

**Grado en Ingeniería Informática de Gestión y Sistemas de
Información**

**Desarrollo de una aplicación de gestión y seguimiento de logros en
videojuegos**

Memoria Final

Carlos Castillo Rivero

TUTOR: Alfredo Rueda Unsain

Curso 2022-2023

Abstract

The project consists of developing a mobile app that introduces, as main features, the management and personalization of the achievements of various video games by its users, among others. The identity of the application is given by the custom achievement creation functionality of the video player community, which differs from other related market applications.

Resum

El projecte consisteix en el desenvolupament d'una aplicació mòbil que introdueix, com a funcionalitats principals, la gestió i personalització dels assoliments de diversos videojocs per part dels usuaris, entre altres. La identitat de l'aplicació ve donada per la funcionalitat de creació d'assoliments personalitzats per part de la comunitat de videojugadors, la qual cosa la diferencia de les altres aplicacions relacionades del mercat.

Resumen

El proyecto consiste en el desarrollo de una aplicación móvil que introduce, como funcionalidades principales, la gestión y personalización de los logros de diversos videojuegos por parte de los usuarios, entre otras. La identidad de la aplicación viene dada por la funcionalidad de creación de logros personalizados por parte de la comunidad de videojugadores, lo que la diferencia de las demás aplicaciones relacionadas del mercado.

Índice

Glosario de términos	II
1. Introducción	4
2. Marco Teórico y Análisis de Referentes	5
2.1 Contexto.....	5
2.2 Antecedentes	6
2.3 Necesidades de información.....	8
3. Objetivos y Alcance	10
4. Metodología.....	12
5. Desarrollo	15
5.1 Análisis y modelado de los datos	15
5.2 Requerimientos Funcionales.....	16
5.3 Requerimientos Tecnológicos	17
5.4 Ingeniería de software	17
5.5 Proceso de desarrollo.....	20
5.6 Proceso de Testing	26
6. Análisis de resultados	36
7. Conclusiones.....	38
8. Posibles ampliaciones	39
9. Bibliografía.....	40

Glosario de términos

- **API:** Conjunto de definiciones y protocolos que se usa para integrar la lógica de aplicaciones entre ellas.
- **Gamificación:** Aplicación de mecánicas y dinámicas propias de videojuegos a contextos ajenos a estos.
- **Achiever:** Tipología de jugador que se caracteriza por completar al máximo posible los videojuegos.
- **Socializer:** Tipología de jugador que se caracteriza por buscar la interacción con otros jugadores durante su experiencia de juego.
- **Xbox360:** Consola creada por Microsoft el año 2005. Esta introdujo el sistema XBOX Live lo que permitió a los usuarios jugar online y descargar diversos contenidos.
- **Atari 2600:** Consola fabricada por Atari el año 1977, que introdujo con éxito los softwares de videojuego en cartuchos intercambiables.
- **PSN:** Play Station Network. Plataforma desarrollada por Sony enfocada a la distribución digital de videojuegos y dar soporte a los videojuegos multijugador online de sus consolas.
- **Framework:** Esquema de trabajo que define la base de proyectos de desarrollo de software.
- **SOLID:** Acrónimo que hace referencia a los principios acuñados por Michael Feathers. Este hace referencia a la elaboración de un software eficaz; con código limpio, flexible, mantenible y reutilizable; y que sea escalable.
- **OAuth2:** Protocolo de autorización de usuarios.
- **APK:** Formato de archivo que contiene la información de una aplicación para poder ser instalada en dispositivos móviles.
- **Back-end:** Faceta de una plataforma que se encarga de acceder a los datos de un software o dispositivo, la cual no pueden acceder los usuarios directamente. En ella se sitúa la lógica de la aplicación y se encarga de gestionar la información requerida.
- **Front-end:** Faceta de una plataforma con la que interactúan los usuarios.
- **Spring boot:** Herramienta para desarrollar software basado en Spring.
- **AWS:** Amazon Web Services. Filial de Amazon que aporta un gran número de servicios en la nube como almacenaje o monitorización.

- **Json Web Tokens:** Framework basado en JSON que garantiza la seguridad y validez de los datos enviados entre aplicaciones mediante el uso y creación de tokens.
- **Oracle:** Desarrollado por Oracle corporation, Oracle es un servicio de almacenaje y recuperación de información relacionada en bases de datos.
- **React:** Biblioteca de lenguaje Javascript utilizada para el desarrollo en front-end de un software.
- **Cloudwatch:** Es uno de los servicios de AWS enfocado a la monitorización del uso en una aplicación por parte de los usuarios.
- **S3:** Es uno de los servicios de AWS enfocado al almacenaje de archivos en la nube.
- **RDS:** Es uno de los servicios de AWS enfocado a la configuración, operación y escalado de bases de datos relacionadas en la nube.
- **UML:** Lenguaje Unificado de modelado. Se utiliza en la etapa de modelado de software y permite generar un diseño de la lógica base de la plataforma a desarrollar.
- **TI:** Tecnologías de la información. Estas tecnologías son con las que interactúan los usuarios para usar los distintos sistemas de información.
- **GitHub:** Herramienta que sirve para alojar proyectos de software en la nube además de contar con un sistema de control de versiones.
- **IDE:** siglas que hacen referencia a “integrated development environment”, que traducido al castellano significa entorno de desarrollo integrado.
- **IntelliJ IDEA:** Desarrollado por JetBrains, IntelliJ es un IDE enfocado al desarrollo de software en ordenadores.
- **IGDB:** Base de datos en línea que recopila una gran variedad de videojuegos y los datos referentes a estos.

1. Introducción

El proyecto nace de la imposibilidad por parte de la comunidad de videojugadores de llevar un registro de logros personalizado por ellos mismos, debido a que las soluciones de mercado actuales como por ejemplo Steam o PSN, así como las plataformas que usan sus API para ello, están totalmente automatizadas, por lo que no permiten ninguna personalización y/o modificación.

Consiste en el desarrollo de una aplicación móvil que introduce, como funcionalidades principales, la gestión y personalización de los logros de diversos videojuegos por parte de los usuarios; y una funcionalidad innovadora, con la cual no cuentan las demás soluciones del mercado, la creación de nuevos logros personalizados por parte de la comunidad de videojugadores. Además de la implementación de otras funcionalidades que, tratándolas como un conjunto, hacen que la aplicación proporcione más elementos útiles y atractivos para los usuarios.

Por otro lado, esta aplicación es una alternativa móvil a las plataformas predominantes actuales, las cuales prestan sus servicios en ordenadores. Este factor es útil en cuanto a accesibilidad y comodidad para los usuarios del producto, pues lo tienen al alcance de la mano y disponible en cualquier situación.

2. Marco Teórico y Análisis de Referentes

2.1 Contexto

Para empezar, hace falta tener claro que es un logro en el ámbito de los videojuegos. Un logro, también llamado trofeo, es un tipo de recompensa que se le otorga al jugador al superar retos específicos planteados en un videojuego. Hay que diferenciarlos de las misiones o niveles de un videojuego, puesto que estos tienen influencia dentro del juego, mientras que los logros se gestionan, en la mayoría de las ocasiones, externamente a este.

La presencia de logros en los videojuegos ayuda a la motivación y fidelización de los usuarios, aunque también sirven para que estos se vean reconocidos por otros jugadores gracias a la posibilidad de compartir sus perfiles en las plataformas pertinentes, lo cual tiene una vertiente social importante. Otra utilidad de los logros, pese a no ser concretamente en videojuegos, es la de ser usados para gamificar diferentes elementos cotidianos como el aprendizaje, el trabajo, etc.

Los sistemas de logros nacieron el año 1980 en algunos juegos de la consola Atari 2600, creados por Activision. El primero de todos fue en un videojuego de carreras llamado "Dragster" y el logro a superar era terminar una carrera en menos de seis segundos. Por aquel entonces no había sistemas gestores de logros, por lo que los usuarios debían sacar una fotografía de su televisor, donde se viese claramente que habían superado ese desafío, y enviarla por correo a Activision, los cuales premiarían con un parche conmemorativo a los jugadores.



Ilustración 1. Parche Activision. Fuente[1]

La creación y despliegue de los logros en las diversas plataformas aparece por primera vez en 2005 con la aparición de la XBOX 360 [2], la cual introducía un sistema de logros e insignias para los videojuegos de su catálogo. Este hecho popularizó los sistemas de logros en las demás plataformas del momento.

2.2 Antecedentes

Actualmente la gestión de los logros en videojuegos se produce en las plataformas de distribución de estos, aunque hay otras soluciones en el mercado que juntan los datos de todas estas plataformas en una. Este tipo de plataformas muestran los datos de los perfiles de los usuarios referentes a los videojuegos que juegan, ya sean datos de logros, horas jugadas, etc.

Todas ellas prestan sus servicios en computadoras; las soluciones en plataformas móviles presentan un comportamiento muy parecido al segundo tipo de plataformas descrito en el párrafo anterior, pero adaptadas a una interfaz para este tipo de dispositivos.

A continuación, se describen diferentes soluciones o productos que comparten una similitud con el proyecto a desarrollar. Dado el gran abanico de servicios, se ejemplifica cada uno con el más representativo.

