



UNIVERSIDAD DE MÁLAGA



INGENIERÍA INFORMÁTICA

**GAMINGHUB: UN ESPACIO CENTRALIZADO PARA TUS
BIBLIOTECAS DE JUEGOS**

**GAMINGHUB: A CENTRALICED SPACE FOR YOUR GAME
LIBRARIES**

Realizado por
IGNACIO POSTIGO PÉREZ

Tutorizado por
EDUARDO GUZMÁN DE LOS RISCOS

Departamento
**LENGUAJE Y CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE MÁLAGA**

MÁLAGA, SEPTIEMBRE 2024



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
INFORMÁTICA
INGENIERÍA INFORMÁTICA

**GAMINGHUB: UN ESPACIO
CENTRALIZADO PARA TUS BIBLIOTECAS
DE JUEGOS**

**GAMINGHUB: A CENTRALICED SPACE
FOR YOUR GAME LIBRARIES**

Realizado por
Ignacio Postigo Pérez

Tutorizado por
Eduardo Guzmán de los Riscos

Departamento
Lenguaje y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, SEPTIEMBRE 2024



UNIVERSIDAD
DE MÁLAGA



Resumen

En estas últimas décadas, la industria del videojuego ha sufrido una enorme evolución, lo que ha traído al mercado numerosos juegos nuevos con características muy variadas y distintivas. No obstante, la diversidad de plataformas tanto físicas (consolas) como digitales (tiendas virtuales) también ha incrementado, dejando a los usuarios con bibliotecas muy divididas que dificultan mantener un control de sus juegos y de sus avances.

Este proyecto intenta centralizar la totalidad de los juegos del usuario, consiguiendo así unificar y organizar su biblioteca de videojuegos. Con este propósito, se ha creado una aplicación web que utiliza una diversa variedad de tecnologías interconectadas entre sí. Concretamente se ha utilizado la librería de JavaScript, React para la construcción del frontend. El framework de JavaScript, Express.js, para el backend y finalmente, una base de datos relacional en MySQL.

Palabras clave: React, Express.js, Node.js, JavaScript, MySQL, videojuego.

Abstract

In recent decades, the video game industry has undergone a massive evolution, bringing numerous new games to the market with very diverse and distinctive features. However, the diversity of both physical (consoles) and digital (virtual stores) platforms has also increased, leaving users with highly fragmented libraries, making it difficult to keep track of their games and progress.

This project aims to centralize the entirety of the user's games, thereby unifying and organizing their video game library. To achieve this, a web application has been developed using various interconnected technologies. Specifically, the JavaScript library React has been used for frontend development. The JavaScript framework Express.js for the backend, and finally, a relational database in MySQL.

Keywords: React, Express.js, Node.js, JavaScript, MySQL, video game.

Índice

Resumen	1
Abstract	1
Índice.....	1
Introducción.....	3
1.1 Motivación.....	3
1.2 Objetivos.....	4
1.3 Antecedentes.....	4
1.4 Organización del documento.....	6
Tecnologías y herramientas utilizadas.....	7
2.1 HTML (HyperText Markup Language).....	7
2.2 CSS (Cascading Style Sheets) + Tailwind.....	8
2.3 JavaScript.....	8
2.4 Node.js.....	9
2.5 MySQL + MySQL Workbench 8.0	9
2.6 Bcrypt.js	10
2.7 JWT (JSON Web Tokens).....	10
2.8 Visual Studio Code + ESLint.....	11
2.9 Vite	11
2.10 React.....	12
2.11 Express.....	13
2.11 Git + GitHub.....	13
2.13 Postman.....	14
2.14 Jest.....	14
2.15 Algunas librerías destacables.....	14
Especificación y Análisis.....	17
3.1 Actores del sistema	17
3.2 Requisitos del sistema	19
3.3 Casos de uso.....	22
Diseño del sistema.....	77
4.1 Estructura general.....	77
4.2 Diagrama de arquitectura.....	78
4.3 Diseño de la base de datos	80
Implementación y pruebas.....	85
5.1 Primeros pasos	85

5.1.1	Instalación de Node.js y NPM.....	85
5.1.2	Instalación de MySQL y MySQL Workbench	86
5.1.3	Instalación de Postman.....	86
5.1.4	Instalación Git	86
5.1.5	Creación del repositorio de GitHub	87
5.1.6	Registro en Cloudfire.....	87
5.2	Estructura	87
5.2.1	Estructura general	87
5.2.2	Seguridad	90
5.3	Metodología de trabajo.....	93
5.4	Pruebas.....	94
5.4.1	Importancia de las pruebas.....	94
5.4.2	Tipos de pruebas.....	95
5.4.3	Pruebas del servidor y la base de datos.....	96
5.4.4	Pruebas del cliente.....	100
	Conclusiones y Trabajos Futuros	103
6.1	Conclusiones	103
6.2	Líneas Futuras.....	105
6.2.1	Despliegue en la web.....	105
6.2.2	Soporte para múltiples idiomas	105
6.2.3	Integración multiplataforma.....	105
6.2.4	Mejora en la búsqueda de juegos	106
6.2.5	Ampliación de funcionalidades sociales.....	106
6.2.6	Tener una base de datos de juegos propia.....	106
6.2.7	Mejoras en seguridad	107
6.2.8	Implementación de modo oscuro.....	107
	Referencias	109
	Manual de Usuario.....	113
A.1	Páginas accesibles por todos los usuarios.....	113
A.2	Perfil de usuario registrado	117
A.3	Biblioteca de un usuario.....	122
A.4	Juego de un usuario.....	125
A.5	Panel de administrador.....	128
	Manual de Instalación	131

1

Introducción

1.1 Motivación

Desde la aparición del primer videojuego comercial en 1972 [1] la industria del videojuego no ha dejado de evolucionar, hasta convertirse en una de las más importantes, alcanzando un valor de 245.900 millones de dólares [2].

Con el desarrollo de las tecnologías y la informática, los efectos gráficos y la fluidez jugable fueron mejorando, moldeando así a jugadores cada vez más críticos y selectivos, hasta el punto de que, en la actualidad, existen incluso premios oficiales, como los Game Awards [3], en los que galardonan a videojuegos en función de diferentes ámbitos de este, como: su narrativa, diseño artístico, banda sonora e incluso, por su mensaje.

Si a este aumento de jugadores críticos, le sumamos la aparición de numerosas plataformas físicas, consolas como la PlayStation o Xbox, y digitales, como Steam o Epic Games Store, obtenemos un continuo flujo de compras e información que dificultan al jugador mantener un orden y conocimiento sobre los juegos que tiene en su poder.

Con este proyecto, se pretende dar solución a esta problemática, creando una plataforma para unificar todas las bibliotecas o colecciones del usuario en un solo lugar digital centralizado, además, de poder clasificarlos y ordenarlos de diversas formas tales como: por orden alfabético, plataforma, estado e incluso por etiquetas.

Además de unificar las bibliotecas, la plataforma también ofrece diversos parámetros para establecer el seguimiento tales como horas jugadas, calificaciones, comentarios a modo de notas personales o progreso del título.

La aplicación también ofrecerá un factor social con el que poder compartir información sobre los juegos del usuario, incluso información del propio usuario mostrando un perfil de jugador con sus propias etiquetas y estadísticas.

1.2 Objetivos

El objetivo de este proyecto es ofrecer a los jugadores una plataforma para gestionar su biblioteca de juegos que unifique tanto aquellos de plataformas físicas como digitales. Gracias a esto, podrán tener en una sola página su catálogo completo de videojuegos ordenados mediante nombres, plataformas, estados y etiquetas de uso propio, donde podrán determinar: puntuación, género, etc.

Esta aplicación web está destinada a personas que manejan una amplia colección de juegos de diferentes orígenes. Además, ofrece la posibilidad de una experiencia social, al buscar a otros usuarios en función del tipo de juegos que suban a la plataforma y poder visualizar sus perfiles.

1.3 Antecedentes

El principal motivo de existencia de este proyecto es la necesidad de centralizar los videojuegos de múltiples plataformas en una sola y la libertad de poder editar de forma cómoda y eficaz diferentes parámetros con el fin de facilitar dicho proceso de organización.

Es fácil encontrar webs en las que puedas crear manualmente listas de juegos, categorizarlas y de algún modo organizarlas, pero, son muy tediosas, no aportan una ayuda excesiva y comúnmente, no aportan un factor social enfocado en el sector de los videojuegos.

Sin embargo, existe una aplicación web que se asemeja a esta idea y es "HowLongToBeat" [4]. En dicha aplicación web es posible tener colecciones de videojuegos de una forma simple, ofreciendo la posibilidad de añadir los videojuegos eligiendo su plataforma física y/o tienda digital. El principal problema es el objetivo que persigue, HowLongToBeat busca ofrecer un recopilatorio de videojuegos en el que se muestra su duración aproximada aportada por los usuarios (como su propio nombre indica, siendo su traducción al castellano "Cuánto se tarda en ganar"), dividiéndolos en consolas, historia principal, logros, etc. pero a su vez, no te permite moldear demasiado el videojuego a gusto del usuario.

Debido a su gran similitud, se procederá a listar una serie de puntos fuertes y débiles:

1. Puntos fuertes:

- a. Gran catálogo de juegos.
- b. Posibilidad de almacenar los juegos en el perfil a modo de biblioteca.
- c. Diferenciación por plataformas digitales y físicas.
- d. Sistema de amigos.
- e. Foros de discusión.
- f. Buscador eficiente.
- g. Permite introducir información personal.
- h. Permite añadir identificadores de usuario de diversas plataformas.

