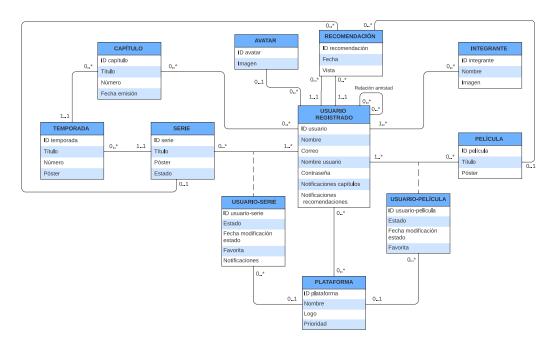


Figura 5. Modelo conceptual.

Para entender mejor la solución que se acaba de plantear, a continuación se expondrá detalladamente el motivo de las clases y de sus atributos, así como las relaciones que existen entre las diferentes clases.

3.3.2.1.1. Clases y atributos

- Clase SERIE: tipo de contenido multimedia que se ofrece en las plataformas streaming. Se almacenarán solo las series que tienen (o han tenido) relación con algún usuario registrado. Toda la información de esta clase será importada directamente desde la API de TMDB y se actualizará periódicamente de manera automática.
 - o ID serie: identificador único de una serie.
 - Título: nombre por el que se conoce a la serie.
 - Póster: imagen que se usa para anunciar la serie. Se almacenará para poder mostrarla en la aplicación sin necesidad de realizar una petición a la API de TMDB.
 - Estado: situación actual de una serie. Los posibles valores son: Rumoreada, Planeada, Piloto, En producción, Post producción, Estrenada, Cancelada, Volverá a emitirse y Finalizada. Es necesario saber el estado

en el que se encuentra la serie para determinar el estado de la relación entre la serie y el usuario.

- Clase TEMPORADA: forma en la que se divide el contenido de una serie.
 - o ID temporada: identificador único de una temporada.
 - o Título: nombre por el que se conoce a una temporada.
 - Número: número que designa el orden de estreno de la temporada en una serie.
 - Póster: imagen que se usa para anunciar la temporada. Se almacenará para poder mostrarla en la aplicación sin necesidad de realizar una petición a la API de TMDB.
- Clase CAPÍTULO: pieza audiovisual que está incluida en una temporada. Es necesario guardar los capítulos de las temporadas almacenadas para poder indicar a un usuario logueado el siguiente capítulo que le toca ver de una serie que está siguiendo.
 - o ID capítulo: identificador único de un capítulo.
 - Título: nombre por el que se conoce a un capítulo.
 - Número: número que designa el orden de estreno del capítulo en una temporada.
 - Fecha emisión: fecha en el que el capítulo se ha emitido (o se emitirá).
 Este atributo se registra para poder indicarle a un usuario registrado que tiene más capítulo disponibles para ver de una serie que sigue.
- Clase PELÍCULA: tipo de contenido multimedia que se ofrece en las plataformas streaming. Se almacenarán solo las películas que tienen (o han tenido) relación con algún usuario registrado. Toda la información de esta clase será importada directamente desde la API de TMDB y se actualizará periódicamente de manera automática.
 - o ID película: identificador único de una película.
 - o Título: nombre por el que se conoce a una película.
 - Póster: imagen que se usa para anunciar la película. Se almacenará para poder mostrarla en la aplicación sin necesidad de realizar una petición a la API de TMDB.
- Clase INTEGRANTE: persona que forma parte del reparto o del equipo (dirección, producción, guion, cámara, etc.) implicada en una serie o película. Solo se almacenarán los integrantes que están (o han estado) marcados como favoritos de algún usuario registrado.
 - o ID integrante: identificador único de un integrante.
 - Nombre: nombre y apellido por el que se conoce a un integrante.
 - o Imagen: foto que se usa para reconocer al integrante.
- Clase USUARIO REGISTRADO: usuario que dispone de una cuenta en la aplicación.
 - o ID usuario: identificador único de un usuario registrado.
 - o Nombre: nombre (o nombre y apellido) del usuario registrado.

- Correo: dirección de correo electrónico del usuario registrado. Debe ser exclusivo, por lo que no puede haber usuarios diferentes con el mismo correo registrado.
- Nombre usuario: nombre por el cual el usuario registrado quiere que se le conozca en la aplicación. Al igual que el correo, es un atributo exclusivo.
- Contraseña: conjunto de caracteres que permiten la autenticación del usuario. Derivado de un requisito de seguridad, esta deberá tener una longitud de entre 8-26 caracteres e incluir una letra mayúscula, una letra minúscula, un número y un símbolo.
- Notificaciones capítulos: atributo que sirve para indicar si el usuario desea recibir notificaciones sobre la disponibilidad de nuevos capítulos de las series que sigue.
- Notificaciones recomendaciones: atributo que sirve para indicar si el usuario desea recibir notificaciones sobre las recomendaciones que ha recibido de sus amigos.
- Clase AVATAR: imagen que un usuario registrado puede asignarse como foto de perfil de entre una selección que se le ofrecerá.
 - o ID avatar: identificador único del avatar.
 - o Imagen: imagen asociada al avatar seleccionado por el usuario registrado.
- Clase USUARIO-SERIE: representa la relación que existe entre un usuario registrado y una serie.
 - o ID usuario-serie: identificador único de la relación entre el usuario registrado y la serie.
 - Estado: estado en la que se encuentra la relación entre un usuario registrado y una serie. Los diferentes estados se reflejan en la Figura 7. Cabe mencionar que aunque en el diagrama se ha incluido un estado inicial "Sin relación", este no es un valor posible de este atributo, ya que como bien dice su nombre, esto significa que no existe relación entre la serie y el usuario registrado. Por tanto, los posibles valores son: Siguiendo, Esperando, Pendiente, Abandonada y Vista.
 - Fecha modificación estado: fecha en la que se ha realizado la última modificación del estado.
 - Favorita: atributo que sirve para indicar si el usuario registrado tiene marcada la serie como favorita.
 - Notificaciones: atributo que sirve para indicar si el usuario desea recibir notificaciones sobre la disponibilidad de nuevos capítulos de esta serie en concreto.
- Clase USUARIO-PELÍCULA: representa la relación que existe entre un usuario registrado y una película.
 - ID usuario-película: identificador único de la relación entre el usuario registrado y la película.
 - Estado: estado en la que se encuentra la relación entre un usuario registrado y una película. Los diferentes estados se reflejan en la Figura 8.
 Cabe mencionar que aunque en el diagrama se ha incluido un estado

inicial "Sin relación", este no es un valor posible de este atributo, ya que como bien dice su nombre, esto significa que no existe relación entre la película y el usuario registrado. Por tanto, los posibles valores son: Viendo, Pendiente y Vista.

- Fecha modificación estado: fecha en la que se ha realizado la última modificación del estado.
- Favorita: atributo que sirve para indicar si el usuario registrado tiene marcada la película como favorita.
- Clase PLATAFORMA: plataforma streaming. Se almacenarán solo las plataformas que están operativas en España. Toda la información de esta clase será importada directamente desde la API de TMDB y se actualizará periódicamente de manera automática.
 - o ID plataforma: identificador único de la plataforma.
 - o Nombre: nombre de la plataforma.
 - Logo: imagen que representa a la plataforma.
 - o Prioridad: nivel de importancia de la plataforma.
- Clase RECOMENDACIÓN: serie o película que ha sido enviada de un usuario registrado a otro usuario registrado con el cual mantiene una relación de amistad.
 - o ID recomendación: identificador único de la recomendación.
 - o Fecha: fecha en el que se ha enviado la serie o película.
 - Vista: atributo que indica si el usuario logueado que ha recibido la recomendación ya ha interactuado con esta. Si el usuario logueado ya ha visto la recomendación que ha recibido, esta no se mostrará de inicio.

3.3.2.1.2. Relaciones

En este apartado se explicará el sentido de las relaciones que se pueden observar en el modelo conceptual de la Figura 5.

- Relación TEMPORADA SERIE: cada temporada pertenece a una única serie. Una serie puede tener 0 (en el caso de que no se haya estrenado aún) o más temporadas.
- Relación CAPÍTULO TEMPORADA: cada capítulo pertenece a una única temporada. Una temporada puede tener 0 (en el caso de que no se conozcan aún los episodios que tendrá la temporada) o más capítulos.
- Relación SERIE USUARIO REGISTRADO: una serie tiene (o ha tenido) mínimo un usuario registrado con algún tipo de relación, ya que sino no estaría guardada en la base de datos. Un usuario registrado puede tener o no tener relaciones con series.
- Relación CAPÍTULO USUARIO REGISTRADO: un capítulo puede haber sido visto por 0 o más usuarios registrados. Un usuario registrado puede haber visto 0 o más capítulos.
- Relación PELÍCULA USUARIO REGISTRADO: una película tiene (o ha tenido) mínimo un usuario registrado con algún tipo de relación, ya que sino no estaría

guardada en la base de datos. Un usuario registrado puede tener o no tener relaciones con series.

- Relación USUARIO REGISTRADO AVATAR: un usuario registrado puede tener o no tener un avatar asignado. Un avatar puede estar asignado a 0 o más usuarios registrados.
- Relación USUARIO REGISTRADO RECOMENDACIÓN: esta es una relación doble.
 - Un usuario registrado puede haber enviado o no una recomendación a un amigo. Una recomendación la ha enviado un único usuario registrado.
 - Un usuario registrado puede haber recibido o no una recomendación de un amigo. Una recomendación la ha recibido un único usuario registrado.
- Relación USUARIO REGITRADO USUARIO REGISTRADO: un usuario registrado puede tener 0 más relaciones de amistad con otros usuarios registrados. Es una relación recíproca, por lo que si un usuario registrado A es amigo de un usuario registrado B, el usuario registrado B también es amigo del usuario registrado A.
- Relación USUARIO REGISTRADO INTEGRANTE: un usuario registrado puede tener guardado como favorito a 0 o más integrantes. Un integrante tiene (o ha tenido) mínimo un usuario registrado que lo tiene guardado en sus favoritos, ya que sino no estaría almacenado en la base de datos.
- Relación USUARIO REGISTRADO PLATAFORMA: un usuario registrado puede haber registrado que tiene contratadas 0 o más plataformas. Una misma plataforma puede estar registrada por 0 o más usuarios registrados.
- Relación USUARIO-SERIE PLATAFORMA: una usuario puede indicar que está siguiendo una serie en una plataforma o no indicarlo. Una plataforma puede estar indicada en 0 más relaciones de un usuario registrado y con una serie.
- Relación USUARIO-PELÍCULA PLATAFORMA: una usuario puede indicar que está viendo una película en una plataforma o no indicarlo. Una plataforma puede estar indicada en 0 más relaciones de un usuario registrado y con una película.
- Relación RECOMENDACIÓN SERIE: una recomendación puede ser o no (sería una película) de una serie. Una serie puede haber sido enviada en 0 o más recomendaciones.
- Relación RECOMENDACIÓN PELÍCULA: una recomendación puede ser o no (sería una serie) de una película. Una película puede haber sido enviada en 0 o más recomendaciones.

3.3.2.2. Modelo relacional

Una vez finalizado el modelo conceptual, lo traducimos a un modelo relacional mediante el siguiente algoritmo:

1. Escoger una clave principal para cada clase.

- 2. Cada relación del modelo conceptual se convierte en una tabla/relación del modelo relacional.
- 3. Cada relación del modelo conceptual se convierte en una tabla/relación del modelo relacional, con sus atributos más las claves de las clases que relaciona.
- 4. Se selecciona una clave principal para cada una de las nuevas tablas/relaciones.
- 5. Se estudia la posibilidad de fusionar algunas de las tablas/relaciones resultantes. Las candidatas son aquellas que tienen la misma clave, pero se debe tener en cuenta su significado y la conveniencia de su fusión.

TIPOGRAFÍA

Tras aplicar el algoritmo que se acaba de detallar, las tablas resultantes son las que se listarán a continuación, pero previamente se mostrarán unas aclaraciones en la Tabla 2.

PRIMARY KEY	Columna/s que identifican de manera única cada registro en una tabla	Subrayado
FOREIGN KEY	Columna/s que identifican de manera única una fila de otra tabla	Negrita

Tabla 2. Aclaraciones tablas base de datos.

- SERIE (ID serie, Título, Póster, Estado)

DEFINICIÓN

- TEMPORADA (ID temporada, Título, Número, Póster, ID serie)
 - o ID serie es una foreign key que viene de la clase SERIE (ID serie).
- CAPÍTULO (<u>ID capítulo</u>, Título, Número, Fecha emisión, **ID temporada**)
 - ID temporada es una foreign key que viene de la clase TEMPORADA (ID temporada).
- PELÍCULA (ID película, Título, Póster)
- INTEGRANTE (ID integrante, Nombre, Imagen)
- USUARIO REGISTRADO (<u>ID usuario</u>, Nombre, Correo, Nombre de usuario, Contraseña, Notificaciones capítulos, Notificaciones recomendaciones, <u>ID avatar</u>)
 - o ID avatar es una *foreign key* que viene de la clase AVATAR (ID avatar).
- AVATAR (ID Avatar, Imagen)
- USUARIO-SERIE (<u>ID usuario, ID serie</u>, Estado, Fecha modificación estado, Favorita, Notificaciones, <u>ID plataforma</u>)
 - o ID usuario es una foreign key que viene de la clase USUARIO (ID usuario).
 - o ID serie es una *foreign key* que viene de la clase SERIE (ID serie).
 - ID plataforma es una foreign key que viene de la clase PLATAFORMA (ID plataforma).
- USUARIO-PELÍCULA (<u>ID usuario, ID película</u>, Estado, Fecha modificación estado, Favorita, ID plataforma)

- o ID usuario es una *foreign key* que viene de la clase USUARIO (ID usuario).
- o ID película es una foreign key que viene de la clase PELÍCULA (ID película).
- ID plataforma es una foreign key que viene de la clase PLATAFORMA (ID plataforma).
- PLATAFORMA (<u>ID plataforma</u>, Nombre, Logo, Prioridad)
- RECOMENDACIÓN (<u>ID recomendación</u>, Fecha, Vista, **ID usuario A**, **ID usuario B**, **ID película**, **ID serie**)
 - ID usuario A es una foreign key que viene de la clase USUARIO REGISTRADO (ID usuario).
 - o ID usuario B es una *foreign key* que viene de la clase USUARIO REGISTRADO (ID usuario).
 - o ID película es una *foreign key* que viene de la clase PELÍCULA (ID película).
 - o ID serie es una foreign key que viene de la clase SERIE (ID serie).