Plataformas de distribución de videojuegos

Estos servicios están enfocados principalmente en la distribución de videojuegos, aunque lo realmente interesante en relación con el proyecto es su sistema de logros y su integración con los diversos videojuegos.

Un ejemplo de este tipo de plataformas es Steam. Desarrollada por Valve Corporation, Steam es una plataforma de distribución de videojuegos que, entre otros servicios menos relevantes, permite la compra de videojuegos digitales e incorpora un sistema de seguimiento de logros para la mayoría de estos.



Ilustración 2. Logo de Steam. Fuente: [3]

Esta cuenta con una API especialmente diseñada para integrar los logros de un juego con la aplicación. En primer lugar, los desarrolladores del videojuego definen los logros en la plataforma y posteriormente hacen la integración a nivel lógico en el videojuego utilizando la API mencionada.

Aplicaciones de seguimiento de experiencia en videojuegos

Este tipo de aplicaciones están enfocadas principalmente en la gestión de perfiles de usuario y su experiencia en distintos videojuegos.

Los productos de este tipo más destacables son aquellas que permiten crear listas personalizadas por el usuario donde especifica los diferentes detalles de sus videojuegos jugados o aplicaciones que integran las API de distintas plataformas para llevar un registro conjunto de los datos en cada una de ellas. La primera opción permite cierta libertad a los usuarios para escoger lo que estos deseen, mientras que la otra solo permite vincular las cuentas que el usuario quiera, por lo que si algo no está en una de esas cuentas no se ve reflejado en su perfil.

Un ejemplo de ello es la plataforma Exophase.



Ilustración 3. Logo de Exophase. Fuente: [4]

Exophase es una plataforma web creada por Mike Bendel, la cual cuenta con cuatro millones de logros y más de cien mil juegos en su base de datos.

Su funcionalidad más destacable es la de crear perfiles de usuario personalizables donde se muestran los logros obtenidos en las plataformas que este vincule a su cuenta de Exophase. Esto genera una tarjeta de usuario que se puede customizar con los elementos deseados.

Aplicaciones de gamificación de elementos cotidianos

Hay una gran variedad de este tipo de aplicaciones, su función, principalmente, es la de motivar o ayudar al usuario a cumplir objetivos que se proponga mediante la gamificación de estos. La gamificación se puede aplicar de diversas maneras, pero el caso que relativo al proyecto es el de gamificar mediante la introducción de logros.

Para ejemplificar este tipo de aplicaciones se describe la aplicación Habitica.

Esta aplicación, creada por la empresa HabitRPG, permite crear logros personalizados sobre elementos cotidianos del día a día. Ver las noticias, levantarse a la hora que te propongas y hacer ejercicio, son algunos ejemplos de logros que un usuario puede introducir en la aplicación.



Ilustración 4. Logo de Habitica. Fuente: [5]

La intención de esta aplicación se asemeja un poco a lo que se pretende innovar con el proyecto, pues la personalización de los logros, en este caso en la vida real, ayuda a motivar a los usuarios para realizar esas tareas.

2.3 Necesidades de información

Durante el desarrollo de la aplicación es necesario consultar la documentación de los distintos frameworks y servicios, ya que estos contienen la información necesaria para poder realizar las distintas implementaciones correctamente. Los más destacables son los siguientes:

- React [6]
- AWS y sus servicios Cloud watch y S3 [7]
- Spring Boot Framework [8]
- Oracle [9]

Por otro lado, dado que un objetivo es el de implementar una funcionalidad de creación de logros personalizados, hace falta adquirir conocimiento sobre como diseñar un logro correctamente, tanto a nivel lógico como integración en la aplicación. Para ello se debe consultar el framework de diseño y evaluación de logros en videojuegos elaborado por Juhon Hamari y Veiko Eranti [10].

Finalmente, también se debe consultar la documentación sobre las API de las distintas plataformas de distribución de videojuegos actuales, con el objetivo de realizar una correcta integración de estas con la aplicación.

3. Objetivos y Alcance

El cliente principal y usuario final del producto son todos los videojugadores, prestando especial atención a los dos siguientes: los de tipología *achiever* y *socializer* [11], los cuales representan un 90% de estos. Pese a ser estos anteriores el público potencial, los desarrolladores de videojuegos también entran dentro del rango de cliente, pues hay funcionalidades específicamente diseñadas para ellos.

Teniendo esto en cuenta se definen los siguientes objetivos principales:

- Desarrollar una aplicación en la cual el usuario sea capaz de gestionar y mantener un seguimiento de los logros en diversos videojuegos.
- Permitir a los usuarios de la aplicación una gestión libre y personalizada del sistema de logros.
- Permitir a los usuarios de la aplicación la creación de logros para distintos videojuegos.
- Integrar las API de otras plataformas para que el usuario pueda vincular sus diversas cuentas y se muestren los logros ya obtenidos y verificados.
- Crear una interfaz visual atractiva e intuitiva para los usuarios.

Objetivos secundarios:

- Implementar la recuperación de la contraseña por parte del usuario. Este puede recuperar la contraseña utilizando su correo electrónico.
- Crear una APK funcional del producto.
- Dar la posibilidad a los usuarios de comunicarse entre ellos mediante chats y foros de comunidad.
- Permitir a los desarrolladores de videojuegos verificados introducir sus juegos y logros desarrollados.

Los objetivos se miden en función de la calidad de los algoritmos o elementos propios de cada funcionalidad implementada. Para comprobar esta calidad se debe tener en cuenta la satisfacción del usuario, ya sea mediante pequeñas encuestas o mediante la monitorización

del uso de las diversas funcionalidades, así como el testing realizado durante el desarrollo del proyecto.

Además de los objetivos de producto anteriormente descritos, se presenta un objetivo personal para el proyecto:

- Tener una aplicación completa que permita demostrar los conocimientos adquiridos durante el transcurso de la carrera y demostrar la capacidad de trabajar en un entorno full stack profesional.

Este objetivo también es medible dado los siguientes parámetros:

- Testing correcto de las funcionalidades desarrolladas.
- Adaptabilidad y escalabilidad de la aplicación. Para ello se tiene en cuenta el número de clases del proyecto; la arquitectura de software utilizada y sus patrones;

4. Metodología

La metodología de trabajo que se da en el proyecto sigue los estándares Agile y unos principios de diseño de carácter SOLID, esto permite un fácil mantenimiento y escalabilidad de la aplicación.

Los principios que caracterizan este tipo de metodología son:

- Las personas sobre los procesos y herramientas
- Un software funcional sobre una documentación exhaustiva
- Una participación activa del cliente durante el proceso de desarrollo
- La capacidad de pivotar en caso de cambios e imprevistos

La metodología Agile a utilizar se basa en el proceso por tablero y tarjetas Kanban, en estos constan los diversos hitos que se llevan a cabo en cada etapa del proyecto y su recorrido en el tablero hasta que se vean finalizados. A la vez cuenta con un sistema de puntuación para cada hito, para no sobrecargar de trabajo a los desarrolladores y solventar los posibles cuellos de botella que se produzcan durante el proceso de desarrollo.

A continuación, se muestra un ejemplo visual de la organización del tablero en la etapa del “Anteproyecto”, para ello se utiliza la herramienta integrada de Github “Boards”.

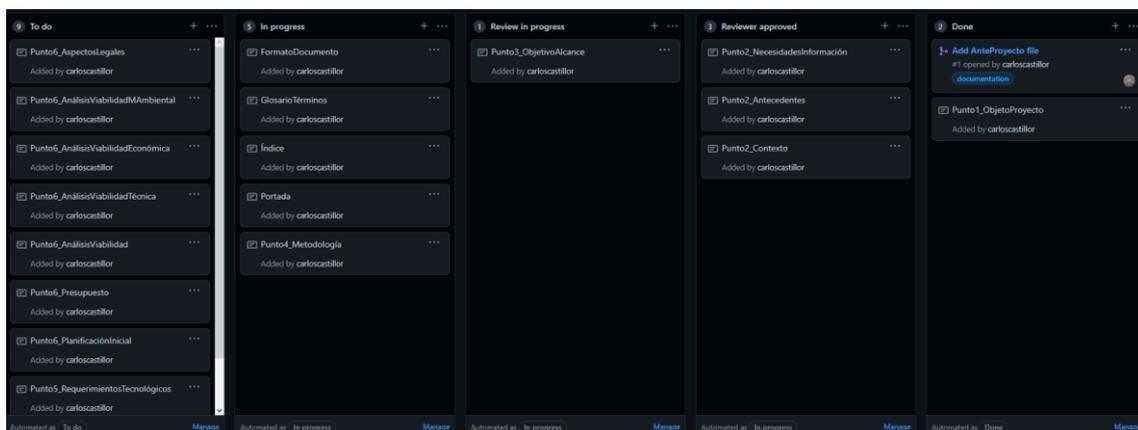


Ilustración 5. Ejemplo Boards. Fuente: propia

El proceso empieza creando los hitos con toda la información necesaria y asignándoles los siguientes parámetros:

- Una puntuación referente a la dificultad o tiempo requeridos por los desarrolladores para llevarlos a cabo.
- El encargado de realizar el hito
- El tipo de tarea a realizar

Una vez creados los hitos correspondientes y colocados en la columna “Por hacer”, estos recorren el tablero de izquierda a derecha teniendo en cuenta que cumplan los requisitos necesarios para acceder a la siguiente columna. Los requisitos de cada columna se enumeran a continuación:

1. Por hacer: En esta columna se establecen los hitos por primera vez y se les asignan los parámetros descritos anteriormente. Una vez seleccionados los hitos a desarrollar se mueven a la siguiente columna.
2. En progreso: En esta columna están presentes los hitos que se están desarrollando en el momento actual. Una vez se hayan implementado se mueven a la siguiente columna.
3. Revisión en progreso: En esta columna están presentes los hitos pendientes de testeo y aprobación posterior. Una vez se hayan testado y verificado su correcta implementación pueden moverse a la siguiente columna.
4. Revisión aprobada: En esta columna se encuentran los hitos pendientes de ser integrados en la rama principal donde se encuentra el proyecto funcional y más actualizado posible. Una vez unida la implementación a la rama principal y verificado que funcione correctamente se pueden mover los hitos a la última columna.
5. Acabado: En esta columna se encuentran los hitos dados por finalizados y correctamente verificado su funcionamiento.