2. Puntos débiles:

- a. No es posible editar datos de los videojuegos.
- b. No es posible añadir ni filtrar por etiquetas.
- c. Poco intuitiva.
- d. El factor social se centra en conversaciones y no tanto en mostrar los juegos y gustos personales.

En términos generales, a pesar de distar de lo que se desea cumplir con este proyecto, sería una alternativa aceptable, ya que cuenta con funciones interesantes y una aplicación web sólida y competitiva.

La intención de este proyecto no es competir con HowLongToBeat, pues el tiempo de desarrollo es muy limitado, además de no perseguir el mismo objetivo, pero es útil para recabar ideas sobre el proyecto.

1.4 Organización del documento

Este documento está estructurado de la siguiente manera para facilitar su lectura y comprensión:

1. **Introducción:** Se expone la motivación del proyecto, los objetivos perseguidos y los antecedentes que justifican su desarrollo.
2. **Tecnologías y herramientas utilizadas:** Se describen las principales tecnologías empleadas para el desarrollo de la plataforma, junto con las herramientas auxiliares utilizadas durante el proceso.
3. **Especificación y análisis:** Se detallan los actores involucrados en el sistema, los requisitos funcionales y no funcionales, y los casos de uso que guían la interacción con la aplicación.
4. **Diseño del sistema:** Incluye la estructura general del sistema, el diagrama de arquitectura y el diseño de la base de datos.
5. **Implementación y pruebas:** Expone el proceso de desarrollo del sistema y las pruebas realizadas para asegurar su correcto funcionamiento.
6. **Conclusiones y trabajos futuros:** Presenta un resumen de los resultados obtenidos y las posibles mejoras o extensiones futuras del proyecto.
7. **Anexos:**
 - **Anexo A: Manual de usuario:** Proporciona instrucciones detalladas sobre cómo utilizar y navegar por la plataforma.
 - **Anexo B: Manual de instalación:** Explica el proceso paso a paso para instalar y configurar la aplicación en un entorno local.

2

Tecnologías y herramientas utilizadas

En la elaboración de este proyecto se han usado principalmente HTML, Express, MySQL, React, y CSS. Además, se ha utilizado otras tecnologías auxiliares como webs de alojamiento, visual estudio, git, postman, etc. las cuales se encuentran a continuación.

2.1 HTML (HyperText Markup Language)

HTML es el lenguaje más utilizado para estructurar el contenido en la web. Es fundamental en la creación de páginas web, ya que define la estructura básica y el contenido que se muestra en el navegador. Cada elemento de la aplicación, desde los encabezados y párrafos hasta las imágenes y enlaces, está compuesto utilizando etiquetas HTML, lo que permite organizar y presentar la información de manera coherente y accesible para el usuario.

Fue creado por Tim Berners-Lee en 1991 como la base para la creación de documentos en la World Wide Web [5]. A lo largo de los años, HTML ha evolucionado, y la versión actual, HTML5, es ahora el estándar para el desarrollo web [6].

2.2 CSS (Cascading Style Sheets) + Tailwind

CSS es un lenguaje de diseño gráfico que se encarga de dar estilo visual a la aplicación web. Permite definir estilos como colores, tipografías, espaciados y la disposición de los elementos en la página y todo lo relacionado con la estética. En este proyecto, además de CSS, se ha utilizado *Tailwind CSS*, un "utility-first CSS framework" de CSS que proporciona una serie de clases utilitarias predefinidas. [7]

Tailwind facilita la aplicación de estilos directamente en los elementos HTML a través de clases, lo que acelera el desarrollo y asegura consistencia en el diseño. A diferencia de otros frameworks CSS más tradicionales, Tailwind no impone un diseño específico, sino que ofrece flexibilidad total al desarrollador, aprovechando la potencia y simplicidad de las clases utilitarias.

2.3 JavaScript

JavaScript es un lenguaje de programación interpretado que se utiliza principalmente para crear contenido dinámico en aplicaciones web, lo que permite que las páginas web sean interactivas, esto permite que las acciones del usuario dentro de la página web sean procesadas en tiempo real mostrando así el contenido modificado al instante.

Es uno de los lenguajes de programación más populares y esenciales en el desarrollo web moderno.

Lo más interesante de JavaScript quizás sea el hecho de que puede usarse tanto para desarrollo frontend como backend, este proyecto es un claro ejemplo de ello.

2.4 Node.js

Node.js es un entorno de ejecución para JavaScript que permite utilizar este lenguaje en el lado del servidor. Está basado en el motor V8 de Google Chrome, lo que le proporciona velocidad y eficiencia en la ejecución del código.

Gracias a su arquitectura asíncrona y basada en eventos, Node.js es capaz de manejar múltiples solicitudes de manera simultánea sin bloquear el hilo principal, lo que lo convierte en una opción ideal para aplicaciones web de alta concurrencia. Esto permite que el servidor procese múltiples solicitudes de usuarios de manera rápida y eficiente [8].

En este proyecto, Node.js se ha utilizado para gestionar el backend de la aplicación. Además, junto con Node.js, se ha utilizado el framework Express, del cual se hablará en otro apartado de este documento [9].

2.5 MySQL + MySQL Workbench 8.0

MySQL es un sistema de gestión de bases de datos relacional que utiliza SQL (Structured Query Language) como su lenguaje principal para la gestión y consulta de datos. Es una de las bases de datos más populares para aplicaciones web, sobre todo si hablamos de bases de datos relacionales.

La elección de utilizar MySQL para este proyecto se debe a la naturaleza de los datos que se manejan en la misma, además de su integración fluida con Node.js, lo que permite que las operaciones en la base de datos se realicen de manera rápida y segura a través de la API desarrollada en el backend [10].

Además, para facilitar la administración de la base de datos, se ha utilizado *MySQL Workbench 8.0*, una herramienta visual que permite diseñar, modelar, administrar, y mantener bases de datos MySQL.

MySQL Workbench ofrece una interfaz gráfica de usuario (GUI) que simplifica a ejecución de consultas SQL entre muchas funcionalidades [11].

2.6 Bcrypt.js

Bcrypt.js es una biblioteca de JavaScript que se utiliza para aplicar hash a contraseñas, un proceso fundamental para garantizar la seguridad de las credenciales de los usuarios en una aplicación web. A diferencia de simplemente almacenar contraseñas en texto plano, bcrypt.js transforma las contraseñas en una serie de caracteres cifrados que son prácticamente imposibles de revertir, protegiendo así las contraseñas incluso si los datos se ven comprometidos.

Esta biblioteca utiliza el algoritmo bcrypt, que genera una "sal" aleatoria cada vez que se almacena una contraseña. Esta sal se combina con la contraseña antes de ser aplicado el hash, lo que asegura que incluso contraseñas idénticas no resulten en hashes iguales, incrementando significativamente la seguridad [12].

En este proyecto, bcrypt.js se ha utilizado para asegurar las contraseñas de los usuarios, garantizando que estas no puedan ser leídas ni utilizadas por personas no autorizadas.

2.7 JWT (JSON Web Tokens)

JWT es un estándar de autenticación que permite la transmisión segura de información entre el cliente y el servidor en formato JSON. Es especialmente útil en aplicaciones web sin estado (stateless), donde la autenticación del usuario se gestiona mediante tokens en lugar de sesiones almacenadas en el servidor.

Un token JWT está compuesto por tres partes: el encabezado (header), la carga útil (payload), y la firma (signature). El encabezado especifica el algoritmo de cifrado utilizado, la carga útil contiene la información que se desea transmitir (como el identificador del usuario y su rol), y la firma se utiliza para verificar que el token no ha sido alterado [13].

En este proyecto, JWT se ha utilizado para autenticar a los usuarios una vez que han iniciado sesión. Tras validar las credenciales del usuario (utilizando bcrypt para comparar la contraseña), el servidor genera un token JWT. Este token se almacena en el LocalStorage del navegador del cliente y tiene un tiempo de expiración asociado. Esto permite que el usuario pueda realizar solicitudes autenticadas sin necesidad de volver a iniciar sesión hasta que el token expire.

2.8 Visual Studio Code + ESLint

Visual Studio Code es un editor de código que destaca por su rendimiento, facilidad de uso, y su extenso marketplace de extensiones. Desarrollado por Microsoft, este editor soporta una amplia gama de lenguajes de programación y herramientas de desarrollo, lo que lo convierte en una opción ideal para el desarrollo de este proyecto, gracias a su integración fluida con tecnologías como Git, HTML, CSS, JavaScript, y Node.js.

En este proyecto, Visual Studio Code ha sido utilizado como el entorno principal de desarrollo debido a sus características avanzadas, como la integración de terminal, herramientas de depuración, control de versiones, y la posibilidad de instalar extensiones personalizadas que mejoran el flujo de trabajo.

Una de las extensiones clave utilizadas es *ESLint*, una herramienta que permite identificar y corregir problemas en el código JavaScript y JSX. ESLint ayuda a mantener un código limpio y libre de errores comunes, aplicando reglas de estilo y promoviendo buenas prácticas de programación. Además, proporciona retroalimentación en tiempo real a medida que se escribe el código. [14]

2.9 Vite

Vite es una herramienta de construcción y un servidor de desarrollo de nueva generación para proyectos web. Desarrollada por Evan You, el creador de Vue.js, Vite está diseñada para ofrecer una experiencia de desarrollo rápida y

eficiente, especialmente en aplicaciones modernas que utilizan frameworks como React.

A diferencia de otras herramientas de construcción tradicionales como Webpack, Vite aprovecha los módulos de ES (ESM) nativos en el navegador para proporcionar un entorno de desarrollo instantáneo, eliminando la necesidad de procesos de empaquetado costosos y lentos durante la fase de desarrollo. Esto permite a los desarrolladores trabajar de manera más ágil, con tiempos de inicio rápidos y recarga en caliente casi instantánea, lo que mejora significativamente el flujo de trabajo [15].

