

Aplicación de recomendación de videojuegos: Monkeys Together Strong

Alejandro Roldán Soblechero

Grado en Ingeniería Informática

Desarrollo multiplataforma de aplicaciones móviles

Carles Sànchez Rosa

Jordi Almirall López

Carles Garrigues Olivella

06/2024



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Aplicación de recomendación de videojuegos</i>
Nombre del autor:	<i>Alejandro Roldán Soblechero</i>
Nombre del consultor/a:	<i>Carles Sànchez Rosa Jordi Almirall López</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	06/2024
Titulación:	<i>Grado en Ingeniería Informática</i>
Área del Trabajo Final:	<i>Desarrollo multiplataforma de aplicaciones móviles</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Ionic, Vue, FastAPI</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>Este trabajo presenta el desarrollo de una aplicación móvil multiplataforma para la recomendación de videojuegos, diseñada para ofrecer sugerencias ajustadas a las preferencias de los jugadores.</p> <p>El proyecto se realizó utilizando una metodología ágil personalizada que implicó ciclos cortos de trabajo, evaluación continua y adaptaciones a las circunstancias para alcanzar los objetivos del proyecto. Este enfoque permitió mantener flexibilidad y agilidad en el desarrollo, asegurando una respuesta rápida a los cambios y necesidades emergentes.</p> <p>La investigación y diseño de la aplicación se basaron en un estudio de las preferencias y comportamientos de los jugadores, empleando teorías sobre los diferentes tipos de jugadores. Esto ayudó a alinear el desarrollo de la aplicación con las expectativas de los potenciales usuarios.</p> <p>Para la implementación, se utilizó una combinación de tecnologías modernas. Se eligieron Ionic y Vue.js para el desarrollo del frontend, lo que permitió crear una aplicación multiplataforma de manera eficiente, basada en un único código base. Para el backend, se emplearon FastAPI y SQLAlchemy, lo que facilitó un desarrollo rápido y flexible de una API robusta y eficiente.</p> <p>El proyecto concluyó con un producto mínimo viable (MVP), que, aunque más simple de lo inicialmente planeado, cumplió con la mayoría de los objetivos técnicos y funcionales, proporcionando recomendaciones útiles y personalizadas. Quedaron pendientes varias áreas de mejora, tanto en la interfaz como en el sistema de recomendaciones, debido a limitaciones de</p>	

tiempo. Además, se identificaron propuestas futuras para continuar mejorando y expandiendo las capacidades de la aplicación.

Abstract (in English, 250 words or less):

This work presents the development of a cross-platform mobile application for video game recommendations, designed to offer suggestions tailored to the preferences of players.

The project was conducted using a customized agile methodology that involved short work cycles, continuous evaluation, and adaptations to circumstances to achieve the project's objectives. This approach allowed for maintaining flexibility and agility in the development process, ensuring a quick response to emerging changes and needs.

The research and design of the application were based on a study of players' preferences and behaviors, employing theories about several types of players. This helped to align the application development with the expectations of potential users.

For the implementation, a combination of modern technologies was used. Ionic and Vue.js were chosen for frontend development, which allowed for efficient creation of a cross-platform application based on a single codebase. For the backend, FastAPI and SQLAlchemy were used, facilitating the rapid and flexible development of a robust and efficient API.

The project concluded with a minimum viable product (MVP) that, although simpler than initially planned, met most of the technical and functional objectives, providing useful and personalized recommendations. Several areas for improvement were left pending, both in the interface and the recommendation system, due to time constraints. Additionally, future proposals were identified to continue improving and expanding the capabilities of the application.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	1
1.3 Enfoque y método seguido.....	2
1.4 Planificación del Trabajo.....	2
1.5 Breve resumen de productos obtenidos.....	4
1.6 Breve descripción de los otros capítulos de la memoria.....	4
2. Diseño centrado en el usuario.....	6
2.1 Usuarios.....	6
2.1.1 Métodos de indagación.....	6
2.1.1.1 Estudio de Richard Bartle sobre los diferentes tipos de jugadores (1).....	6
2.1.1.2 Entrevistas.....	6
2.1.1.3 Resumen de Resultados y Conclusiones.....	7
Conclusiones.....	8
2.1.2 Perfiles de usuario.....	9
2.2 Diseño conceptual.....	11
2.2.1 Point of view statements.....	11
2.2.2 Flujos de interacción.....	12
2.3 Prototipado.....	14
2.3.1 <i>Sketches</i>	14
2.3.2 Prototipo horizontal.....	16
2.4 Evaluación.....	21
2.4.1 Guion de la sesión de pruebas.....	21
2.4.2 <i>Feedback</i> de las pruebas.....	22
2.4.3 Puntos de mejora.....	22
3. Diseño técnico.....	23
3.1. Definición de los casos de uso.....	23
3.2. Diseño de la arquitectura.....	25
3.2.1. Base de datos.....	25
3.2.2. Diagrama de entidades y clases.....	25
3.2.3. Diagrama de arquitectura.....	26
4. Desarrollo.....	27
4.1. Herramientas y stack tecnológico.....	27
4.1.1. <i>Frameworks</i> y librerías principales.....	27
4.1.2. Lenguajes de programación.....	27
4.1.3. IDE y herramientas de desarrollo.....	28
4.1.4. Base de datos.....	28
4.1.5. Plataforma de despliegue en producción.....	29
5. Pruebas.....	30
5.1 Pruebas manuales.....	30
5.2 Pruebas automáticas.....	30
5.2.1 Tecnologías utilizadas.....	31
5.3 Plan de Pruebas.....	33
6. Conclusiones.....	34

6.1. Logro de objetivos	34
6.2. Metodología y planificación	34
6.3. Trabajo Futuro	34
7. Glosario	35
8. Bibliografía	36
9. Anexos	37
9.1. Configuración del entorno y construcción de la aplicación	37
9.1.1. Estructura de los proyectos.....	37
9.1.2. Configuración de la aplicación móvil.....	37
9.1.3. Configuración del <i>backend</i>	38
9.2. Manual de usuario.....	40

Lista de figuras

Figura 1. Lista de tareas.....	3
Figura 2. Diagrama de Gantt.....	4
Figura 3. Login y Onboarding.....	12
Figura 4. Añadir juego a una lista de usuario.....	12
Figura 5. Modificar preferencias.....	13
Figura 6. Onboarding, preferencias.....	14
Figura 7. Login / Signup.....	14
Figura 8. Búsqueda.....	15
Figura 9. Home.....	15
Figura 10. Información del juego.....	15
Figura 11. Perfil.....	15
Figura 12. Signup.....	16
Figura 13. Login.....	16
Figura 14. Onboarding, get started.....	17
Figura 15. Onboarding, preferencias.....	17
Figura 16. Home.....	18
Figura 17. Perfil.....	18
Figura 18. Categoría o genero.....	19
Figura 19. Búsqueda.....	19
Figura 20. Información del juego.....	20
Figura 21. Información del juego.....	20
Figura 22. Diagrama de base de datos.....	25
Figura 23. Diagrama de entidades del frontend.....	25
Figura 24. Diagrama de entidades del backend.....	26
Figura 25. Diagrama de arquitectura.....	26
Figura 26. Ejecución <i>tests</i> unitarios.....	31
Figura 27. Ejecución <i>test</i> unitarios del <i>backend</i>	32
Figura 28. <i>Backend endpoints</i>	32
Figura 29. Pantalla de onboarding.....	40
Figura 30. Pantalla de login/signup.....	40
Figura 31. Pantalla de detalle de juego.....	41
Figura 32. Pantalla principal.....	41
Figura 33. Pantalla de resultado de búsqueda.....	42
Figura 34. Pantalla de búsqueda.....	42
Figura 35. Pantalla de perfil.....	43
Figura 36. Pantalla de género.....	43

1. Introducción

1.1 Contexto y justificación del Trabajo

La industria de los videojuegos ha crecido hasta convertirse en la mayor industria del entretenimiento, superando a la del cine y la música juntas. Este auge ha resultado en una gran cantidad y variedad de títulos disponibles, que, junto con el coste de éstos, plantea un desafío para los usuarios a la hora de consumir contenido que se alinee con sus gustos y expectativas.

Sin embargo, el panorama actual de las herramientas de recomendación no es el ideal para los aficionados de los videojuegos. Por un lado, predominan las plataformas generalistas de entretenimiento, las cuales no abordan adecuadamente el nicho específico de los videojuegos o incluso lo pasan por alto. Por otro lado, las pocas soluciones especializadas en videojuegos suelen estar influenciadas por los intereses de grandes corporaciones del sector, lo que sesga las sugerencias hacia títulos promocionados, en lugar de reflejar las preferencias de los usuarios.

Por lo tanto, surge una necesidad de una herramienta de recomendación de videojuegos centrada en el usuario e independiente. Con esta solución, no solo se espera mejorar la manera en que los jugadores descubren nuevos títulos, sino también dar visibilidad a una variedad más amplia de videojuegos, incluidos aquellos menos conocidos o independientes.

1.2 Objetivos del Trabajo

El desarrollo de esta aplicación tiene como finalidad abordar la necesidad de los usuarios de obtener sugerencias personalizadas, fiables e independientes en la selección de videojuegos. Los objetivos específicos de este proyecto se detallan a continuación:

- Investigación y análisis de usuario: Identificar las necesidades, preferencias y comportamientos de los jugadores, para asegurar que la aplicación cumpla con las expectativas de los usuarios.
- Diseño centrado en el usuario: Crear una experiencia de usuario eficaz a través de una interfaz de usuario intuitiva y accesible, que facilite la interacción con la aplicación.
- Desarrollo de un sistema de recomendación personalizada: Desarrollar un sistema para proporcionar recomendaciones personalizadas, basadas en las preferencias y el historial de juego de cada usuario.

- Integración con IGDB y otras fuentes: Implementar una integración con IGDB (Internet Game Database) entre otras bases de datos y fuentes de información sobre videojuegos, para construir una base de conocimiento que alimente el sistema de recomendación y asegure la vigencia de las sugerencias.
- Independencia de las recomendaciones: Mantener la objetividad en las sugerencias, ofreciendo recomendaciones basadas únicamente en la relevancia para el usuario.
- Evaluación y mejora continua: Implementar un sistema de *feedback* dentro de la aplicación que permita a los usuarios valorar la relevancia de las recomendaciones, para mejorar continuamente los algoritmos de recomendación y la experiencia del usuario.
- Soporte multiplataforma móvil: Asegurar que la aplicación esté disponible para dispositivos iOS y Android, además de ofrecer una versión web accesible como Progressive Web App (PWA), garantizando una experiencia coherente en todas las plataformas.

1.3 Enfoque y método seguido

Para desarrollar un nuevo producto multiplataforma, utilizaremos un conjunto de tecnologías que nos permitan un desarrollo con un enfoque multiplataforma eficiente:

- Ionic: Framework de desarrollo de aplicaciones móviles que nos permitirá crear una aplicación multiplataforma para iOS, Android y PWA, compartiendo una base de código común para la eficiencia en el desarrollo.
- Vue.js: Framework de JavaScript para la interfaz de usuario, es progresivo y simple, además de tener soporte integrado dentro de Ionic.
- Capacitor: Entorno de ejecución nativa para aplicaciones móviles desarrolladas con Ionic, siendo una alternativa moderna a Cordova.

Para la metodología de trabajo, dado que el proyecto será llevado a cabo por un único desarrollador, se adoptará una metodología ágil personalizada. Esta metodología estará enfocada en mantener la flexibilidad a lo largo del desarrollo, permitiendo iteraciones rápidas y adaptaciones según sea necesario.

La planificación del proyecto se dividirá en iteraciones o *sprints* cortos, permitiendo una evaluación continua del progreso y la capacidad de adaptar el plan de trabajo en función de los resultados y el *feedback* obtenido.

1.4 Planificación del Trabajo

Para la planificación del trabajo del proyecto, se ha estimado una duración total de 15 semanas. Para simplificar los cálculos, se ha decidido dedicar 2.5 horas al día, de lunes a domingo, 17.5h semanales.

Nombre	Duración	Esfuerzo	Inicio	Fin
☐ TFG	105días	262.5horas	28/02/2024	11/06/2024
☐ Plan de Trabajo	14días	35horas	28/02/2024	12/03/2024
Definición de objetivos y alcance	5días	12.5horas	28/02/2024	03/03/2024
Investigación y recolección de recursos	4días	10horas	04/03/2024	07/03/2024
Aplicación demo	2días	5horas	08/03/2024	09/03/2024
Planificación y cronograma	1día	2.5horas	10/03/2024	10/03/2024
Redacción del Plan de Trabajo	2días	5horas	11/03/2024	12/03/2024
Hito: Documento del Plan de Trabajo	0día	0hora	12/03/2024	12/03/2024
☐ Diseño y Arquitectura	28días	70horas	13/03/2024	09/04/2024
Investigación de usuarios	5días	12.5horas	13/03/2024	17/03/2024
Recopilación y análisis de requisitos	5días	12.5horas	18/03/2024	22/03/2024
Diseño conceptual	6días	15horas	23/03/2024	28/03/2024
Prototipado inicial	6días	15horas	29/03/2024	03/04/2024
Diseño técnico	6días	15horas	04/04/2024	09/04/2024
Hito: Prototipo y Diseño técnico	0día	0hora	09/04/2024	09/04/2024
☐ Implementación	42días	105horas	10/04/2024	21/05/2024
Configuración del entorno de desarrollo	3días	7.5horas	10/04/2024	12/04/2024
Implementación de funcionalidades básicas	15días	37.5horas	13/04/2024	27/04/2024
Integración de sistemas y desarrollo de características avanza	15días	37.5horas	28/04/2024	12/05/2024
Pruebas unitarias y de integración	9días	22.5horas	13/05/2024	21/05/2024
Hito: Versión funcional de la aplicación	0día	0hora	21/05/2024	21/05/2024
☐ Entrega Final	21días	52.5horas	22/05/2024	11/06/2024
Pruebas finales y de usabilidad	8días	20horas	22/05/2024	29/05/2024
Corrección de errores y ajustes	4días	10horas	30/05/2024	02/06/2024
Preparación de documentación final	8días	20horas	03/06/2024	10/06/2024
Compilación y empaquetado de la entrega	1día	2.5horas	11/06/2024	11/06/2024
Hito: Entrega final de la aplicación y documentación	0día	0hora	11/06/2024	11/06/2024

Figura 1. Lista de tareas

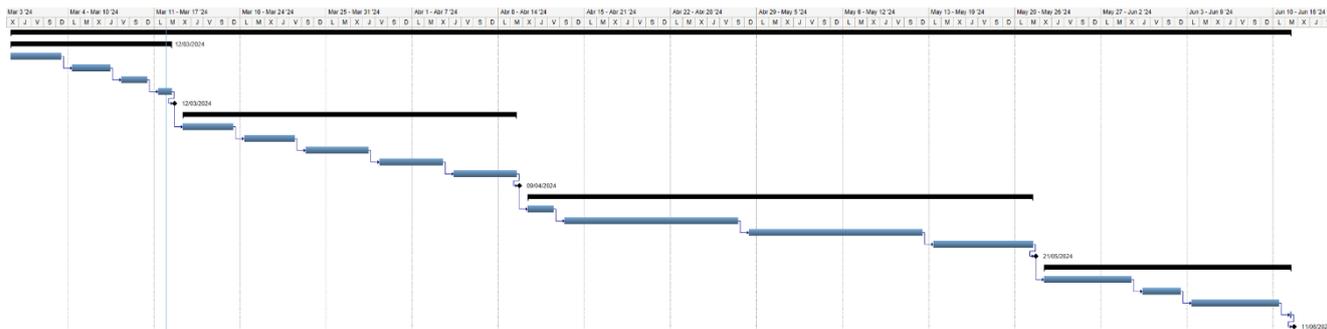


Figura 2. Diagrama de Gantt

1.5 Breve resumen de productos obtenidos

Al completar el proyecto se obtendrán los siguientes elementos:

- **Aplicación Móvil Desarrollada:** La aplicación final de recomendación de videojuegos, tanto la versión funcional empaquetada, como el código fuente.
- **Memoria del proyecto:** El informe que documenta el trabajo de desarrollo de la aplicación, junto con sus productos intermedios: plan de trabajo, requisitos, diseño conceptual, diseño técnico, pruebas, etc.
- **Presentación del proyecto:** Una presentación en video que comunica los aspectos clave del proyecto, objetivos, decisiones de diseño, etc., junto con una demostración de la aplicación.