- R USUARIOREGISTRADO INTEGRANTE (ID usuario, ID integrante)

- ID usuario es una foreign key que viene de la clase USUARIO REGISTRADO (ID usuario).
- o ID integrante es una *foreign key* que viene de la clase INTEGRANTE (ID integrante).

R_USUARIOREGISTRADO_PLATAFORMA (<u>ID usuario</u>, <u>ID plataforma</u>)

- ID usuario es una foreign key que viene de la clase USUARIO REGISTRADO (ID usuario).
- ID plataforma es una foreign key que viene de la clase PLATAFORMA (ID plataforma).

R_CAPÍTULO_USUARIO (ID usuario, ID plataforma)

- o ID capítulo es una *foreign key* que viene de la clase CAPÍTULO (ID capítulo).
- ID usuario es una foreign key que viene de la clase USUARIO REGISTRADO (ID usuario).
- R_USUARIOREGISTRADO_USUARIOREGISTRADO (<u>ID usuario A, ID usuario B</u>, Fecha solicitud, Fecha aceptada)
 - o ID usuario A es una *foreign key* que viene de la clase USUARIO REGISTRADO (ID usuario).
 - o ID usuario B es una *foreign key* que viene de la clase USUARIO REGISTRADO (ID usuario).

3.3.2.3. Creación de la base de datos

A partir de los resultados obtenidos en el modelo relacional, ya es posible la creación de la base de datos relacional. En el apéndice del documento Apéndice A. Creación base de datos, se muestra el código SQL que se ha utilizado para la creación de la base de datos, la cual se puede ver representada visualmente en la Figura 6.

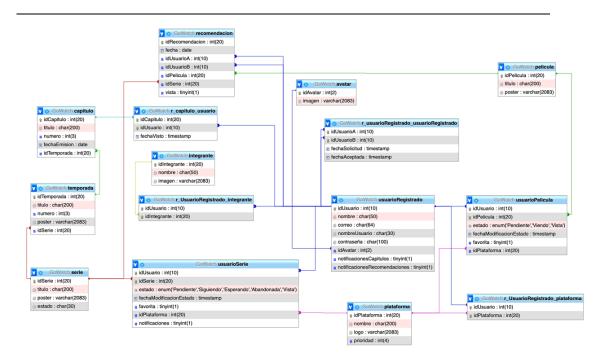


Figura 6. Representación visual de la base de datos.

3.3.3. Diagramas de estado

En este apartado se representarán los diagramas de estado que ayudarán a comprender mejor los posibles valores del atributo "Estado" tanto de la clase USUARIO-SERIE como de la clase USUARIO-PELÍCULA.

3.3.3.1. Diagrama de estados de la relación entre una serie y un usuario registrado

Se mostrará un diagrama de los posibles estados de la relación entre una serie y un usuario registrado (ver Figura 7), seguido de las descripciones de cada uno de estos estados (ver Tabla 3) y de las transiciones entre estados (ver Tabla 4).

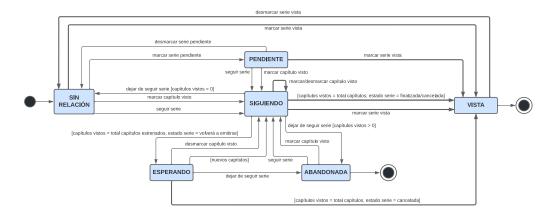


Figura 7. Diagrama de estados de la relación entre una serie y un usuario registrado.

ESTADO	DESCRIPCIÓN
SIN RELACIÓN	El usuario registrado y la serie no tienen ningún tipo de relación.
PENDIENTE	El usuario registrado tiene la serie marcada como pendiente para ver en un futuro.
SIGUIENDO	El usuario registrado está siguiendo la serie.
ESPERANDO	El usuario registrado ha visto todos los capítulos estrenados de la serie, pero esta aun no ha finalizado, por lo que está esperando a que salgan nuevos capítulos de la serie.
ABANDONADA	El usuario registrado ha dejado de seguir una serie que ya había empezado a ver.
VISTA	El usuario registrado ha visto todos los capítulos de la serie, la cual ya ha finalizado o ha sido cancelada.

Tabla 3. Descripción de estados del diagrama de estados de la relación entre una serie y un usuario registrado.

TRANSICIÓN	DESCRIPCIÓN
marcar serie vista	El usuario registrado ha marcado la serie, cuyo estado es finalizada o cancelada, como vista. Se marcarán todos los capítulos de la serie como vistos.
marcar serie pendiente	El usuario registrado ha guardado una serie (con la cual no tenía relación) para ver en un futuro.
marcar capítulo visto	El usuario registrado ha marcado un capítulo como visto. Si el usuario registrado no estaba siguiendo la serie, automáticamente pasará a seguirla.
seguir serie	El usuario registrado ha indicado que está siguiendo la serie.
marcar/desmarcar capítulo visto	El usuario registrado está siguiendo la serie y cada vez que ve un capítulo lo marca como visto. Si se equivoca de capítulo también puede desmarcarlo, lo cual conllevará que se elimine la relación entre el usuario registrado y el capítulo de la base de datos.
[capítulo vistos = total capítulos estrenados,	El usuario registrado tiene marcados como vistos todos los capítulos estrenados de la serie pero esta volverá a

estado serie = volverá a emitirse]	emitirse. Por tanto, el usuario registrado está esperando a que salgan nuevos capítulos de esa serie.
[capítulos vistos = total capítulos, estado serie = finalizada/cancelada]	El usuario registrado tiene marcados como vistos todos los capítulos de la serie y además esta ya ha finalizado o ha sido cancelada. Por tanto, el usuario registrado ya ha visto la serie completa.
[capítulos vistos = total capítulos, estado serie = cancelada]	El usuario registrado tiene marcados como vistos todos los capítulos de la serie y mientras esperaba nuevos capítulos, esta ha sido cancelada. Por tanto, el usuario registrado ya ha visto la serie completa.
dejar de seguir serie, [capítulos vistos = 0]	El usuario registrado ha dejado de seguir la serie que estaba siguiendo, pero aun no había marcado como visto ningún capítulo. Se eliminará la relación entre el usuario registrado y la serie de la base de datos.
dejar de seguir serie, [capítulos vistos > 0]	El usuario registrado ha dejado de seguir la serie que estaba siguiendo, pero ya había marcado como visto algún capítulo y todavía tenía capítulos disponibles para ver. Se conservarán los capítulos vistos por el usuario registrado.
desmarcar capítulo visto	El usuario registrado estaba esperando nuevos capítulos pero desmarca un capítulo como visto, por lo tanto tiene más capítulos para ver.
[nuevos capítulos]	Se han estrenado nuevos capítulos de la serie por la cual el usuario registrado estaba esperando ya que se había visto todos los capítulos previos. Se notificará al usuario registrado si tiene este tipo de notificaciones activadas.
dejar de seguir serie	El usuario registrado deja de seguir la serie por la cual esperaba nuevos capítulos. Se conservarán los capítulos vistos por el usuario registrado.
desmarcar serie vista	El usuario registrado desmarca la serie que tenía marcada como vista. Se desmarcarán todos los capítulos de la serie como vistos y se eliminará la relación entre la serie y el usuario registrado.

Tabla 4. Descripción de transiciones del diagrama de estados de la relación entre una serie y un usuario registrado.

3.3.3.2. Diagrama de estados de la relación entre una película y un usuario registrado

Se mostrará un diagrama de los posibles estados de la relación entre una serie y un usuario registrado (ver Figura 8), seguido de las descripciones de cada uno de estos estados (ver Tabla 5) y el significado de las transiciones entre estados (ver Tabla 6).

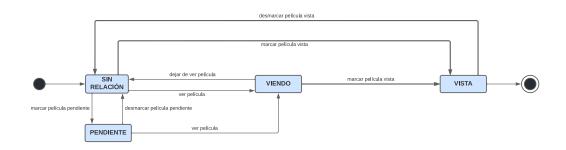


Figura 8. Diagrama de estados de la relación entre una película y un usuario registrado.

ESTADO	DESCRIPCIÓN
SIN RELACIÓN	El usuario registrado y la película no tienen ningún tipo de relación.
PENDIENTE	El usuario registrado tiene la película marcada como pendiente para ver en un futuro.
VIENDO	El usuario registrado está viendo la película.
VISTA	El usuario registrado ha acabado de ver la película.

Tabla 5. Descripción de estados del diagrama de estados de la relación entre una película y un usuario registrado.

TRANSICIÓN	DESCRIPCIÓN
marcar película pendiente	El usuario registrado ha marcado la película como pendiente de ver en un futuro.
desmarcar película pendiente	El usuario registrado ha decidido que ya no querrá ver esa película. Se eliminará la relación entre el usuario registrado y la película de la base de datos.
ver película	El usuario registrado ha indicado que ha empezado a ver la película.
marcar película vista	El usuario registrado ha marcado la película como vista porque ya ha terminado de ver la película.
dejar de ver película	El usuario registrado ya no quiere seguir viendo la película. Se eliminará la relación entre el usuario registrado y la película de la base de datos.

Tabla 6. Descripción de transiciones del diagrama de estados de la relación entre una película y un usuario registrado.

3.3.4. Diagramas de secuencia

En este apartado se va a mostrar un diagrama de secuencia que represente un requisito funcional por cada módulo de la aplicación.

3.3.4.1. Perfil

Se va a representar el requisito "RF1. Un usuario no registrado que disponga de una Samsung *Smart TV* podrá registrarse en la aplicación" (ver Figura 9).

3.3.4.2. Amigos

Se va a representar el "RF10. Un usuario logueado que ha recibido una solicitud de amistad puede aceptarla o rechazarla. En cualquier caso, se notificará al usuario emisor de la solicitud" (ver Figura 10).

3.3.4.3. Tendencias

Se va a representar el "RF12. Cualquier usuario podrá consultar las tendencias diarias. Es decir, podrá ver las series, películas e integrantes más populares de ese día" (ver Figura 11).

3.3.4.4. Búsqueda

Se va a representar el "RF15. Cualquier usuario podrá acceder directamente desde la aplicación a las plataformas *streaming* en las que se encuentran disponibles las series y películas" (ver Figura 12). Para entender el diagrama de la Figura 12, se debe dar por supuesto que el usuario ya está visualizando los detalles de la serie o película, siendo uno de los detalles el listado de plataformas *streaming* en las que se oferta.

3.3.4.5. Mis series

Se va a representar el "RF18. Un usuario logueado podrá marcar (o desmarcar) que ha visto un capítulo" (ver Figura 13). En el escenario que se está representando en la Figura 13, el usuario logueado ya tiene cargados en la interfaz de GoWatch todos los detalles de la serie, incluyendo sus temporadas y capítulos.

3.3.4.6. Mis películas

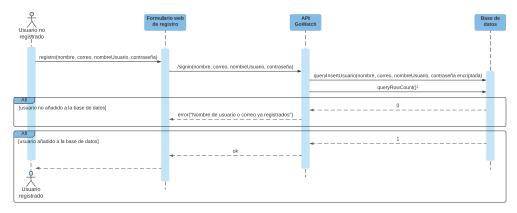
Se va a representar el "RF28. Un usuario logueado podrá marcar (o desmarcar) que está viendo una película. Opcionalmente podrá indicar en qué plataforma *streaming* la está visualizando" (ver Figura 14). En el escenario que se está representando en la Figura 14, el usuario logueado ya tiene cargados en la interfaz de GoWatch todos los detalles de la película, estando entre estos las plataformas *streaming* en las que la película está disponible.

3.3.4.7. Favoritos

Se va representar el "RF36. Un usuario logueado podrá consultar las series, películas e integrantes que tiene marcadas como favoritos" (ver Figura 15).

3.3.4.8. Recomendaciones

Se va a representar el "RF37. Un usuario logueado podrá enviar recomendaciones de series o películas a sus amigos" (ver Figura 16). Para comprender el diagrama de la Figura 16, se debe entender que el usuario logueado ya tenía cargado en la interfaz de GoWatch todos los detalles del contenido multimedia (serie o película) que va a enviar.



¹queryRowCount(): comprueba cuantas filas de la base de datos se han visto afectadas por la operación anterio

Figura 9. Diagrama de secuencia "RF1. Un usuario no registrado que disponga de una Samsung *Smart TV* podrá registrarse en la aplicación".

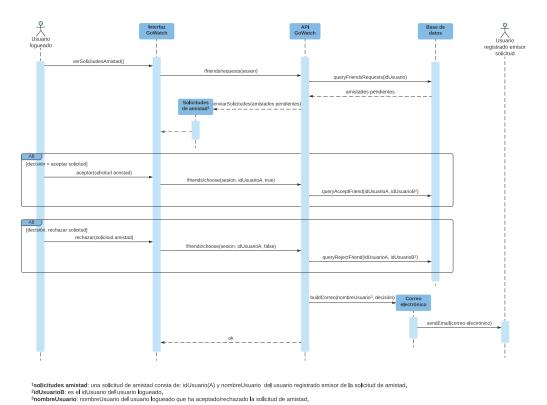
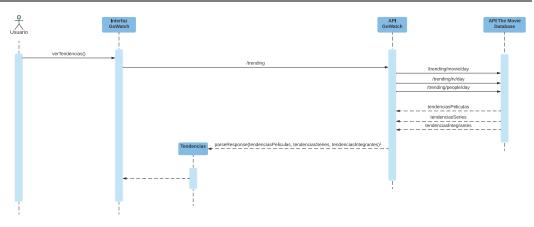


Figura 10. Diagrama de secuencia "RF10. Un usuario logueado que ha recibido una solicitud de amistad puede aceptarla o rechazarla. En cualquier caso, se notificará al usuario emisor de la solicitud".