2.10 React

React es una biblioteca de JavaScript de código abierto desarrollada por Meta, diseñada para construir interfaces de usuario). React permite a los desarrolladores crear componentes de interfaz de usuario reutilizables, lo que facilita la construcción y el mantenimiento de interfaces complejas y dinámicas [16].

En este proyecto, React ha sido la herramienta principal para el desarrollo del frontend.

Una innovación clave de React es el *Virtual DOM* (Document Object Model Virtual), una representación ligera del DOM real que React utiliza para mejorar el rendimiento. En lugar de renderizar todo el DOM cada vez que hay un cambio, React compara el Virtual DOM con una versión previa, y solo actualiza los elementos que han cambiado. Esto reduce considerablemente el tiempo de renderizado y mejora la eficiencia de la aplicación [17].

React se integra perfectamente con otras herramientas y tecnologías utilizadas en este proyecto, como *Vite* para el entorno de desarrollo rápido, *Tailwind CSS* para

el estilizado de componentes, y *ESLint* para asegurar un código limpio y libre de errores.

2.11 Express

Express es un framework web minimalista para Node.js, diseñado para facilitar la construcción de aplicaciones y APIs web. Debido a su sencillez y flexibilidad, Express es una de las herramientas más populares para el desarrollo backend en aplicaciones web modernas [18].

En este proyecto, Express ha sido utilizado para gestionar el backend, actuando como el intermediario entre el frontend y la base de datos. Gracias a su arquitectura basada en middleware, Express permite manejar solicitudes HTTP de manera eficiente, organizando las rutas y controladores de la aplicación de forma modular. Esto facilita la gestión y mantenimiento del código, permitiendo que cada parte de la aplicación sea desarrollada y depurada de manera independiente.

2.11 Git + GitHub

Git es un sistema de control de versiones distribuido que permite a los desarrolladores realizar un seguimiento de los cambios en el código fuente a lo largo del tiempo. Es una herramienta esencial para gestionar el desarrollo de software, especialmente en proyectos colaborativos donde varios desarrolladores trabajan simultáneamente en diferentes partes del código. Git permite crear "commits" que registran el estado del código en un momento dado, facilitando la reversión a versiones anteriores en caso de errores y la comparación de cambios entre versiones [19].

GitHub, por su parte, es una plataforma de alojamiento para repositorios Git que proporciona herramientas adicionales para la colaboración y la gestión de proyectos. En este proyecto, GitHub ha sido utilizado para alojar el repositorio

del código fuente, facilitando el acceso remoto y el control de versiones mediante el plugin de Visual Studio Code.

El uso de Git y GitHub ha sido fundamental para asegurar un desarrollo controlado gracias a los “commits”, además, de tener una copia de seguridad del código en la nube [20].

2.13 Postman

Postman es una herramienta muy útil para el desarrollo y prueba de APIs, ampliamente utilizada por desarrolladores para facilitar la creación, prueba y documentación de APIs RESTful. Su interfaz intuitiva permite enviar solicitudes HTTP, configurar parámetros, encabezados y cuerpos de solicitud, y ver las respuestas del servidor de manera clara y organizada [21].

En este proyecto se ha utilizado para depurar errores, pudiendo detectar de forma eficaz si el error proviene de los datos backend o del cómo se tratan en el frontend.

2.14 Jest

Jest es un framework de pruebas de JavaScript desarrollado por Meta, diseñado para asegurar la calidad del código mediante la realización de pruebas unitarias, pruebas de integración y pruebas de extremo a extremo. Es una herramienta especialmente popular en proyectos que utilizan React, aunque también es compatible con otros entornos de JavaScript [22].

En este proyecto, Jest ha sido utilizado principalmente para realizar pruebas unitarias sobre el frontend.

2.15 Algunas librerías destacables

React Router Dom: Esta librería se utiliza para la navegación entre las diferentes vistas de la aplicación, permitiendo un enrutamiento dinámico y modular.

React Modal: Se emplea para la creación y gestión de modales en la aplicación, facilitando la implementación de ventanas emergentes.

React Select: Esta librería proporciona componentes selectores personalizables, mejorando la experiencia de usuario en formularios y otros elementos interactivos.

SweetAlert2: Se utiliza para mostrar alertas personalizadas y con mejor estilo visual que las alertas nativas del navegador.

Axios: Utilizado para realizar solicitudes HTTP tanto en el frontend como en el backend, facilitando la comunicación con APIs y la gestión de datos asíncronos.

Dotenv: Utilizado en el backend para la gestión de variables de entorno, asegurando que la configuración sensible no esté embebida en el código fuente.

DaisyUI: Plugin para Tailwind CSS que provee componentes preconstruidos y estilizados, acelerando el desarrollo de la interfaz de usuario.

3

Especificación y Análisis

3.1 Actores del sistema

Los actores del sistema son entidades que interactúan con el sistema para realizar una o varias funciones. Los actores son un concepto fundamental cuando hablamos de análisis y diseño del desarrollo software.

En este proyecto existen 3 actores claramente diferenciables, los cuales tienen una escala jerárquica bien marcada (Figura 1), a continuación, se procederá a explicar cómo puede interactuar cada actor con el sistema.

1) Invitado

- a) Son usuarios que no han iniciado sesión en GamingHub.
- b) Disponen de un buscador que les da acceso a perfiles y sus respectivas bibliotecas y juegos siempre y cuando el dueño las haya establecido su perfil como público.

- i) En caso de que los perfiles sean privados, podrán ver una información mínima solamente del perfil.
- c) Pueden registrarse y/o iniciar sesión para pasar a ser usuarios o administradores.

2) Usuario

- a) Son los usuarios que han iniciado sesión en GamingHub.
- b) Disponen de todos los privilegios que los invitados.
- c) Pueden modificar su perfil.
- d) Pueden elegir si son perfiles públicos o privados.
- e) Pueden enviar, aceptar y rechazar peticiones de amistad.
 - i) Pueden ver toda la información de un perfil privado si es amigo de dicho usuario.
- f) Puede buscar, añadir, eliminar y editar juegos de su biblioteca.
- g) Puede utilizar filtros de búsqueda y ordenación en su biblioteca.
- h) Pueden encontrar usuarios buscando por juegos o etiquetas asignadas a los juegos.
- i) Puede cerrar sesión.
- j) Puede eliminar su cuenta permanentemente.
 - i) Una cuenta eliminada puede ser registrada de nuevo, pero pierde los datos.

3) Administrador

- a) Puede hacer todo lo que hace un usuario y, por consiguiente, un invitado.
- b) Tiene acceso a todos los perfiles independientemente de si es privado o si no son amigos.
- c) Tiene acceso al panel de administrador.
 - i) Puede buscar usuarios, ver su id y eliminar permanentemente sus cuentas.
 - ii) Puede crear nuevas etiquetas o eliminar alguna actual.

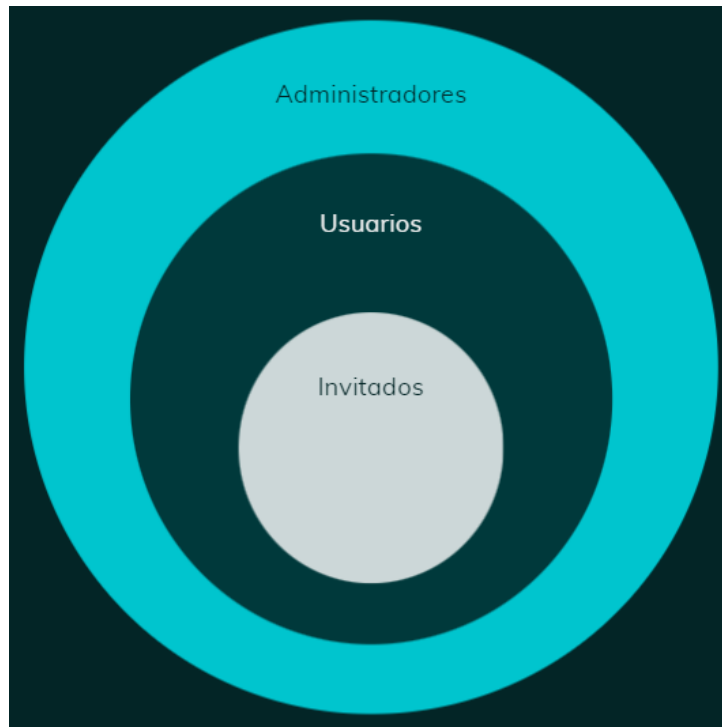


Figura 1: Diagrama de Venn sobre la jerarquía de permisos de los actores.

3.2 Requisitos del sistema

Los requisitos de un sistema son una descripción detallada de las funcionalidades y características que debe cumplir una aplicación o software para satisfacer las necesidades de sus usuarios y cumplir con los objetivos del proyecto. Estos requisitos sirven como base para el desarrollo, asegurando que todos los involucrados (desarrolladores, usuarios, clientes) tengan una comprensión clara y común de lo que se espera del sistema.

A continuación, se procederá a desglosar los requisitos del proyecto en cuestión:

3.2.1. Requisitos Funcionales

Dividiremos los requisitos funcionales en categorías según las funcionalidades principales de la aplicación.

1. Requisitos Funcionales para Invitados

RF 1.1. Un invitado debe poder acceder a la plataforma sin iniciar sesión.

RF 1.2. Un invitado debe poder buscar usuarios.