1.6 Breve descripción de los otros capítulos de la memoria

Los capítulos restantes de la memoria cubren los aspectos del diseño y desarrollo del proyecto, así como la evaluación de los productos obtenidos:

- **Diseño centrado en el usuario:** Se detallan las distintas etapas del proceso, incluyendo la identificación de los usuarios objetivo, la recopilación de requisitos, el diseño conceptual, la creación de prototipos y las pruebas de usabilidad.
- **Diseño técnico:** Se detallan las especificaciones técnicas del proyecto, tales como la definición de casos de uso y el diseño de la arquitectura del sistema, utilizando los requisitos recopilados.
- **Desarrollo:** Se documenta el proceso de implementación del proyecto, incluyendo las herramientas, *frameworks*, etc. utilizados, y se justifican las decisiones técnicas tomadas.
- **Pruebas:** Este capítulo aborda las estrategias de prueba implementadas para verificar la funcionalidad de la aplicación, asegurando que cumpla con las expectativas.
- **Conclusiones:** Este capítulo resume las lecciones aprendidas, evalúa el cumplimiento de los objetivos, analiza la efectividad de la planificación y metodología, y propone futuras líneas de investigación o desarrollo no cubiertas en el proyecto.

- **Glosario:** Se definen los términos y acrónimos clave utilizados.
- **Anexos:** Contiene material complementario al contenido principal de la memoria.

2. Diseño centrado en el usuario

En este apartado, exploramos el diseño centrado en el usuario, un enfoque clave en el desarrollo de nuestra aplicación que se enfoca en las necesidades y experiencias de los usuarios para crear una interfaz intuitiva y accesible. Este método asegura que la aplicación sea relevante y eficiente, optimizando la experiencia del usuario final.

2.1 Usuarios

2.1.1 Métodos de indagación

Para la fase de investigación de usuarios, y debido a nuestras limitaciones de recursos, hemos usado el estudio de Richard Bartle sobre los diferentes tipos de jugadores en MUDs (Multi-User Dungeon) (1) como nuestro punto de partida. Este enfoque teórico nos ofrece una clasificación preliminar que facilita una comprensión inicial de nuestros usuarios. Además, para complementar esta base teórica, y garantizar que nuestra aplicación responda a las necesidades reales de los usuarios, hemos decidido realizar entrevistas con usuarios reales.

2.1.1.1 Estudio de Richard Bartle sobre los diferentes tipos de jugadores (1)

Basándonos en la taxonomía del estudio de Richard Bartle y teniendo en cuenta nuestro contexto de recomendación de videojuegos, hemos identificado dos arquetipos de jugadores principales.

El jugador **Hardcore** se basa en los tipos Killers y Achievers, quienes se caracterizan por su enfoque en la competencia y el logro dentro de los entornos de juego. Estos jugadores buscan desafíos, dominio y reconocimiento en sus experiencias de juego, lo que los distingue como un grupo dedicado y altamente comprometido.

Por otro lado, el jugador **Casual** se inspira en los tipos Socialisers y Explorers, quienes priorizan la exploración, la narrativa y las interacciones sociales sobre la competitividad. Este grupo disfruta de los videojuegos como una forma de relajación y socialización, prefiriendo experiencias más accesibles y menos intensivas en tiempo.

Al adoptar estos dos arquetipos basados en la clasificación de Richard Bartle, buscamos establecer un marco de referencia inicial para estudiar y entender a nuestros usuarios.

2.1.1.2 Entrevistas

Aunque el estudio de Bartle ofrece *insights* valiosos, es importante involucrar a personas reales para obtener un entendimiento profundo y contextual de las necesidades y expectativas específicas de los usuarios. Las entrevistas permiten explorar en detalle cómo cada tipo de jugador interactúa con las plataformas de recomendación y qué esperan de ellas. Este método directo de indagación nos ayuda a validar y ampliar nuestras hipótesis iniciales con datos reales y actuales.

Las entrevistas en persona y online se hicieron siguiendo el siguiente guion:

- **Introducción**
 - Breve explicación del propósito de la entrevista y asegurar la confidencialidad.
- **Preguntas demográficas**
 - ¿Qué edad tienes?
 - ¿Estado civil?
 - ¿Tienes hijos? ¿De qué edad?
 - ¿En qué trabajas?
 - Cuéntame un poco sobre cómo te manejas con la tecnología
- **Preguntas sobre Hábitos de Juego**
 - ¿Cómo describirías tu estilo de juego y qué tipo de videojuegos prefieres?
 - ¿Cuánto tiempo dedicas a jugar videojuegos semanalmente?
- **Preguntas sobre Descubrimiento y Selección de Juegos**
 - ¿Cómo sueles encontrar nuevos juegos para jugar?
 - ¿Qué factores son más importantes para ti al elegir un nuevo juego?
- **Experiencia con Sistemas de Recomendación**
 - ¿Has utilizado alguna vez aplicaciones o servicios de recomendación? Si es así, ¿qué te gustó y qué no te gustó de ellos?
- **Necesidades y Preferencias**
 - Describe una característica que considerarías esencial en una aplicación de recomendación de videojuegos ideal.
 - ¿Cómo prefieres interactuar con las recomendaciones de juegos? ¿Te gusta explorar opciones o prefieres una selección curada?
- **Retroalimentación y Mejoras**
 - ¿Qué funcionalidad añadirías a una aplicación de recomendación de videojuegos para mejorar tu experiencia?
- **Cierre**
 - Agradecimiento por la participación y oportunidad para comentarios finales.

2.1.1.3 Resumen de Resultados y Conclusiones

A partir de las respuestas y usando los arquetipos definidos, obtenemos lo siguiente:

- Jugadores *Hardcore*
 - Prefieren juegos con alta complejidad y profundidad estratégica.
 - Valoran los sistemas de clasificación y las recompensas que reconocen sus habilidades y logros.
 - Dedicar varias horas semanales al juego, a menudo en sesiones prolongadas.
 - Buscan constantemente nuevos desafíos y juegos que satisfagan su competitividad.
 - Prefieren recomendaciones personalizadas basadas en su historial de juegos y preferencias.
 - Les gustaría tener funcionalidades que les permitan comparar su rendimiento con el de otros jugadores.
 - Aprecian las reseñas detalladas y análisis profundos de juegos.
 - Frustraciones: Recomendaciones genéricas, falta de opciones avanzadas de filtrado y búsqueda.

- Jugadores Casuales
 - Prefieren juegos fáciles de aprender pero difíciles de dominar, con sesiones de juego cortas.
 - Valoran los juegos sociales y cooperativos.
 - Juegan para relajarse y desestresarse, no necesariamente para completar retos o competir.
 - Están interesados en una amplia gama de géneros, y a menudo dependen de recomendaciones de amigos.
 - Prefieren una interfaz sencilla con recomendaciones claras y concisas, y desean saber la duración de los juegos de antemano.
 - Frustraciones: Recomendaciones que no coinciden con su tiempo limitado para jugar, interfaces complicadas.

Conclusiones

- Recomendaciones personalizadas: Los usuarios desean que las recomendaciones se adapten a sus preferencias, tanto a los géneros en los que están interesados como a los juegos que juegan o recomendaciones pasadas.
- Descubrimiento de nuevos juegos: Además de ajustarse a sus preferencias conocidas, los usuarios están interesados en descubrir nuevos juegos que podrían no haber considerado por sí mismos. La aplicación debe equilibrar entre ofrecer lo familiar y presentar opciones novedosas.
- *Feedback* y mejora continua: La capacidad de proporcionar *feedback* directo sobre las recomendaciones, lo que se utiliza para mejorar la precisión del sistema de recomendación. La aplicación debe facilitar un mecanismo sencillo para que los usuarios expresen su satisfacción o insatisfacción con las recomendaciones recibidas.
- Claridad de las recomendaciones: Es importante para los usuarios entender por qué se les recomienda un juego específico. La aplicación debería proporcionar detalles de las recomendaciones, como similitudes con juegos previamente disfrutados o tendencias actuales.

- Detalle de las recomendaciones: Ofrecer información de los juegos recomendados, como calificaciones, duración, etc.

2.1.2 Perfiles de usuario

Teniendo en cuenta lo descubierto sobre los usuarios en la fase de investigación anterior, vamos a crear un perfil para cada protopersona basada en los arquetipos identificados.

Protopersona que representa al jugador *Hardcore*

<p style="text-align: center;">Susana Calvin</p> 	<p style="text-align: center;">Comportamientos</p> <ul style="list-style-type: none"> • Familiarizado con la tecnología y la utiliza frecuentemente. • Se mantiene al tanto de las últimas tendencias y lanzamientos en el mundo de los videojuegos a través de medios digitales como Twitter y YouTube. • Una parte importante de su tiempo libre lo dedica a jugar a videojuegos.
<p style="text-align: center;">Demografía</p> <ul style="list-style-type: none"> • 31 años • Soltera • Profesional en el sector tecnológico • Vive en un piso en la ciudad 	<p style="text-align: center;">Necesidades y objetivos</p> <ul style="list-style-type: none"> • Buscar constantemente nuevos juegos que probar. • Estar informado de novedades y tendencias. • Ajustar y personalizar los criterios de recomendación según sus intereses cambiantes, como enfocarse en géneros específicos, desarrolladores favoritos o modos de juego.

Protopersona que representa al jugador Casual

Íñigo Montoya



Comportamientos

- Dispone de poco tiempo libre debido a sus responsabilidades laborales y familiares.
- Sus sesiones de juego son esporádicas y breves, encajando en los escasos momentos libres que encuentra.
- Confía en las recomendaciones de amigos, siendo selectivo en cómo usa su limitado tiempo libre.

Demografía

- 43 años
- Casado
- Padre de un hijo de 9 años
- Teletrabaja
- Vive en una urbanización cerrada en las afueras

Necesidades y objetivos

- Método rápido y confiable para descubrir juegos, sin tener que dedicar mucho tiempo a la búsqueda o selección.
- Desea recibir sugerencias de juegos basadas en sus patrones de juego anteriores y las preferencias de sus amigos.

2.2 Diseño conceptual

Vamos a desarrollar los *point of view statements* para cada uno de los perfiles identificados en la fase de investigación de usuarios: el jugador *hardcore*, Susana Calvin y el jugador casual, Íñigo Montoya. Estos enunciados nos ayudaran a definir las necesidades específicas de los usuarios y orientar los flujos de interacción de la aplicación

2.2.1 Point of view statements

Para el jugador *hardcore*, Susana Calvin:

- Los jugadores *hardcore* como Susana necesitan una manera de explorar nuevos juegos que cumplan con sus expectativas porque están en constante búsqueda de nuevas experiencias que coincidan con su dedicación y experiencia.
- Los jugadores *hardcore* como Susana necesitan estar informados sobre los últimos lanzamientos y tendencias en el mundo de los videojuegos porque, dada su dedicación a esta afición, no quieren perderse ningún juego que pueda ser de su interés.
- Los jugadores *hardcore* como Susana necesitan una forma de adaptar y personalizar las recomendaciones de juegos porque sus intereses son amplios y, dependiendo del momento, pueden centrarse en temáticas específicas, reflejando la evolución de sus intereses a lo largo del tiempo.

Para el jugador casual, Íñigo Montoya:

- Los jugadores casuales como Íñigo necesitan una forma rápida y fiable de descubrir juegos que se ajusten a sus limitaciones de tiempo y preferencias porque desean optimizar su tiempo libre y no quieren invertir mucho tiempo en la búsqueda o selección de nuevos títulos.
- Los jugadores casuales como Íñigo necesitan dar *feedback* a las recomendaciones, por ejemplo ignorar sugerencias, porque dadas sus limitaciones de tiempo, desean ser selectivos y asegurarse de que las recomendaciones futuras sean cada vez más relevantes.

2.2.2 Flujos de interacción

A continuación, definimos los principales diagramas de flujo que representan las interacciones del usuario con la aplicación.