Cabe destacar que el proceso de desarrollo se divide en diversas etapas. En cada una de ellas se desarrollan diversos hitos, los cuales siguen una secuencia lógica. Estas etapas son las siguientes:

- 1ª Etapa: Hace referencia a la entrega del anteproyecto. Empieza el día 5 de noviembre y acaba el día 11 de febrero. Se centra en la búsqueda y recolecta de información para preparar el comienzo del proceso de desarrollo, así como la generación de la documentación necesaria para ello.
- 2ª Etapa: Hace referencia a la entrega de la memoria intermedia del proyecto. Empieza el 11 de febrero y acaba el día 22 de abril. Se centra en, principalmente, la realización e implementación del back-end de la aplicación; una primera aproximación del front-end; y la integración de algunos servicios con el back-end.
- 3ª Etapa: Hace referencia a la entrega de la memoria final del proyecto. Empieza el 22 de abril y acaba el 15 de junio. Se focaliza en la implementación total del front-end; la integración de los servicios que no hayan sido integrados en la 2ª etapa; y la integración de las API de otras plataformas.
- 4ª Etapa: Hace referencia a la defensa del proyecto. Empieza el 15 de junio y acaba del 4 al 15 de julio. En esta etapa se elabora el guion y se ensaya la defensa del producto final frente al jurado.

Todo el proceso viene acompañado de reuniones periódicas para analizar el transcurso del desarrollo en un periodo de tiempo concreto. En estas reuniones se comprueban el correcto funcionamiento de las distintas implementaciones, la necesidad de descartar o pivotar una tarea y finalmente se extraen conclusiones.

Las reuniones tienen un acta asociada en la cual constan los siguientes puntos:

- Objetivos de la reunión
- Temas tratados
- Conclusiones
- Tareas o elementos necesarios para la próxima reunión

5. Desarrollo

5.1 Análisis y modelado de los datos

En la siguiente imagen se pueden observar las distintas clases del dominio de la aplicación, sus relaciones y los atributos propios de cada clase.

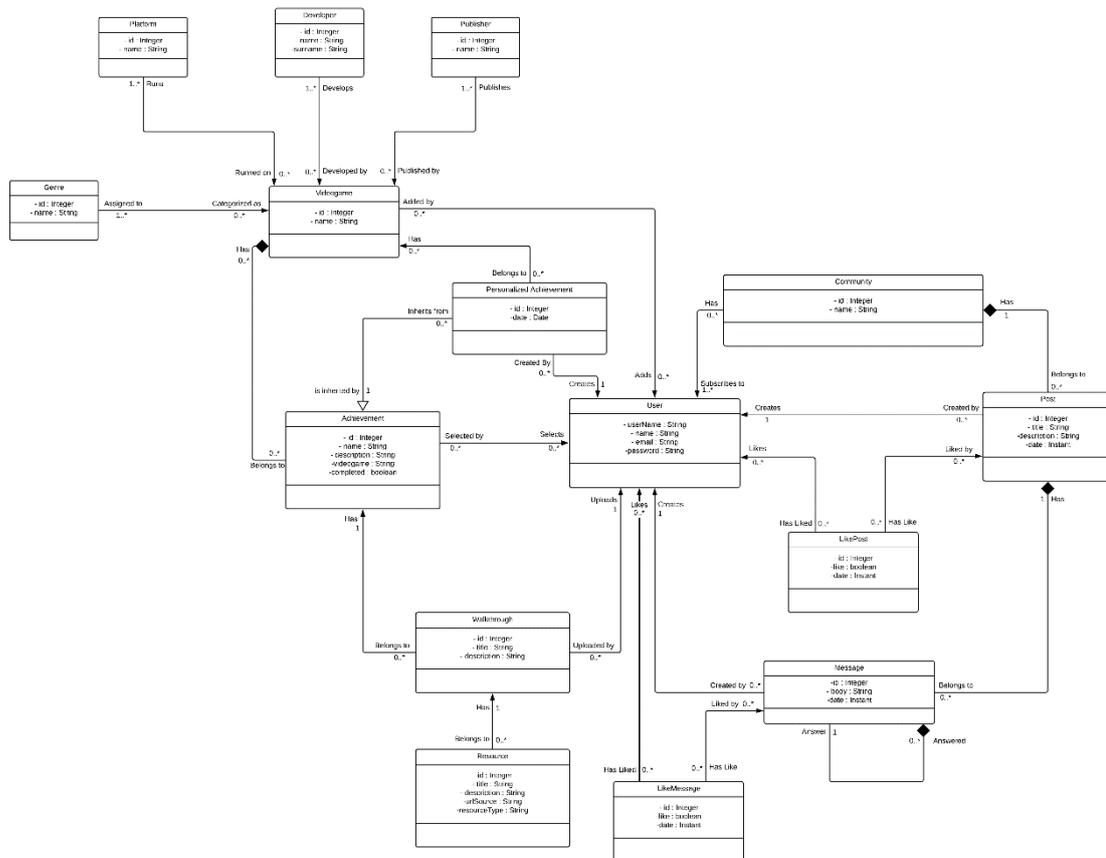


Ilustración 6. Diagrama UML. Fuente: propia

En este diagrama de clases UML se pueden observar las clases principales y las relaciones que hay entre estas. Cada clase representa un objeto o entidad del dominio y se pueden apreciar sus atributos propios.

Por otro lado, se detalla la cardinalidad y el tipo de relación entre estas clases, ya sean herencias o composiciones.

5.2 Requerimientos Funcionales

Este tipo de requerimientos son aquellos que expresan la esencia del software del producto. Estos dictaminan como se interactúa con el entorno y que información ha de gestionar.

- Permitir el registro e inicio de sesión desde la interfaz de inicio de la aplicación. Esta información se almacena en una base de datos y se encripta por motivos de seguridad.
- Implementar una navegación fluida y lógica entre las pestañas y secciones de la aplicación. Esta navegación ha de ser lo más cómoda posible para el usuario
- Implementar una pestaña con el listado personalizado donde consten los videojuegos escogidos por el usuario.
- Implementar una pestaña de creación y gestión de logros personalizados por los usuarios.
- Implementar una herramienta para la ayuda en la realización de los diversos logros.
- Implementar un listado completo de la mayoría de los videojuegos existentes con los logros de cada uno ya establecidos
- Implementar una sección para el perfil de cada usuario donde conste la información relativa a este.
- Implementar una sección de comunidad dividida por juegos
- Implementar una sección de chat en las pestañas requeridas.
- Implementar una sección de logros mensuales en la cual se planteen distintos retos a cumplir en los diez videojuegos más populares del mes anterior.
- Guardar los datos e información modificados durante una sesión del usuario en la base de datos.
- Permitir modificar la información del perfil de un usuario.
- Permitir a los usuarios vincular las cuentas de otras plataformas y se vea reflejada su información en la aplicación.

5.3 Requerimientos Tecnológicos

Este tipo de requerimientos representan las necesidades relativas a TI, software y servicios a las cuales el proyecto se ve sometido. Estos dictaminan como y con que se ha de realizar el desarrollo del proyecto.

- Utilizar React para la implementación del Front-End de la aplicación.
- Utilizar Spring Framework y el lenguaje de programación Java para la realización del Back-End de la aplicación.
- Utilizar Heroku como servicio cloud de la aplicación.
- Utilizar Amazon RDS para Oracle y el lenguaje SQL para el almacenamiento de datos e información en la base de datos.
- Utilizar un sistema OAuth2, que funcione utilizando Json Web Tokens, para asegurar una autenticación verídica de los usuarios.
- Generar una APK del producto para la evaluación del correcto funcionamiento final.
- Utilizar el servicio S3 de AWS en caso de ser necesario espacio extra de almacenaje.
- Utilizar el servicio Cloudwatch de AWS para monitorizar y representar visualmente el uso de la aplicación por parte de los usuarios.
- Utilizar IntelliJ para el desarrollo de la aplicación.
- Utilizar GitHub como repositorio del código de la aplicación y control de versiones.