RF 1.3. Un invitado debe poder visualizar los perfiles públicos de otros usuarios.

RF 1.4. Un invitado debe poder visualizar las librerías de usuarios con perfiles públicos.

RF 1.5. Un invitado debe poder visualizar los juegos de las librerías de usuarios con perfiles públicos.

RF 1.6. Un invitado debe poder filtrar los juegos de las librerías de usuarios con perfiles públicos.

RF 1.7. Un invitado debe poder ordenar los juegos de las librerías de usuarios con perfiles públicos.

RF 1.8. Un invitado debe poder ver información básica de perfiles privados.

RF 1.9. Un invitado debe poder registrarse en la aplicación.

RF 1.10. Un invitado debe poder iniciar sesión en la aplicación.

2. Requisitos Funcionales para Usuarios

RF 2.1. Un usuario debe poder modificar su perfil (foto de perfil, fecha de nacimiento, país, descripción, etiquetas y redes sociales).

RF 2.2. Un usuario debe poder definir si su perfil es público o privado y modificarlo en cualquier momento.

RF 2.3. Un usuario debe poder enviar solicitudes de amistad.

RF 2.4. Un usuario debe poder aceptar solicitudes de amistad.

RF 2.5. Un usuario debe poder rechazar solicitudes de amistad.

RF 2.6. Un usuario debe poder cerrar sesión.

RF 2.7. Un usuario debe poder eliminar su cuenta permanentemente.

RF 2.8. Un usuario debe poder acceder a su biblioteca.

RF 2.9. Un usuario debe poder buscar juegos nuevos.

RF 2.10. Un usuario debe poder añadir juegos a su biblioteca.

RF 2.10.1 Un usuario debe poder elegir la plataforma del juego (de una lista o introducirla a mano).

RF 2.11. Un usuario debe poder eliminar juegos en su biblioteca.

RF 2.12. Un usuario debe poder editar los juegos de su biblioteca (estado, comentario, nota y porcentajes de historia, juego completo y logros).

RF 2.13. Un usuario debe poder buscar juegos dentro de su biblioteca (por título o etiqueta).

RF 2.14 Un usuario debe poder ordenar los juegos dentro de su biblioteca (por título, etiqueta o estado).

RF 2.15. Un usuario debe poder añadir a favoritos un juego de su biblioteca.

RF 2.16. Un usuario debe poder eliminar a favoritos un juego de su biblioteca.

RF 2.18. Un usuario debe poder añadir etiquetas al juego.

RF 2.19. Un usuario debe poder eliminar etiquetas al juego.

3. Requisitos Funcionales para Administradores

RF 3.1. Un administrador debe poder acceder a todos los perfiles, independientemente de su estado.

RF 3.2. Un administrador debe poder eliminar permanentemente cuentas de usuarios.

RF 3.3. Un administrador debe poder crear etiquetas.

RF 3.4. Un administrador debe poder eliminar etiquetas.

3.2.2. Requisitos No Funcionales

RNF 1. La base de datos será relacional y estará basada en SQL.

RNF 2. Las contraseñas se almacenarán de forma segura mediante algoritmos de cifrado.

3.3 Casos de uso

Los casos de uso son describen como interactúan los actores con el sistema con el fin de lograr un objetivo, en este caso, cumplir un requisito funcional.

Título	El nombre del caso de uso que describe brevemente la acción.
Actor	Las entidades que interactúan con el sistema, nombrados con anterioridad.
Descripción	Un resumen del propósito del caso de uso, indicando lo que el actor quiere lograr mediante la interacción con el sistema.
Precondiciones	Las condiciones que deben cumplirse antes de que el caso de uso pueda comenzar.
Postcondiciones	Los resultados que deben cumplirse una vez que el caso de uso ha terminado, independientemente de si la interacción fue exitosa o no.
Escenarios principales	La secuencia principal de pasos que realiza el actor para alcanzar su objetivo en condiciones normales, sin interrupciones ni errores.
Escenarios alternativos (Si procede)	Variaciones del flujo principal, incluyendo cómo manejar excepciones, errores o caminos alternativos.

3.3.1. Caso de uso para RF.1.1: Acceso a la plataforma sin iniciar sesión.

Título	Acceso a la plataforma sin iniciar sesión.
Actor	Invitado.
Descripción	Este caso de uso describe cómo un invitado puede acceder a la plataforma sin necesidad de iniciar sesión. El propósito es permitir que cualquier usuario potencial explore la plataforma, visualice contenido público sin la necesidad de crear una cuenta o autenticarse.
Precondiciones	El sistema debe estar disponible y accesible a través del navegador con la aplicación lanzada en local, o en su defecto, una URL pública.
Postcondiciones	El sistema debe estar disponible y accesible a través del navegador con la aplicación lanzada en local, o en su defecto, una URL pública.
Escenarios principales	<p>Escenario 1: Acceso a la página principal.</p> <ul style="list-style-type: none"> • El invitado abre un navegador web e ingresa la URL de la plataforma (por ejemplo, http://localhost:5173). • El sistema carga la página principal de la plataforma y la muestra al invitado. • El invitado visualiza la página principal con acceso a las secciones "Login", "Regístrate aquí", y "Buscar usuario".
Escenarios alternativos (Si procede)	Escenario alternativo 1: Error al cargar la plataforma.

	<ul style="list-style-type: none"> • Si el sistema encuentra un problema al cargar la página principal, se muestra un mensaje de error.
--	--

3.3.2. Caso de uso para RF.1.2: Buscar usuarios.

Título	Buscar usuarios.
Actor	Invitado.
Descripción	Este caso de uso describe cómo un invitado puede buscar perfiles de usuarios en la plataforma sin necesidad de iniciar sesión.
Precondiciones	El invitado debe haber accedido a la plataforma sin iniciar sesión (como se describe en RF 1.1).
Postcondiciones	El invitado puede ver al usuario que coincide con su búsqueda en caso de que exista.
Escenarios principales	<p>Escenario 1: Búsqueda de usuarios.</p> <ul style="list-style-type: none"> • El invitado accede a la barra de búsqueda en la página principal. • El invitado introduce un nombre. • El invitado es redirigido a la página del usuario encontrado.
<p>Escenarios alternativos</p> <p>(Si procede)</p>	<p>Escenario alternativo 1: Sin resultados.</p> <ul style="list-style-type: none"> • Si no hay coincidencias, el sistema muestra un mensaje indicando "¡Error! Usuario no encontrado". • El invitado pulsa en aceptar y vuelve a la página principal.

3.3.3. Caso de uso para RF.1.3: Visualizar perfiles públicos.

Título	Visualización de perfiles públicos.
Actor	Invitado.
Descripción	Este caso de uso describe cómo un invitado puede visualizar los perfiles públicos de otros usuarios.
Precondiciones	El invitado debe haber accedido a la plataforma y haber realizado una búsqueda de usuarios.
Postcondiciones	El invitado podrá ver el perfil público de un usuario.
Escenarios principales	Escenario 1: Visualización de un perfil público. <ul style="list-style-type: none">• El invitado busca un usuario existente.• El sistema muestra el perfil público del usuario seleccionado.• El invitado puede ver la información disponible en el perfil.
Escenarios alternativos (Si procede)	No procede.

3.3.4. Caso de uso para RF.1.4: Visualización de librerías de usuarios públicos.

Título	Visualización de librerías de usuarios públicos.
Actor	Invitado.
Descripción	Este caso de uso describe cómo un invitado puede visualizar las librerías de juegos de usuarios con perfiles públicos.

Precondiciones	El invitado debe estar visualizando un perfil público (como se describe en RF 1.3).
Postcondiciones	El invitado puede ver la lista de juegos en la librería del usuario.
Escenarios principales	<p>Escenario 1: Visualización de la librería de un usuario público.</p> <ul style="list-style-type: none"> • El invitado hace clic en la sección de librería del perfil del usuario. • El sistema muestra la lista de juegos en la librería del usuario. • El invitado puede ver los títulos y detalles básicos de los juegos en la librería.
Escenarios alternativos (Si procede)	<p>Escenario alternativo 1: Librería vacía.</p> <ul style="list-style-type: none"> • Si la librería está vacía, el sistema muestra un mensaje indicando que no hay juegos en la librería del usuario.

3.3.5. Caso de uso para RF.1.5: Visualización de juegos en las librerías de usuarios con perfiles públicos.

Título	Visualización de juegos en las librerías de usuarios con perfiles públicos.
Actor	Invitado.
Descripción	Este caso de uso describe cómo un invitado puede ver los detalles específicos de un juego dentro de la biblioteca de un usuario con un perfil público.

	Los detalles incluyen información como el estado del juego, la plataforma, la puntuación, los comentarios, y las etiquetas asociadas al juego.
Precondiciones	El invitado debe estar visualizando la biblioteca de un usuario con perfil público (como se describe en RF 1.4).
Postcondiciones	El invitado podrá ver información detallada sobre un juego específico, incluyendo progreso, puntuación, comentarios, y etiquetas.
Escenarios principales	<p>Escenario 1: Visualización de los detalles de un juego en la librería de un perfil público.</p> <ul style="list-style-type: none"> • El invitado accede al perfil de un usuario público. • El invitado selecciona la opción para ver la librería de juegos del usuario. • El sistema muestra la lista de juegos en la librería del usuario. • El invitado selecciona un juego específico de la lista. • El sistema muestra los detalles del juego, que incluyen: <ul style="list-style-type: none"> Plataforma, Estado, Puntuación,

	<p>Comentarios,</p> <p>Historia completada,</p> <p>Juego completado,</p> <p>Logros completados,</p> <p>Horas totales jugadas,</p> <p>Etiquetas.</p>
<p>Escenarios alternativos</p> <p>(Si procede)</p>	No procede.