Proceso de *login* y registro / *onboarding*

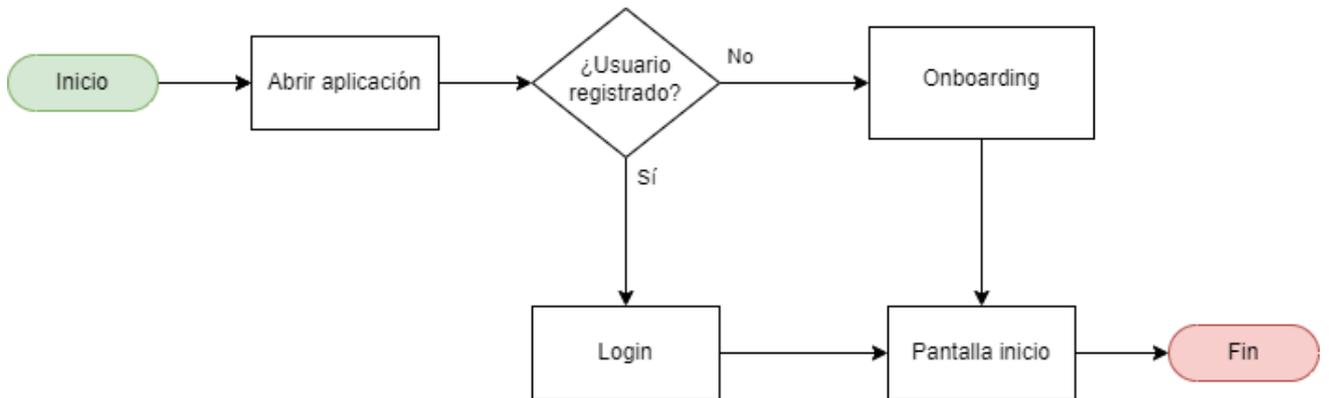


Figura 3. Login y Onboarding

Proceso para añadir juegos a las listas de preferencias del usuario

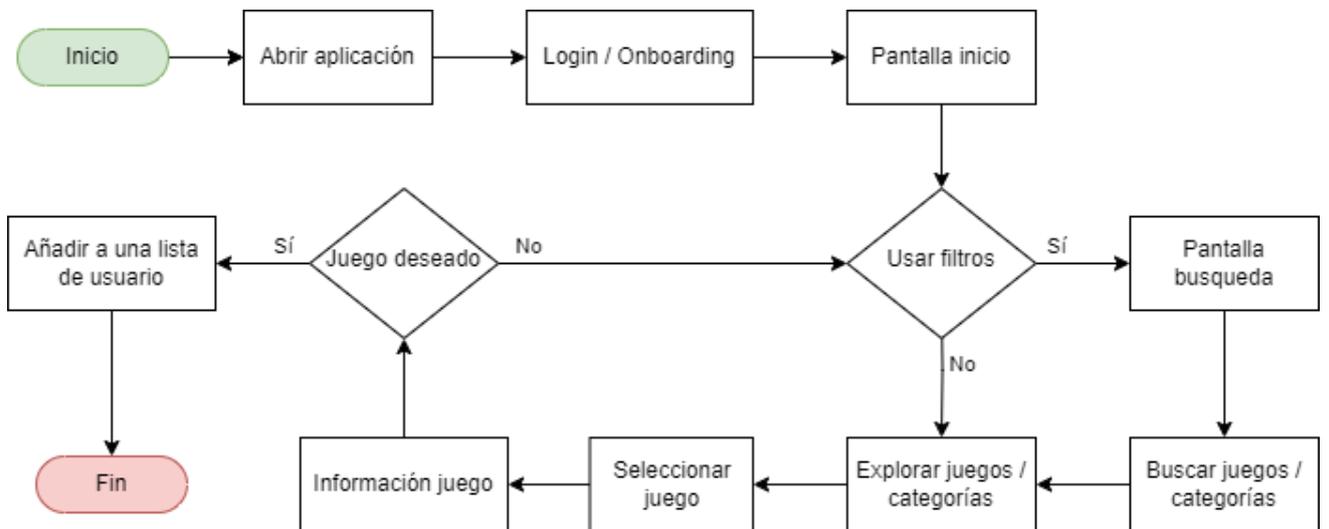


Figura 4. Añadir juego a una lista de usuario

Proceso para la modificación de las preferencias del perfil de usuario

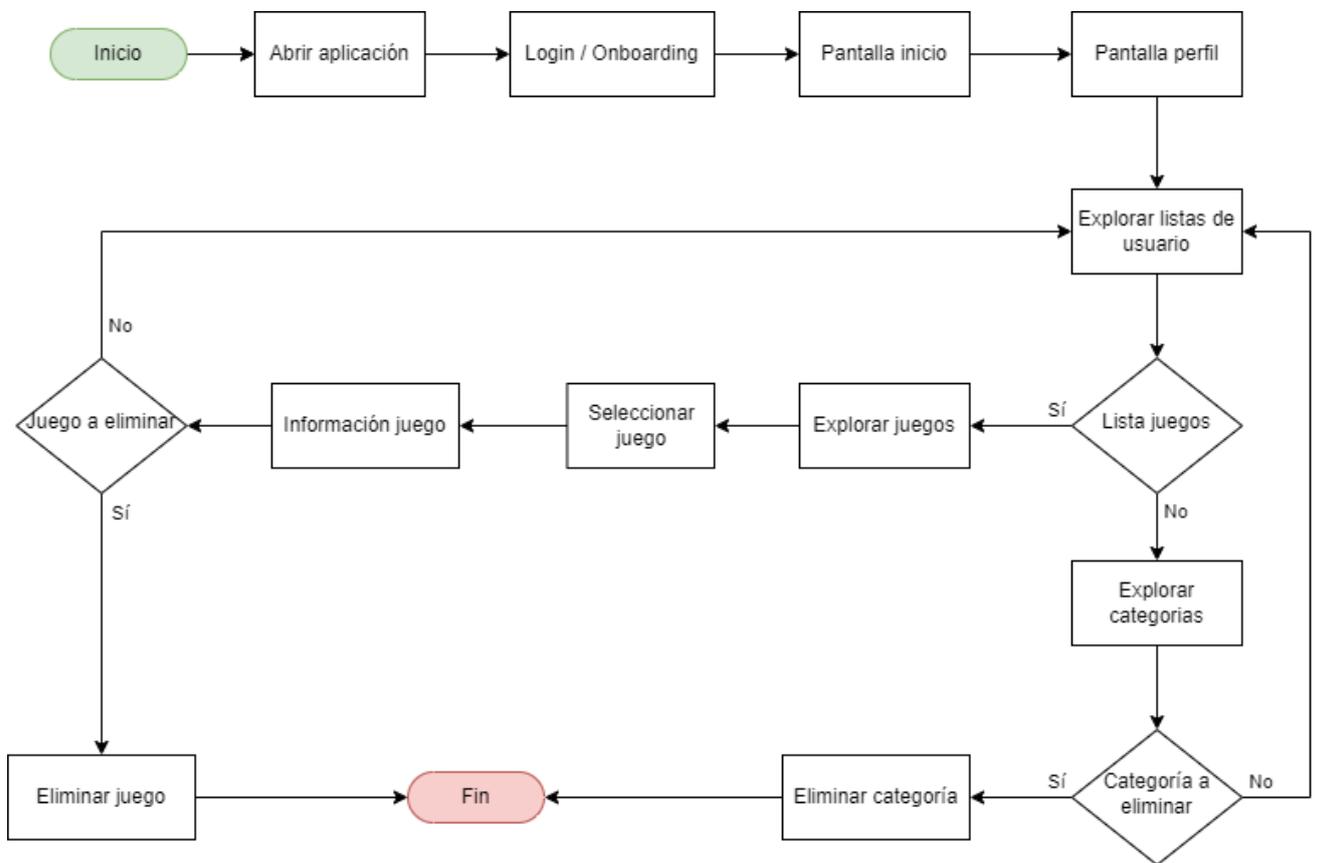


Figura 5. Modificar preferencias

2.3 Prototipado

2.3.1 Sketches

En esta sección presentamos los *sketches* con la primera aproximación al diseño de la aplicación, ayudando a conceptualizar las ideas principales en torno a la interacción y la estructura de la interfaz.

La primera vez que entremos en la aplicación, durante el registro (*onboarding*), es conveniente indicar unas preferencias iniciales para que las recomendaciones sean relevantes, aunque siempre se tendrá la opción de modificarlas en el perfil de usuario.

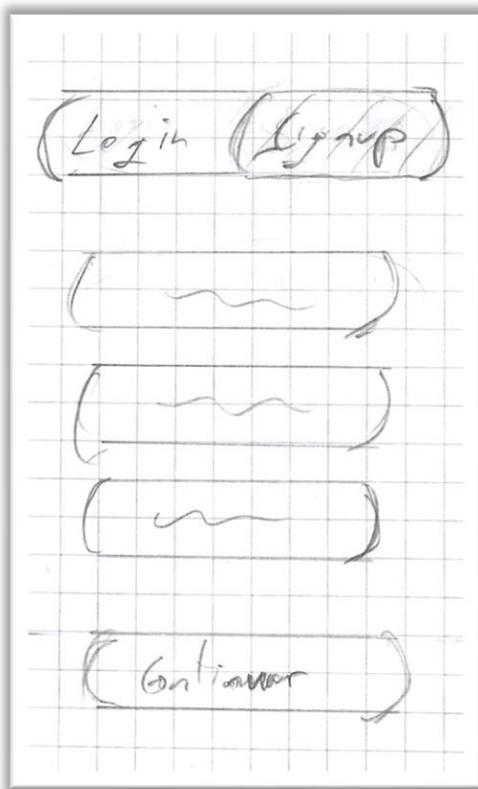


Figura 7. Login / Signup

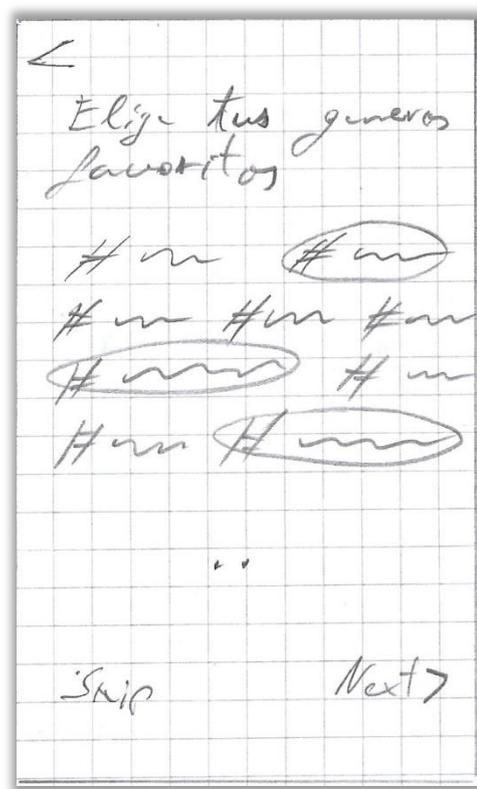


Figura 6. Onboarding, preferencias

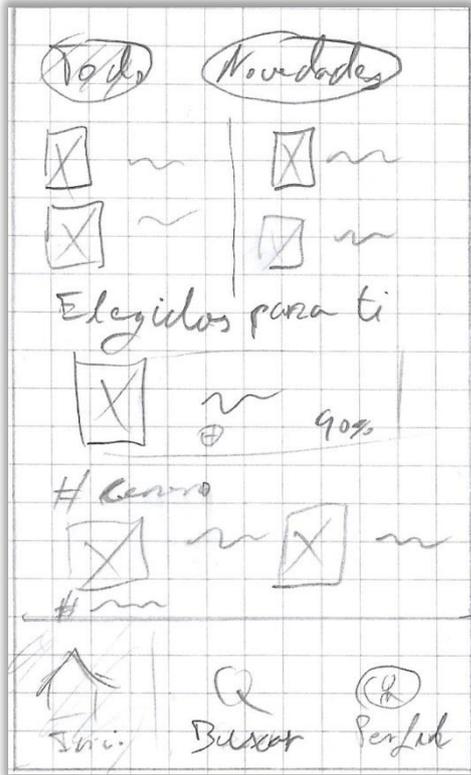


Figura 9. Home

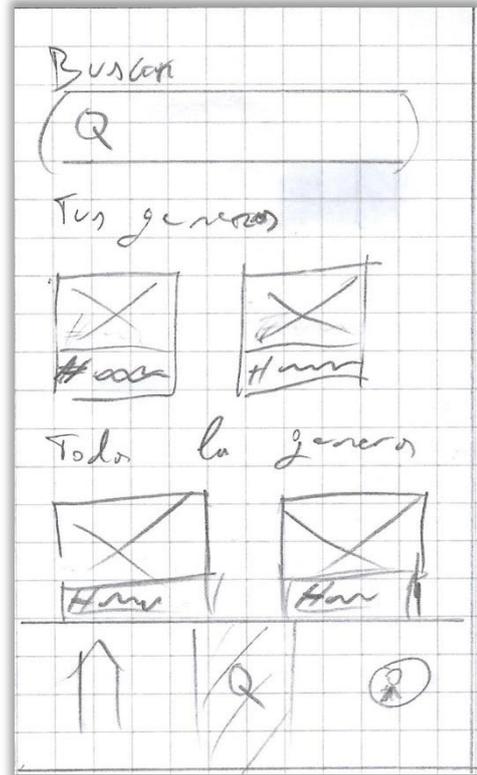


Figura 8. Búsqueda

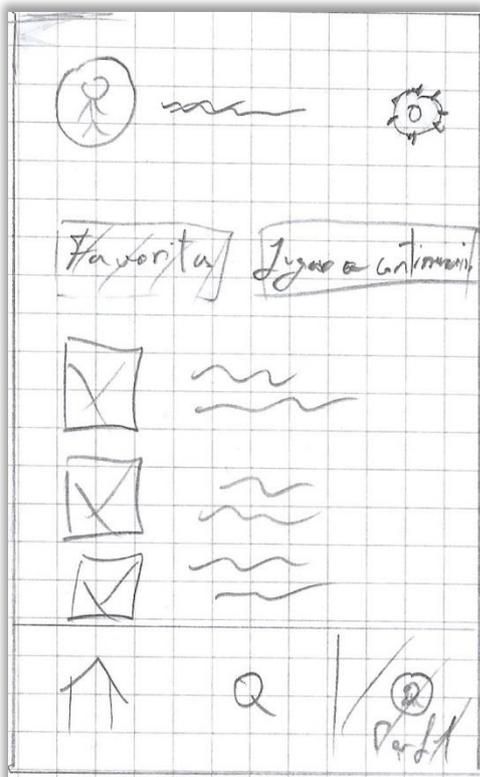


Figura 11. Perfil

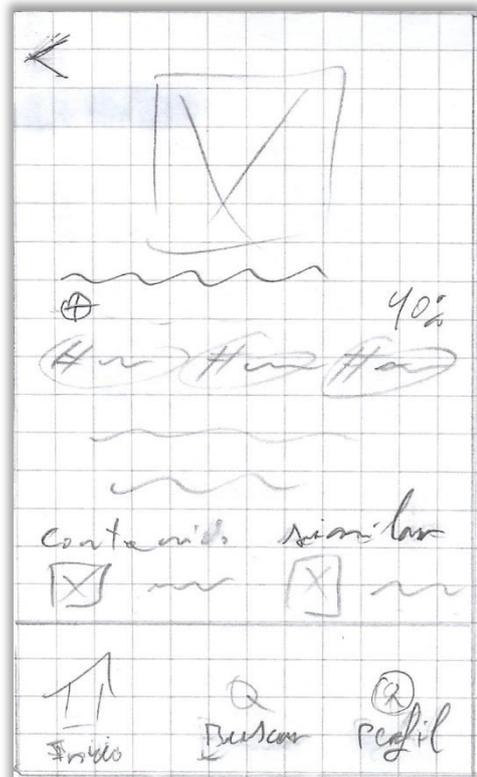


Figura 10. Información del juego

2.3.2 Prototipo horizontal

Mediante el prototipo horizontal de alta fidelidad, avanzaremos sobre las ideas planteadas en los bocetos iniciales, permitiéndonos obtener un entendimiento más detallado de los objetivos y necesidades identificados en los apartados anteriores.