5.4 Ingeniería de software

La arquitectura de software es de tipo “capas” divididas en capa de persistencia, dominio, presentación y base de datos. Se incluye la base de datos como una capa adicional dado que la arquitectura se basa en la de Spring Boot.

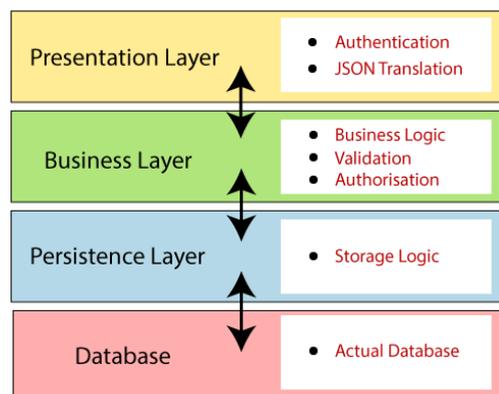


Ilustración 7. Spring Boot Architecture. Fuente: [12]

Este tipo de arquitectura tiene las siguientes ventajas:

- Separar los servicios en alto y bajo nivel.
- Reducir el acoplamiento y las dependencias entre entidades.
- Mejora la cohesión.
- Incrementa la reutilización y la claridad del código.
- Propicia una mejor adaptabilidad y escalabilidad.
- Permite encapsular y descomponer la complejidad.

El flow que siguen los datos e información del software se basa en la arquitectura que se puede observar a continuación.

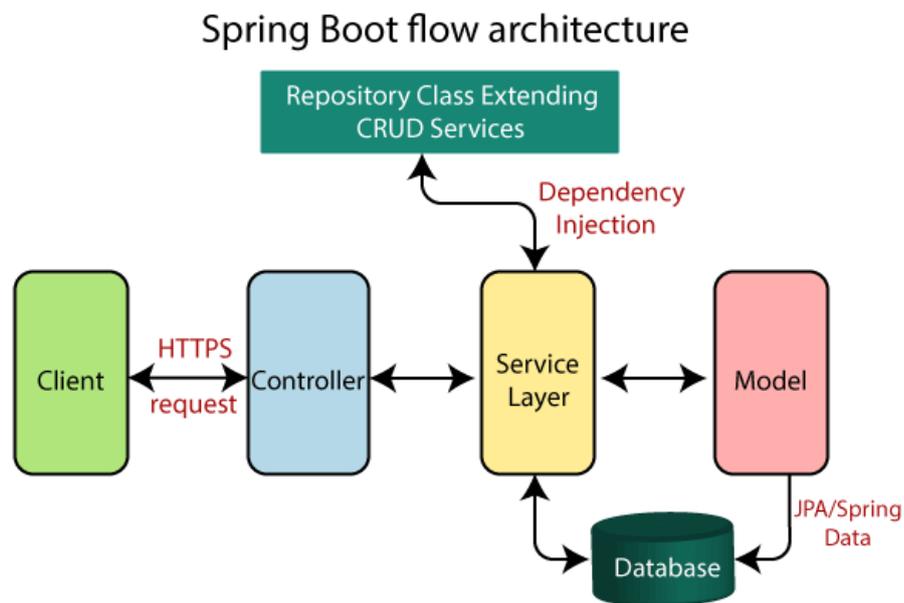


Ilustración 8. Spring Boot Flow Architecture. Fuente: [12]

El cliente lleva a cabo una petición HTTP, ya sea GET o POST, la cual es gestionada por el controlador. En caso de ser necesario, esta llama a los servicios lógicos necesarios, para que la capa de aplicación o negocio ejecute los procesos determinados.

A esta estructura se le ha de sumar la integración con dos API, la de Steam y la de IGDB.

API de IGDB

Esta API permite obtener información y datos de una gran variedad de videojuegos. Su base de datos cuenta con más de 235.000 videojuegos y se mantiene en constante actualización y mantenimiento, por lo que resulta una fuente de datos imprescindible para el proyecto.

Las consultas a esta API están bastante simplificadas, pues la empresa desarrolladora ha creado un wrapper para realizarlas. Este wrapper implementa funciones que permiten la búsqueda de datos específicos de los distintos videojuegos o el filtrado de estos según diversos parámetros. Esto es muy conveniente para el proyecto, ya que se busca que la capa de dominio cuente con objetos como el propio género de los videojuegos, y este tipo de mecanismos agilizan y optimizan mucho el proceso de obtención de datos.

Además, el uso de la API es totalmente gratis para proyectos sin ánimo de lucro, lo que resulta especialmente beneficiario para este proyecto.

API de Steam

Esta API permite consultar y obtener información sobre los datos del catálogo completo de Steam. Esto permite consultar y recoger los distintos logros de los videojuegos que cuenten con este tipo de datos, ya que la existencia de logros en cada uno depende del desarrollador del videojuego.

A partir de estos datos, se pueden asociar con la información ya existente de cada videojuego en la base de datos.

Base de datos Oracle

Se ha optado por una base de datos OracleXE, ya que es una edición gratuita y de fácil instalación, lo cual desde el punto de vista económico y de accesibilidad es muy beneficiario para el proyecto.

Además, dado el gran volumen de datos a almacenar en esta base de datos, OracleXE es idónea, ya que su capacidad para manejar cargas de trabajo de alta demanda, así como su robustez y estabilidad, son aspectos cruciales para garantizar esta optimización en un entorno local.

Otro factor relevante para su elección es la posibilidad de escalabilidad y compatibilidad con otras bases de datos Oracle, ya que permite la migración a otras ediciones en caso de ser necesario. Esto se ha tenido en cuenta, ya que el hecho de tenerla en el cloud de Amazon Web Services, o también llamado Amazon RDS, cuesta una gran suma de dinero, pero en caso de disponer de este se podría trasladar la base de datos sin dificultades.

5.5 Proceso de desarrollo

El proyecto ha usado control de versiones mediante GitHub durante el proceso de desarrollo, el siguiente enlace es la organización de Github donde están, por separado, el proyecto referente al fron-end y el del back-end: [<https://github.com/tfgInfo>].

El proceso de desarrollo empieza con la programación básica de la aplicación. En esta primera etapa se desarrollan los siguientes puntos:

- **Capa Dominio:** Se definen y programan las clases y entidades que representan los objetos del dominio de la aplicación. En cada una de las clases se estipulan las anotaciones de “@Entity” y “@Table”, las cuales permiten al sistema JPA reconocer dichas clases y crearlas en la base de datos correspondiente. Además, se aplican las anotaciones pertinentes en los atributos de cada clase, los cuales permiten la generación automática de columnas en la base de datos. En casos específicos, ha sido necesario establecer relaciones de unión entre columnas o tablas.
- **Capa Persistencia:** Se definen y programan los repositorios que tienen como función la comunicación con la base de datos. Estos repositorios extienden de la interfaz JPAREpository, lo cual implica que la lógica relacionada con las consultas a la base de datos se gestiona mediante dicho sistema de persistencia. Si bien la librería proporciona un conjunto de consultas por defecto, en ocasiones se requiere de la definición de consultas adicionales en repositorios específicos, adaptadas a los requerimientos particulares del proyecto.
- **Capa Controlador:** Los controladores REST constituyen el componente encargado de interactuar con la capa de presentación o front-end de la aplicación. Cada clase del dominio tiene un controlador propio, el cual se encarga de comunicarse con las

clases de servicio o repositorios para las peticiones Post, Get u otras operaciones que sean requeridas por el front-end.

Una vez realizada la primera etapa de programación básica de la aplicación, se empieza la segunda. Esta se caracteriza por implementar los distintos servicios enfocados a la integración con las distintas API y base de datos que utiliza la aplicación.

- **Servicio Api IGDB:** La Api de IGDB se ha implementado mediante un servicio que integra el wrapper proporcionado por la empresa. Este wrapper permite tanto recoger los juegos de su base de datos, como filtrar información específica de los mismos. Para la obtención de estos datos, primero se han obtenido los datos referentes a los objetos que representan los datos de un videojuego, como el género o los desarrolladores, y una vez obtenidos e introducidos en la base de datos se han utilizado estos y la consulta para obtener los videojuegos para crear e introducir el objeto videojuego en la base de datos.
- **Servicio Api Steam:** La implementación de la Api de Steam ha sido más complicada. Ya que para obtener los datos de los logros que proporciona la Api se han de realizar diversas consultas. Por un lado, primero se ha de recoger la totalidad de los “appIds”, lo que representan el ID de los distintos objetos, de la tienda de Steam. Una vez recogidos, y con tal de optimizar la última consulta, se ha realizado una consulta específica para recoger los datos primarios de cada objeto, donde se nos proporciona el tipo de este. Este tipo permite definir si el objeto recogido es un juego, un DLC, una banda sonora, entre otros; con el objetivo de filtrar los que son juegos completos y así ahorrar consultas finales. Esta consulta final recoge por cada appId filtrada como juego completo, los datos de este, concretamente la información relativa a los logros de cada videojuego en caso de contar con ellos.