*parseResponse: este método procesa las respuestas recibidas de la API de The Movie Database, quedándose únicamente con la información relevante y devuelve un objeto que contiene los 3 tipos de tendencias.

Figura 11. Diagrama de secuencia "RF12. Cualquier usuario podrá consultar las tendencias diarias. Es decir, podrá ver las series, películas e integrantes más populares de ese día".

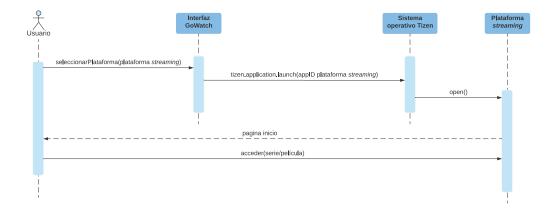
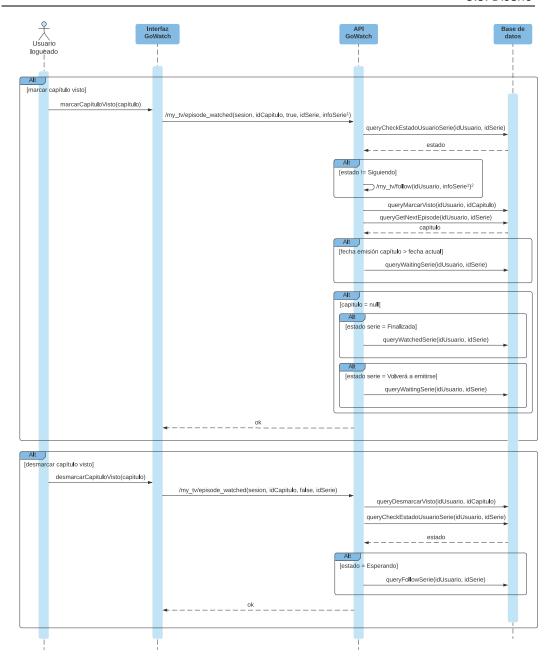


Figura 12. Diagrama de secuencia "RF15. Cualquier usuario podrá acceder directamente desde la aplicación a las plataformas *streaming* en las que se encuentran disponibles las series y películas".



infoSerie: número de temporadas, título, póster, estado serie.

/my_tvifollow: se llama a ese endpoint de la API GoWatch porque además de cambiar el estado de la clase usuarioSerie a "Siguiendo", se encarga de comprobar que la serie esté registrada en la base de datos y si no está la añade, con todas sus temporadas y capítulos. No hace falta llamar al endpoint en el caso de desmarcar el capítulo y con el estado = "Esperando" ya que en ese ecenario la serie seguro que ya estaba registrada, por lo que solo hace falta actualizar el estado a "Siguiendo".

Figura 13. Diagrama de secuencia "RF18. Un usuario logueado podrá marcar (o desmarcar) que ha visto un capítulo".

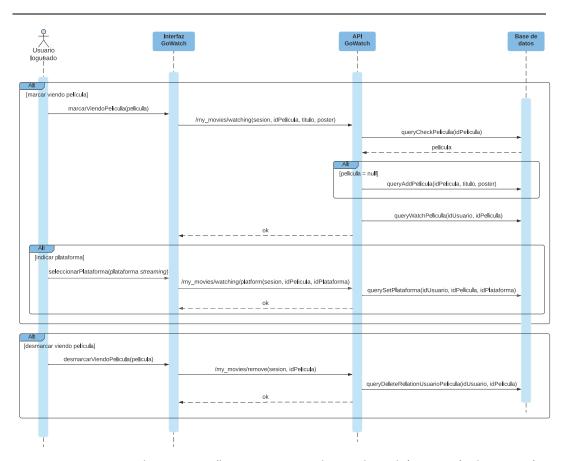


Figura 14. Diagrama de secuencia "RF28. Un usuario logueado podrá marcar (o desmarcar) que está viendo una película. Opcionalmente podrá indicar en qué plataforma *streaming* la está visualizando".

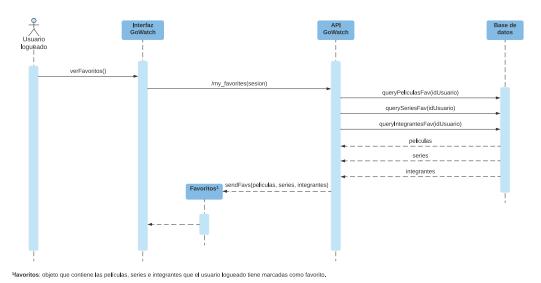


Figura 15. Diagrama de secuencia "RF36. Un usuario logueado podrá consultar las series, películas e integrantes que tiene marcadas como favoritos".

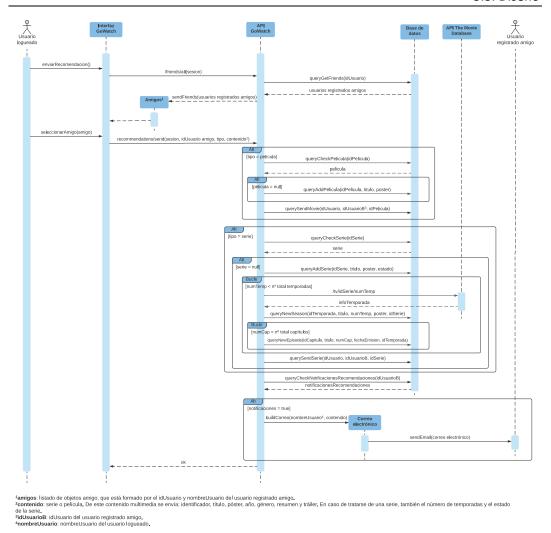


Figura 16. Diagrama de secuencia "RF37. Un usuario logueado podrá enviar recomendaciones de series o películas a sus amigos".

3.3.5. Diseño arquitectónico

En la Figura 17 se puede ver representada la arquitectura que tendrá el sistema.

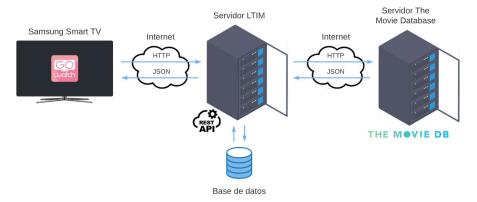


Figura 17. Arquitectura del sistema.

En el televisor Samsung *Smart TV* se instalará la aplicación basada en HTML5 creada con Tizen Studio. Desde el lado del cliente, es decir el televisor, se realizarán peticiones a través de métodos HTTP a la API RESTful de GoWatch alojada en el servidor LTIM. A su vez, desde la API RESTful de GoWatch también se podrán realizar peticiones a la API de TMDB, así como interactuar con la base de datos que también se alojará en el servidor LTIM.

3.3.6. Diseño modular

El diseño modular reduce la complejidad de un problema, favorece su entendimiento y facilita la detección y corrección de errores. Por estos motivos, tanto la aplicación como la API de GoWatch se dividirán en los siguientes módulos:

- 1. Perfil
- 2. Amigos
- 3. Tendencias
- 4. Búsqueda
- 5. Mis series
- 6. Mis películas
- 7. Favoritos
- 8. Recomendaciones

Cada módulo agrupa una serie de funcionalidades que quedaron definidas previamente en la sección 3.2.2. Requisitos funcionales.

3.4. Implementación

En esta sección se va a explicar cómo se ha llevado a cabo la implementación tanto de la API como de la aplicación de GoWatch.

3.4.1. API RESTful

Antes de comenzar con la explicación de cómo se han implementado los diferentes módulos en la parte *backend*, habrá un pequeño inciso en cómo se usa la API de TMDB y una vista general a la estructura del proyecto que contiene la API de GoWatch.

3.4.1.1. Uso de la API de The Movie Database

Es cierto que la API de TMDB es de uso gratuito, pero es imprescindible disponer de una API Key para poder emplearla. Para obtener esta API Key, lo primero es crearse una cuenta de desarrollador en The Movie Database [9] y después debes aportarles la información que solicitan, como por ejemplo el nombre de la aplicación, el uso que le vas a dar y cierta información sobre ti. Una vez enviada esta información, su equipo decide si aprueba tu solicitud y te proporcionan una API Key única que es la que debes utilizar cuando realizas las peticiones.

La única condición que exigen es que en tu aplicación les debes dar reconocimiento y si vas a hacer uso de la información de la disponibilidad de las series y películas, también debes dar reconocimiento a JustWatch, ya que es de donde ellos obtienen esa información.

3.4.1.2. Estructura del proyecto

En esta sección se mostrará (de manera simplificada) la estructura del proyecto que contiene la API RESTful de GoWatch y se explicarán los rasgos más importantes de los diferentes archivos y carpetas que lo componen.

```
gowatch
|-- app.js
|-- config.js
|-- constants.js
|-- database.js
|-- mail_templates
|-- node modules
|-- package-lock.json
|-- package.json
|-- public
|-- routes
  |-- friends.js
  |-- movie.js
   |-- my favorites.js
   |-- my movies.js
   |-- my_tv.js
   |-- person.js
   |-- profile.js
   |-- recommendations.js
  |-- search.js
   |-- trending.js
    `-- tv.js
|-- statics_updates_register
|-- update statics.js
`-- util.js
```

 app.js: es el archivo principal del proyecto ya que es en este donde se crea el servidor web de Node.js. Cuando se está ejecutando el servidor en un entorno de desarrollo el servidor escucha en el puerto 80, mientras que si se está ejecutando en un entorno de desarrollo el puerto seleccionado es el 3000.

Hace uso del *framework* Express, que permite la utilización de *middlewares* para responder a las peticiones que recibe el servidor y también definir las diferentes rutas que gestionará el servidor. Además de Express, los módulos que se utilizan son:

o body-parser. *Middleware* de Node.js que permite convertir los datos que llegan en las peticiones al servidor en objetos JSON.

o cookie-parser. *Middleware* de Node.js que para manipular las *cookies* que se envían con las peticiones que se realizan al servidor.

También se llama desde aquí a initScheduledJobs(), método dentro del módulo update_statics.js que se encarga de actualizar periódicamente la información obtenida de la API de TMDB que está almacenada en la base de datos.

- **config.js**: módulo que contiene parámetros de configuración que son necesarios a lo largo de todo el proyecto.
- **constants.js**: módulo que contiene valores constantes de los que se hacen uso en los diferentes archivos del proyecto.
- **database.js**: módulo que contiene los métodos para establecer y terminar la conexión con la base de datos. Para ello hace uso del módulo mysgl de Node.js.
- mail_templates: esta carpeta contiene plantillas HTML de los diferentes correos que se pueden enviar a los usuarios registrados en GoWatch. Se pueden ver ejemplos de estos correos en el Apéndice del documento Apéndice B. Correos.
- node_modules: módulos de Node.js que se han instalado en el directorio del proyecto.
- package-lock.json y package.json: archivos de configuración del proyecto.
 Incluye valores como el nombre del proyecto, la versión y lo más importante, las dependencias del proyecto.
- **public**: carpeta que contiene archivos HTML, JavaScript y CSS, además de las imágenes. Para poder servirlos es necesaria la instrucción express.static(dirname + '/public') en el fichero app.js.
- routes: es donde realmente se plasman los diferentes módulos definidos previamente. En cada archivo que contiene esta carpeta, se ha definido un objeto router mediante express.Router() que se encarga de gestionar las peticiones de ese módulo concreto. Este objeto router se exporta y se recoge desde app.js gracias a Express.
- **statics_updates_register**: contiene diferentes archivos CSV para cada tipo de actualización de datos que permiten llevar un control de las actualizaciones automáticas periódicas y detectar si se están llevando a cabo correctamente.
- update_statics.js: este módulo se encarga de actualizar, periódicamente y de manera automática, los datos obtenidos a través de la API de TMDB que están almacenados en la base de datos. Contiene el método initScheduledJobs(), que se llama desde app.js y encapsula los siguientes métodos:
 - o updateMovies(): los días 1 y 15 a las 00:00h (UTC) de cada mes actualiza la tabla pelicula.
 - o updateTvShows (): cada día a las 00:00h (UTC) actualiza la tabla serie. Si de una serie hay nuevas temporadas, las tablas temporada y capitulo también se actualizan. Si el estado de una serie ha pasado de "Volverá a emitirse" a "Cancelada", actualiza todas las relaciones de esa serie con los usuarios que estaban con estado "Esperando" a estado "Vista".
 - o updateTvSeasonsAndEpisodes (): cada domingo a las 00:00h (UTC) actualiza las tablas temporada y capitulo. Es necesario que se realicen también estas actualizaciones separadas ya que la información que se recibe de las temporadas y capítulos con la actualización de su serie

puede cambiar a lo largo del tiempo. También puede darse que se reciba información incompleta, ya que aunque aun no se haya estrenado una temporada, si TMDB ya la tiene registrada, su API ya devuelve la información que tenga de esta.

- o updatePeople(): el día 1 de cada mes a las 00.00h (UTC) actualiza la tabla integrante.
- o updatePlatforms (): el día 1 de cada mes a las 00.00h (UTC) actualiza la tabla plataforma.
- o updateNewEpisodes(): cada día a las 00:00h (UTC) comprueba la tabla capítulo y se queda con todos los capítulos que coinciden con la fecha actual. Obtiene la serie a la que pertenece cada capítulo y actualiza el estado de la relación de esa serie con todos los usuarios registrados que tienen una relación con esa serie a "Siguiendo". Para cada usuario, comprueba si tiene las notificaciones de capítulos (general) y de esa serie en concreto activada y de ser así, envía un correo avisando de la disponibilidad de nuevos capítulos. El correo se puede ver en la Figura 42 del Apéndice B. Correos.