Enlace al prototipo:

<https://www.figma.com/proto/ILd8bYJIMt7NuROOTNkOdL/Prototipo-Monkeys-Together-Strong?page-id=0%3A1&type=design&node-id=4-2803&viewport=722%2C517%2C0.46&t=6wRYXSN68VFCWdfd-1&scaling=min-zoom&starting-point-node-id=4%3A2803&mode=design>

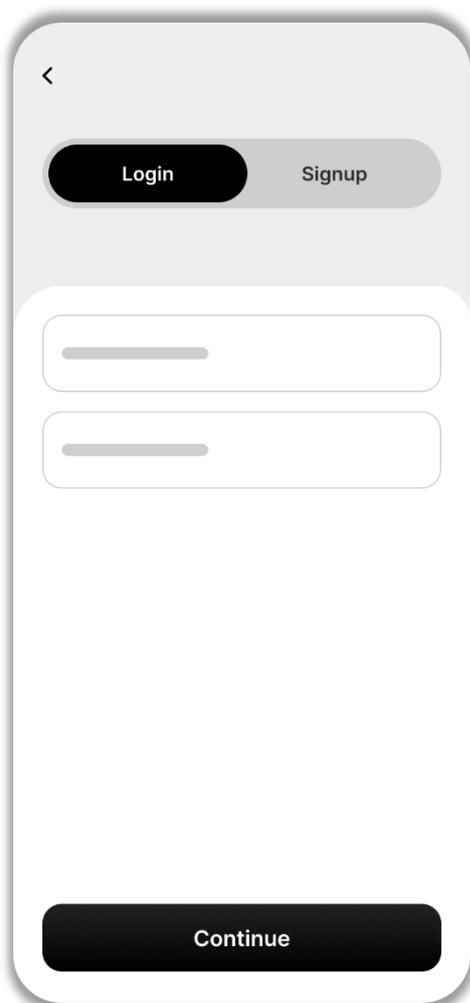


Figura 13. Login

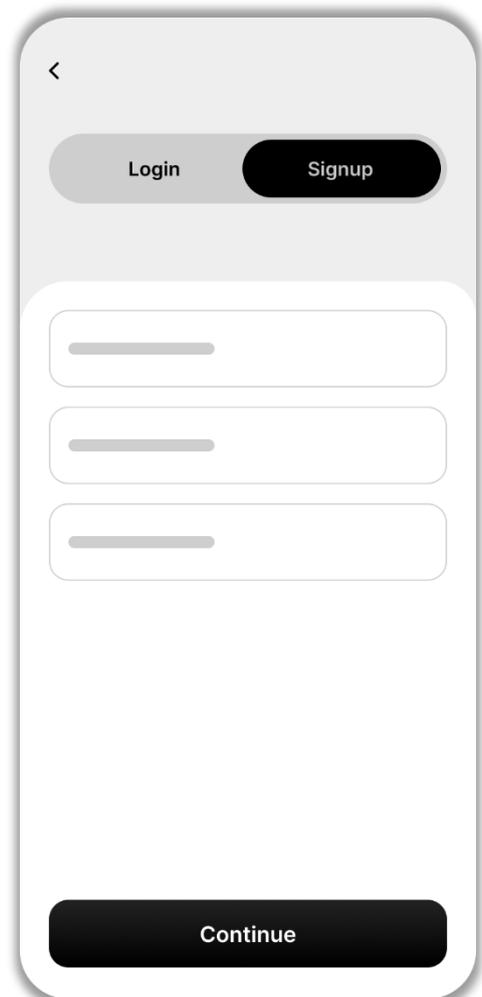


Figura 12. Signup

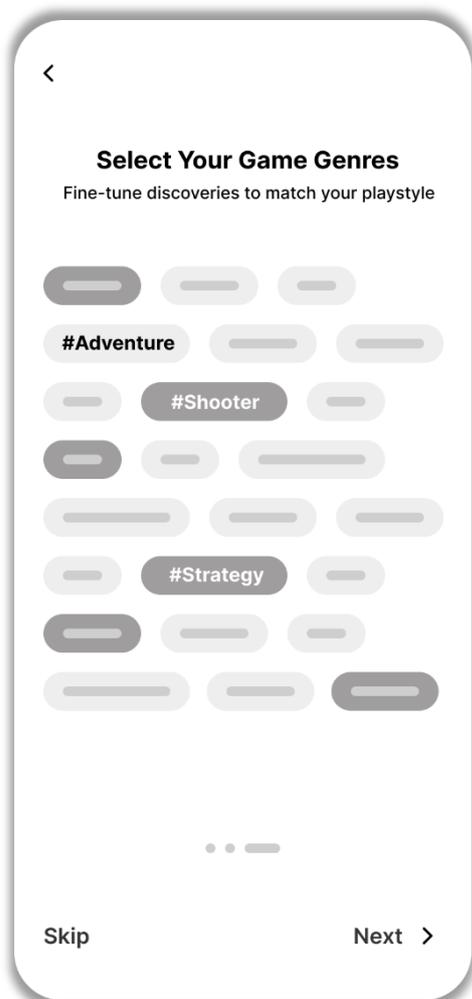


Figura 15. Onboarding, preferencias

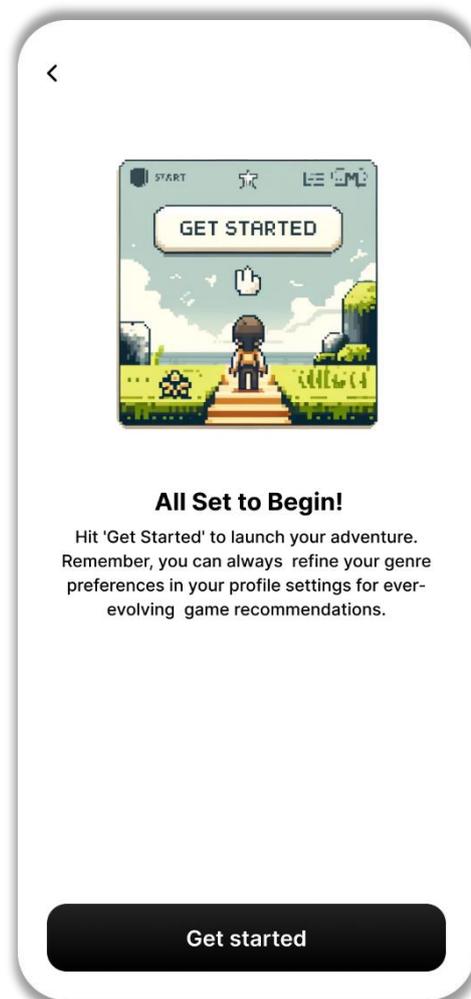


Figura 14. Onboarding, get started

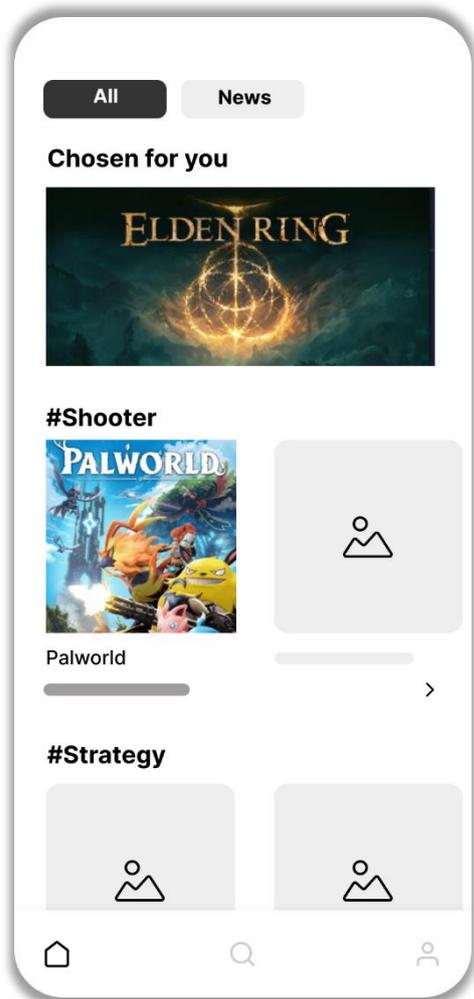


Figura 16. Home

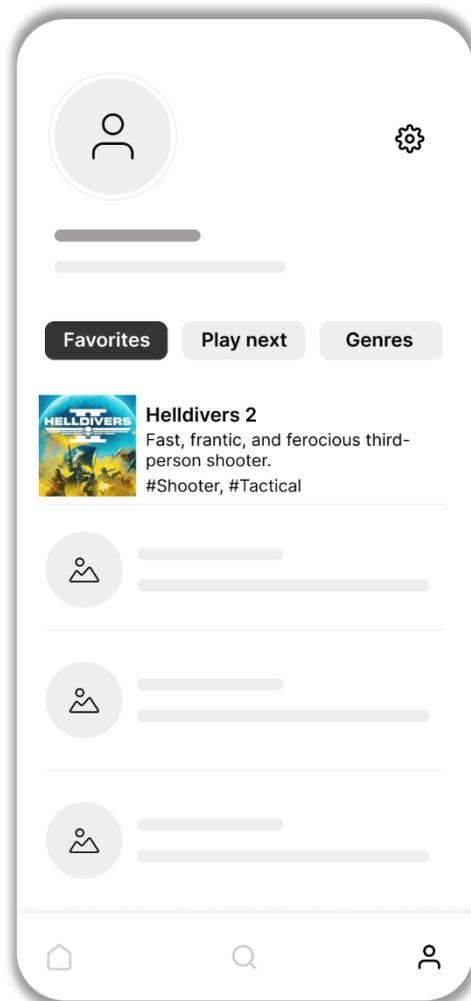


Figura 17. Perfil

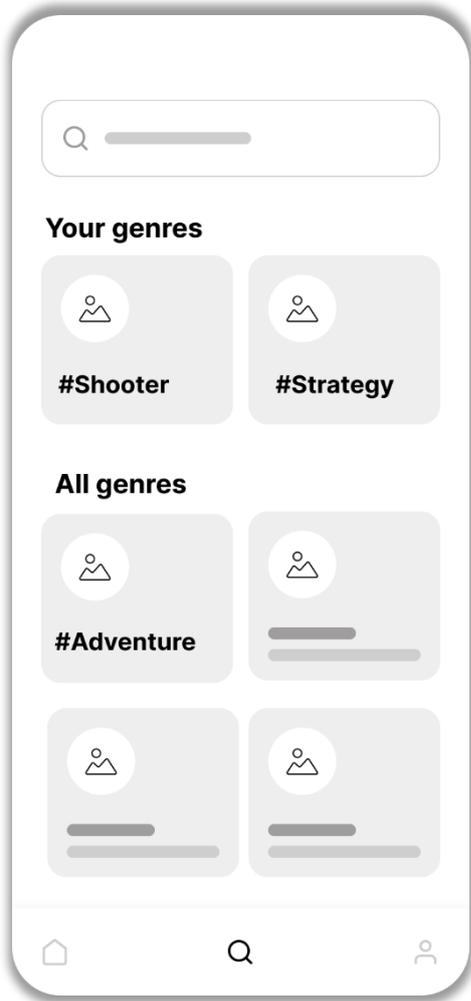


Figura 19. Búsqueda

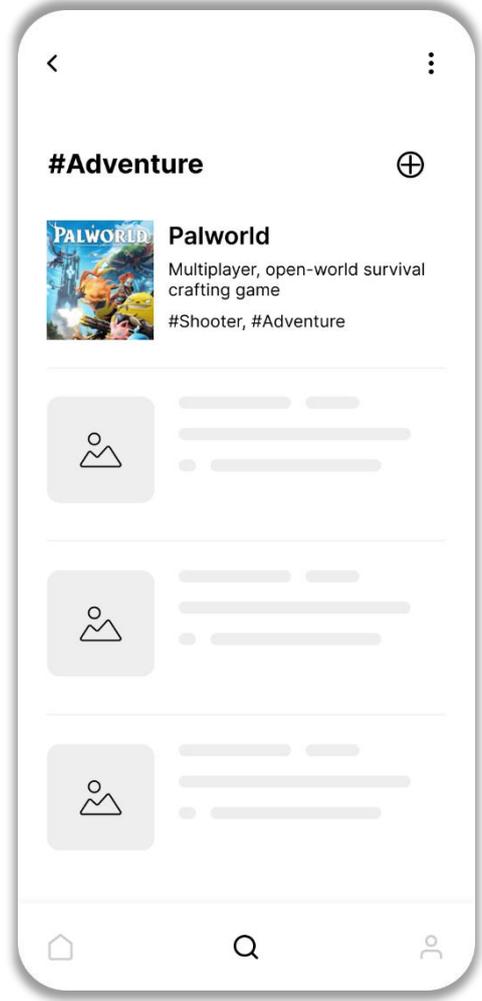


Figura 18. Categoría o género

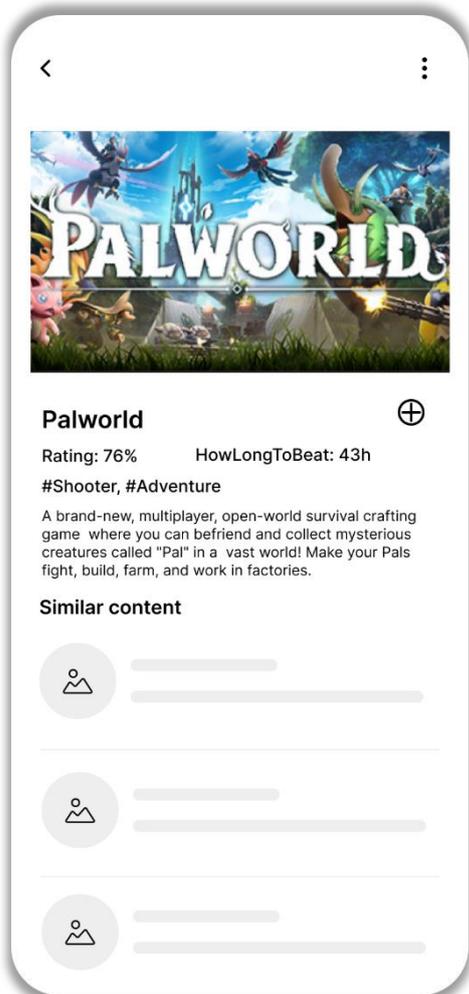


Figura 20. Información del juego

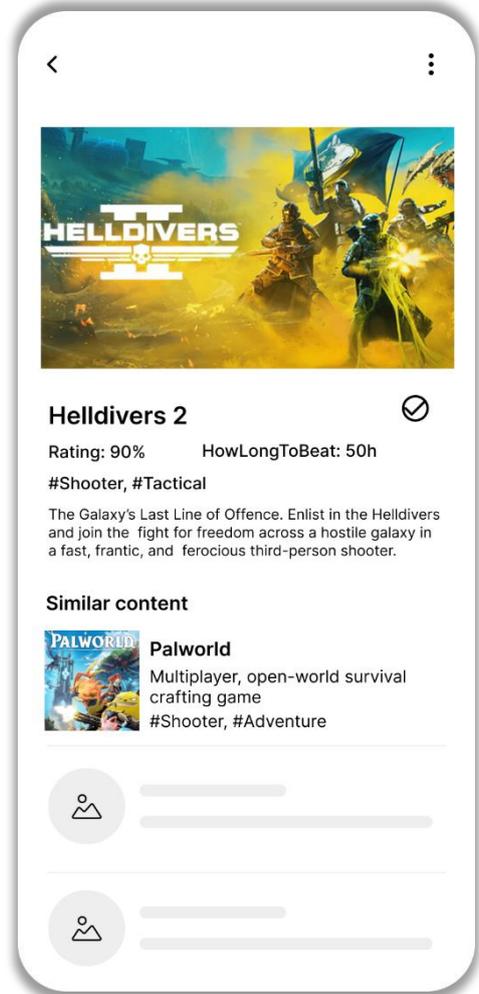


Figura 21. Información del juego

2.4 Evaluación

Llevaremos a cabo la evaluación del prototipo mediante pruebas con usuarios reales para validar los *insights* obtenidos en las fases de investigación previas e identificar oportunidades de mejora

2.4.1 Guion de la sesión de pruebas

- Introducción
 - Breve explicación del propósito, el procedimiento a seguir y asegurar la confidencialidad.

- Cuestionario demográfico
 - ¿Qué edad tienes?
 - ¿Estado civil?
 - ¿Tienes hijos? ¿De qué edad?
 - ¿En qué trabajas?
 - Cuéntame un poco sobre cómo te manejas con la tecnología
 - ¿Cuánto juegas a la semana?

- Tareas asignadas para el test:
 1. Registro en la aplicación:
Acceder a la aplicación y completar el proceso de registro.

 2. Búsqueda de un juego:
Utilizar la función de búsqueda para encontrar un juego de su género preferido.

 3. Añadir un juego a favoritos:
Desde la pestaña "Inicio", explorar y seleccionar un juego para añadirlo a la lista de favoritos.

 4. Modificar preferencias en el perfil:
Ir al perfil de usuario y ajustar los juegos favoritos.

- Cuestionario final sobre las tareas realizadas
 - ¿Hubo algún paso en el proceso de registro que te resultara confuso o tedioso?
 - ¿Qué filtros echas en falta para la búsqueda y exploración de juegos y géneros?
 - ¿Cuánto tardaste en encontrar los controles para añadir juegos y géneros a favoritos?
 - ¿Has tenido dificultades en encontrar las preferencias a modificar en el perfil?
 - ¿Hubo algún momento que no supieras cómo proceder o qué hacer a continuación?
 - ¿Algún comentario sobre el uso de la aplicación?

2.4.2 Feedback de las pruebas

Teniendo en cuenta los perfiles de usuarios de las fases de investigación previas, podemos clasificar a nuestro usuario de pruebas en el arquetipo de jugador casual (Íñigo Montoya)

	Inputs Positivos	Inputs Negativos
Concepto general	Diseño de interfaz limpio y eficiente que facilita la navegación.	
Onboarding	El proceso de registro es directo y sencillo.	
Búsqueda de Juego	La interfaz de búsqueda es clara y no requiere de esfuerzo adicional para entenderla.	Se echan en falta filtros por género en la pantalla de búsqueda.
Añadir a Favoritos	Controles intuitivos.	Se echan en falta opciones para gestionar favoritos directamente sin tener que entrar en el detalle del juego.
Modificar Preferencias		Necesidad de entrar en el detalle de los juegos para ajustar las preferencias de favoritos.
Otros	La aplicación es intuitiva y fácil de usar.	Sugerencia de añadir accesos directos para la gestión de favoritos y preferencias desde las pantallas principales.

2.4.3 Puntos de mejora

- Filtros por género en la pantalla de búsqueda:
La funcionalidad de búsqueda se optimizará con la integración de filtros por género, mejorando así la precisión y la rapidez con la que los usuarios pueden encontrar juegos de su interés.
- Gestión directa de favoritos:
Se implementará la opción de añadir o eliminar juegos de la lista de favoritos de manera directa desde cualquier listado, simplificando el proceso de personalización del usuario.
- Facilitar la modificación de preferencias en el perfil:
Optimizar el área del perfil para que los usuarios puedan ajustar sus preferencias de juegos y géneros más fácilmente, eliminando la necesidad de visitar la página de detalle de cada juego. Se implementará esta mejora, reconociendo la importancia de una gestión de preferencias sencilla y accesible en una aplicación centrada en recomendaciones.

3. Diseño técnico

3.1. Definición de los casos de uso

A continuación se definen los casos de uso correspondiente a los flujos principales de la aplicación.

Caso de Uso 1: Registro de usuario / *Onboarding*

Identificador	CU-001
Nombre	Registro de usuario / <i>Onboarding</i>
Prioridad	Alta