Cabe destacar que la consulta final relativa a los datos de los logros se ha realizado para los primeros cincuenta mil juegos del catálogo, ya que, pese a haber sido optimizado, el tiempo relativo a la obtención de los datos sigue siendo muy elevado, ya que la API de Steam solo permite doscientas consultas cada cinco minutos y para completar la totalidad de los juegos se requiere de más de cinco días sin parar de

hacer consultas. Aparte de estas doscientas consultas cada cinco minutos, la API de Steam tiene un límite cada diez mil consultas, lo que corta el flujo de operaciones de la aplicación. Para sortear este problema se ha implementado un sistema caché el cual almacena los videojuegos consultados para que en la siguiente consulta al recargar la URL que activa el controlador no se vuelvan a recoger los que ya estén añadidos.

Se han evaluado otras opciones para realizar esta obtención de datos como las plataformas SteamDB o AchievementStats.com las cuales también cuentan con API para recoger estos datos. El inconveniente, y el porqué se han descartado estas opciones, es que o bien requieren de una inversión monetaria o se necesita realizar consultas a la API de Steam igualmente para sincronizar los datos obtenidos de dichas plataformas.

- **Servicio de Oauth2 Google:** Para la implementación del sistema Oauth2 de Google se ha creado la App en la consola de Google, lo que ha permitido obtener el ID y secreto de cliente para poder iniciar sesión con el servicio en la aplicación, los cuales se han definido en las propiedades del proyecto para que funcione correctamente. Además, se han definido las rutas de origen y de redirección para que Google las tenga como enlaces válidos.

Una vez realizada la sincronización con Google, se ha definido una clase seguridad mediante la anotación “@Configuration” la cual implementa los métodos proporcionados por la librería Oauth2 que proporciona Spring Boot, y permite acceder al inicio de sesión de Google. Una vez el usuario escoge con que correo electrónico inicia sesión, se utiliza JWT para crear y almacenar el token de la sesión del usuario, el cual es utilizado en gran parte de la aplicación para recoger los datos almacenados referentes al usuario. Cabe destacar que, si el usuario no existe en la base de datos en el momento de iniciar sesión, se crea un nuevo usuario automáticamente, pero si ya existe, simplemente lo comprueba y lo marca como sesión iniciada.

La tercera etapa es la referente al desarrollo de la capa de presentación y la lógica del conjunto de funcionalidades específicas en el back-end. Para definir esta etapa se proponen los siguientes puntos, los cuales hacen referencia a cada funcionalidad.

- **Navegación:** Para navegar por la aplicación se ha creado un componente barra de navegación donde constan los distintos apartados de esta, así como un desplegable para el inicio de sesión y el perfil del usuario.
- **Inicio y cierre de sesión:** La funcionalidad de iniciar o cerrar sesión se ha programado en el componente principal de la aplicación y se activa en la barra de navegación de la aplicación, concretamente en el desplegable de opciones. Posteriormente a su activación, este redirige al usuario a la ruta del servidor back-end que se encarga de la gestión de este sistema. Si el usuario no ha iniciado sesión en la app, este no puede interactuar con ella, sino que aparece una alerta en pantalla que especifica que debe iniciar sesión. Una vez la sesión haya sido iniciada, en el mismo desplegable se muestra la opción de cierre de sesión donde estaba la de inicio. Cuando el usuario inicia sesión, el front end recoge el input y manda una consulta al back-end, el cual se encarga de generar el token y marcar al usuario como que ha iniciado sesión, en caso de que ya esté en la base de datos, sino lo almacena. Una vez la sesión haya sido iniciada, el usuario ya puede interactuar con la aplicación y se cargan los datos respectivos a este en las secciones específicas.
- **Listado de videojuegos:** La funcionalidad de listado de videojuegos muestra en la ventana de la aplicación un listado paginado y ordenado por ID de los videojuegos que hay almacenados en la base de datos. Se ha elaborado un paginado de 20 unidades por página para optimizar la carga de los videojuegos en pantalla, ya que el volumen de datos a mostrar es muy amplio. Debido a este gran volumen de datos, a su vez se ha elaborado un buscador de texto para agilizar el seleccionado de videojuegos por nombre. Además, la funcionalidad cuenta con un filtrado por género para que el usuario pueda escoger, en caso de que le resulte más cómodo, los videojuegos con el rasgo escogido, además esta funcionalidad ayuda al usuario a poder descubrir juegos nuevos del género deseado. Al clicar un objeto videojuego del listado se redirige al usuario a una ventana donde consta toda la información del videojuego escogido. En esta ventana se pueden observar la siguiente información:

- Botón de añadir o eliminar: El botón implementa la funcionalidad para añadir o eliminar de la lista de videojuegos añadidos del usuario. Esto almacena en la base de datos la relación entre el usuario y el videojuego, para que cuando se elabore la petición en el back-end pueda cargar los diversos videojuegos en su sección de lista, la cual se describe más adelante.
- Datos generales de cada videojuego:
 - Imagen
 - Nombre
 - Género/s
 - Desarrollador/es
 - Editorial
 - Plataforma/s
- Logros: El listado de logros de cada videojuego extraído de la API de Steam, por lo tanto, los juegos que no pertenezcan o estén disponibles en la tienda no cuentan con logros propios. Estos logros pueden marcarse o desmarcarse, lo que permite al usuario tener un seguimiento de los logros completados para cada videojuego.
- Logros personalizados: El listado de logros personalizados de cada videojuego. Este listado es realizado por los usuarios de la aplicación al crear un logro nuevo. Además, estos logros pueden marcarse o desmarcarse, lo que permite al usuario tener un seguimiento de los logros personalizados completados para cada videojuego.
- Ranking: El ranking de los 10 logros personalizados con más me gustas del videojuego en cuestión. Esta funcionalidad se explica más adelante.
- Creación de logros: El botón para crear logros personalizados permite al usuario crear un logro para el videojuego escogido. Se detalla a continuación.
- **Creación de logros:** Al dar clic en el botón para crear logro, el usuario es conducido a una URL donde se muestra en pantalla un formulario que consta de los parámetros nombre y descripción, donde el usuario ha de especificar ambos parámetros para poder crear dicho logro. La introducción de estos parámetros tiene ciertas restricciones para que sean válidos, como un mínimo de caracteres. Al crear el logro, este se almacena en la base de datos relacionándolo con el usuario en cuestión. Esta relación se muestra en el logro al mostrarse en la ventana del propio videojuego, donde se muestra, aparte del nombre y descripción, qué usuario lo ha creado.

- **Ranking:** Esta funcionalidad muestra el top 10 de los logros creados para un videojuego concreto con más me gustas. Estos me gustas son atribuidos por los usuarios de la aplicación mediante el botón que aparece en cada uno de los logros creados. Al dar clic, el me gusta se relaciona tanto con el usuario como con el logro y sube en uno su valor, lo que repercute en el ranking mostrado. Además, este me gusta puede ser eliminado en caso de que el usuario así lo desee.
- **Lista del usuario:** Esta funcionalidad se encarga de mostrar en pantalla, una vez el usuario haya accedido a dicha sección, el conjunto de videojuegos añadidos mediante el botón de añadir. Esta funcionalidad llama al controlador del back-end encargado de la gestión de videojuegos y recoge de la base de datos aquellos videojuegos que hayan sido asociados con el usuario para así cargarlos en pantalla. En este caso, y a diferencia del listado completo de videojuegos, los videojuegos se muestran con su imagen propia, en caso de que exista en la API de IGDB.
- **Perfil del usuario:** El perfil del usuario permite a este ver sus estadísticas, en ellas se observan los siguientes datos:
 - Nombre
 - Número de juegos en la lista
 - Número de me gustas totales de sus logros creados
 - Número de logros completados

Además, aparte de estos datos se muestran, seguido de estos, un número de desplegables relativo a los juegos donde el usuario ha creado logros, y al desplegarse muestra dichos logros junto a un botón para modificarlos. Esta modificación permite al usuario cambiar los datos almacenados para el logro seleccionado y, una vez terminado, permite guardarlos en la base de datos o cancelar la operación.

- **Últimos juegos y logro:** En la ventana de inicio, si el usuario ha iniciado sesión correctamente, se muestran los últimos videojuegos que ha añadido a su lista y los últimos logros que haya completado. En el caso de que no haya añadido ningún videojuego o no haya completado ningún logro, aparece una alerta que indica que debe hacer dichas acciones para que se muestren en ambas secciones la información pertinente.

5.6 Proceso de Testing

El proceso de testing propuesto tiene como objetivo brindar una experiencia fluida y satisfactoria para los usuarios. Al seguir una metodología Agile, se han propuesto distintas pruebas a lo largo del proceso de desarrollo con el objetivo de comprobar el correcto funcionamiento de las distintas funcionalidades. Estas pruebas se realizan al dar por finalizada una funcionalidad a los usuarios que representan la muestra ideal.

Esta muestra ideal es representada por un grupo sesgado de la totalidad de videojugadores del mundo. Este sesgo se ha producido teniendo en cuenta los siguientes parámetros:

- **Tipología de videojugador:** La tipología de videojugador que representa la muestra ideal es la de “achiever”. Esto es debido a que, como se ha detallado en el punto de objetivos, el público principal de la aplicación es de esta tipología, por lo que las pruebas deben de hacerse con usuarios de esta.
- **Edad:** Pese a que la edad no es un limitante como tal, si que es cierto que el público objetivo de la aplicación se concentra en un rango de edades de entre 15 a 30 años. Por ello los usuarios que realicen las pruebas debe encontrarse en este rango de edades.