Es posible planificar las ejecuciones de estos métodos gracias al método schedule del módulo node-cron de Node.js. Tanto en estos métodos como en el resto de módulos que necesiten obtener información de la API de TMDB, se va a usar el módulo axios de Node.js para lanzar las peticiones HTTP de manera asíncrona.

Por cada uno de estos métodos, hay un fichero CSV correspondiente en la carpeta statics_updates_register en el que se registra la fecha de la actualización, el número de elementos actualizados y los errores (si ha habido). Para escribir en estos ficheros se ha hecho uso del módulo objects-to-csv de Node.js.

util.js: agrupa funciones que se utilizan a lo largo de todo el proyecto en un mismo archivo para así evitar repetir el mismo código. Las funciones que se pueden encontrar aquí son procesar determinadas respuestas de la API de TMDB y construir el formato de respuesta deseado, convertir formatos de tiempo o fechas, ordenar arrays según determinados criterios y enviar los diferentes correos. Para esto último se ha usado el módulo nodemailer de Node.js y se ha creado una cuenta de Google para la aplicación que figura como emisor de estos correos.

3.4.1.3. Implementación de los módulos

La URL base de la API es "https://alumnes-ltim.uib.es/gowatch/", los objetos router de Express que tiene cada módulo se encargan de gestionar los diferentes *endpoints* o rutas.

3.4.1.3.1. Perfil

Este módulo está implementado en el archivo profile.js. Consta de una serie de rutas que permiten llevar a cabo las diferentes funcionalidades de este módulo. Para todos los métodos de tipo POST en este módulo y en el resto, el procedimiento que se sigue para validar el cuerpo de la petición es el siguiente:

- Se crea un método validateInput() que dependiendo del tipo de petición tiene definido un esquema concreto y se encarga de validarlo con el módulo joi de Node.js.
- Cuando llega una petición, se llama a este método pasándole el cuerpo que el cliente ha enviado en la petición. Si la validación es correcta se prosigue con normalidad, sino se devuelve una respuesta con código de estado 400 (Bad Request) y un mensaje que indica el porqué del error.

Las rutas de este módulo son:

- POST /signin. Crea un nuevo usuario registrado siempre que no exista ya en la base de datos un usuario registrado con el nombre de usuario o dirección de correo electrónico que se han pasado en el cuerpo de la petición. Para gestionar las contraseñas, se ha utilizado el módulo bcrypt de Node.js La contraseña se guarda cifrada en la base de datos con el método bcrypt.hash() especificando un factor de trabajo de 12.
- POST /login. Loguea al usuario comprobando primero que exista el nombre de usuario y de ser así, comprueba que la contraseña almacenada de ese usuario registrado se corresponda con la indicada en la petición con el método bcrypt.compare(). Si las contraseñas no coinciden, se devuelve una respuesta con código de estado 403 (Forbidden). Si las contraseñas coinciden, el idUsuario se añade a la cookie "logged" pasada con la petición y si esta no existía, se crea la cookie con el idUsuario. La cookie contiene un array con los idUsuario de los usuarios logueados. Esta cookie tiene una caducidad de 10 años y se le han agregado medidas de seguridad que prohíben acceder a la cookie a través de JavaScript y también que si la conexión no se produce a través de HTTPS, el navegador rechace la cookie.
- **POST** /access. Permite saber quién de todos los usuarios logueados en un mismo televisor está usando la aplicación. Para ello, crea una cookie "activeUser" que tiene los mismos atributos de seguridad que la cookie "logged", pero sin embargo esta se elimina una vez se sale de la aplicación.
- **GET /logout**. Cierra la sesión del usuario logueado eliminando la *cookie* "activeUser" y borrando el idUsuario correspondiente de la *cookie* "logged". Si la *cookie* "logged" queda vacía tras borrar el idUsuario, también elimina la *cookie*.
- **DELETE** /remove_user. Además de hacer lo mismo que cuando se cierra la sesión, se borran de la base de datos todas las referencias a este usuario registrado.
- GET /users. Devuelve el nombreUsuario e imagen de avatar de los usuarios logueados en el televisor que realiza la petición. Obtiene los usuarios logueados gracias a la cookie "logged".
- **GET /avatars**. Devuelve todos los avatares registrados en la base de datos.
- **GET /new_platforms**. Devuelve las plataformas *streaming* registradas en la base de datos que el usuario logueado (activo) no tiene añadidas en su lista de plataformas. En este módulo y en todos los otros, se identifica el usuario logueado (activo) gracias a la *cookie* "activeUser".
- **GET /profile**. Devuelve el nombre de usuario, correo, avatar, listado de plataformas y el estado de las notificaciones del usuario logueado (activo).
- POST /profile/change_username. Permite cambiar el nombre de usuario del usuario logueado (activo) siempre que el nuevo nombre que elija no esté siendo usado ya por otro usuario registrado.

- **GET /profile/request/change_email**. Envía un correo de solicitud de cambio de dirección de correo electrónico al usuario logueado (activo). El correo se puede ver en la Figura 43 del Apéndice B. Correos.
- PUT /profile/change_email. Con los datos introducidos en el formulario de solicitud de cambio de dirección de correo electrónico, comprueba primero que exista ese nombre de usuario y luego que la dirección de correo introducida no esté asociada a otro usuario registrado. Si se cumplen esas condiciones, efectúa el cambio y sino devuelve los mensajes de error correspondientes.
- GET /profile/request/change_password. Envía un correo de solicitud de cambio de contraseña al usuario logueado (activo). El correo se puede ver en la Figura 44 del Apéndice B. Correos.
- POST /profile/change_password. Con los datos introducidos en el formulario de solicitud de contraseña, comprueba primero que exista el nombre de usuario introducido y de ser así actualiza su contraseña. Al igual que en el proceso de registro, antes de guardar la contraseña en la base de datos, se le aplica bcrypt.hash() con un factor de trabajo de 12.
- **POST /profile/change_avatar**. Cambia el avatar del usuario logueado (activo) por el que se indica en la petición.
- **POST /profile/change_platforms**. Dependiendo de si en el cuerpo de la petición se indica en un parámetro *add* o *delete*, añade o elimina plataformas *streaming* de la lista de plataformas del usuario logueado (activo).
- POST /profile/notifications/episodes. Dependiendo de si en el cuerpo de la petición se indica en un parámetro true o false, activa o desactiva las notificaciones de capítulos nuevos del usuario logueado (activo).
- **POST** /profile/notifications/recommendations. Dependiendo de si en el cuerpo de la petición se indica en un parámetro *true* o *false*, activa o desactiva las notificaciones de recomendaciones de amigos del usuario logueado (activo).

3.4.1.3.2. Amigos

Este módulo está implementado en el archivo friends.js. Consta de las siguientes rutas que permiten llevar a cabo las diferentes funcionalidades de este módulo:

- POST /friends/add. A partir del nombre de usuario que se envía en la petición, se comprueba primero que exista un usuario registrado con ese nombre de usuario y luego que el usuario logueado (activo) y este otro usuario registrado no sean ya amigos o no haya una solicitud de amistad pendiente entre ellos. Si se cumplen esas condiciones, se registra la nueva relación de amistad entre estos usuarios (con el atributo fechaAceptada a null) y se envía un correo de solicitud de amistad al usuario que el usuario logueado (activo) quiere añadir como amigo. El correo se puede ver en la Figura 45 del Apéndice B. Correos.
- **GET** /friends/requests. Devuelve todas las solicitudes de amistad que el usuario logueado (activo) tiene pendientes de aceptar o rechazar (las que tienen el atributo fechaAceptada a null).
- POST /friends/choose. Dependiendo de si en el cuerpo de la petición se indica en un parámetro true o false, se acepta o se rechaza una amistad que el usuario logueado (activo) tenía pendiente con el usuario que envió esa solicitud. En ambos casos, se envía un correo al usuario emisor de la solicitud informando de la decisión. Los correos de solicitud de amistad aceptada y solicitud de amistad

- rechazada se pueden ver, respectivamente en la Figura 46 y Figura 47 del Apéndice B. Correos.
- POST /friends/delete. Elimina la relación de amistad entre el usuario logueado (activo) y el usuario registrado amigo que se indica en la petición. Al usuario registrado amigo se le envía un correo para informarle de que el usuario logueado (activo) ha dejado de ser su amigo. El correo se puede ver en la Figura 48 del Apéndice B. Correos.
- **GET** /friends/all. Devuelve todos los amigos que el usuario logueado (activo) tiene en GoWatch.

3.4.1.3.3. Tendencias

Este módulo está implementado en el archivo trending.js. Consta únicamente de la ruta **GET** /trending. En esta se realizan peticiones a la API de TMDB mediante axios.get() a cada uno de los *endpoints* que proporcionan las tendencias de películas, series e integrantes. Una vez se han obtenido todas las respuestas, se llama al método parseResponse(), que se encarga de devolver un único objeto que contiene un array de películas, un array de series y un array de integrantes. Estos arrays se han rellenado, respectivamente, con los métodos parseMovie(), parseTV() y parsePerson() que se encuentran en el fichero util.js. Estos métodos se quedan con la información necesaria de las respuestas de la API de TMDB y la devuelven en el formato deseado.

3.4.1.3.4. Búsqueda

Este módulo está implementado entre los ficheros search.js, tv.js, movie.js y person.js. Consta de las siguientes rutas que permiten llevar a cabo las diferentes funcionalidades de este módulo:

- **POST** /search. La API de TMDB tiene diferentes *endpoints* para cada tipo de búsqueda. Dependiendo del tipo de búsqueda (todo, solo series, solo películas o solo integrantes) que se pase en la petición, internamente se decide a qué *endpoint* se debe realizar la petición mediante axios.get(). Una vez se obtiene la respuesta, se llama al método parseResponse() para devolver solo la información necesaria y en el formato esperado.
- **POST** /tv. Devuelve los detalles de la serie cuyo ID se indica en el cuerpo de la petición. Se realiza la petición a la API de TMDB mediante axios.get() y la respuesta se adapta con el método parseResponse().
- POST /tv_seasons. Devuelve los detalles de una temporada concreta de una serie. En el cuerpo de la petición se debe indicar el ID de la serie y el número de la temporada que se quiere consultar. Se realiza la petición a la API de TMDB mediante axios.get() y la respuesta se adapta con el método parseResponse(). Para obtener el tráiler de la película se utiliza la API de Youtube [12], para la cual también hay que obtener una API Key.
- **POST /movie**. Devuelve los detalles de la película cuyo ID se indica en el cuerpo de la petición. Se realiza la petición a la API de TMDB mediante axios.get() y la respuesta se adapta con el método parseResponse(). Para obtener el tráiler de la película también se usa la API de Youtube.

- **POST /person**. Devuelve los detalles del integrante cuyo ID se indica en el cuerpo de la petición. Se realiza la petición a la API de TMDB mediante axios.get() y la respuesta se adapta con el método parseResponse().

3.4.1.3.5. Mis series

Este módulo está implementado en el fichero $my_tv.js$. Consta de las siguientes rutas que permiten llevar a cabo las diferentes funcionalidades de este módulo:

- **GET /my_tv**. Devuelve todas las series con las que el usuario logueado (activo) tiene relación.
- POST /my_tv/details. Devuelve información de la relación del usuario logueado (activo) con la serie concreta que se indica en el cuerpo de la petición. Esta información es: el estado de la relación de la serie, si tiene la serie marcada como favorita, los episodios que ha visto de esa serie y la plataforma streaming en la que la está siguiendo.
- POST /my_tv/follow. Lo primero que hace es comprobar si la serie que el usuario logueado (activo) quiere seguir está registrada en la base de datos. Si no lo está, esta se añade a la base de datos con todas sus temporadas y capítulos. Para obtener la información de las temporadas y los capítulos, se ha creado el método getInfoSeasons(), que se encarga de realizar una petición a la API de TMDB por cada temporada que tenga la serie. Cuando se han obtenido todas las respuestas, se guarda en la base de datos la información necesaria de cada temporadas y de sus capítulos.

Por último, si el usuario ya tenía una relación con esa serie, se cambia el estado a "Siguiendo" y se actualiza la fecha de modificación del estado. Si no existía una relación, se crea una nueva entre el usuario logueado (activo) y la serie con estado "Siguiendo".

- POST /my_tv/follow/platform. Actualiza la plataforma streaming en la cual el usuario logueado (activo) está siguiendo la serie. Si el usuario logueado (activo) no tenía registrada esa plataforma, se añade la nueva relación entre estos.
- POST /my_tv/unfollow. Si el usuario logueado (activo) había visto capítulos de la serie que quiere dejar de seguir, se actualizará el estado de la relación a "Abandonada". Si no había visto ningún capítulo, se elimina la relación entre la serie y el usuario logueado (activo).
- POST /my_tv/save. Al igual que en '/my_tv/follow', primero se comprueba si la serie está registrada y sino lo está se añade con todas sus temporadas y capítulos. Por último, si el usuario ya tenía una relación con esa serie, se cambia el estado a "Pendiente" y se actualiza la fecha de modificación del estado. Si no existía una relación, se crea una nueva entre el usuario logueado (activo) y la serie con estado "Pendiente".
- **POST** /my_tv/unsave. Elimina la relación entre la serie que se indica en el cuerpo de la petición y el usuario logueado (activo).
- **POST** /my_tv/episode_watched. Dependiendo de si en el cuerpo de la petición se indica en un parámetro *true* o *false*, se marca que el usuario logueado (activo) ha visto o no ha visto un capítulo.

En caso de que el parámetro venga con valor *true*, lo primero que se hace es comprobar si el usuario logueado (activo) está siguiendo la serie. Si no la estaba siguiendo, el servidor actúa como cliente y se hace a si mismo una petición HTTP

POST a la ruta "/my_tv/follow" enviando los datos necesarios que había recibido en el cuerpo de la petición y con un parámetro extra para indicar el idUsuario del usuario logueado (activo). Si la estaba siguiendo, no es necesario comprobar si está registrada en la base de datos, porque esto ya se habría comprobado previamente cuando solicitó seguir la serie.