Descripción	Registro inicial del usuario a través de la pestaña "signup", donde se configuran las preferencias iniciales.
Actores	Usuario nuevo
Pre-Condiciones	La aplicación está instalada y el usuario no está registrado.
Iniciado por	Usuario selecciona la pestaña "signup".
Flujo	1. El usuario abre la aplicación y selecciona "signup". 2. El usuario ingresa información requerida para el registro. 3. El usuario selecciona preferencias iniciales. 4. El sistema registra al usuario y almacena las preferencias. 5. Se confirma el registro y se redirige al usuario al inicio.
Notas	Este es el único proceso de registro para nuevos usuarios.

Caso de Uso 2: Añadir juegos y géneros a preferencias

Identificador	CU-002
Nombre	Añadir juegos y géneros a preferencias
Prioridad	Alta
Descripción	Usuarios añaden juegos y géneros a sus preferencias desde cualquier parte de la aplicación.
Actores	Usuario registrado
Pre-Condiciones	Usuario debe estar registrado y haber iniciado sesión.
Iniciado por	Usuario decide añadir juegos o géneros.
Flujo	1. Usuario navega dentro de la aplicación 2. En cualquier momento, el usuario puede añadir juegos y géneros a sus preferencias. 3. El sistema actualiza las preferencias en tiempo real. 4. Se confirma la adición en la interfaz.
Notas	Los usuarios pueden añadir y eliminar juegos a sus preferencias desde cualquier lugar dentro de la aplicación.

Caso de Uso 3: Modificar y consultar preferencias

Identificador	CU-003
Nombre	Modificar y consultar preferencias
Prioridad	Alta
Descripción	Usuarios consultan y modifican sus preferencias desde la pantalla de perfil.

Actores	Usuario registrado
Pre-Condiciones	Usuario debe estar registrado y haber iniciado sesión.
Iniciado por	Usuario accede a su perfil.
Flujo	<ol style="list-style-type: none"> 1. Usuario selecciona la opción de perfil. 2. Consulta o modifica juegos y géneros preferidos. 3. El sistema actualiza la base de datos con las nuevas preferencias. 5. Se muestra una notificación de éxito.
Notas	Permite personalización continua y consulta de preferencias.

Caso de Uso 4: Buscar y explorar juegos

Identificador	CU-004
Nombre	Buscar y explorar juegos
Prioridad	Alta
Descripción	Usuarios buscan y exploran juegos por nombre o género desde la pantalla de búsqueda.
Actores	Usuario registrado
Pre-Condiciones	Usuario debe estar registrado y haber iniciado sesión.
Iniciado por	Usuario accede a la pantalla de búsqueda.
Flujo	<ol style="list-style-type: none"> 1. Usuario ingresa términos de búsqueda o selecciona un género. 2. El sistema muestra resultados basados en la entrada del usuario. 3. Usuario puede explorar los resultados y seleccionar juegos para ver más detalles.
Notas	Facilita la exploración y descubrimiento de juegos basados en intereses específicos.

Caso de Uso 5: Consultar detalles de juegos

Identificador	CU-005
Nombre	Consultar detalles de juegos
Prioridad	Alta
Descripción	Usuarios consultan detalles completos de juegos desde cualquier listado de juegos.
Actores	Usuario registrado
Pre-Condiciones	Usuario debe estar registrado y haber iniciado sesión.
Iniciado por	Usuario selecciona un juego desde cualquier listado en la aplicación.
Flujo	<ol style="list-style-type: none"> 1. Usuario elige un juego de interés. 2. El sistema muestra una página con detalles completos del juego, incluyendo descripción, <i>rating</i>, imágenes, etc. 3. Usuario puede agregar el juego a su lista de favoritos.
Notas	Este caso de uso es importante para ofrecer información relevante sobre cada juego.

3.2. Diseño de la arquitectura

3.2.1. Base de datos

El siguiente diagrama de base de datos muestra cómo se almacenan las credenciales de los usuarios y sus preferencias en una base de datos relacional.

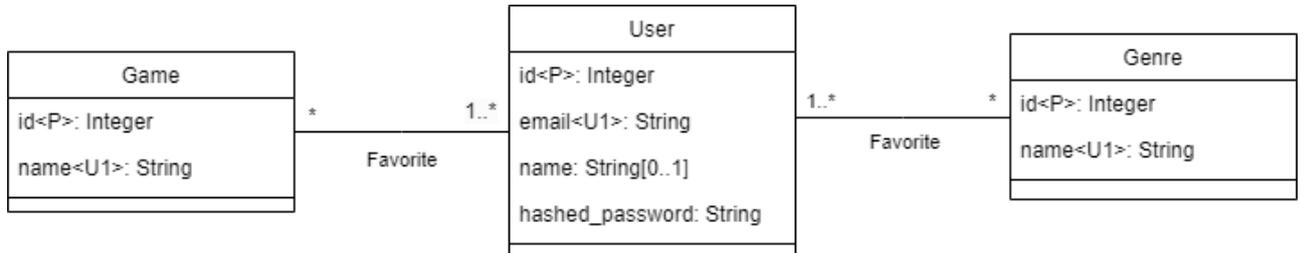


Figura 22. Diagrama de base de datos

3.2.2. Diagrama de entidades y clases

Nuestra aplicación utiliza un enfoque multiparadigma, combinando la programación orientada a objetos y funcional para manejar diversas necesidades técnicas. El diagrama de clases UML presentado refleja esta integración, mostrando las entidades, módulos y sus interrelaciones dentro del sistema.

En el diagrama del *frontend*, hemos agrupado los diversos componentes en entidades representativas clave, simplificando así la visualización de cómo interactúan las diferentes partes del sistema.

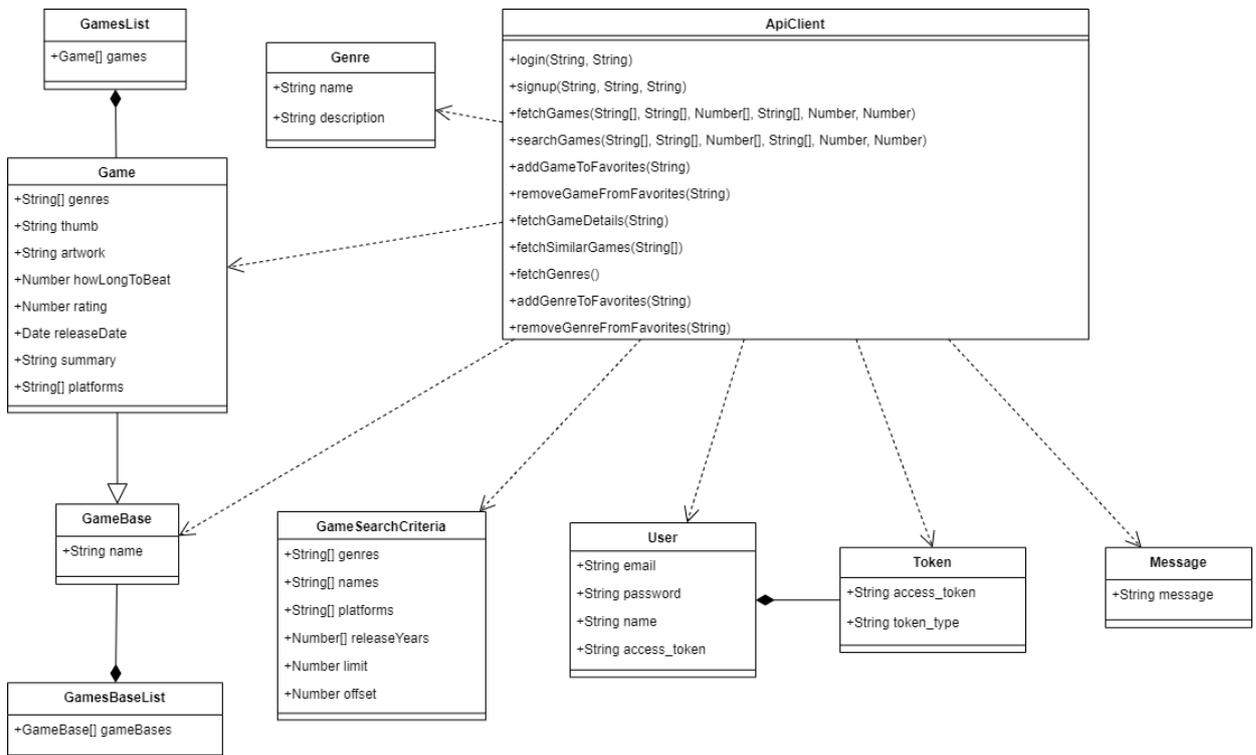


Figura 23. Diagrama de entidades del frontend

En el diagrama del *backend*, los módulos correspondientes a las rutas de la API se han consolidado en una única entidad representativa, *APIRouter*, así como los modelos y *schemas*, para proporcionar una visión más clara y simplificada de cómo se manejan las interacciones dentro de la arquitectura.