3.3.6. Caso de uso para RF.1.6: Filtrar juegos en las librerías de usuarios con perfiles públicos.

Título	Filtrar juegos en las librerías de usuarios con perfiles públicos.
Actor	Invitado.
Descripción	Este caso de uso describe cómo un invitado puede filtrar los juegos dentro de la biblioteca de un usuario con un perfil público. El invitado puede aplicar filtros por título o etiquetas.
Precondiciones	El invitado debe estar visualizando la biblioteca de un usuario con perfil público (como se describe en RF 1.4).
Postcondiciones	El invitado podrá ver una lista filtrada de juegos en función de los criterios seleccionados (por título o etiqueta).
Escenarios principales	Escenario 1: Filtrado de juegos por título.

	<ul style="list-style-type: none"> • El invitado accede al perfil de un usuario público. • El invitado selecciona la opción para ver la librería de juegos del usuario. • El sistema muestra la lista de juegos en la librería del usuario. • El invitado utiliza la barra de búsqueda para introducir el título o parte del título de un juego. • El sistema filtra los juegos en función del texto introducido y muestra solo aquellos juegos que coincidan con el criterio de búsqueda. <p>Escenario 2: Filtrado de juegos por etiqueta.</p> <ul style="list-style-type: none"> • El invitado accede a la librería de juegos de un usuario público. • El sistema muestra la lista completa de juegos. • El invitado selecciona una etiqueta de la lista de etiquetas asociadas a los juegos. • El sistema filtra la lista de juegos para mostrar únicamente los juegos que tienen asociada la etiqueta seleccionada.
<p>Escenarios alternativos (Si procede)</p>	<p>Escenario alternativo 1: No coincide la búsqueda.</p>

	<ul style="list-style-type: none"> • Si no se encuentran juegos que coincidan con los criterios de filtrado, el sistema mostrará un mensaje indicando que no hay resultados disponibles para los criterios seleccionados. • Si el perfil del usuario cambia a privado mientras el invitado está aplicando un filtro, el acceso a la información será restringido y se mostrará un mensaje indicando que la información ya no está disponible.
--	---

3.3.7. Caso de uso para RF.1.7: Ordenar juegos en las librerías de usuarios con perfiles públicos.

Título	Ordenar juegos en las librerías de usuarios con perfiles públicos.
Actor	Invitado.
Descripción	Este caso de uso describe cómo un invitado puede ordenar los juegos dentro de la biblioteca de un usuario con un perfil público. El invitado puede aplicar criterios de ordenación como el nombre, la plataforma, o el estado del juego.
Precondiciones	El invitado debe estar visualizando la biblioteca de un usuario con perfil público (como se describe en RF 1.4).

Postcondiciones	El invitado podrá ver una lista ordenada de juegos en función de los criterios seleccionados (por nombre, plataforma o estado).
Escenarios principales	<p data-bbox="735 421 1310 454">Escenario 1: Ordenar juegos por nombre</p> <ul data-bbox="783 506 1433 1211" style="list-style-type: none"> <li data-bbox="783 506 1433 607">• El invitado accede al perfil de un usuario público. <li data-bbox="783 658 1433 759">• El invitado selecciona la opción para ver la librería de juegos del usuario. <li data-bbox="783 810 1433 911">• El sistema muestra la lista de juegos en la librería del usuario. <li data-bbox="783 963 1433 1064">• El invitado selecciona la opción para ordenar los juegos por nombre. <li data-bbox="783 1115 1433 1211">• El sistema reordena la lista de juegos en orden alfabético por nombre. <p data-bbox="735 1263 1355 1296">Escenario 2: Ordenar juegos por plataforma</p> <ul data-bbox="783 1348 1433 1968" style="list-style-type: none"> <li data-bbox="783 1348 1433 1449">• El invitado accede a la librería de juegos de un usuario público. <li data-bbox="783 1500 1433 1601">• El sistema muestra la lista completa de juegos. <li data-bbox="783 1653 1433 1753">• El invitado selecciona la opción para ordenar los juegos por plataforma. <li data-bbox="783 1805 1433 1968">• El sistema reordena la lista de juegos en orden alfabético del nombre de la plataforma.

	<p>Escenario 3: Ordenar juegos por estado</p> <ul style="list-style-type: none"> • El invitado accede a la librería de un usuario público. • El sistema muestra la lista completa de juegos. • El invitado selecciona la opción para ordenar los juegos por estado (completado, en progreso, pendiente). • El sistema reordena la lista de juegos en función del estado seleccionado.
<p>Escenarios alternativos (Si procede)</p>	<p>Escenario alternativo 1: Librería vacía.</p> <ul style="list-style-type: none"> • Si la librería está vacía, el sistema muestra un mensaje indicando que no hay juegos en la librería del usuario.

3.3.8. Caso de uso para RF.1.8: Ver información básica de perfiles privados

Título	Visualización de información básica de perfiles privados.
Actor	Invitado.
Descripción	Este caso de uso describe cómo un invitado puede ver solo la información básica de un perfil privado en la plataforma. La información disponible será limitada en comparación con los perfiles públicos.
Precondiciones	El invitado debe haber accedido a la plataforma y haber realizado una búsqueda de usuarios. El

	perfil del usuario encontrado debe estar configurado como privado.
Postcondiciones	El invitado podrá ver únicamente la información básica del perfil privado, como el nombre de usuario y una descripción limitada.
Escenarios principales	<p>Escenario 1: Visualización de un perfil privado.</p> <ul style="list-style-type: none"> • El invitado busca un usuario a través de la barra de búsqueda. • El sistema encuentra el perfil del usuario, que está configurado como privado. • El sistema muestra la versión básica del perfil, que incluye información limitada.
Escenarios alternativos (Si procede)	No procede.

3.3.9. Caso de uso para RF.1.9: Registro en la aplicación.

Título	Registro en la aplicación.
Actor	Invitado.
Descripción	Este caso de uso describe cómo un invitado puede registrarse en la aplicación creando una nueva cuenta mediante un formulario de registro que incluye nombre de usuario, correo electrónico,

	contraseña, confirmación de contraseña, país y fecha de nacimiento.
Precondiciones	El invitado debe estar en la página principal de la plataforma y debe tener acceso al formulario de registro.
Postcondiciones	El invitado se convertirá en un usuario registrado y podrá iniciar sesión en la aplicación.
Escenarios principales	<p>Escenario 1: Proceso de registro.</p> <ul style="list-style-type: none"> • El invitado accede a la página principal de la plataforma. • El invitado hace clic en la opción "Regístrate aquí". • El sistema muestra un formulario de registro solicitando los siguientes datos: <ul style="list-style-type: none"> ○ Nombre de usuario. ○ Correo electrónico. ○ Contraseña. ○ Confirmación de contraseña. ○ País (mediante un menú desplegable). ○ Fecha de nacimiento. • El invitado completa el formulario con toda la información solicitada. • El invitado hace clic en "Registrar".

	<ul style="list-style-type: none"> • El sistema valida la información proporcionada. • Si la información es válida, el sistema crea una nueva cuenta para el invitado. • El sistema confirma el registro y muestra un mensaje indicando que la cuenta ha sido creada exitosamente.
<p>Escenarios alternativos (Si procede)</p>	<p>Escenario alternativo 1: Error en el registro, datos en uso.</p> <ul style="list-style-type: none"> • Si el nombre de usuario o el correo electrónico ya están en uso, el sistema muestra un mensaje indicando que debe ingresar un nombre de usuario o correo electrónico diferente. • El invitado corrige la información y vuelve a enviar el formulario. <p>Escenario alternativo 2: Error en el registro, las contraseñas no coinciden.</p> <ul style="list-style-type: none"> • Si las contraseñas no coinciden, el sistema muestra un mensaje indicando el error. • El invitado corrige la información y vuelve a enviar el formulario. <p>Escenario alternativo 3: Error en el registro, el correo no tiene formato de correo.</p>

	<ul style="list-style-type: none"> • Si, el correo no tiene formato de correo, el sistema muestra un mensaje indicando el error. • El invitado corrige la información con un correo válido y vuelve a enviar el formulario. <p>Escenario alternativo 4: Información incompleta.</p> <ul style="list-style-type: none"> • Si el invitado no completa todos los campos obligatorios, el sistema muestra un mensaje de advertencia solicitando que complete la información faltante antes de continuar.
--	---

3.3.10. Caso de uso para RF.1.10: Iniciar sesión en la aplicación.

Título	Iniciar sesión en la aplicación.
Actor	Invitado.
Descripción	Este caso de uso describe cómo un invitado puede iniciar sesión en la aplicación proporcionando sus credenciales de usuario registrado (usuario/correo electrónico y contraseña).
Precondiciones	El invitado debe haber creado previamente una cuenta en GamingHub.
Postcondiciones	El invitado se convierte en un usuario autenticado y podrá acceder a las funcionalidades disponibles para usuarios registrados.
Escenarios principales	Escenario 1: Proceso de inicio de sesión.

	<ul style="list-style-type: none"> • El invitado accede a la página principal de la plataforma. • El invitado hace clic en la opción "Login". • El sistema muestra un formulario de inicio de sesión solicitando las siguientes credenciales: <ul style="list-style-type: none"> ◦ Usuario o correo electrónico. ◦ Contraseña. • El invitado completa el formulario con sus credenciales y hace clic en "Iniciar sesión". • El sistema verifica las credenciales proporcionadas. • Si las credenciales son correctas, el sistema autentica al usuario y redirige al perfil de su cuenta.
<p>Escenarios alternativos (Si procede)</p>	<p>Escenario alternativo 1: Credenciales incorrectas.</p> <ul style="list-style-type: none"> • Si el correo electrónico o la contraseña son incorrectos, el sistema muestra un mensaje de error indicando que las credenciales no coinciden con ninguna cuenta registrada. • El invitado tiene la opción de volver a intentar introducir las credenciales correctas o de solicitar el restablecimiento de la contraseña.