Tras marcar el capítulo visto, se comprueba si este capítulo era el último estrenado llamando al método <code>getNextEpisode()</code>. Si este método devuelve un capítulo con fecha de emisión mayor a la fecha actual, se actualiza el valor de la relación del usuario con la serie a "Esperando". Si lo ha que ha devuelto el método es <code>null</code>, se actualizará el valor de la relación del usuario con la serie dependiendo del estado actual de dicha serie. Si la serie se encuentra en estado "Finalizada" o "Cancelada", el nuevo estado de la relación será "Vista". Por otro lado, si el estado de la serie es "Volverá a emitirse", el estado de la relación pasará a ser "Esperando".

POST /my_tv/watched. Primero comprueba si la serie que se indica en el cuerpo de la petición está registrada, sino lo está se, añade con todas sus temporadas y capítulos.

Por último, si el usuario logueado (activo) ya tenía una relación con esa serie, se cambia el estado a "Vista", se actualiza la fecha de modificación del estado y se marca que el usuario ha visto todos los capítulos de la serie. Si no existía una relación, se crea una nueva entre el usuario logueado (activo) y la serie con estado "Vista" y se marcan todos los capítulos como vistos.

- **POST /my_tv/watched**. Elimina la relación entre la serie que se indica en el cuerpo de la petición y el usuario logueado (activo). También se eliminan el registro de todos los capítulos de esa serie que estaban vistos por el usuario.
- POST /my_tv/notifications. Dependiendo de si en el cuerpo de la petición se indica en un parámetro true o false, activa o desactiva las notificaciones del usuario logueado (activo) sobre nuevos capítulos de la serie en concreto que se indica en el cuerpo de la petición.

3.4.1.3.6. Mis películas

Este módulo está implementado en el fichero my_movies.js. Consta de las siguientes rutas que permiten llevar a cabo las diferentes funcionalidades de este módulo:

- **GET /my_movies**. Devuelve todas las películas con las que el usuario logueado (activo) tiene relación.
- **POST /my_movies/details**. Devuelve información de la relación del usuario logueado (activo) con la película concreta que se indica en el cuerpo de la petición. Esta información es: el estado de la relación de la serie, si tiene la película marcada como favorita y la plataforma *streaming* en la que la está viendo.
- **POST** /my_movies/watching. Lo primero que hace es comprobar si la película que el usuario logueado (activo) quiere ver está registrada en la base de datos. Si no lo está, esta se añade a la base de datos.

Por último, si el usuario ya tenía una relación con esa película, se cambia el estado a "Viendo" y se actualiza la fecha de modificación del estado. Si no existía una relación, se crea una nueva entre el usuario logueado (activo) y la película con estado "Viendo".

- POST /my_movies/watching/platform. Actualiza la plataforma streaming en la cual el usuario logueado (activo) está viendo la película. Si el usuario logueado (activo) no tenía registrada esa plataforma, se añade la nueva relación entre estos.
- POST /my_movies/save. Al igual que en '/my_movies/watching', primero se comprueba si la película está registrada y sino lo está se añade. Si el usuario ya tenía una relación con esa película, se cambia el estado a "Pendiente" y se actualiza la fecha de modificación del estado. Si no existía una relación, se crea una nueva entre el usuario logueado (activo) y la película con estado "Pendiente".
- POST /my_movies/watched. Primero comprueba si la película está registrada y sino lo está se añade. Si el usuario ya tenía una relación con esa película, se cambia el estado a "Vista" y se actualiza la fecha de modificación del estado. Si no existía una relación, se crea una nueva entre el usuario logueado (activo) y la película con estado "Vista".
- **POST /my_movies/remove**. Elimina la relación entre la película que se indica en el cuerpo de la petición y el usuario logueado (activo).

3.4.1.3.7. Favoritos

Este módulo está implementado en el fichero my_favorites.js. Consta de las siguientes rutas que permiten llevar a cabo las diferentes funcionalidades de este módulo:

- **GET /my_favorites**. Devuelve todas las series, películas e integrantes que el usuario logueado (activo) tiene marcados como favorito.
- POST /my_favorites/like. En el cuerpo de la petición se indica el tipo de contenido que el usuario logueado (activo) quiere marcar como favorito y el ID de este. En caso de tratarse de un integrante, primero comprueba si este integrante está registrado en la base de datos, y si no lo está lo añade. Con las series y películas no es necesario comprobarlo porque se habría comprobado previamente cuando se creó la relación entre el usuario y la serie o película. Tras hacer la comprobación, ya se marca como favorito del usuario el contenido. En el caso de las series y películas, simplemente cambia a *true* el campo "favorita". Si se trata de un integrante, crea una nueva relación entre el usuario logueado (activo) y el integrante.
- **POST** /my_favorites/dislike. Desmarca como favorito del usuario logueado (activo) la serie, película o integrante que se indica en el cuerpo de la petición. En el caso de las series y películas, cambia a *false* el campo "favorita", mientras que si se trata de un integrante, elimina la relación entre el usuario y el integrante.
- **POST /my_favorites/integrante**. Comprueba si el usuario logueado (activo) tiene como favorito al integrante que se indica en el cuerpo de la petición.

3.4.1.3.8. Recomendaciones

Este módulo está implementado en el fichero recommendations.js. Consta de las siguientes rutas que permiten llevar a cabo las diferentes funcionalidades de este módulo:

POST /recommendations/send. Lo primero que hace es comprobar que la serie
o película que se quiere recomendar esté registrada en la base de datos, y si no
lo está la añade. En el caso de ser una serie, también se añadirán sus temporadas
y capítulos llamando al método getInfoSeasons (), que las obtendrá de la API
de TMDB.

Tras la comprobación, se crea una recomendación enviada por el usuario logueado (activo) hacia el usuario registrado que se ha indicado en el cuerpo de la petición. Si este usuario registrado tiene activadas las notificaciones de recomendaciones, se le enviará un correo indicando que el usuario logueado (activo) le ha enviado una recomendación y también los detalles de esta. El correo se puede ver en la Figura 49 del Apéndice B. Correos.

- GET /recommendations/new. Devuelve las recomendaciones nuevas que el usuario logueado (activo) ha recibido. Estas se distinguen porque tienen el campo "vista" a false.
- **GET /recommendations/all**. Devuelve todas las recomendaciones que el usuario logueado (activo) ha recibido.
- **POST** /recommendations/seen. Cambia el campo "vista" a *true* de la recomendación que se indica en el cuerpo de la petición.
- **POST /recommendations/delete**. Elimina la recomendación que se indica en el cuerpo de la petición.

3.4.2. Aplicación GoWatch

Antes de comenzar con la explicación de la implementación los diferentes módulos en la parte *frontend*, se expondrá la estructura que tiene el proyecto de la aplicación y se comentarán los factores que se han tenido en cuenta para el diseño y desarrollo de la interfaz de usuario.

3.4.2.1. Estructura del proyecto

El proyecto de la aplicación web creada siguiendo las instrucciones explicadas en el , tiene la siguiente estructura:

```
GoWatchApp
|-- JavaScript Resources
|-- lib
|-- scripts
|-- styles
|-- config.xml
|-- GO_WATCH_LOGO.png
|-- GoWatchApp.wgt
`-- index.html
```

Las carpetas JavaScript Resources y lib, las crea el propio Tizen Studio al crear el proyecto. Contiene los diferentes recursos que son necesarios para hacer uso JavaScript, CAPH y JQuery (librería de JavaScript). Las otras carpetas y archivos también aparecen por defecto con la creación del proyecto con código de ejemplo.

En la carpeta scripts están todos los archivos JavaScript que se han creado durante el desarrollo de la aplicación. La carpeta styles contiene un único archivo CSS que es el que definirá las características visuales de la aplicación.

El archivo config.xml es el que contiene la información de configuración del proyecto, que contiene información por defecto de Tizen además de los privilegios de la ¡Error! No se encuentra el origen de la referencia.

La imagen GO_WATCH_LOGO.png es el icono de la aplicación (ver Figura 18) que se mostrará en el menú *Smart Hub* del televisor Samsung *Smart TV*. Por especificaciones de Samsung, las medidas de esta imagen deben ser 117x117 y en formato png.

El archivo GoWatchApp.wgt es el archivo que se crea al compilar y empaquetar la aplicación, el que realmente se instala en el dispositivo donde se vaya a instalar la aplicación. Además de contener todos los componentes del proyecto, también incluye los certificados de autor y de distribuidor.

El proyecto tiene un solo archivo HTML con diferentes secciones definidas y se va mostrando una o otra dependiendo de la parte de la aplicación en la que se encuentre el usuario.



Figura 18. Logo de la aplicación GoWatch.

3.4.2.2. Interfaz de usuario

La Interacción Persona-Ordenador es una disciplina que estudia los aspectos relacionados con el proceso interactivo entre personas y ordenadores. El factor humano es uno de los conceptos que maneja esta disciplina y hace referencia a los aspectos que determinan la manera en que las personas hacen uso de la tecnología. Se han aplicado los siguientes conceptos:

- Percepción del color: se ha utilizado un contraste alto entre los colores del fondo (negro) y el color de la letra (blanco), a excepción de ciertos títulos que tienen el color característico de la aplicación rgb(231,161,168) ■. Para ciertas situaciones, como por ejemplo aceptar/rechazar solicitud de amistad, se usa un código redundante al tener cada opción un color llamativo y una forma distintiva, además de la aplicación de los colores de manera lógica ya que la opción aceptar es verde (positivo) y la opción rechazar es roja (negativo).
- Percepción y atención: para separar el contenido que se muestra en la pantalla se han usado espacios en blanco y cambios en los tipos de letra, bien sea por un tamaño de letra diferente o un color diferente. Se proporciona retroalimentación continua al usuario marcando el borde del elemento en el que tiene puesto el puesto con el color característico de la aplicación.

3.4.2.3. Implementación de los módulos

Cada módulo definido se puede ver reflejado en las diferentes secciones que tiene la aplicación. Se puede acceder a cada una de estas secciones a través del *sidebar* que se encuentra en el lateral izquierdo de la interfaz de usuario.

CAPH se ha utilizado en todos los módulos principalmente para gestionar el foco de la aplicación. Para indicar que un elemento de la interfaz de usuario puede tener foco, se

debe indicar el atributo focusable cuando se crea el elemento HTML. En la Figura 19 se muestra la estructura básica de esa gestión del foco. En cada caso se comprueba cuál es el elemento con el foco con el comando \$(event.currentTarget) y la pantalla en la que se encuentra el usuario para llevar a cabo una acción concreta. CAPH además determina de manera automática cual es el nuevo elemento que debe tener el foco cuando el usuario pulsa una tecla del mando a distancia para desplazarse. También se ha usado CAPH cada vez que se muestra una lista en la interfaz de usuario se ha hecho mediante el método caph.List().

```
$.caph.focus.activate(function(nearestFocusableFinderProvider, controllerProvider) {
    controllerProvider.onFocused(function(event, originalEvent) {
        // acción a realizar cuando un elemento tiene el foco
    });

controllerProvider.onBlurred(function(event, originalEvent) {
        // acción a realizar cuando un elemento acaba de perder el foco
    });

controllerProvider.onSelected(function(event, originalEvent) {
        // acción a realizar cuando un elemento ha sido seleccionado
        });

""
```

Figura 19. Fragmento de código de estructura básica de gestión del foco con CAPH.

También es muy importante con respecto al mando a distancia la tecla de retorno, porque es la que determina el cierre de la aplicación. Para ello se ha añadido un *listener* a esa tecla (tizenhwkey) y en función de la pantalla de la aplicación en la que se encuentre el usuario, se vuelve a la pantalla previa o se sale de la aplicación mediante el comando tizen.application.getCurrentApplication().exit().

Por último, destacar que en todos los módulos se utilizan métodos ajax de JQuery para realizar el intercambio de datos con la API de GoWatch.

3.4.2.3.1. Perfil

Las funciones de este módulo se reflejan tanto en la sección de perfil como en la primera pantalla de selección de usuario. En esta pantalla se detecta el tipo de usuario que va a usar la aplicación, si un usuario logueado o uno no registrado. Cuando un usuario no registrado utiliza la aplicación, solo tiene disponibles las secciones de "Tendencias" y "Búsqueda". Aquí también se ofrecen las opciones de iniciar sesión y de registrarse en la aplicación. La peculiaridad de esta última es que no se realiza en la propia aplicación, sino que en la aplicación se muestra un código QR para que un usuario no registrado lo pueda escanear con su dispositivo móvil y se le abrirá una página web con un formulario de registro. El formulario tiene el aspecto que se puede ver en la Figura 20 y se encuentra en la página web "https://alumnes-ltim.uib.es/gowatch/html/signin.html". Se ha realizado el registro de esta manera para facilitar dicha actividad al usuario, ya que rellenar todos estos campos a través del mando a distancia sería una tarea tediosa y aumenta la probabilidad de que el usuario que se quiere registrar introdujera mal alguno de los campos por la falta de costumbre de utilizar el mando para escribir texto.

El formulario controla mediante JavaScript que se cumplan todos los requisitos para que el registro sea válido. Si algún campo no es válido aparece un mensaje de error en el

propio campo para que el usuario se percate y lo pueda corregir. También vale la pena destacar el mensaje que aparece antes del botón "Confirmar", en el cual se declara que el usuario consiente el uso de *cookies* al realizar el registro. Finalmente, si el registro se realiza correctamente, al usuario le aparece una pantalla de confirmación, mientras que si ha ocurrido algún problema este se mostrará en la misma pantalla.

Registro en GoWatch

Nombre
Introduce tu nombre (apellidos opcionales)
Nombre de usuario
Introduce tu nombre de usuario
Correo electrónico
Introduce tu dirección de correo electrónico
Contraseña (Requisistos: entre 8-26 caracteres, minúscula, mayúscula, número, caracter especial)
Introduce tu nueva contraseña
Confirma la contraseña
Introduce tu nueva contraseña
Al registrarte en GoWatch aceptas el uso de cookies en tu televisor para poder ofrecerte contenido personalizado
CONFIRMAR

Figura 20. Formulario de registro en la aplicación GoWatch.