Teniendo en cuenta el contexto de la aplicación y la muestra ideal propuesta, se han definido las siguientes pruebas, las cuales permiten abarcar los diferentes aspectos y funcionalidades de la aplicación.

- **Prueba de inicio de sesión:** Se ha evaluado si el usuario es capaz de iniciar y cerrar sesión correctamente a través del sistema Oauth2 de Google, así como la asignación de un token JWT al hacerlo. Esta prueba consta de los siguientes pasos:
 - Verificar que el usuario no puede interactuar con la aplicación si no ha iniciado sesión.
 - Dar clic al desplegable de opciones.
 - Dar clic a la opción de iniciar sesión.
 - Proseguir con el inicio de sesión de Google.
 - Escoger el correo electrónico con el que se desea iniciar sesión.

- Verificar que si el usuario no está almacenado en la base de datos se almacene o si ya lo está que lo marque como sesión iniciada.
- Verificar que el usuario recibe un token de sesión.
- Verificar que el usuario es redirigido al Inicio de la aplicación.
- Dar clic al desplegable de opciones.
- Dar clic a la opción de cerrar sesión.
- Verificar que el usuario ha sido marcado como sesión cerrada.
- Verificar que el token de acceso se elimina.
- Verificar que el usuario es redirigido a la pantalla de inicio.

Resultado esperado: El usuario no puede interactuar con los elementos de la aplicación antes de iniciar sesión. Una vez la inicie con el correo electrónico deseado sin problemas, si ya está en la base de datos, lo marca como sesión iniciada y si no lo está que cree un usuario nuevo. El usuario recibe un token de sesión para tener constancia de este en el contexto de la aplicación y es redirigido a la pantalla de inicio. El proceso de cerrar sesión se espera lo mismo, pero al revés.

Resultado real: El usuario no puede interactuar con los elementos de la aplicación sin haber iniciado sesión. Encontrar el botón de iniciar sesión no ha sido tan rápido como lo esperado, pero al interactuar con este el proceso ha sido como el esperado.

- **Prueba de navegación:** Se ha comprobado que el usuario pueda navegar por la aplicación satisfactoriamente, que se muestren correctamente los componentes respectivos a lo que el usuario ha clicado y la información de estos. Esta prueba consta de los siguientes pasos:
 - Dar clic a la sección de la base de datos de videojuegos.
 - Verificar que se redirige al usuario correctamente y se muestra el listado de videojuegos.
 - Dar clic a la sección de la lista del usuario.
 - Verificar que se redirige al usuario correctamente y se muestra, en caso de que haya, los videojuegos añadidos.
 - Dar clic a la sección de estadísticas.
 - Verificar que se redirige al usuario correctamente y se muestran los datos de su perfil.

Resultado esperado: El usuario es redirigido correctamente a todas las URL que haya seleccionado y se muestran los datos concretos de cada una de estas.

Resultado real: El usuario ha sido redirigido correctamente a todas las URL que ha seleccionado y se han mostrado los datos concretos de cada una de estas.

- **Prueba de creación de logros:** Se ha comprobado que el usuario crear logros personalizados para un videojuego en concreto y se almacene correctamente. Esta prueba consta de los siguientes pasos:
 - Dar clic al videojuego donde se pretende crear un nuevo logro.
 - Dar clic al botón de crear logro.
 - Verificar que el usuario es redirigido al formulario de creación de logros.
 - Introducir el nombre y la descripción del logro.
 - Verificar que si el nombre o descripción no cumplen los requisitos no permite continuar.
 - Dar clic al botón de guardar.
 - Verificar que el logro ha sido almacenado en la base de datos.
 - Verificar que el usuario es redirigido al videojuego al cual ha creado el logro.
 - Verificar que el nuevo logro está presente en el apartado de logros del videojuego.

Resultado esperado: El usuario ha podido crear un logro personalizado para el videojuego escogido. El formulario no le ha permitido guardar el logro en caso de no cumplir los requisitos establecidos. Al finalizar, el usuario puede ver el logro que ha creado en el apartado de logros del videojuego en cuestión.

Resultado real: El usuario ha podido encontrar el botón fácilmente y ha accedido al formulario sin problemas. En el formulario ha establecido los campos de nombre y descripción a su parecer, aunque le ha parecido que debe de haber unos requisitos más estrictos. Al finalizar el logro se puede observar en el apartado de logros del videojuego, pero al usuario le ha sido complicado percibir que el apartado en cuestión es un desplegable.

- **Prueba de marcar logros:** Se ha comprobado que el usuario pueda gestionar los logros correctamente, mediante la checkbox presente en cada uno de estos. Esta prueba consta de los siguientes pasos:
 - Dar clic al videojuego donde se quiera marcar el logro como completado.
 - Dar clic al apartado de logros y verificar que se despliega.
 - Verificar que se muestren los logros del videojuego en cuestión, en caso de que el videojuego cuente con estos, sino que se muestre el texto de “No hay logros disponibles”.
 - Dar clic al logro que se quiere marcar como completado.
 - Verificar que el clic marca la checkbox y se almacena la relación entre el usuario y el logro en la base de datos.

Resultado esperado: El usuario puede marcar el logro como completado al desplegar el apartado de logros de un videojuego. Al hacerlo, la relación entre el usuario y el logro queda almacenada en la base de datos para futuras operaciones.

Resultado real: Al usuario le ha sido complicado percibir que el apartado de logros es un desplegable, pero al darle clic se han podido observar los logros concretos de ese videojuego. Al darle clic a uno de ellos lo ha establecido como completado y ha almacenado la relación entre el usuario y el logro en la base de datos.

- **Prueba de lista del usuario:** Se ha comprobado que el usuario pueda añadir o eliminar correctamente videojuegos a su lista personal y que estos, al navegar a dicha lista, se muestren correctamente. Esta prueba consta de los siguientes pasos:
 - Dar clic al videojuego que se pretende añadir.
 - Dar clic al botón de añadir a la lista.
 - Verificar que se almacena en la base de datos la relación entre el videojuego y el usuario.
 - Verificar que el botón de añadir a la lista cambia por eliminar de la lista.
 - Dar clic a la sección de mi lista.
 - Verificar que el videojuego añadido se muestre en pantalla mediante su carátula, en caso de haber disponible, y nombre.
 - Dar clic al videojuego en cuestión.

- Verificar que el usuario es redirigido a los detalles del videojuego.
- Dar clic al botón de eliminar de la lista.
- Verificar que se elimina de la base de datos la relación entre el videojuego y el usuario.
- Verificar que el botón de eliminar de la lista cambia por añadir a la lista.

Resultado esperado: El usuario puede clicar al botón de añadir correctamente y esto hace que la relación entre el usuario y el videojuego se almacene en la base de datos. Al dar clic al botón de mi lista, en pantalla se pueden observar los videojuegos que tiene el usuario en su lista, entre los cuales está el que acaba de añadir. El usuario accede al videojuego en cuestión otra vez, pero el botón ha cambiado de añadir a eliminar, esto provoca que al volverse a clicar elimine la relación entre ambos de la base de datos.

Resultado real: El usuario ha podido añadir a la lista el videojuego escogido sin problemas, aunque al dar clic en el botón de añadir, como el cambio de botón es instantáneo, le ha dado clic seguidamente a eliminar de la lista directamente. Se ha verificado que ambas acciones han funcionado correctamente, y al volverlo añadir se ha comprobado que en la sección de mi lista se muestra la carátula del nuevo videojuego añadido con su nombre debajo.

- **Prueba de listado global:** Se ha comprobado que el usuario pueda filtrar tanto por nombre como por género los distintos videojuegos del listado, y que este pueda acceder a su información correctamente. Además, dado al gran volumen de datos que utiliza la aplicación, se ha comprobado que el paginado y los filtros aplicados funcionen correctamente y no entorpezcan la experiencia del usuario. Esta prueba consta de los siguientes pasos:
 - Dar clic a la sección de Videojuegos BD.
 - Escoger un Género del desplegable.
 - Verificar que se muestren solo juegos de ese género.
 - Escribir un nombre en el buscador de texto.
 - Verificar que el videojuego aparece, aunque no esté todo el nombre escrito.
 - Verificar que no se puede clicar el botón de página anterior.
 - Dar clic al botón de siguiente página.
 - Verificar que se muestran los 20 videojuegos siguientes.

- Dar clic al botón de página anterior.
- Verificar que se muestra los 20 videojuegos anteriores.

Resultado esperado: El usuario puede filtrar los videojuegos tanto por su género como por su nombre, además el paginado implementado muestra los videojuegos que siguen a continuación o los anteriores, en caso de ser posible, respectivamente.

Resultado real: El usuario ha podido filtrar los videojuegos del listado según el género escogido. A su vez, el paginado ha funcionado correctamente, mostrando los siguientes o anteriores juegos al pulsar el botón respectivo, siempre que los hubiera. Por último, el filtrado por texto ha funcionado, pero el usuario ha tenido ciertos problemas para que se muestre el videojuego deseado, ya que al filtrar el texto comprueba letra a letra y los cambios que se muestran en la lista lo han confundido al principio.