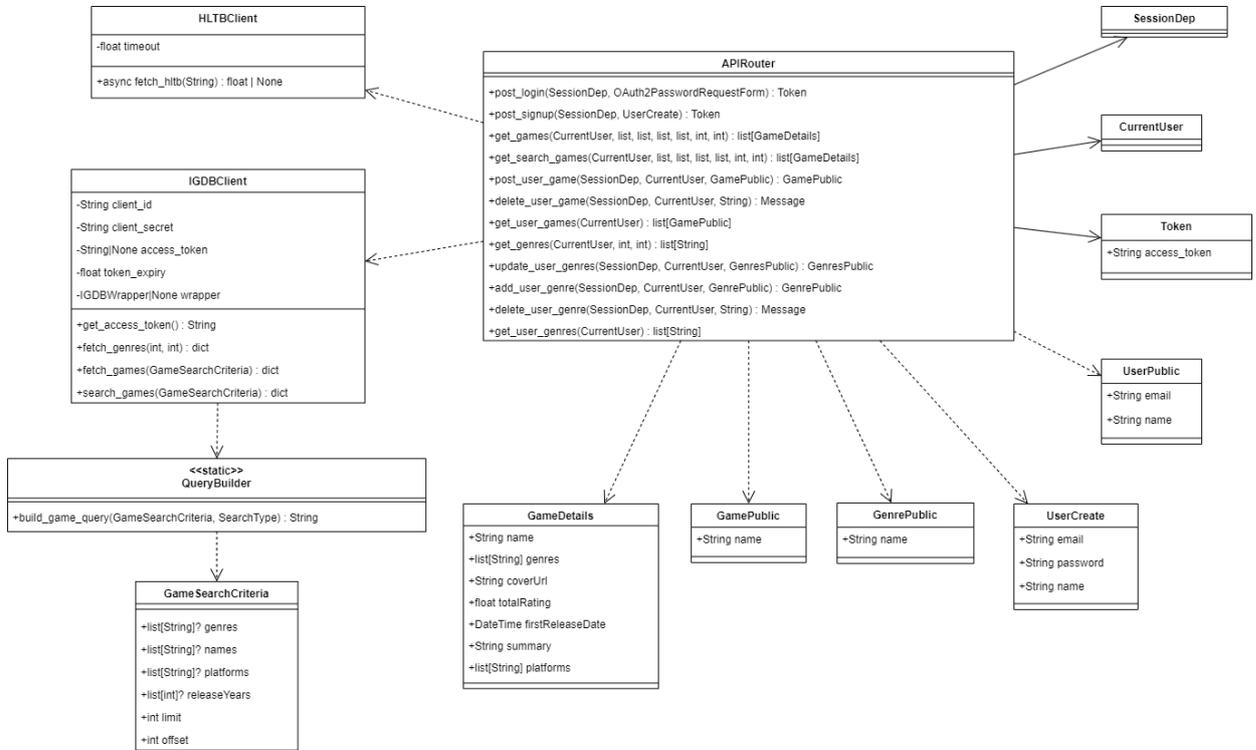


Figura 24. Diagrama de entidades del backend

3.2.3. Diagrama de arquitectura

El siguiente diagrama de arquitectura representa cómo la aplicación móvil se comunica con el *backend* a través de una API, mostrando las interacciones con la base de datos para la gestión de datos y la autenticación mediante JWT, así como las consultas a servicios externos como IGDB y HowLongToBeat.

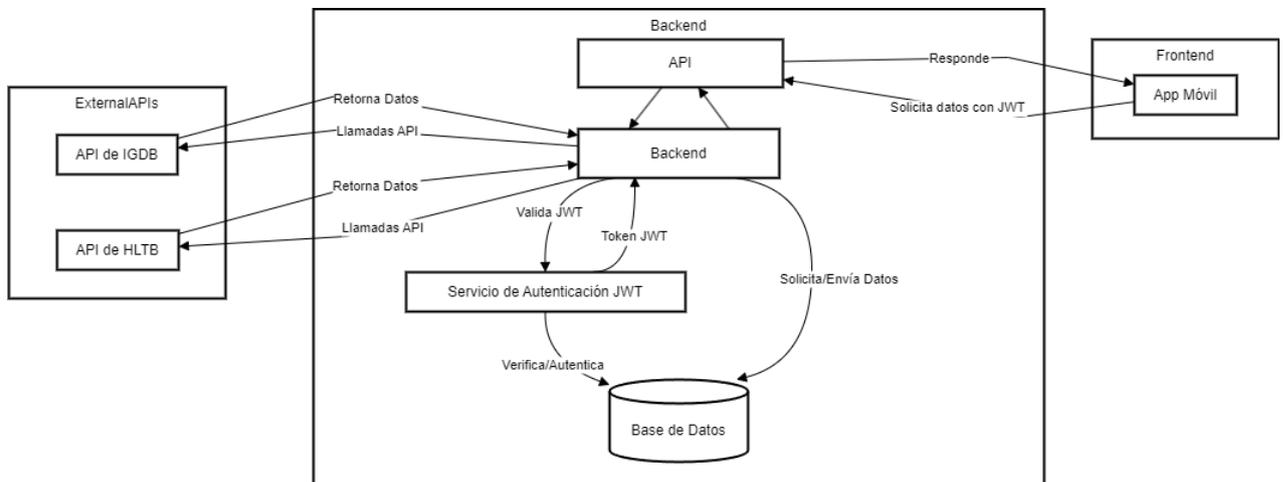


Figura 25. Diagrama de arquitectura

4. Desarrollo

Durante esta fase, se ha completado la primera iteración de la implementación del diseño desarrollado previamente, que culminó en la creación de un prototipo horizontal de alta fidelidad. El objetivo ha sido transformar este diseño en una aplicación móvil funcional. Aunque es una versión preliminar y todavía quedan tareas pendientes, se ha seguido un enfoque específico para garantizar que se pueda completar el proyecto con éxito en su entrega final.

4.1. Herramientas y stack tecnológico

El *stack* tecnológico se escogió con el objetivo de lograr un desarrollo eficiente y productivo para una aplicación multiplataforma.

4.1.1. Frameworks y librerías principales

Frontend

- **Ionic con Capacitor:** Proporciona una plataforma estable para desarrollar aplicaciones móviles con experiencia nativa iOS y Android. Capacitor permite el acceso a funcionalidades nativas del dispositivo, como cámaras y almacenamiento local. Utilizando un código base único.
- **Vue.js:** Framework que facilita la creación de interfaces de usuario reactivas y modulares. Su arquitectura basada en componentes permite un desarrollo más organizado y mantenible.
- **Axios:** Librería utilizada para realizar solicitudes HTTP, simplificando la comunicación con el *backend* y servicios externos.

Backend

- **FastAPI:** Framework moderno y rápido para la creación de APIs en Python, que ofrece características avanzadas para construir interfaces de backend eficientes y escalables.
- **SQLModel:** Librería que facilita la interacción con bases de datos SQL combinando SQLAlchemy y Pydantic para la gestión de modelos y validaciones.

4.1.2. Lenguajes de programación

- **TypeScript:** Utilizado en el frontend, lenguaje que extiende JavaScript con tipos estáticos, lo cual ayuda a mejorar la calidad del código, facilitando el desarrollo y mantenimiento.
- **Python:** Usado en el backend por su simplicidad y eficacia, adecuado para el rápido desarrollo de aplicaciones web y APIs.

4.1.3. IDE y herramientas de desarrollo

- **IntelliJ IDEA:** Entorno de desarrollo integrado que ofrece soporte avanzado para TypeScript e integración con *frameworks* como Vue.js. Proporciona funcionalidades de refactorización, así como sugerencias contextuales que mejoran la productividad. Además, tiene un sistema de depuración, JavaScript Debugger, el cual se conecta al servidor de desarrollo local de Ionic (Ionic serve) junto con Google Chrome.

Frontend

- **Vite:** Herramienta de construcción utilizada por su integración predeterminada con Vue.js. Proporciona un servidor rápido y sencillo que facilita el desarrollo, además de automatizar los *tests*.
- **Node.js y npm:** Node.js proporciona el entorno de ejecución para JavaScript en el servidor, mientras que npm se utiliza para la gestión de paquetes, facilitando la instalación y actualización de las dependencias del proyecto.
- **Ionic CLI:** Herramienta de línea de comandos que simplifica la creación, construcción y despliegue de aplicaciones Ionic. Ofrece comandos integrados para automatizar tareas comunes y optimizar el flujo de trabajo. Además de proporcionarnos el *scaffolding* de la aplicación.
- **Android Studio:** Utilizado para realizar pruebas en el emulador de Android, siendo un requisito de Ionic para la construcción y pruebas en dispositivos Android.

Backend

- **Poetry:** Herramienta de gestión de dependencias y empaquetado en Python, que simplifica la configuración de proyectos y la gestión de librerías.
- **Uvicorn:** Un servidor ASGI ligero recomendado para ejecutar aplicaciones FastAPI, optimizado para operaciones asincrónicas y de alto rendimiento.

4.1.4. Base de datos

SQLite: Utilizada en la fase de desarrollo por su simplicidad, integración con la biblioteca estándar de Python, y por proporcionar una base de datos ligera basada en disco, lo que elimina la necesidad de configurar un servidor de base de datos externo.

PostgreSQL: Seleccionada para los entornos de prueba y producción debido a su robustez como sistema de gestión de bases de datos relacional, que opera bajo el modelo cliente-servidor. Destaca por su estabilidad, extensas funcionalidades y su condición de código abierto.

4.1.5. Plataforma de despliegue en producción

Heroku: Tanto el backend como la base de datos de producción se alojan en Heroku, una plataforma como servicio (PaaS) que facilita el despliegue rápido y la gestión simplificada de aplicaciones en la nube. Heroku ofrece una integración directa con PostgreSQL, lo que permite una gestión eficiente de la base de datos en el mismo entorno que la aplicación. Además, ofrece un plan gratuito para estudiantes por medio de GitHub Student Developer Pack.

Enlace a la documentación de la API:

<https://monkeys-together-strong-6be19d4d341e.herokuapp.com/docs>

5. Pruebas

En esta etapa, se describen las pruebas realizadas y las acciones previstas en el plan de pruebas. Se detallarán los métodos utilizados hasta ahora y las estrategias previstas para asegurar el correcto funcionamiento de la aplicación.

5.1 Pruebas manuales

Para asegurarse de que la aplicación cumple con las expectativas de funcionalidad y usabilidad, se han realizado pruebas manuales en emuladores Android y en entornos web. Estas pruebas se han centrado en validar la funcionalidad básica y la usabilidad de la aplicación en un entorno controlado.

5.2 Pruebas automáticas

Para complementar las pruebas manuales, se implementaron varias pruebas automáticas, proporcionando un método eficiente y sistemático para asegurar que la aplicación cumple con los requisitos especificados y, al desarrollarse en paralelo, agilizan el proceso de desarrollo.

- **Mocks para simular el *backend*:** El uso de *mocks* nos ha permitido simular el comportamiento del backend sin necesidad de tener una infraestructura completa en funcionamiento. Esto facilita el desarrollo y las pruebas de la aplicación en fases tempranas, asegurando que podamos centrarnos en la implementación de funcionalidades sin depender de un *backend* completamente desarrollado.
- **Pruebas unitarias** Las pruebas unitarias se realizaron en paralelo a la implementación de la aplicación. Estas pruebas verifican funcionalidades específicas e independientes del sistema. Esta práctica es útil para detectar errores de forma temprana, asegurar la corrección del código y facilitar la refactorización, mejorando así la calidad y mantenibilidad del software.
- **Pruebas End-to-End (E2E):** Estas pruebas simulan la interacción completa del usuario con la aplicación, verificando que todos los componentes y sistemas funcionen correctamente en conjunto. Son especialmente útiles para identificar problemas de integración que no se detectan con pruebas unitarias.

5.2.1 Tecnologías utilizadas

Frontend

- **Vitest:** Framework de pruebas unitarias para JavaScript y TypeScript. Integra de forma nativa con Vite.
- **Cypress:** Framework de pruebas *end-to-end* (E2E) que permite escribir, ejecutar y depurar pruebas para aplicaciones web. Simula la interacción del usuario en un navegador real. Es compatible con Vite.
- **Vue Test Utils:** Librería oficial para probar componentes Vue.js. Proporciona utilidades para montar y renderizar componentes en un entorno de prueba. Se usa junto con test runners como Vitest y Cypress.
- **Axios Mock Adapter:** Librería para simular peticiones HTTP hechas con Axios. Permite interceptar y definir respuestas para las solicitudes. Utilizado para simular el *backend* durante el desarrollo y en las pruebas manuales y automáticas.

Backend

- **Pytest:** Un framework de pruebas para Python que permite escribir scripts de pruebas sencillos. Es ideal para probar las APIs desarrolladas con FastAPI.
- **FastAPI's Swagger UI / ReDoc with OpenAPI:** FastAPI aprovecha el estándar OpenAPI para generar automáticamente documentación detallada de la API, que es accesible a través de herramientas como Swagger UI y ReDoc. Estas interfaces permiten una interacción fácil y directa con la API, facilitando la realización de pruebas manuales y la comprensión de la API desde el navegador.

Junto con el código fuente de la aplicación se incluye un directorio “tests”, el cual contiene un conjunto de *tests* unitarios desarrollados en Vitest (“tests/unit”) . Y un conjunto de tests E2E desarrollados en Cypress (“tests/e2e”). De la misma manera, en el *backend* se incluyen varios *test* unitarios implementados con Pytest, también en su correspondiente directorio “tests”.