3.3.11. Caso de uso para RF.2.1: Modificar el perfil.

Título	Modificar el perfil.
Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede modificar su perfil en la plataforma. Las modificaciones incluyen la actualización de la foto de perfil, fecha de nacimiento, país, descripción, etiquetas y redes sociales.
Precondiciones	El usuario debe estar autenticado y haber accedido a la página de su perfil.
Postcondiciones	El usuario podrá ver los cambios reflejados en su perfil tras realizar la modificación.
Escenarios principales	<p>Escenario 1: Modificación de la foto de perfil.</p> <ul style="list-style-type: none"> • El usuario accede a su perfil y hace clic en "Actualizar imagen de perfil". • El sistema permite al usuario seleccionar una nueva imagen desde su dispositivo. • El usuario selecciona la imagen y confirma. • El sistema carga la nueva imagen de perfil y la muestra actualizada en la página de perfil del usuario. <p>Escenario 2: Modificación de la fecha de nacimiento.</p> <ul style="list-style-type: none"> • El usuario hace clic en "Modificar fecha de nacimiento".

	<ul style="list-style-type: none">• El sistema permite al usuario cambiar la fecha de nacimiento seleccionando una nueva fecha.• El usuario selecciona la fecha y confirma.• El sistema actualiza la fecha de nacimiento.• El sistema actualiza la edad en el perfil del usuario. <p>Escenario 3: Modificación del país.</p> <ul style="list-style-type: none">• El usuario hace clic en "Modificar país".• El sistema muestra un menú desplegable con opciones de países.• El usuario selecciona el país y confirma.• El sistema actualiza el país en el perfil del usuario. <p>Escenario 4: Modificación de la descripción.</p> <ul style="list-style-type: none">• El usuario hace clic en "Modificar descripción".• El sistema muestra un campo de texto editable donde el usuario puede escribir su nueva descripción.• El usuario escribe la nueva descripción y confirma.
--	--

	<ul style="list-style-type: none"> • El sistema actualiza la descripción en el perfil del usuario. <p>Escenario 5: Modificación de las etiquetas.</p> <ul style="list-style-type: none"> • El usuario hace clic en "Modificar etiquetas". • El sistema permite al usuario agregar o eliminar etiquetas relacionadas con su perfil. • El usuario añade o elimina las etiquetas y confirma. • El sistema actualiza las etiquetas en el perfil del usuario. <p>Escenario 6: Modificación de redes sociales.</p> <ul style="list-style-type: none"> • El usuario hace clic en "Redes Sociales". • El sistema permite al usuario agregar o modificar enlaces a sus redes sociales. • El usuario modifica o añade sus redes sociales y confirma. • El sistema actualiza las redes sociales en el perfil del usuario.
<p>Escenarios alternativos</p> <p>(Si procede)</p>	<ul style="list-style-type: none"> • No procede.

3.3.12. Caso de uso para RF.2.2: Definir si el perfil es público o privado.

Título	Definir si el perfil es público o privado.
--------	--

Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede definir si su perfil es público o privado, permitiendo que otros usuarios vean o no la información detallada de su perfil, y cómo puede modificar esta configuración en cualquier momento.
Precondiciones	El usuario debe estar autenticado y haber accedido a la página de su perfil.
Postcondiciones	El perfil del usuario será accesible públicamente o estará restringido según la configuración seleccionada.
Escenarios principales	<p>Escenario 1: Definir el perfil como público o privado.</p> <ul style="list-style-type: none"> • El usuario accede a su perfil. • El sistema muestra la opción de marcar el perfil como "Público" o "Privado" mediante una "checkbox". • El usuario selecciona la opción deseada. • El sistema guarda la configuración seleccionada y actualiza el estado del perfil del usuario.
Escenarios alternativos (Si procede)	No procede.

3.3.13. Caso de uso para RF.2.3: Enviar solicitudes de amistad.

Título	Enviar solicitudes de amistad.
Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede enviar una solicitud de amistad a otro usuario en la plataforma. La solicitud quedará pendiente hasta que el destinatario la acepte o rechace.
Precondiciones	<p>El usuario debe estar autenticado.</p> <p>El usuario debe haber accedido al menú de amigos.</p> <p>El usuario debe buscar al usuario que quiere agregar en el buscador de amigos.</p>
Postcondiciones	La solicitud de amistad será enviada y quedará pendiente de aceptación o rechazo por parte del usuario receptor.
Escenarios principales	<p>Escenario 1: Enviar una solicitud de amistad.</p> <ul style="list-style-type: none"> • El usuario accede al menú de amigos y selecciona la opción "Buscar Amigos". • El sistema muestra un buscador de amigos. • El usuario introduce el nombre del usuario que desea agregar. • El sistema muestra el listado de usuarios que coinciden. • El usuario hace clic en el botón "Enviar solicitud de amistad" del usuario deseado.

	<ul style="list-style-type: none"> El sistema confirma que la solicitud ha sido enviada y muestra un mensaje indicando que está pendiente de aceptación.
Escenarios alternativos (Si procede)	<p>Escenario alternativo 1: El usuario ya es amigo.</p> <ul style="list-style-type: none"> Si el usuario ya es amigo del destinatario, el sistema no muestra al usuario en la lista

3.3.14. Caso de uso para RF.2.4: Aceptar solicitudes de amistad.

Título	Aceptar solicitudes de amistad.
Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede aceptar una solicitud de amistad recibida de otro usuario en la plataforma. Al aceptar la solicitud, ambos usuarios se convierten en amigos y pueden acceder a funcionalidades adicionales disponibles para amigos.
Precondiciones	<p>El usuario debe estar autenticado.</p> <p>El usuario debe haber recibido al menos una solicitud de amistad.</p> <p>El usuario debe haber accedido al menú de solicitudes de amistad.</p>
Postcondiciones	<p>La solicitud de amistad será aceptada.</p> <p>Ambos usuarios aparecerán en la lista de amigos mutuamente.</p>

Escenarios principales	<p>Escenario 1: Aceptar una solicitud de amistad.</p> <ul style="list-style-type: none"> • El usuario accede al menú de solicitudes de amistad. • El sistema muestra una lista de todas las solicitudes de amistad pendientes. • El usuario hace clic en el botón "Aceptar solicitud". • El sistema actualiza el estado de la solicitud a "Aceptada". • El sistema añade al solicitante a la lista de amigos del usuario y viceversa. • El sistema muestra un mensaje confirmando que la solicitud ha sido aceptada.
Escenarios alternativos (Si procede)	No procede.

3.3.15. Caso de uso para RF.2.5: Rechazar solicitudes de amistad.

Título	Rechazar solicitudes de amistad.
Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede rechazar una solicitud de amistad recibida de otro usuario en la plataforma. Al rechazar la solicitud,

	esta se elimina y el usuario solicitante no aparecerá en la lista de amigos.
Precondiciones	<p>El usuario debe estar autenticado.</p> <p>El usuario debe haber recibido al menos una solicitud de amistad.</p> <p>El usuario debe haber accedido al menú de solicitudes de amistad.</p>
Postcondiciones	<p>La solicitud de amistad será aceptada.</p> <p>Ambos usuarios aparecerán en la lista de amigos mutuamente.</p>
Escenarios principales	<p>Escenario 1: Rechazar una solicitud de amistad.</p> <ul style="list-style-type: none"> • El usuario accede al menú de solicitudes de amistad. • El sistema muestra una lista de todas las solicitudes de amistad pendientes. • El usuario hace clic en el botón "Rechazar solicitud". • El sistema elimina la solicitud de amistad. • El sistema muestra un mensaje confirmando que la solicitud ha sido rechazada.
Escenarios alternativos (Si procede)	No procede.

3.3.16. Caso de uso para RF.2.6: Cerrar sesión.

Título	Cerrar sesión.
--------	----------------

Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede cerrar sesión en la plataforma utilizando el botón accesible en la barra de navegación. Antes de completar la acción, el sistema solicita confirmación para cerrar sesión.
Precondiciones	El usuario debe estar autenticado en la plataforma.
Postcondiciones	El usuario es redirigido a la página de inicio de sesión como invitado. Su sesión queda cerrada.
Escenarios principales	<p>Escenario 1: Cerrar sesión.</p> <ul style="list-style-type: none"> • El usuario está autenticado en la plataforma. • El usuario accede a la barra de navegación y selecciona la opción "Cerrar sesión". • El sistema muestra un cuadro de diálogo de confirmación con el mensaje: "¿Estás seguro? Cerrarás sesión en la aplicación". • El usuario confirma haciendo clic en "Sí, cerrar sesión". • El sistema cierra la sesión del usuario y redirige a la página de inicio o de inicio de sesión. • El sistema muestra un mensaje confirmando que la sesión ha sido cerrada.