- **Prueba de me gusta:** Se ha comprobado el sistema de me gustas de la aplicación funcione correctamente. Esta funcionalidad permite a los usuarios dar me gustas o quitarlos a los logros personalizados de cada videojuego. Esta prueba consta de los siguientes pasos:
 - Dar clic al videojuego al que se quiera añadir o quitar un me gusta.
 - Abrir el desplegable del apartado de logros.
 - Dar clic al botón de me gusta de uno de los logros personalizados del videojuego.
 - Verificar que la relación entre el me gusta y el usuario se almacena en la base de datos.
 - Verificar que el botón cambia el texto a No me gusta.
 - Verificar que a la propiedad me gustas del logro se le suma uno al valor.
 - Dar clic al botón de no me gusta de uno de los logros personalizados del videojuego.
 - Verificar que la relación entre el me gusta y el usuario se elimina de la base de datos.
 - Verificar que el botón cambia el texto a Me gusta.
 - Verificar que a la propiedad de me gustas del logro se le resta uno al valor.

Resultado esperado: El usuario puede añadir o quitar me gustas a los logros personalizados del videojuego. Esta acción añade o elimina en el almacenamiento de la base de datos la relación entre el me gusta y el usuario; le suma o resta uno al valor de me gustas del logro personalizado, y, por último, cambia el texto del botón respectivamente.

Resultado real: El usuario ha podido dar me gusta y no me gusta a los logros personalizados escogidos, las relaciones entre el usuario; el me gusta se han almacenado o eliminado correctamente de la base de datos; y el botón ha modificado su texto correctamente.

- **Prueba del ranking:** Se ha verificado el correcto funcionamiento del ranking de los logros creados por usuarios, teniendo en cuenta que al añadirse o quitar cada me gusta, las posiciones del ranking varíen, en caso de ser necesario. Esta prueba consta de los siguientes pasos:
 - Dar clic al videojuego del que se quiera observar su ranking.
 - Verificar que el ranking contiene, en caso de tenerlos, los 10 logros con más me gustas en ese videojuego.
 - Dar clic al botón de me gusta de cualquier logro personalizado.
 - Verificar que el ranking se actualiza, en caso de que la modificación lo permita.
 - Dar clic al botón de no me gusta de cualquier logro personalizado.
 - Verificar que el ranking se actualiza, en caso de que la modificación lo permita.

Resultado esperado: El usuario puede ver el ranking correctamente en el videojuego escogido, y al clicar el botón de me gusta o no me gusta de alguno de los logros personalizados de dicho videojuego este se ve actualizado.

Resultado real: El usuario ha podido observar el ranking del videojuego escogido correctamente y al realizar la acción de añadir me gustas o quitarlos se ha modificado y actualizado las posiciones en el ranking.

- **Prueba de perfil:** Se ha verificado que las estadísticas mostradas en la ventana del perfil del usuario sean las correctas y reflejen los datos del usuario adecuadamente a medida que se realicen cambios en el contexto de la aplicación. Además, se ha tenido en cuenta que cuando el usuario entre para ver el perfil de otro usuario al seleccionarlo, se muestre su información correctamente. Esta prueba consta de los siguientes pasos:
 - Dar clic en el desplegable de opciones.
 - Seleccionar el botón de estadísticas.
 - Verificar que el usuario es redirigido a la URL que muestra su perfil.
 - Verificar que los datos mostrados del usuario son los correctos.
 - Dar clic a cualquier videojuego con logros personalizados.
 - Desplegar el apartado de logros.
 - Verificar que se muestre el nombre del usuario que ha creado dichos logros.
 - Dar clic al nombre de un usuario distinto al actual.
 - Verificar que el usuario es redirigido a la URL que muestra el perfil del usuario escogido.
 - Verificar que los datos mostrados son los del usuario escogido.

Resultado esperado: El usuario puede acceder a su perfil al clicar el botón de estadísticas. En este perfil se muestran el número de me gustas totales que tiene el conjunto de sus logros creados, el número de logros creado y el número de videojuegos añadidos a su lista. Seguido a estos datos se puede observar un conjunto de desplegables referentes a cada videojuego donde el usuario ha creado logros; estos desplegables contiene todos los logros creados.

Al clicar el nombre de otro usuario, el usuario actual es redirigido al perfil de estadísticas del que ha sido seleccionado y se muestra toda la información relativa a este.

Resultado real: El usuario ha sido capaz de acceder a su perfil y comprobado que los datos mostrados eran los correctos. Al clicar en el nombre de un usuario distinto, se ha redirigido al usuario actual al perfil del seleccionado y se han podido observar todas sus estadísticas.

- **Prueba de Inicio:** Se ha comprobado el correcto funcionamiento de la pantalla de inicio. En esta se pueden observar los últimos seis videojuegos añadidos a la lista del usuario y los últimos cinco logros superados. Esta prueba consta de los siguientes pasos:
 - Efectuar el inicio de sesión.
 - Verificar que en el apartado de últimos videojuegos añadidos aparezca un texto haciendo referencia a que todavía no se han añadido videojuegos a la lista.
 - Verificar que en el apartado de últimos logros completados aparezca un texto haciendo referencia a que todavía no se han completado logros en ningún videojuego.
 - Añadir seis videojuegos a la lista del usuario.
 - Marcar como completados cinco logros escogidos.
 - Dar clic al nombre de la aplicación en la barra de navegación.
 - Verificar que se redirige al usuario a la pantalla de inicio de la aplicación.
 - Verificar que se muestran los nuevos videojuegos y logros en el apartado correspondiente.
 - Añadir un nuevo videojuego a la lista.
 - Marcar como completado un nuevo logro.
 - Verificar que se muestran el videojuego y logro añadidos recientemente en el apartado correspondiente, y que se han eliminado del apartado los más antiguos.

Resultado esperado: El usuario puede ver un texto que hace referencia a que no se han añadido videojuegos a su lista o no se han completado logros todavía. En caso de añadir o marcar cualquiera de ambos, este texto se sustituye por el videojuego o logro añadidos. En caso de superar el límite de seis videojuegos o cinco logros, el más antiguo de cada apartado se sustituye por el que se acaba de añadir.

Resultado real: El usuario ha visto el texto referente a que no hay objetos a mostrar en ambos apartados. Al añadir los videojuegos y logros, y regresar a la ventana de inicio, estos se han mostrado sin problema. Al añadir los dos objetos nuevos, se han sustituido por los más antiguos que se habían añadido con anterioridad.

- **Prueba modificar logro personalizado:** Se ha comprobado que el usuario pueda modificar los logros personalizados creados por él mismo desde el panel de estadísticas. Esta prueba consta de los siguientes pasos:
 - Dar clic al desplegable del videojuego al cual pertenece el logro personalizado que se quiera modificar.
 - Dar clic al botón de modificar en el logro personalizado seleccionado.
 - Verificar que el usuario es redirigido al formulario de modificación de logros personalizados.
 - Modificar el nombre y descripción ya establecidos.
 - Verificar que si el nombre o descripción no cumplen los requisitos no permite continuar.
 - Dar clic al botón de guardar.
 - Verificar que el logro se ha actualizado en la base de datos.

Resultado esperado: El usuario puede modificar los logros que haya creado en su panel de estadísticas. Al dar clic en el botón de modificar del logro personalizado escogido, se redirige al usuario a la URL de modificación de logro, donde este puede cambiar el nombre y descripción que había establecido con anterioridad. Al dar clic al botón de guardar, se actualiza el logro modificado en la base de datos.

Resultado real: El usuario ha podido acceder al formulario de modificación desde su perfil de estadísticas. Al realizar las modificaciones, ha clicado al botón de guardar y se ha actualizado en la base de datos.

6. Análisis de resultados

Para hablar del análisis de resultados es importante saber qué requisitos se habían planteado al principio del proyecto. Partiendo desde ese contexto, se puede deducir que la implementación de las funcionalidades principales y más representativas de la aplicación se han llevado a cabo satisfactoriamente, aunque se han descartado algunas de estas.

En primer lugar, la funcionalidad que permite a los usuarios iniciar y cerrar sesión en la aplicación mediante el sistema OAuth2 Google, ha sido implementado exitosamente. Además, en la lista de requerimientos se planteaba la posibilidad de permitir el registro de los usuarios, pero se ha obviado esta parte debido a la simpleza y conveniencia de iniciar sesión con la cuenta de Google para el usuario.

A nivel de navegación y usabilidad, la aplicación presenta una navegación entre pestañas lógica y comprensible para los usuarios, lo que facilita la exploración de la aplicación y proporciona una buena experiencia en general.

En cuanto a las funcionalidades principales de la aplicación, las implementadas funcionan correctamente y permiten a los usuarios experimentar de primera mano el objetivo del proyecto, la gestión y seguimiento de los logros de los videojuegos que estos escojan.