Ejecución de pruebas unitarias con ‘npx vitest’:

```
✓ tests/unit/utils/games.spec.ts (3)
✓ tests/unit/utils/handleError.spec.ts (2)
✓ tests/unit/utils/validator.spec.ts (2)
✓ tests/unit/preferences/games.spec.ts (6)
✓ tests/unit/homePage.spec.ts (1)

Test Files 5 passed (5)
Tests 14 passed (14)
Start at 21:00:28
Duration 1.94s (transform 621ms, setup 0ms, collect 1.02s, tests 52ms, environment 1.76s, prepare 1.28s)

PASS Waiting for file changes...
press h to show help, press q to quit
```

Figura 26. Ejecución *tests* unitarios

Ejecución de pruebas unitarias en el *backend* con 'pytest -vv'

```
tests/persistence/test_game_crud.py::test_create_game PASSED
tests/persistence/test_game_crud.py::test_add_favorite_game PASSED
tests/persistence/test_game_crud.py::test_remove_favorite_game PASSED
tests/persistence/test_game_crud.py::test_update_favorite_games PASSED
tests/persistence/test_game_crud.py::test_is_favorite_game PASSED
tests/persistence/test_game_crud.py::test_get_game_by_name PASSED
tests/persistence/test_user_crud.py::test_create_user PASSED
tests/persistence/test_user_crud.py::test_get_user_by_email PASSED
===== 8 passed in 0.41s =====
```

Figura 27. Ejecución *test* unitarios del *backend*

Documentación interactiva de los *endpoints* en formato OpenAPI:

auth

- POST** /login Login
- POST** /signup Signup

genres

- GET** /genres Read Genres
- GET** /user/genres Read User Genres
- PUT** /user/genres Update User Genres
- POST** /user/genres Add User Genre
- DELETE** /user/genres/{genre_name} Delete User Genre

games

- GET** /games Read Games
- GET** /games/search Search Games
- GET** /user/games Read User Games
- POST** /user/games Add User Game
- DELETE** /user/games/{game_name} Delete User Game

Figura 28. *Backend endpoints*

5.3 Plan de Pruebas

En el plan de pruebas previsto se incluyen las siguientes acciones:

- Pruebas E2E: Se implementarán varias pruebas E2E utilizando la herramienta Cypress. Estas pruebas intentarán cubrir todos los flujos críticos de la aplicación de manera automatizada, predecible y repetible.
- Refuerzo de pruebas unitarias: Se continuará ampliando el conjunto de pruebas unitarias para mejorar la cobertura de estas. El objetivo es asegurar que cada funcionalidad clave de la aplicación cuente con al menos una prueba unitaria asociada que verifique su correcto funcionamiento.
- Pruebas de Integración: Se realizarán pruebas de integración para confirmar la correcta interacción entre todos los componentes de la aplicación y el *backend*.
- Pruebas funcionales: Se tomará como referencia los casos de uso definidos en la sección del diseño técnico para validar los requisitos principales.

6. Conclusiones

Este proyecto ha ofrecido valiosas lecciones sobre el desarrollo y la implementación de una aplicación de recomendación de videojuegos centrada en el usuario. A pesar de los desafíos enfrentados, se lograron la mayoría de los objetivos propuestos, aunque con ciertas limitaciones en la complejidad del sistema de recomendación y las funcionalidades colaborativas, que no se incluyeron en el alcance del proyecto inicial.

6.1. Logro de objetivos

El proyecto ha logrado establecer un producto mínimo viable funcional, que ha servido como una primera iteración operativa de la aplicación. Este MVP ha permitido alcanzar el objetivo principal de proporcionar recomendaciones personalizadas a los usuarios, aunque con una complejidad algo menor a la originalmente planeada. La simplicidad del sistema de recomendación implementado refleja las restricciones de tiempo y recursos más que una limitación en la capacidad técnica. Además, se reconoce que la interfaz de usuario, si bien es completamente funcional, podría ser enriquecida en futuras versiones para mejorar la experiencia del usuario y la estética general de la aplicación.

6.2. Metodología y planificación

El seguimiento de una metodología ágil personalizada permitió adaptaciones constantes al plan de trabajo, lo cual fue importante para el cumplimiento de los objetivos. Las iteraciones cortas y la capacidad de adaptación demostraron ser adecuadas para un proyecto gestionado por un solo desarrollador.

6.3. Trabajo Futuro

Para futuras iteraciones del proyecto, se propone explorar las siguientes áreas:

- Mejora del sistema de recomendación: Integrar algoritmos más complejos.
- Funcionalidades colaborativas: Implementar un sistema de listas compartidas y recomendaciones colaborativas, permitiendo a los usuarios compartir y discutir sus juegos favoritos dentro de la aplicación.
- Optimización de la interfaz de usuario: Mejorar la interfaz, sobre todo en estética.

En conclusión, aunque el proyecto se considera que ha sido un éxito en términos de la implementación de las funcionalidades básicas, aún existen oportunidades de mejora para expandir y profundizar en las características ofrecidas.

7. Glosario

Adaptive (Adaptativo): Describe la práctica de diseñar múltiples versiones de un contenido o servicio para distintos tamaños de pantalla o dispositivos, eligiendo el diseño más adecuado en función del dispositivo que accede al contenido, lo que permite una experiencia más personalizada.

ASGI (Asynchronous Server Gateway Interface): Especificación para aplicaciones web en Python que permite la comunicación asíncrona entre el servidor y la aplicación, optimizando el manejo de conexiones simultáneas.

Boilerplate: Código repetitivo que se suele reutilizar con pocas o nulas modificaciones.

CLI (Command Line Interface): Interfaz basada en texto que permite a los usuarios interactuar con el sistema o aplicaciones mediante comandos introducidos en un terminal o consola.

Framework: Estructura conceptual y técnica que proporciona funcionalidades y herramientas predefinidas para facilitar el desarrollo de software, al proporcionar una base sobre la cual construir y organizar el código.

Hardcore: es un adjetivo referido a una forma intensa o comprometida de realizar una actividad.

IDE (Integrated Development Environment): Aplicación que ofrece un conjunto de herramientas integradas para el desarrollo, incluyendo editor de texto, compilador y depurador.

OpenAPI: Anteriormente conocida como Swagger, es una especificación estándar para describir interfaces de programación de aplicaciones (APIs) RESTful. Esta especificación permite a los desarrolladores detallar todos los aspectos de una API, facilitando la integración y automatización del uso de servicios web.

PWA (Progressive Web Application): Aplicación web que utiliza tecnologías modernas para ofrecer una experiencia de usuario similar a las aplicaciones nativas. Diseñadas para ser rápidas y adaptativas, permiten un desarrollo y mantenimiento eficiente al ser compatibles con múltiples plataformas.

Responsive (Adaptable): Se refiere a un enfoque que permite que el contenido se ajuste y funcione de manera óptima en diferentes dispositivos y tamaños de pantalla, mediante el uso de técnicas como media *queries* de CSS y unidades de medida relativas, que proporcionan una flexibilidad y adaptabilidad automáticas.

Scaffolding: Generación automática de código para crear la estructura básica de un proyecto, facilitando el inicio rápido de nuevos proyectos al proporcionar archivos de configuración y código básico.

8. Bibliografía

1. **Bartle, Richard.** Hearts, Clubs, Diamonds, Spades: Players Who Suit MUDs. [En línea] 1996. <http://mud.co.uk/richard/hcnds.htm>.
2. **Ionic Team.** Ionic Docs. [En línea] 2024. <https://ionicframework.com/docs/>.
3. **Krug, Steve.** *Don't make me think, revisited (3ra ed.)*. s.l. : New Riders, 2014.
4. **Ramírez, Sebastián.** FastAPI Docs. [En línea] 2024. <https://fastapi.tiangolo.com/>.
5. —. SQLAlchemy Docs. [En línea] 2024. <https://sqlmodel.tiangolo.com/>.

9. Anexos

9.1. Configuración del entorno y construcción de la aplicación

9.1.1. Estructura de los proyectos

- mobile-app/: Código fuente de la aplicación móvil.
 - o test/: Tests unitarios y de integración e2e.
 - o android/: Código autogenerado para Android.
 - o ios/**: Código autogenerado para iOS.
- backend/: Código fuente del backend.
 - o test/: Tests unitarios

9.1.2. Configuración de la aplicación móvil

Configuración de entornos

Se utilizan ficheros `.env` para gestionar la configuración en diferentes entornos. Las variables a configurar son:

- VITE_API_BASE_URL: URL base del backend de la aplicación.
- VITE_USE MOCKS: Define si se usa el servicio mock para simular el backend (false para usar el backend real).

Requisitos Previos

- Node.js y npm
- Ionic CLI
- Android Studio

Instalación de dependencias globales

Instalar las herramientas necesarias globalmente:

```
npm install -g @ionic/cli@latest native-run@latest
```

Configuración Inicial

- Instalar dependencias:

```
cd mobile-app
npm install
```

- Configurar variables de entorno en Windows:

Estas variables son necesarias para que las herramientas de desarrollo localicen el SDK de Android y ejecuten Android Studio y el emulador correctamente:

ANDROID_HOME: Apunta al directorio del SDK de Android. Necesario para que las herramientas de desarrollo como Gradle encuentren las herramientas del SDK:

```
C:\Users\Usuario\AppData\Local\Android\Sdk
```

CAPACITOR_ANDROID_STUDIO_PATH: Ruta al ejecutable de Android Studio. Necesario para abrir Android Studio desde la línea de comandos con Ionic/Capacitor CLI:

```
C:\Users\Usuario\AppData\Local\Programs\Android Studio\bin\studio64.exe
```

Construir y probar la aplicación Android:

- Construir y sincronizar la Aplicación:

```
npm run build  
npx cap sync
```

- Abrir en Android Studio:

```
npx cap open android
```

- Ejecutar en el emulador:

```
npx cap run android
```

Navegador:

- Ejecutar el servidor de desarrollo local:

```
ionic serve
```

- Simular dispositivo móvil:
 - o Abrir las herramientas de desarrollo del navegador (F12).
 - o Seleccionar la vista adaptable y elegir un dispositivo móvil.

9.1.3. Configuración del *backend*

Configuración de la base de datos

La aplicación utiliza varios ficheros `.env`` para gestionar la configuración de la base de datos en diferentes entornos. Se utiliza la variable ``ENV_MODE`` del fichero `.env`` para determinar cuál de los ficheros de entorno cargar (``development``, ``testing``, ``production``)

- Desarrollo:

Se utiliza el fichero ``.env.development`` que configura la aplicación para crear una base de datos SQLite en el directorio ``data``. `SQLModel` crea automáticamente la base de datos y las tablas correspondientes al iniciar la aplicación.

- Producción y pruebas:

Se utilizan los ficheros ``.env.production`` y ``.env.testing`` respectivamente. Estos configuran la conexión a una base de datos PostgreSQL preexistente. Es necesario asegurarse que la instancia de la base de datos esté creada para que `SQLModel` pueda generar las tablas al inicio.

Requisitos Previos

- Python 3.11
- Poetry

Instalación de dependencias globales

Instalar Poetry globalmente usando `pipx`:

```
pipx install poetry
```

Configuración Inicial

- Instalar dependencias:

```
cd backend  
poetry install
```

Ejecución local del *backend*

- Activar el entorno virtual:

```
poetry shell
```

- Iniciar el servidor de desarrollo local:

```
uvicorn app.main:app --reload
```

Documentación interactiva de la API

Acceder a la documentación interactiva generada por FastAPI, la cual utiliza OpenAPI para definir y permitir la prueba de endpoints directamente desde el navegador:

- Swagger UI:

```
http://localhost:8000/docs
```

- ReDoc:

```
http://localhost:8000/redoc
```

9.2. Manual de usuario

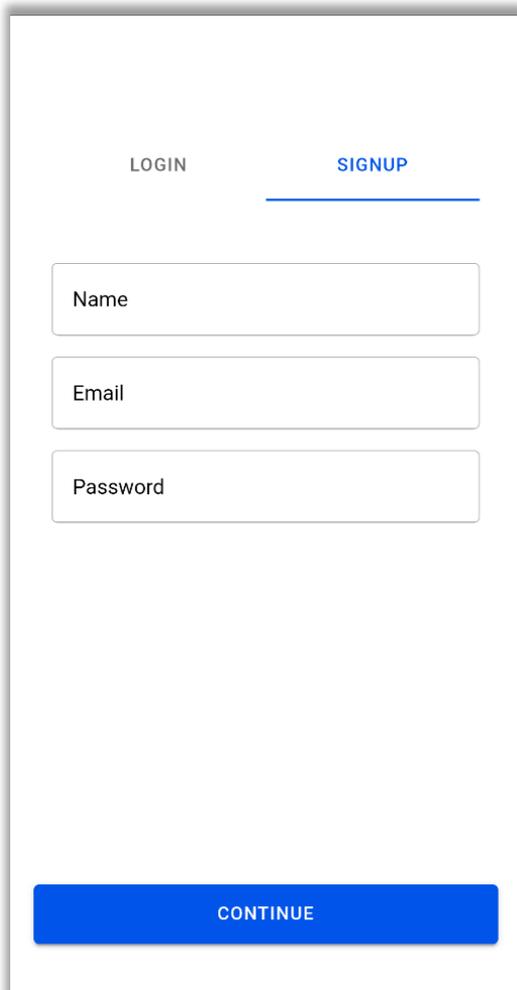
Este manual describe el uso de la aplicación móvil de recomendación de videojuegos, que adapta sus sugerencias a las preferencias del usuario.

1. Pantalla de registro (*Signup*)

Complete este formulario introduciendo su nombre, email y contraseña para crear su cuenta.

2. Pantalla de *onboarding*

Inmediatamente después del registro, se le pedirá seleccionar sus géneros de videojuegos preferidos. Esta información es importante para personalizar las recomendaciones. Es posible ajustar sus preferencias en cualquier momento desde la configuración de su perfil.