<p>Escenarios alternativos</p> <p>(Si procede)</p>	<p>Escenario alternativo 1: No cerrar sesión.</p> <ul style="list-style-type: none"> • El usuario está autenticado en la plataforma. • El usuario accede a la barra de navegación y selecciona la opción "Cerrar sesión". • El sistema muestra un cuadro de diálogo de confirmación con el mensaje: "¿Estás seguro? Cerrarás sesión en la aplicación". • El usuario confirma haciendo clic en "Cancelar". • El usuario mantiene la sesión actual.
--	--

3.3.17. Caso de uso para RF.2.7: Eliminar cuenta permanentemente.

<p>Título</p>	<p>Eliminar cuenta permanentemente.</p>
<p>Actor</p>	<p>Usuario.</p>
<p>Descripción</p>	<p>Este caso de uso describe cómo un usuario puede eliminar su cuenta de la plataforma de forma permanente. Una vez eliminada, todos los datos asociados con la cuenta se borrarán de forma irreversible.</p>
<p>Precondiciones</p>	<p>El usuario debe estar autenticado.</p> <p>El usuario debe haber accedido perfil de su cuenta.</p>
<p>Postcondiciones</p>	<p>La cuenta del usuario se eliminará permanentemente, y todos los datos asociados con</p>

	<p>dicha cuenta se borrarán. El usuario será redirigido a la página de inicio como invitado.</p>
<p>Escenarios principales</p>	<p>Escenario 1: Eliminar cuenta permanentemente</p> <ul style="list-style-type: none"> • El usuario accede a la sección de su perfil y selecciona la opción "Eliminar mi cuenta". • El sistema muestra un cuadro de diálogo de confirmación con el mensaje: "¿Estás seguro? Todos tus datos se eliminarán permanentemente". • El usuario hace clic en "Sí, eliminar cuenta". • El sistema elimina todos los datos del usuario, incluyendo sus juegos en la biblioteca, amigos y configuraciones. • El sistema cierra la sesión automáticamente y redirige al usuario a la página de inicio como invitado.
<p>Escenarios alternativos (Si procede)</p>	<p>Escenario 1: Cancelar eliminar cuenta permanentemente.</p> <ul style="list-style-type: none"> • El usuario accede a la sección de su perfil y selecciona la opción "Eliminar mi cuenta".

	<ul style="list-style-type: none"> • El sistema muestra un cuadro de diálogo de confirmación con el mensaje: "Cancelar". • El usuario no pierde sus datos.
--	--

3.3.20. Caso de uso para RF.2.10: Agregar juegos a la biblioteca.

Título	Agregar juegos a la biblioteca.
Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede agregar un juego a su biblioteca personal en la plataforma. Los juegos agregados podrán visualizarse en su biblioteca personal.
Precondiciones	El usuario debe estar autenticado. El usuario debe estar en la biblioteca.
Postcondiciones	El juego será añadido a la biblioteca del usuario.
Escenarios principales	<p>Escenario 1: Agregar un juego desde el catálogo de juegos</p> <ul style="list-style-type: none"> • El usuario accede al catálogo de juegos disponible en la plataforma. • El sistema muestra una lista de juegos disponibles. • El usuario usa el buscador para encontrar el que busca. • El usuario selecciona el juego que desea agregar a su biblioteca.

	<ul style="list-style-type: none"> • El sistema muestra un panel para elegir plataforma. • El usuario hace clic en el botón "Ok". • El sistema confirma que el juego ha sido agregado y lo refleja en la biblioteca del usuario.
Escenarios alternativos (Si procede)	No procede.

3.3.21. Caso de uso para RF.2.8: Acceder a la biblioteca

Título	Acceder a la biblioteca.
Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede acceder a su biblioteca de juegos en la plataforma.
Precondiciones	El usuario debe estar autenticado en la plataforma.
Postcondiciones	El usuario podrá ver la lista completa de juegos que ha agregado a su biblioteca.
Escenarios principales	<p>Escenario 1: Acceder a la biblioteca</p> <ul style="list-style-type: none"> • El usuario accede a la plataforma después de iniciar sesión. • El usuario selecciona la opción "GamingHub" en la barra de navegación.

	<ul style="list-style-type: none"> • El sistema muestra la biblioteca personal del usuario, que incluye todos los juegos que ha agregado. • El usuario puede ver detalles de los juegos, como el título, la plataforma, el estado y la puntuación.
Escenarios alternativos (Si procede)	<p>Escenario alternativo 1: Acceder a la biblioteca vacía</p> <ul style="list-style-type: none"> • El usuario accede a la plataforma después de iniciar sesión. • El usuario selecciona la opción "GamingHub" en la barra de navegación. • El sistema muestra la biblioteca personal del usuario vacía mostrando un aviso de que no hay juegos.

3.3.22. Caso de uso para RF.2.9: Buscar juegos nuevos.

Título	Buscar juegos nuevos.
Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede buscar juegos nuevos dentro de la plataforma para agregarlos a su biblioteca. Los juegos pueden buscarse mediante su título o parte de este.
Precondiciones	El usuario debe estar autenticado en la plataforma.

	El usuario debe acceder al buscador de juegos en la biblioteca.
Postcondiciones	El usuario verá una lista de juegos que coinciden con los criterios de búsqueda y podrá seleccionar uno o varios juegos para añadirlos a su biblioteca
Escenarios principales	<p>Escenario 1: Búsqueda de juegos por título</p> <ul style="list-style-type: none"> • El usuario accede a la biblioteca y selecciona la opción "Añadir juegos". • El sistema muestra un campo de búsqueda. • El usuario introduce el título del juego que desea buscar. • El sistema busca juegos que coinciden con el título ingresado y muestra una lista de resultados.
Escenarios alternativos (Si procede)	No procede.

3.3.23. Caso de uso para RF.2.10: Añadir juegos a la biblioteca

Título	Añadir juegos a la biblioteca.
Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede añadir juegos nuevos a su biblioteca personal en la plataforma. Los juegos pueden seleccionarse desde una lista de resultados de búsqueda y añadirse a la biblioteca del usuario.

Precondiciones	<p>El usuario debe estar autenticado en la plataforma.</p> <p>El usuario debe acceder al buscador de juegos en la biblioteca.</p>
Postcondiciones	<p>El juego será añadido a la biblioteca del usuario y aparecerá en su biblioteca.</p>
Escenarios principales	<p>Escenario 1: Añadir un juego a la biblioteca.</p> <ul style="list-style-type: none"> • El usuario accede a la biblioteca y selecciona la opción "Añadir juegos". • El usuario realiza una búsqueda para encontrar un juego nuevo. • El sistema muestra una lista de resultados que coinciden con la búsqueda. • El usuario selecciona un juego de la lista de resultados. • El sistema muestra un mensaje para seleccionar una plataforma mediante un desplegable. • El usuario pulsa "OK". • El sistema confirma que el juego ha sido agregado a la biblioteca del usuario. <p>Escenario 2: Añadir un juego con plataforma otros.</p> <ul style="list-style-type: none"> • El usuario accede a la biblioteca y selecciona la opción "Añadir juegos".

	<ul style="list-style-type: none"> • El usuario realiza una búsqueda para encontrar un juego nuevo. • El sistema muestra una lista de resultados que coinciden con la búsqueda. • El usuario selecciona un juego de la lista de resultados. • El sistema muestra un mensaje para seleccionar una plataforma mediante un desplegable. • El usuario elige otros y le aparece una caja de texto para elegir una plataforma a mano. • El usuario pulsa "OK". <p>El sistema confirma que el juego ha sido agregado a la biblioteca del usuario con la plataforma personalizada.</p>
<p>Escenarios alternativos (Si procede)</p>	<p>Escenario alternativo 1: Añadir un juego a la biblioteca.</p> <ul style="list-style-type: none"> • El usuario accede a la biblioteca y selecciona la opción "Añadir juegos". • El usuario realiza una búsqueda para encontrar un juego nuevo. • El sistema muestra una lista de resultados que coinciden con la búsqueda.

	<ul style="list-style-type: none"> • El usuario selecciona un juego de la lista de resultados. • El sistema muestra un mensaje para seleccionar una plataforma mediante un desplegable. • El usuario pulsa "Cancelar". • Su biblioteca permanece intacta.
--	---

3.3.24. Caso de uso para RF.2.11: Eliminar juegos en la biblioteca.

Título	Eliminar juegos en la biblioteca.
Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede eliminar juegos de su biblioteca personal en la plataforma. Al eliminar un juego, se borrarán todos los datos relacionados con dicho juego en la biblioteca del usuario.
Precondiciones	El usuario debe estar autenticado. El usuario debe haber accedido a su biblioteca personal.
Postcondiciones	El juego será eliminado de la biblioteca del usuario y ya no aparecerá en su biblioteca.
Escenarios principales	Escenario 1: Eliminar un juego de la biblioteca. <ul style="list-style-type: none"> • El usuario accede a su biblioteca personal. • El sistema muestra una lista de todos los juegos en la biblioteca del usuario.

	<ul style="list-style-type: none"> • El usuario selecciona el icono de papelera para eliminar un juego específico. • El sistema muestra un cuadro de diálogo de confirmación con el mensaje: "¿Estás seguro de eliminar?". • El usuario confirma haciendo clic en "Sí, eliminarlo". • El sistema elimina el juego de la biblioteca del usuario y muestra un mensaje confirmando que el juego ha sido eliminado.
<p>Escenarios alternativos (Si procede)</p>	<p>Escenario alternativo 1: No eliminar un juego de la biblioteca.</p> <ul style="list-style-type: none"> • El usuario accede a su biblioteca personal. • El sistema muestra una lista de todos los juegos en la biblioteca del usuario. • El usuario selecciona el icono de papelera para eliminar un juego específico. • El sistema muestra un cuadro de diálogo de confirmación con el mensaje: "¿Estás seguro de eliminar?". • El usuario cancela haciendo clic en "No, cancelar". • La biblioteca se mantiene intacta.