Ahora ya en la propia sección "Perfil" de la aplicación, se muestra información sobre el usuario logueado (activo). En esa sección tiene disponibles botones para cambiar su nombre de usuario, avatar, dirección de correo electrónico y contraseña. Estas 2 últimas, también se realizan a través de formularios web a los cuales el usuario logueado puede acceder a través de un correo que habrá recibido tras pulsar alguno de dichos botones. Se ha realizado de esta manera para mejorar la usabilidad al igual que en el caso del registro, pero también para mejorar la seguridad, ya que sino cualquier persona con acceso al televisor podría cambiar estos campos de los usuarios logueados ya que estos no necesitan ningún tipo de autorización para acceder con normalidad a la aplicación. El formulario web del cambio de dirección de correo electrónico y de contraseña se pueden ver, respectivamente, en las figuras Figura 21 y Figura 22, y sus páginas web son "https://alumnes-ltim.uib.es/gowatch/html/change_email.html" y "https://alumnesltim.uib.es/gowatch/html/change_password.html". Ambos formularios también validan los campos mediante JavaScript y al igual que el anterior, si el cambio se realiza correctamente, al usuario le aparece una pantalla de confirmación, mientras que si ha ocurrido algún problema este se mostrará en la misma pantalla.

Nombre de usuario Introduce tu nombre de usuario Nueva dirección de correo electrónico Introduce tu nueva dirección de correo electrónico Confirma la dirección de correo electrónico Introduce tu nueva dirección de correo electrónico Confirma la dirección de correo electrónico

Figura 21. Formulario de cambio de dirección de correo electrónico.

Cambio de contraseña



Figura 22. Formulario de cambio de contraseña.

También, dentro de esta misma sección el usuario logueado (activo) puede modificar su lista de plataformas *streaming* y su preferencia sobre las notificaciones de recomendaciones y de nuevos capítulos. Además, desde aquí se puede cambiar de cuenta, cerrar la sesión y eliminar la cuenta. Cada opción conlleva unas acciones diferentes, pero todas desencadenan en que se vuelve a la pantalla inicial de selección de tipo de usuario.

3.4.2.3.2. Amigos

En esta sección el usuario logueado (activo) visualiza un listado de sus amigos (avatar y nombre de usuario) y las solicitudes de amistad que ha recibido, junto con las opciones de aceptarla o rechazarla para cada una. También tiene las opciones de eliminar amigos y de añadir nuevos. Para esto último debe introducir en un campo de texto el nombre de usuario del usuario registrado al que quiere añadir como amigo. Una vez recibida la respuesta a esta petición por parte de la API, se muestra en la interfaz un mensaje temporal con la respuesta recibida. Esta respuesta bien puede indicar que la solicitud de amistad se ha enviado correctamente o que ha ocurrido un error, como por ejemplo que no exista un usuario registrado con el nombre de usuario introducido.

3.4.2.3.3. Tendencias

En esta sección el usuario puede ver las 20 series, películas e integrantes más populares del día. En la parte inferior de la interfaz se muestra un listado con las tendencias del tipo de contenido seleccionado y la parte superior cambia dinámicamente según el contenido en el que el usuario tenga puesto el foco. El usuario puede consultar cada uno de estos contenidos para conocer sus detalles.

Si se trata de una serie o película, se muestra el título, el año, la calificación y resumen de esta, además de una imagen alternativa a la del póster. Si se trata de un integrante, se muestra el género de este, su departamento principal y 3 series o películas por las que es reconocido.

3.4.2.3.4. Búsqueda

Inicialmente se muestra únicamente un campo de texto en el que el usuario puede introducir su búsqueda. Cuando envíe la petición, se devolverán los resultado de todo tipo que coincidan con el texto que ha introducido en el campo de texto. Una vez ya ha recibido resultados, el usuario puede filtrar si quiere visualizar series, películas o integrantes.

El usuario puede consultar los detalles de cada uno de estos contenidos al seleccionarlo. En las películas y series se mostrará su disponibilidad en las plataformas *streaming* y se podrá acceder directamente a la aplicación desde ahí gracias al comando tizen.application.launch(applD). El parámetro applD es el ID interno que tiene la aplicación de la plataforma *streaming* en el propio televisor. Para obtenerlo, hay que llamar previamente a tizen.application.getAppsInfo(onListInstalledApps), método que invoca a la función que se le pasa por parámetro con pasándole por parámetro un listado de las aplicaciones que ha encontrado instaladas en el televisor. Sabiendo el nombre de la aplicación, se puede obtener el ID de esta y así obtener el valor de applD.

Si el usuario está logueado, se mostrarán una serie de opciones que el usuario tiene disponibles para interactuar con dicho contenido (marcar favorito, seguir/ver, marcar pendiente, enviar, etc.). Además, si ya tiene una relación con ese contenido, las opciones se muestran diferentes dependiendo del estado de la relación. Para la opción de enviar como recomendación, se controla que solo pueda enviarla a sus amigos ya que solo se le muestran disponibles sus amigos como posibles destinatarios.

En el caso de tratarse de una serie, el usuario logueado solo tiene la opción de seguir la serie si esta ya se ha estrenado, que se comprueba con la fecha de emisión del primer capítulo de la primera temporada. La opción de marcar la serie como vista solo se muestra si el estado de la serie es "Finalizada" o "Cancelada". La opción de marcar como favorita solo está disponible para el usuario logueado si ya tiene una relación con la serie. Además, si consulta las diferentes temporadas de la serie, los capítulos que ya ha visto se muestran diferentes a los que no ha visto. Por eso, si selecciona un capítulo que ya ha visto este se marca como no visto y al contrario, si selecciona un capítulo no visto, se marca como visto.

En el caso de tratarse de una película, el usuario logueado solo tendrá la opción de ver la película y marcarla como vista si esta ya se ha estrenado. La opción de marcar como favorita solo estará disponible para el usuario logueado si ya tiene una relación con la película.

3.4.2.3.5. Mis series

El usuario logueado (activo) puede consultar todas las series con las que tiene relación, que se muestran agrupadas según el estado de la relación entre el usuario logueado (activo) y la serie. También puede consultar los detalles de cada una de estas series al seleccionarla.

Cuando el usuario logueado (activo) está visualizando las series que está siguiendo, se muestra un listado con sus plataformas *streaming* para que así pueda filtrar las series en función de la plataforma en la que la está siguiendo. La plataforma *streaming* en que un usuario logueado está viendo una serie se determina cuando este selecciona una de las plataformas *streaming* que se muestran en el apartado de disponibilidad de la serie. Si el usuario logueado (activo) selecciona otra plataforma *streaming*, se actualiza la plataforma *streaming* en la que la está viendo. Además, para que el usuario logueado (activo) no tenga que desplazarse hasta la temporada en la que se encuentra actualmente para saber el capítulo que le toca ver, en la misma pantalla donde se muestra el listado de todas las series que está siguiendo, se muestra debajo de cada serie el siguiente capítulo representado con el formato Temp. (Temporada) x, cap. (capítulo) y.

3.4.2.3.6. Mis películas

El usuario logueado (activo) puede consultar todas las películas con las que tiene relación, que se muestran agrupadas según el estado de la relación entre el usuario logueado (activo) y la película. También puede consultar los detalles de cada una de estas series al seleccionarla.

Al igual que con las series que el usuario logueado (activo) está siguiendo, en el apartado de las películas que el usuario logueado (activo) está viendo también se muestra un listado con sus plataformas *streaming* aportando esa función de filtrado. La manera en la que se indica la plataforma *streaming* en la que el usuario logueado está viendo la película es exactamente la misma a la explicada con las series, simplemente tiene que seleccionar dicha plataforma en el apartado de disponibilidad cuando consulta sus detalles.

3.4.2.3.7. Favoritos

El usuario logueado (activo) puede consultar todos sus favoritos, que están agrupados dependiendo de si se trata de una serie, película o integrante. También puede consultar los detalles de cada uno de sus favoritos.

En el caso de las series y las películas, estas se pueden encontrar también en las secciones "Mis series" y "Mis películas" respectivamente, ya que se ha especificado el requisito de que debe existir una relación previa entre el usuario y la serie/película para que este la haya podido marcar como favorita. Sin embargo, al tener la posibilidad de marcar las series y películas como favoritos, el usuario logueado (activo) puede encontrar de una manera más fácil y directa estos contenidos que realmente le han gustado.

3.4.2.3.8. Recomendaciones

El usuario logueado (activo) puede ver las recomendaciones nuevas que ha recibido o todas las recomendaciones. Cuando está visualizando las recomendaciones nuevas, puede filtrarlas según el amigo que se la ha enviado. Una vez selecciona cualquiera de estas recomendaciones nuevas para consultar sus detalles, se llama a la ruta de la API que

la marca como vista y de esta manera ya no se considera una recomendación nueva. Si el usuario logueado (activo) decide que quiere ver todas las recomendaciones que ha recibido, dispone de un botón en la parte superior derecha para ello y se le carga un listado con todas las recomendaciones, mostrando de cada una el título, el amigo y la fecha en la que se la ha enviado. Puede borrar las recomendaciones que quiera pulsando el botón "Borrar" y luego simplemente seleccionar las recomendaciones que quiera borrar. Para indicar que ya ha finalizado de borrar, tiene que volver a pulsar el mismo botón que ahora muestra el texto "Listo".

3.5. Instalación

En este apartado se explicará cómo se lleva a cabo la instalación de la aplicación web creada con Tizen Studio en un televisor Samsung *Smart TV*. Los pasos a seguir son:

- 1. Crear una cuenta en Samsung. Se puede hacer en el propio televisor desde Apps.
- 2. Activar el modo desarrollador en el televisor. Sin salir de Apps, se debe introducir "12345" de manera seguida con el mando a distancia. Se abrirá una ventana en la que hay que introducir la dirección IP del ordenador en el que tenemos instalado Tizen Studio y contiene la aplicación. Para que sea posible establecer una conexión entre el ordenador y el televisor, es necesario que ambos estén conectados a la misma red Wi-Fi. Para finalizar la activación del modo desarrollador, se debe reiniciar el televisor.
- 3. Añadir el televisor Samsung *Smart TV* a los dispositivos de Tizen Studio. Hay que abrir la herramienta *Device Manager* de Tizen Studio y pulsar el la opción *Remote Device Manager*. Se abrirá una ventana desde la cual se pueden añadir nuevos dispositivos y hay que pulsar "+" para conectar nuestra Samsung *Smart TV*. A continuación, se debe introducir el nombre que se le quiere dar al televisor y su IP (Configuración > General > Red > Estado de la red > Configuración de IP).
- 4. Desde Tizen Studio, seleccionar el televisor que se acaba de conectar como dispositivo de destino y compilar y empaquetar el proyecto de la aplicación web. Por último, seleccionar *Run As > Tizen Web Application* y comenzará la instalación en el televisor.

Capítulo 4. Resultados

En este capítulo se hará un breve recorrido por la aplicación de GoWatch, para que se puedan ver de manera visual los resultados tras el proceso de implementación.

Al abrir la aplicación, la primera pantalla que se muestra es la que se ve en la Figura 23, sin ningún usuario logueado. Si el usuario que va a usar GoWatch ya está registrado, puede iniciar sesión y ya aparecerá su usuario (ver Figura 24). Los usuarios que se hayan logueado aparecerán de inicio en esta pantalla aunque se cierre la aplicación o se apague el televisor. Si por un contrario, se trata de un usuario no registrado, puede pulsar el botón "No tengo cuenta" y se le mostrará la pantalla de la Figura 25. En esta pantalla se introduce qué es y para qué sirve la aplicación GoWatch, además de las funcionalidades extras de las que disponen los usuarios registrados. Tras esta explicación, se muestra un código QR que al escanearlo redirige al usuario a la página de registro o si no quiere registrarse, puede acceder sin cuenta pulsando el botón "Entrar", pero solo tendrá disponibles las secciones "Búsqueda" y "Tendencias".



Figura 23. Pantalla inicial de la aplicación GoWatch (sin ningún usuario logueado).



Figura 24. Pantalla inicial de la aplicación GoWatch (con un usuario logueado).



Figura 25. Pantalla de registro en GoWatch o de acceso sin cuenta.

Para que un usuario logueado acceda y pase a ser el usuario logueado (activo), simplemente tiene que seleccionar su avatar y ya se le abrirá la aplicación en la sección de "Tendencias" por defecto (ver Figura 26). El *sidebar* que se puede observar en la Figura 26 solo se despliega cuando el foco está en el propio *sidebar*, sino se muestran solo los iconos de las diferentes secciones de la aplicación. Además de los nombres de las secciones cuando está desplegado, también se puede observar en la parte inferior los logos de TMDB y JustWatch, ya al completar el registro en TMDB se establecía la condición de que se debía dar crédito al usarlos como fuente de información.



Figura 26. Pantalla de "Tendencias" (inicial) de la aplicación GoWatch.

Se pueden visualizar las tendencias de series, películas o integrantes. Al tener el foco sobre cualquiera de los contenidos que se muestran en el listado de tendencias, en la parte superior de la pantalla se visualiza la información más relevante de este junto con una imagen alternativa (ver Figura 27).



Figura 27. Pantalla de "Tendencias" (series) de la aplicación GoWatch.

Desde la sección "Perfil" (ver Figura 28) el usuario logueado (activo) puede realizar todas las funciones de dicho módulo, como por ejemplo cambiar su avatar (ver Figura 29).