Cabe destacar que se han descartado o aplazado algunas de las propuestas de funcionalidades iniciales. Por un lado, las funcionalidades que pretendían crear comunidades y permitir a los usuarios comunicarse entre ellos, se ha descartado completamente, debido a que se ha creído conveniente que la aplicación se centre en un único ámbito y esta relación recíproca entre usuarios se dé mediante la apreciación de sus cuentas, ya sea con los me gustas y el ranking, o la capacidad de poder ver sus perfiles mutuamente.

La funcionalidad de permitir a los usuarios introducir guías para la ayuda en la superación de los logros se ha aplazado, pues se ha creído oportuno debido al gran volumen de datos de la base de datos y el coste representativo en el cloud.

Como último requerimiento funcional descartado se encuentra la implementación del sistema de logros mensuales. La razón para aplazar dicha funcionalidad se basa

principalmente en consideraciones de tiempo y priorización. No solo esto, sino que se ha observado que, pese a la gran capacidad de retención de usuarios que implica, el mantenimiento de esta funcionalidad es muy elevado y, para una sola persona, supone una gran cantidad de tiempo y esfuerzo elaborarlo.

Pese a este descarte, se ha decidido implementar una funcionalidad que no suponga un gasto en recursos humanos tan elevado y también fomente esta retención y es el sistema de me gustas y rankings.

En cuanto a los requerimientos tecnológicos, se ha descartado para la entrega de este proyecto todos los servicios relacionados con la nube. Debido al gran volumen de datos que hay almacenados, el precio total por almacenar estos un solo mes asciende a más de mil euros, lo que no permite que esto se pueda producir. Por ende, no tiene sentido que si la base de datos no se encuentra en Amazon RDS la aplicación esté subida a Heroku, ya que no puede haber comunicación con la base de datos.

Para solventar este hecho se ha decidido realizar su implementación en local mediante una base de datos OracleXE, lo que permite crear una base de datos local que pueda comunicarse con el back-end mediante el datasource establecido y el hibernate de Oracle para que el sistema JPA de persistencia pueda gestionar las consultas correctamente.

Por lo que hace a el modelo de datos, se ha variado la idea principal debido a problemas con el sistema JPA y su anotación `@JoinColumn`. Para solventar este hecho se ha optado por hacer que las relaciones, en vez de ser mediante claves foráneas almacenadas en el objeto principal, se produzcan en tablas unión que almacenen las claves de los IDs de ambos objetos. Por poner un ejemplo, en la propuesta inicial, en la tabla videojuego se almacenaban las claves foráneas de sus datos, como género, desarrolladores, etc. En la nueva propuesta se genera una tabla para almacenar esta relación. El acceso a los datos por parte del back-end no se ha visto muy modificado, ya que la consulta se produce mediante sus IDs y estos siguen presentes y plasman correctamente las relaciones.

Por último, se destaca el uso de las API anteriormente mencionadas. La obtención de datos por parte de ambas API ha sido satisfactoria, aunque la API de Steam requiere de mucho tiempo para proporcionar toda la información respectiva a los logros. Aun así, el resultado es correcto y la recogida de los datos es efectiva.

7. Conclusiones

Empezando por el objetivo y uso principal de la aplicación, se puede observar, mediante el análisis del marco teórico, que una aplicación que permita el seguimiento y gestión de logros en los diversos videojuegos es inexistente, por lo que el formular este producto puede ser de utilidad para la industria o los usuarios que consumen ese tipo de productos, focalizándose en los videojugadores de tipología “achiever”.

Por otro lado, y centrándose más en el desarrollo del proyecto, se pueden extraer distintas conclusiones. En cuanto a la elección de la herramienta Java Spring Boot para la construcción e implementación del back-end de la aplicación en contraparte a otras herramientas como PHP, ha permitido el desarrollo satisfactorio de la aplicación, proporcionado una mejor escalabilidad del producto en caso de que en un futuro se quieran añadir nuevas funcionalidades o mejoras. Además, este framework permite una gestión de la concurrencia más efectiva, lo que se traduce en, si la aplicación acaba siendo un éxito, un mejor control y gestión del gran volumen de usuarios.

La integración con las API y el guardado los datos puros de videojuegos y logros en la base de datos ha sido diferencial. Esto es debido a que, en caso de solo recoger los datos sin almacenarlos, mostrándolos únicamente en pantalla, tiene como desventaja el hecho que si la API en cuestión deja de funcionar la aplicación pierde todo el sentido del mundo, ya que no mostrará los datos más importantes de la aplicación, por lo que haberlo hecho hace que no dependa de estas y la aplicación sea más robusta.

Si bien algunos requerimientos iniciales propuestos no se han implementado en esta primera versión, se ha pivotado de dirección para sustituirlos o determinar el porqué de este desprendimiento. Aunque cabe destacar el gran contratiempo que ha supuesto la imposibilidad de contratar los servicios cloud debido a la gran inversión monetaria que supone, ya que la capa gratis de estos servicios no puede soportar el volumen de datos de mi aplicación.

8. Posibles ampliaciones

En este apartado se proponen distintas mejoras o ampliaciones para el producto desarrollado. Todo ello con el objetivo de impulsar la aplicación, ya sea por calidad o experiencia general del producto. A continuación, se describen estas nuevas implementaciones.

- **Listas personalizadas:** esta funcionalidad permite a los usuarios crear listas separadas según este lo crea conveniente, en vez de una sola lista. Esta funcionalidad permite una mayor customización del seguimiento y gestión de los logros, por lo que la experiencia del usuario se ve mejorada por ello.
- **Funcionalidad de seguir usuarios:** Para mejorar la interacción entre usuarios y suplir aún más la funcionalidad eliminada de las comunidades y foro, esta mejora permite a los usuarios seguirse entre ellos y compartir sus listas o perfil con otros. Esto fomenta la retención de usuarios y la captación de nuevos de ellos.
- **Servicio en cloud:** La mejora más importante es la de llevar la aplicación fuera de un entorno local. Se debe invertir una gran cantidad de dinero para ello, por lo que la probabilidad de que esta ampliación se produzca es baja, aunque se puede investigar si se puede retroalimentar esta necesidad de inversión con un sistema de monetización, aunque esto perjudicaría el uso de API, ya que estas solo prestan sus servicios gratuitamente si el producto no tiene ánimo de lucro.
- **Versión móvil:** La creación de una aplicación móvil basada en esta aplicación es un gran factor a tener en cuenta. La portabilidad y comodidad que ofrece un móvil es mucho mejor que la de un ordenador, por lo que la experiencia de uso del usuario se ve mejorada por ello.
- **Aspecto visual:** El aspecto visual de la aplicación puede mejorar mucho mediante librerías especializadas o customización de los elementos de la interfaz. Esto, nuevamente, requiere de una inversión monetaria, en caso de buscar un cambio muy significativo, por lo que su prioridad es más bien baja.

9. Bibliografía

- [1] Activision, 1979. [En línea] [consulta: 6 de febrero de 2023]. Disponible en https://www.atariage.com/2600/archives/activision_patches.html
- [2] Mikael Jakobsson, *The Achievement Machine: Understanding Xbox 260 Achievements in Gaming Practices*. [En línea] [consulta: 3 de diciembre de 2022] Disponible en <http://gamestudies.org/1101/articles/jakobsson>
- [3] Steam, 2003. [En línea] [consulta: 6 de febrero de 2023]. Disponible en <https://1000marcas.net/steam-logo/>
- [4] Exophase, 2006. [En línea] [consulta: 6 de febrero de 2023]. Disponible en https://www.exophase.com/assets/zeal/logos/exologo_bw.png
- [5] Habitica, 2013. [En línea] [consulta: 6 de febrero de 2023]. Disponible en <https://play-lh.googleusercontent.com/8OCsGkPqZe2EgKsbF8o8ew9rhv0EeL8X77AtIrNctXA0x8Zj65OdRebDwIoqpGDqmw4=s180-rw>
- [6] B. Eisenman, *Learning React Native: Building Native Mobile Apps with JavaScript*, 1a ed., Sebastopol, California, O'Reilly Media Inc., 2016. Disponible a <https://pepa.holla.cz/wp-content/uploads/2016/12/Learning-React-Native.pdf>
- [7] *Guides and API References*, Amazon Web Services Inc., Seattle, Washington, 2021. [En línea] [Consulta 5 de enero de 2023] Disponible en <https://docs.aws.amazon.com>
- [8] Craig Walls: *Spring In Action*. Manning Publications, Nueva York, 2019, 5ta edición. ISBN: 1617294942
- [9] Oracle, 1977. [En línea] [consulta: 5 de enero de 2023] Disponible en <https://docs.oracle.com/en/>

[10] Juho Hamari, Veikko Eranti, *Framework for Designing and Evaluating Game Achievements*. [En línea] [consulta: 12 de diciembre de 2022] Disponible en <http://www.digra.org/wp-content/uploads/digital-library/11307.59151.pdf>

[11] R. Bartle: *Hearts, Clubs, Diamonds, Spades: Players Who suit MUDs*, Colchester, Regne Unit, Muse Ltd ,1996. Disponible en <https://mud.co.uk/richard/hcds.htm>

[12] JavaTpoint, 2009. [En línea] [consulta: 20 de marzo de 2023]. Disponible en <https://www.javatpoint.com/spring-boot-architecture>