LOGIN SIGNUP

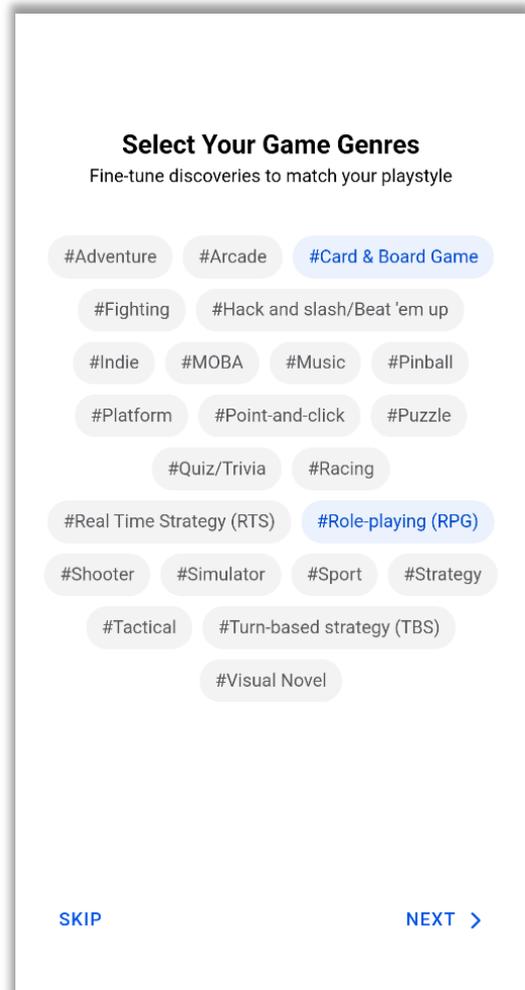
Name

Email

Password

CONTINUE

Figura 30. Pantalla de login/signup



Select Your Game Genres
Fine-tune discoveries to match your playstyle

#Adventure #Arcade **#Card & Board Game**

#Fighting #Hack and slash/Beat 'em up

#Indie #MOBA #Music #Pinball

#Platform #Point-and-click #Puzzle

#Quiz/Trivia #Racing

#Real Time Strategy (RTS) **#Role-playing (RPG)**

#Shooter #Simulator #Sport #Strategy

#Tactical #Turn-based strategy (TBS)

#Visual Novel

SKIP NEXT >

Figura 29. Pantalla de onboarding

3. Pantalla principal (Home)

Una vez configuradas sus preferencias, la pantalla principal mostrará una selección de videojuegos ajustada a sus gustos. Los juegos están organizados en listas según los géneros que tenga seleccionados.

- Pestaña "All": Aquí encontrará todos los juegos recomendados, ordenados por género.
- Pestaña "New": Al seleccionar esta pestaña, se mostrarán exclusivamente los lanzamientos más recientes dentro de los géneros de su interés.

4. Pantalla de detalle de juego

Al seleccionar un juego desde cualquier parte de la aplicación, accederá a su pantalla de detalle. Aquí se presenta información sobre el juego:

Descripción, rating, tiempo medio para completarlo, una lista de juegos similares, etc. Además, en esta pantalla encontrará un botón para añadir o eliminar el juego de favoritos.

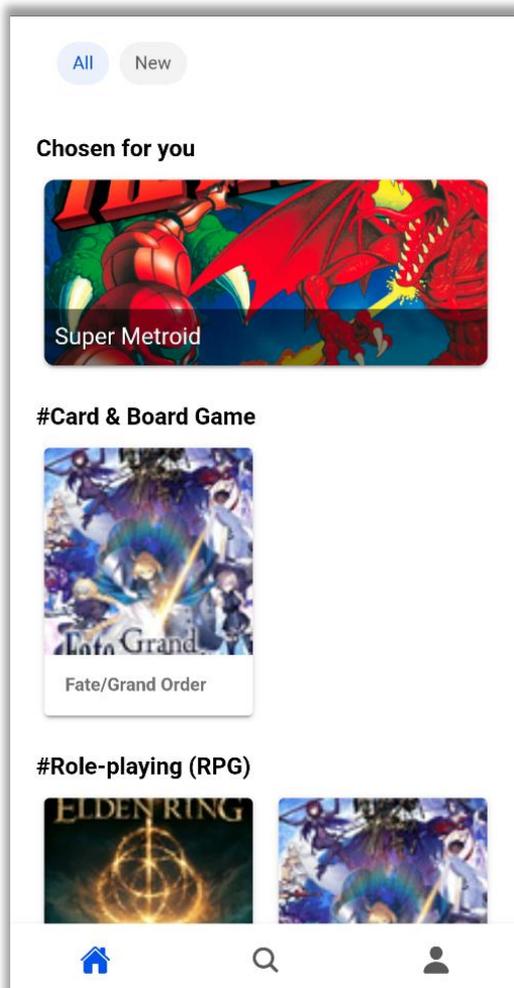


Figura 32. Pantalla principal

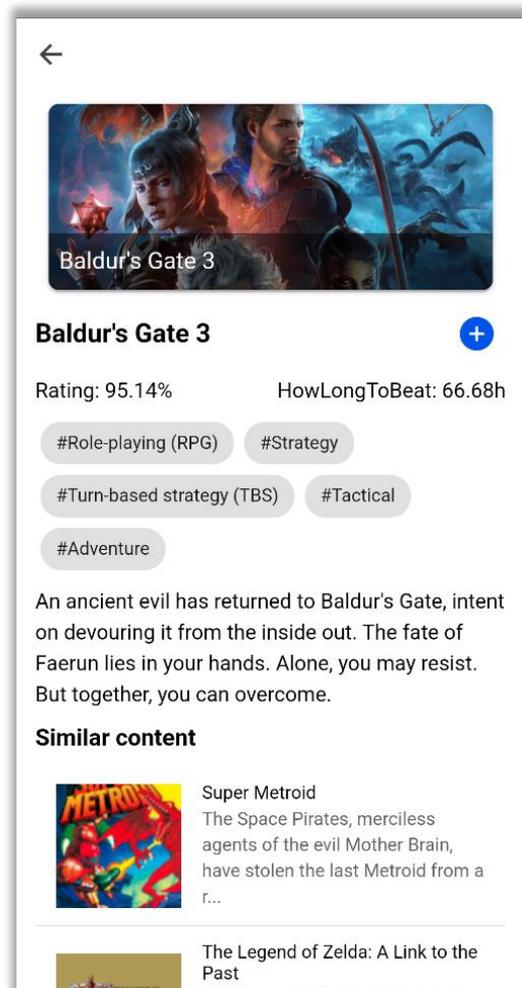


Figura 31. Pantalla de detalle de juego

5. Pantalla de búsqueda

La pantalla de búsqueda permite una exploración más específica a través de un listado de géneros y un campo de búsqueda

- Listado de géneros: Visualice y seleccione de una lista completa de géneros de videojuegos para explorar títulos específicos dentro de esas categorías.
- Campo de búsqueda: Introduzca el nombre de un juego o parte de él para buscar directamente juegos que coincidan con su consulta.

La búsqueda se puede hacer mediante dos pestañas que permiten filtrar los resultados:

- Por nombre de juego: Muestra los juegos que coinciden con las palabras de la búsqueda.
- Por género: Filtra los juegos que corresponden al género seleccionado introducido en la búsqueda.

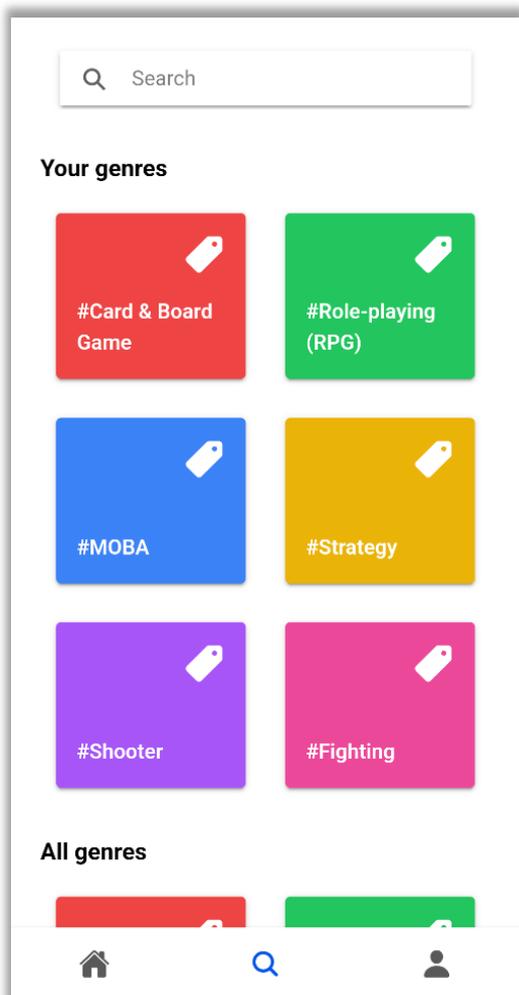


Figura 34. Pantalla de búsqueda

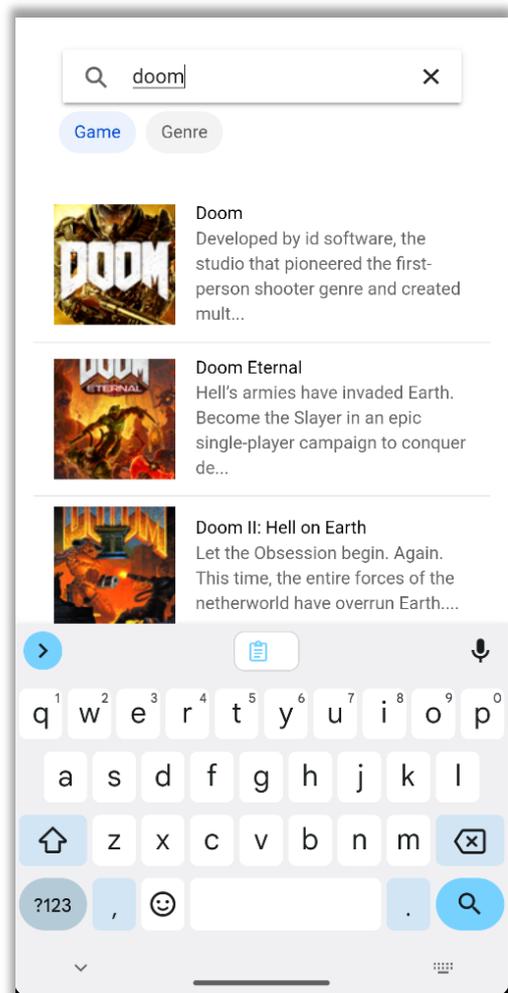


Figura 33. Pantalla de resultado de búsqueda

6. Pantalla de género

Al seleccionar un género, accederá a una pantalla que muestra todos los juegos asociados a ese género. En esta pantalla puede interactuar con los juegos de las siguientes maneras:

- Gestión de favoritos: Deslice el elemento de la lista hacia la izquierda (swipe) para añadir o eliminar juegos de favoritos. Esta acción está disponible en los listados a lo largo de la aplicación.
- Botón de género favorito: En la parte superior de la pantalla, encontrará un botón que le permite añadir o eliminar el género de sus favoritos.

7. Pantalla de perfil

Aquí puede revisar y modificar sus preferencias de usuario:

- Consultar preferencias: Visualice las preferencias actuales, incluyendo los géneros de videojuegos favoritos y los títulos que ha añadido a su lista de favoritos.
- Modificar preferencias: Cambie sus géneros favoritos o actualice la lista de juegos favoritos. Así, permite que la aplicación se adapte continuamente a sus intereses cambiantes y le ofrezca las recomendaciones más relevantes.

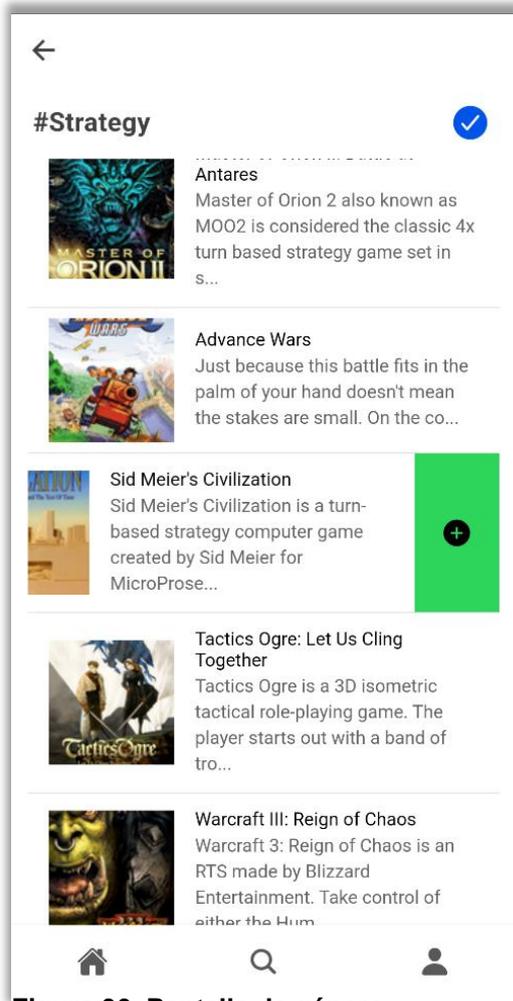


Figura 36. Pantalla de género

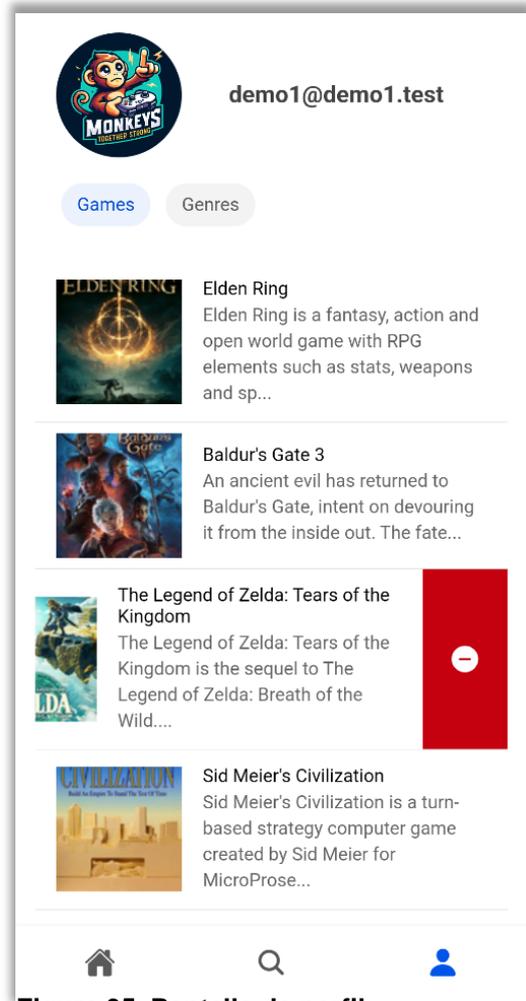


Figura 35. Pantalla de perfil