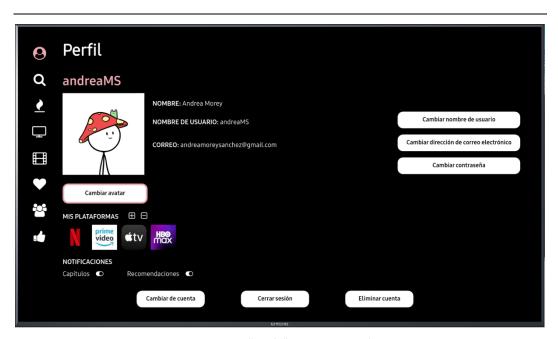


Figura 28. Pantalla de "Perfil" de la aplicación GoWatch.

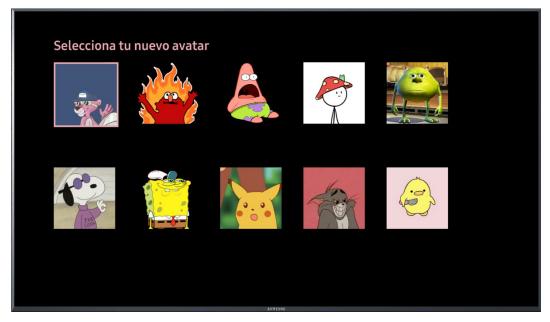


Figura 29. Pantalla de cambio de avatar de la aplicación GoWatch.

En la sección "Búsqueda", aparece un campo de texto para que el usuario pueda introducir su búsqueda con el teclado virtual que se abrirá en la interfaz de usuario al seleccionar el campo de texto (ver Figura 30). Cuando el usuario pulse "Go" en el teclado virtual, este se cerrará y aparecerán los resultados de su búsqueda (ver Figura 31). También aparecerán las opciones de filtrado por tipo de contenido, con las cuales el usuario podrá ver los resultados únicamente del tipo de contenido que ha seleccionado (ver Figura 32).



Figura 30. Pantalla de "Búsqueda" (con teclado virtual) de GoWatch.

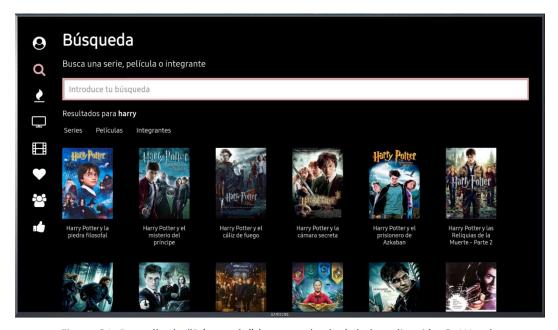


Figura 31. Pantalla de "Búsqueda" (con resultados) de la aplicación GoWatch.

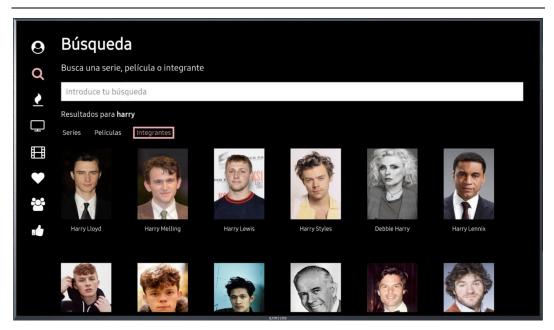


Figura 32. Pantalla de "Búsqueda" (con resultados filtrados por tipo de contenido) de la aplicación GoWatch.

En la sección "Mis series", el usuario logueado (activo) puede consultar las series con las que tiene relación agrupadas por el estado el que se encuentra la relación. En la Figura 33 se pueden ver las series que el usuario logueado (activo) está siguiendo y de cada serie se indica el siguiente capítulo que le toca ver. Al estar en el apartado de las series que está siguiendo, se le muestra un listado de sus plataformas para poder filtrar las series que está siguiendo en caso de haberlo indicado previamente.

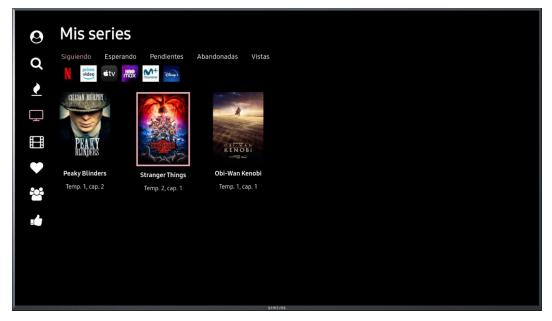


Figura 33. Pantalla de "Mis series" (Siguiendo) de la aplicación GoWatch.

Como el usuario logueado (activo) está siguiendo la serie Stranger Things, se le muestra la opción de seguir serie como "Siguiendo" y con el color característico de la aplicación (ver Figura 34). Además, el usuario tiene marcada la serie como favorita y tiene las notificaciones activadas. También se muestra la plataforma *streaming* en la que el usuario está siguiendo la serie.



Figura 34. Pantalla de la serie Stranger Things de la aplicación GoWatch.

Como el siguiente capítulo que le toca ver al usuario logueado (activo) de esta serie es el capítulo 1 de la temporada 2, si se consultan los detalles de la temporada 1 (ver Figura 35), aparecen todos los capítulos marcados como vistos y el botón de marcar temporada como vista también aparece con el color característico de la aplicación.

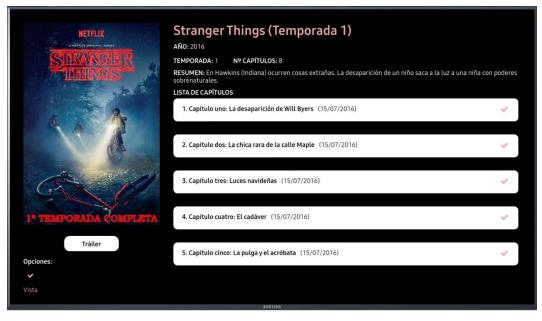


Figura 35. Pantalla de la temporada 1 de Stranger Things de la aplicación GoWatch.

En la sección "Mis películas", el usuario logueado (activo) puede consultar las películas con las que tiene relación agrupadas por el estado el que se encuentra la relación. En la Figura 36 se pueden ver las películas que el usuario logueado (activo) ya ha visto. Si el usuario logueado (activo) consulta los detalles de la película The Batman, le aparece la opción de marcar película vista como "Vista" y con el color característico de la aplicación (ver Figura 37).



Figura 36. Pantalla de "Mis películas" (Vistas) de la aplicación GoWatch.



Figura 37. Pantalla de la película The Batman de la aplicación GoWatch.

En la sección "Favoritos", el usuario logueado (activo) puede consultar las series, películas e integrantes que tiene marcados como favorito (ver Figura 38). Si el usuario logueado

(activo) consulta los detalles de la integrante Zendaya, le aparece la opción de marcar como favorito con el color característico de la aplicación (ver Figura 39).



Figura 38. Pantalla de "Favoritos" (Integrantes) de la aplicación GoWatch.

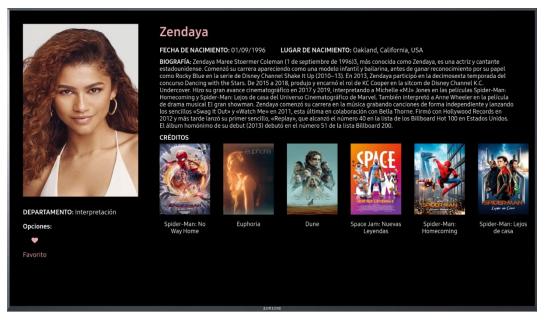


Figura 39. Pantalla de la integrante Zendaya de la aplicación GoWatch.

En la sección "Amigos", el usuario logueado (activo) puede ver un listado de sus amigos y las solicitudes de amistad que ha recibido (ver Figura 40). También puede añadir amigos pulsando el botón "Añadir amigo" e introduciendo el nombre de usuario del usuario registrado que quiere añadir como amigo en el campo de texto que se le mostrará en la interfaz de usuario.

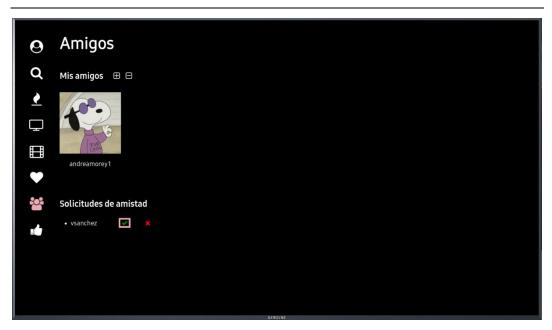


Figura 40. Pantalla de "Amigos" de la aplicación GoWatch.

En la sección "Recomendaciones", el usuario logueado (activo) puede ver las nuevas recomendaciones que ha recibido (ver Figura 41) y un historial de todas las recomendaciones recibidas, las cuales puede borrar si así lo desea.



Figura 41. Pantalla de "Recomendaciones" (nuevas) de la aplicación GoWatch.

Capítulo 5. Conclusiones

Finalmente, se ha conseguido cumplir el objetivo que se había planteado en el capítulo inicial, el desarrollo de GoWatch. Se han podido implementar todos los requisitos funcionales definidos y también se han seguido las restricciones impuestas por los requisitos no funcionales.

La aplicación se ha podido probar en el emulador de Tizen Studio, pero sin embargo no ha sido posible probarla en una Samsung *Smart TV*, porque una vez instalada la aplicación en el televisor del que se disponía, esta no respondía tal y como se esperaba ni tampoco se podía cargar el contenido de manera correcta. Esto se debe a que la aplicación se ha desarrollado en la versión 6.5 de Tizen del año 2022, mientras que el televisor disponible es del año 2016 y con una versión Tizen 2.4.

En el apartado personal, creo que realizar este proyecto me ha aportado muchos conocimientos nuevos, sobretodo la parte *backend* ya que prácticamente no había trabajado en esta parte de las aplicaciones. Además, la implementación de la aplicación me ha permitido poner en práctica y potenciar lo aprendido en el curso de Samsung Dev Spain "Desarrollo de aplicaciones Tizen para *Smart TV*".

Cabe destacar que durante el desarrollo de la aplicación se ha detectado una limitación de la API de TMDB. La API actualmente soporta 50 peticiones/segundo y 20 conexiones por IP con tal de evitar ataques DDoS. Esta restricción ha supuesto un impedimento sobretodo a la hora de cargar una gran cantidad de imágenes simultáneamente, como por ejemplo en los resultados de la búsqueda o cuando se muestran nuevas plataformas streaming que el usuario logueado (activo) puede añadir a su lista. Por este motivo, en ciertas ocasiones se han dado problemas con la carga de imágenes y se obtenían errores 429 (Too Many Requests) por parte de la API de TMDB.

En el apartado de futuras mejoras, la principal mejora que se presenta es la de aumentar el número de potenciales usuarios. Ya que Samsung es una marca que domina el mundo de los *Smart TV* a nivel global, muchos usuarios de diferentes países y con distintos idiomas disponen de estos televisores en sus hogares. Por tanto, la aplicación podría ofrecer a los usuarios la posibilidad de configurar el idioma de la aplicación. Además, como el contenido que ofrecen las plataformas *streaming* es diferente en cada país, sería idóneo que el usuario pueda indicar su país de residencia, para que así se muestre la disponibilidad del contenido en las plataformas *streaming* del país del usuario.

Otra funcionalidad que se podría implementar en el futuro es la de sincronizar el calendario de los usuarios registrados con la aplicación. De esta manera, además de recibir una notificación por correo el día del estreno de nuevos capítulos, el usuario registrado también podría tener marcado en su calendario por adelantado cuándo serán los estrenos de nuevos capítulos de las series que sigue.

Como se ha querido incluir un factor social en la aplicación de GoWatch, también sería interesante centrarse en ampliar las funcionalidades de este tipo.

La primera de ellas sería la de poder reaccionar mediante a emoticonos que expresen emociones diferentes a las recomendaciones recibidas de tus amigos en la aplicación. Así el usuario que la envió puede saber si acertó recomendando dicho contenido y también se consigue aumentar la interacción entre los usuarios registrados amigos.

Otra posible incorporación futura es la de saber qué series o películas están siguiendo/viendo tus amigos en ese momento. Esta idea ha surgido a raíz de la funcionalidad que recientemente incorporó Spotify [13], "Actividad de amigos". Con esto Spotify muestra a los amigos del usuario lo que este está escuchando y le muestra al usuario lo que escuchan sus amigos. Evidentemente, en caso de que un usuario registrado no quisiera compartir dicha información se podría desactivar, tal y como ocurre en Spotify.

Por último, se podrían añadir recomendaciones personalizaciones a los usuarios en función de sus favoritos. Tal y como está la aplicación, se puede ver contenido similar al que se está consultando, por tanto sería aplicar este mismo concepto pero con las series y películas que el usuario tiene marcadas como favoritos. En el caso de los integrantes favoritos, se podrían recomendar series o películas populares de dicho integrante. Para acotar el rango de estas recomendaciones, se tendrían en cuenta factores como que el usuario no haya visto la serie o película y que esta se encuentre disponible en alguna de las plataformas *streaming* que el usuario tiene contratadas.

Capítulo 6. Referencias bibliográficas

A continuación se muestra una listado de todas las citas y referencias, en formato IEEE, que se han empleado a lo largo del documento.

- [1] Evolución del porcentaje de hogares con televisores inteligentes en España entre 2015 y 2021, «statista,» 2022. [En línea]. Available: https://es.statista.com/estadisticas/597365/porcentaje-de-hogares-con-smart-tv-en-espana/. [Último acceso: 19 Julio 2022].
- [2] Carlota Martínez, «Las plataformas provocan la mayor caída del consumo televisivo en 30 años,» *El independiente,* 13 Marzo 2022.
- [3] Rasmus Larsen, «Samsung tops global TV market for 16th straight year,» flatpanelshd, 21 febrero 2022.
- [4] Whip Media, «TV Time,» Whip Media, [En línea]. Available: https://www.tvtime.com. [Último acceso: 21 Julio 2022].
- [5] «CVM Platform,» Whip Media, [En línea]. Available: https://www.whipmedia.com/#. [Último acceso: 21 Julio 2022].
- [6] N. Nachev, «Next Episode,» [En línea]. Available: https://next-episode.net. [Último acceso: 21 Julio 2022].
- [7] «TheTVDB,» Whip Media, [En línea]. Available: https://thetvdb.com/api-information. [Último acceso: 21 Julio 2022].
- [8] «TVmaze,» [En línea]. Available: https://www.tvmaze.com/api. [Último acceso: 21 Julio 2022].
- [9] «The Movie Database,» 2008. [En línea]. Available: https://www.themoviedb.org. [Último acceso: Julio 2022].
- [10] «JustWatch,» [En línea]. Available: https://www.justwatch.com. [Último acceso: 22 Julio 2022].
- [11] The Movie Database, «The Movie Database API,» [En línea]. Available: https://developers.themoviedb.org/3/getting-started/introduction. [Último acceso: 22 Julio 2022].

- [12] Youtube, «Youtube Data API,» [En línea]. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2a hUKEwip6ojtt_H5AhW3hM4BHT2HAQAQFnoECAlQAQ&url=https%3A%2F%2Fde velopers.google.com%2Fyoutube%2Fv3&usg=AOvVaw1vqb6TFqS_49i0ix264s-i. [Último acceso: Julio 2022].
- [13] «Spotify,» [En línea]. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja &uact=8&ved=2ahUKEwiw4vLCpIH6AhW2_7sIHRjgAW4QFnoECAQQAQ&url=http s%3A%2F%2Fwww.spotify.com%2Fes%2F&usg=AOvVaw1I7_auvcmx2BWubLJI4Y gW. [Último acceso: Septiembre 2022].
- [14] «unlayer,» [En línea]. Available: https://unlayer.com. [Último acceso: Agosto 2022].

Apéndice A. Creación base de datos

```
CREATE DATABASE `GoWatch`;
USE `GoWatch`;
CREATE TABLE avatar(
idAvatar INT(2) PRIMARY KEY AUTO_INCREMENT,
imagen VARCHAR(2083) NOT NULL
CREATE TABLE usuarioRegistrado(
idUsuario INT(10) PRIMARY KEY AUTO_INCREMENT,
nombre CHAR(50) NOT NULL,
correo CHAR(64) NOT NULL,
nombreUsuario CHAR(30) NOT NULL,
contraseña CHAR(100) NOT NULL,
notificacionesCapitulos BOOLEAN DEFAULT TRUE,
idAvatar INT(2) NULL,
FOREIGN KEY (idAvatar) REFERENCES avatar(idAvatar)
CREATE TABLE integrante(
idIntegrante INT(20) PRIMARY KEY,
nombre CHAR(50) NOT NULL,
imagen VARCHAR(2083) NOT NULL
);
CREATE TABLE serie(
idSerie INT(20) PRIMARY KEY,
titulo CHAR(200) NOT NULL,
poster VARCHAR(2083) NOT NULL,
estado CHAR(30) NULL
);
CREATE TABLE temporada(
idTemporada INT(20) PRIMARY KEY,
```

```
titulo CHAR(200) NOT NULL,
numero INT(3) NOT NULL,
poster VARCHAR(2083) NOT NULL,
idSerie INT(20) NOT NULL,
FOREIGN KEY (idSerie) REFERENCES serie(idSerie)
);
CREATE TABLE capitulo(
idCapitulo INT(20) PRIMARY KEY,
titulo CHAR(200) NOT NULL,
numero INT(3) NOT NULL,
fechaEmision DATE NULL,
idTemporada INT(20) NOT NULL,
FOREIGN KEY (idTemporada) REFERENCES temporada(idTemporada)
);
CREATE TABLE pelicula(
idPelicula INT(20) PRIMARY KEY,
titulo CHAR(200) NOT NULL,
poster VARCHAR(2083) NOT NULL
);
CREATE TABLE plataforma(
idPlataforma INT(20) PRIMARY KEY,
nombre CHAR(200) NOT NULL,
logo VARCHAR(2083) NOT NULL,
prioridad INT(4) NOT NULL
);
CREATE TABLE usuarioSerie(
idUsuario INT(10) NOT NULL,
idSerie INT(20) NOT NULL,
estado ENUM('Pendiente', 'Siguiendo', 'Esperando', 'Abandonada', 'Vista'),
fechaModificacionEstado TIMESTAMP NULL,
favorita BOOLEAN DEFAULT FALSE,
notificaciones BOOLEAN DEFAULT TRUE,
idPlataforma INT(20) NULL,
PRIMARY KEY ('idUsuario', 'idSerie'),
```

```
FOREIGN KEY (idUsuario) REFERENCES usuarioRegistrado(idUsuario),
FOREIGN KEY (idSerie) REFERENCES serie(idSerie),
FOREIGN KEY (idPlataforma) REFERENCES plataforma(idPlataforma)
CREATE TABLE usuarioPelicula(
idUsuario INT(10) NOT NULL,
idPelicula INT(20) NOT NULL,
estado ENUM('Pendiente', 'Viendo', 'Vista'),
fechaModificacionEstado TIMESTAMP NULL,
favorita BOOLEAN DEFAULT false,
idPlataforma INT(20) NULL,
PRIMARY KEY ('idUsuario', 'idPelicula'),
FOREIGN KEY (idUsuario) REFERENCES usuarioRegistrado(idUsuario),
FOREIGN KEY (idPelicula) REFERENCES pelicula(idPelicula),
FOREIGN KEY (idPlataforma) REFERENCES plataforma(idPlataforma)
);
CREATE TABLE recomendacion (
idRecomendacion INT(20) PRIMARY KEY AUTO_INCREMENT,
fecha TIMESTAMP NOT NULL,
vista BOOLEAN DEFAULT false,
idUsuarioA INT(10) NOT NULL,
idUsuarioB INT(10) NOT NULL,
idPelicula INT(20) NULL,
idSerie INT(20) NULL,
FOREIGN KEY (idUsuarioA) REFERENCES usuarioRegistrado(idUsuario),
FOREIGN KEY (idUsuarioB) REFERENCES usuarioRegistrado(idUsuario),
FOREIGN KEY (idPelicula) REFERENCES pelicula(idPelicula),
FOREIGN KEY (idSerie) REFERENCES serie(idSerie)
);
CREATE TABLE r_UsuarioRegistrado_integrante(
idUsuario INT(10) NOT NULL,
idIntegrante INT(20) NOT NULL,
PRIMARY KEY ('idUsuario', 'idIntegrante'),
FOREIGN KEY (idUsuario) REFERENCES usuarioRegistrado(idUsuario),
FOREIGN KEY (idIntegrante) REFERENCES integrante(idIntegrante)
```

```
);
CREATE TABLE r_UsuarioRegistrado_plataforma(
idUsuario INT(10) NOT NULL,
idPlataforma INT(20) NOT NULL,
PRIMARY KEY ('idUsuario', 'idPlataforma'),
FOREIGN KEY (idUsuario) REFERENCES usuarioRegistrado(idUsuario),
FOREIGN KEY (idPlataforma) REFERENCES plataforma(idPlataforma)
);
CREATE TABLE r_capitulo_usuario(
idCapitulo INT(20) NOT NULL,
idUsuario INT(10) NOT NULL,
fechaVisto TIMESTAMP NULL,
PRIMARY KEY ('idCapitulo', 'idUsuario'),
FOREIGN KEY (idCapitulo) REFERENCES capitulo(idCapitulo),
FOREIGN KEY (idUsuario) REFERENCES usuarioRegistrado(idUsuario)
);
CREATE TABLE r_usuarioRegistrado_usuarioRegistrado(
idUsuarioA INT(10) NOT NULL,
idUsuarioB INT(10) NOT NULL,
fechaSolicitud TIMESTAMP NOT NULL,
fechaAceptada TIMESTAMP NULL,
PRIMARY KEY ('idUsuarioA', 'idUsuarioB'),
FOREIGN KEY (idUsuarioA) REFERENCES usuarioRegistrado(idUsuario),
FOREIGN KEY (idUsuarioB) REFERENCES usuarioRegistrado(idUsuario)
);
```

Apéndice B. Correos

Los correos se han creado a partir de plantillas obtenidas de la página web unlayer [14].

¡Ya puedes seguir viendo tu serie!



Peaky Blinders



Figura 42. Correo de notificación de nuevos capítulos disponibles.



Si no has solicitado un cambio de dirección de correo electrónico, simplemente ignora este correo.

Figura 43. Correo de solicitud de cambio de dirección de correo electrónico.



Figura 44. Correo de solicitud de cambio de contraseña.

¡andreaMS quiere ser tu amigo!



Puedes aceptar o rechazar la solicitud desde la app de GoWatch

Figura 45. Correo de nueva solicitud de amistad recibida.

¡andreamorey1 ha aceptado tu solicitud de amistad!



Figura 46. Correo de aceptación de solicitud de amistad.

andreamorey1 ha rechazado tu solicitud de amistad

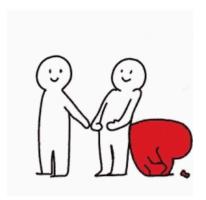


Figura 47. Correo de denegación de solicitud de amistad.

andreaMS ha dejado de ser tu amigo

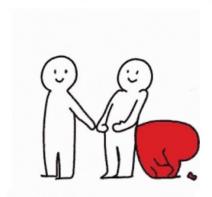


Figura 48. Correo de pérdida de un amigo.

andreamorey1 te ha recomendado



Thor: Love and Thunder

Película

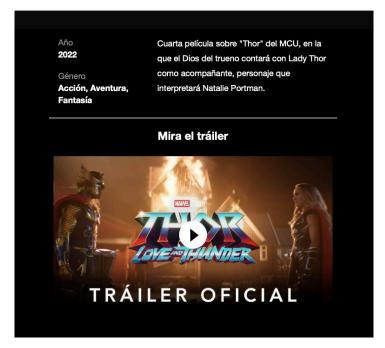


Figura 49. Correo de recomendación recibida.

Apéndice C. Creación de una aplicación web con Tizen Studio

El primer paso para crear una aplicación para el sistema operativo Tizen es descargarse el IDE Tizen Studio. Este se puede descargar de manera gratuita desde la página web oficial de Tizen. La última versión disponible de Tizen Studio es la 4.6, pero este proyecto se ha desarrollado con la versión 4.5.1. Tras descargarlo, será necesario abrir *Package Manager* e instalar *Tizen SDK tools* de la sección *Main SDK y TV Extensions-6.5 y TV Extensions Tools* de la sección *Extension SDK*. También se requiere un certificado de autor y un certificado de distribuidor, que se pueden crear con *Tizen Certificate Manager*.

Cuando ya se hayan realizado todos estos pasos, ya es posible crear el proyecto de la aplicación con Tizen Studio. Los pasos a seguir son:

- 1. Crear un nuevo proyecto Tizen (File > New > Tizen Project).
- Configurar las propiedades del proyecto. Se deben seleccionar *Template* como tipo de proyecto y TV-samsung v6.5 como perfil y versión del proyecto. Lo siguiente es seleccionar el tipo de aplicación, en este caso se ha seleccionado *Web Application*.
- 3. Seleccionar la plantilla de la aplicación. En este caso se ha elegido *Caph-Empty Template for jQuery*. CAPH es un UI *framework* creado para el facilitar el desarrollo de aplicaciones web para Smart TV.
- 4. Asignar un nombre al proyecto.

Una vez creado el proyecto, hay que incluir los privilegios que muestran en la Figura 50 en el archivo de configuración config.xml. Estos privilegios permiten utilizar el mando a distancia, el acceso a Internet y las descargas de ficheros de Internet.

```
<tizen:privilege name="http://tizen.org/privilege/tv.inputdevice"/>
<tizen:privilege name='http://tizen.org/privilege/download'/>
<tizen:privilege name="http://tizen.org/privilege/internet"/>
<access origin="*" subdomains="true"/>
```

Figura 50. Fragmento de código con privilegios Tizen.

En el SDK de Tizen Studio viene incluido un simulador y un emulador para poder testear las aplicaciones antes de probarlo en un televisor Samsung *Smart TV* real. La recomendación de Samsung es probarla en el emulador, ya que el comportamiento de la aplicación puede ser diferente al esperado en el simulador. En mi caso concreto, desarrollando el proyecto desde un MacBook Pro (2020), fue necesaria la instalación del motor de virtualización asistido por hardware HAXM para poder utilizar el emulador.

Cada vez que se quiera ejecutar la aplicación con los nuevos cambios implementados, es necesario compilarla y empaquetarla seleccionando la opción "Build Signed Package" y volver a instalarla en el emulador. Esto puede resultar muy incómodo durante la fase de desarrollo, ya que no es una operación especialmente rápida. Por este motivo Tizen creó

WITs. WITs es una herramienta que permite acelerar y mejorar el desarrollo de las aplicación web para Tizen. Con WITs, no es necesario empaquetar e instalar la aplicación cada vez que se modifica el código, sino que recarga instantáneamente la aplicación con los nuevos cambios mediante *Live Reload*. Las diferencias entre estos dos procesos se pueden observar en la Figura 51.



Figura 51. Proceso convencional de instalación vs WITs.

Para ejecutar la aplicación con WITs, primero hay que lanzar el emulador o el televisor en el que se quiera instalar la aplicación y desde la terminal acceder al directorio en el que se encuentra el proyecto. La primera vez que se use WITs habrá que instalarlo con el comando sudo npm install -g @tizentv/wits. Una vez instalado, habrá que introducir el comando wits -i y rellenar los parámetros de configuración que se piden (ver Figura 52). Finalmente, con el comando wits -s se lanzará la aplicación a través de WITs en el dispositivo que se haya indicado en la configuración.

```
? Input your Device Ip address(If using Emulator, input 0.0.0.0): 0.0.0.0
? Input your Application width (1920 or 1280): 1920
? Input the path of profile.xml: /Users/andreamoreysanchez/tizen-studio-data/profile/profiles.xml
[? Do you want to launch with chrome DevTools?: Yes
```

Figura 52. Parámetros de configuración de WITs.