3.3.25. Caso de uso para RF.2.12: Editar juegos en la biblioteca.

Título	Editar juegos en la biblioteca.
Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede editar los detalles de un juego que ha añadido previamente a su biblioteca personal. Los cambios pueden incluir el estado del juego, la puntuación, comentarios, etiquetas, y el progreso en la historia, logros y horas jugadas.
Precondiciones	El usuario debe estar autenticado. El usuario debe haber accedido a su biblioteca y seleccionado un juego que desee editar.
Postcondiciones	Los cambios realizados por el usuario se guardarán y reflejarán en la biblioteca del usuario.
Escenarios principales	Escenario 1: Editar un juego en la biblioteca. <ul style="list-style-type: none"> • El usuario accede a su biblioteca personal. • El sistema muestra una lista de juegos en la biblioteca del usuario. • El usuario selecciona el botón "Ver más" en un juego específico. • El sistema muestra los detalles del juego, junto con la opción "Editar". • El usuario hace clic en el botón "Editar".

	<ul style="list-style-type: none"> • El sistema permite al usuario modificar la información del juego, incluyendo: <ul style="list-style-type: none"> ○ Estado del juego ○ Puntuación (en una escala de 1 a 10). ○ Comentarios personales. ○ Progreso en la historia. ○ Progreso general del juego. ○ Logros completados. ○ Horas jugadas. ○ Etiquetas asociadas al juego. • El usuario realiza los cambios necesarios y hace clic en "Guardar cambios". • El sistema actualiza el juego en la biblioteca con la nueva información.
<p>Escenarios alternativos (Si procede)</p>	<p>Escenario alternativo 1: Cancelar editar un juego en la biblioteca.</p> <ul style="list-style-type: none"> • El usuario accede a su biblioteca personal. • El sistema muestra una lista de juegos en la biblioteca del usuario. • El usuario selecciona el botón "Ver más" en un juego específico. • El sistema muestra los detalles del juego, junto con la opción "Editar".

	<ul style="list-style-type: none"> • El usuario hace clic en el botón "Editar". • El usuario decide cancelar la edición. • El juego se mantiene intacto.
--	---

3.3.26. Caso de uso para RF.2.12: Buscar juegos dentro de la biblioteca del usuario.

Título	Buscar juegos dentro de la biblioteca del usuario.
Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede buscar juegos en su biblioteca personal utilizando el título o las etiquetas asignadas a los juegos.
Precondiciones	El usuario debe estar autenticado en la plataforma y haber accedido a su biblioteca.
Postcondiciones	El usuario verá una lista de juegos que coinciden con los criterios de búsqueda seleccionados.
Escenarios principales	<p>Escenario 1: Buscar juegos por título.</p> <ul style="list-style-type: none"> • El usuario accede a su biblioteca personal. • El sistema muestra una barra de búsqueda. • El usuario introduce el título o parte del título del juego que desea buscar. • El sistema filtra los juegos y muestra aquellos que coinciden con el criterio de búsqueda. <p>Escenario 2: Buscar juegos por etiqueta.</p> <ul style="list-style-type: none"> • El usuario accede a su biblioteca personal.

	<ul style="list-style-type: none"> • El sistema muestra una barra de búsqueda. • El usuario introduce la etiqueta o parte de la etiqueta asociada a los juegos que desea buscar. • El sistema filtra los juegos y muestra aquellos que coinciden con el criterio de búsqueda.
Escenarios alternativos (Si procede)	<p>Escenario alternativo 1: Buscar juegos sin resultado.</p> <ul style="list-style-type: none"> • El usuario accede a su biblioteca personal. • El sistema muestra una barra de búsqueda. • El usuario introduce la etiqueta o parte de la etiqueta asociada a los juegos que desea buscar.

3.3.27. Caso de uso para RF.2.13: Ordenar juegos dentro de la biblioteca del usuario

Título	Ordenar juegos dentro de la biblioteca del usuario.
Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede ordenar los juegos dentro de su biblioteca por título, plataforma o estado.
Precondiciones	El usuario debe estar autenticado en la plataforma y haber accedido a su biblioteca.

Postcondiciones	El sistema muestra los juegos ordenados según los criterios seleccionados.
Escenarios principales	<p>Escenario 1: Ordenar juegos por nombre</p> <ul style="list-style-type: none"> • El invitado accede al perfil de un usuario público. • El invitado selecciona la opción para ver la librería de juegos del usuario. • El sistema muestra la lista de juegos en la librería del usuario. • El invitado selecciona la opción para ordenar los juegos por nombre. • El sistema reordena la lista de juegos en orden alfabético por nombre. <p>Escenario 2: Ordenar juegos por plataforma</p> <ul style="list-style-type: none"> • El invitado accede a la librería de juegos de un usuario público. • El sistema muestra la lista completa de juegos. • El invitado selecciona la opción para ordenar los juegos por plataforma. • El sistema reordena la lista de juegos en orden alfabético del nombre de la plataforma. <p>Escenario 3: Ordenar juegos por estado</p>

	<ul style="list-style-type: none"> • El invitado accede a la librería de un usuario público. • El sistema muestra la lista completa de juegos. • El invitado selecciona la opción para ordenar los juegos por estado (completado, en progreso, pendiente). • El sistema reordena la lista de juegos en función del estado seleccionado.
Escenarios alternativos (Si procede)	<p>Escenario alternativo 1: Librería vacía.</p> <ul style="list-style-type: none"> • Si la librería está vacía, el sistema muestra un mensaje indicando que no hay juegos en la librería del usuario.

3.3.28. Caso de uso para RF.2.14: Añadir a favoritos un juego de la biblioteca.

Título	Añadir a favoritos un juego de la biblioteca.
Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede añadir un juego a su lista de favoritos desde su biblioteca personal haciendo clic en el icono de corazón desde dentro del juego deseado.
Precondiciones	<p>El usuario debe estar autenticado y haber accedido a su biblioteca personal y al juego que desea añadir a favoritos.</p> <p>El juego no debe estar guardado como favorito.</p>

Postcondiciones	El juego se añadirá a la lista de favoritos del usuario, y el icono de corazón se mostrará resaltado.
Escenarios principales	<p>Escenario 1: Añadir un juego a favoritos.</p> <ul style="list-style-type: none"> • El usuario accede a su biblioteca personal. • El sistema muestra la lista de juegos en la biblioteca. • El usuario accede al juego que desea añadir a favoritos. • El usuario hace clic en el corazón dentro del juego para añadirlo a favoritos. • El sistema resalta el corazón en rojo, indicando que el juego ha sido añadido a favoritos.
Escenarios alternativos (Si procede)	No procede.

3.3.29. Caso de uso para RF.2.15: Eliminar de favoritos un juego de la biblioteca.

Título	Eliminar de favoritos un juego de la biblioteca.
Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede eliminar un juego de su lista de favoritos desde su biblioteca personal haciendo clic en el icono de corazón desde dentro del juego deseado.

Precondiciones	<p>El usuario debe estar autenticado y haber accedido a su biblioteca personal y al juego que desea eliminar de favoritos.</p> <p>El juego debe estar guardado como favorito.</p>
Postcondiciones	El juego será eliminado de la lista de favoritos del usuario, y el icono de corazón se verá blanco.
Escenarios principales	<p>Escenario 1: Eliminar un juego de favoritos</p> <ul style="list-style-type: none"> • El usuario accede a su biblioteca personal • El sistema muestra la lista de juegos en la biblioteca. • El usuario accede al juego que desea eliminar de favoritos. • El usuario hace clic en el corazón dentro del juego para eliminarlo de favoritos. • El sistema desactiva el corazón, indicando que el juego ha sido eliminado de favoritos.
Escenarios alternativos (Si procede)	No procede.

3.3.30. Caso de uso para RF.2.18: Añadir etiquetas al juego.

Título	Añadir etiquetas al juego.
Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede añadir etiquetas a un juego dentro de su

	<p>biblioteca personal al editar los detalles del juego y seleccionar las etiquetas deseadas.</p>
Precondiciones	<p>El usuario debe estar autenticado.</p> <p>El usuario debe haber accedido a su biblioteca y estar editando un juego.</p>
Postcondiciones	<p>El juego tendrá asociadas las nuevas etiquetas seleccionadas por el usuario.</p>
Escenarios principales	<p>Escenario 1: Añadir etiquetas a un juego.</p> <ul style="list-style-type: none"> • El usuario accede a su biblioteca personal. • El sistema muestra la lista de juegos en la biblioteca. • El usuario selecciona un juego y elige la opción "Editar". • El sistema muestra los detalles del juego, incluyendo una sección para añadir etiquetas • El usuario selecciona una o más etiquetas del desplegable. • El usuario selecciona "guardar cambios". • El sistema guarda las etiquetas seleccionadas y las asocia al juego.
Escenarios alternativos (Si procede)	<p>Escenario alternativo 1: Cancelar el añadir etiquetas a un juego.</p> <ul style="list-style-type: none"> • El usuario accede a su biblioteca personal.

	<ul style="list-style-type: none"> • El sistema muestra la lista de juegos en la biblioteca. • El usuario selecciona un juego y elige la opción "Editar". • El sistema muestra los detalles del juego, incluyendo una sección para añadir etiquetas • El usuario selecciona una o más etiquetas del desplegable. • El usuario selecciona "cancelar edición". • El juego se mantiene intacto.
--	--

3.3.31. Caso de uso para RF.2.19: Eliminar etiquetas del juego.

Título	Eliminar etiquetas del juego.
Actor	Usuario.
Descripción	Este caso de uso describe cómo un usuario puede eliminar etiquetas de un juego dentro de su biblioteca personal al editar los detalles del juego y deseleccionar las etiquetas asociadas.
Precondiciones	El usuario debe estar autenticado. El usuario debe haber accedido a su biblioteca y estar editando un juego.
Postcondiciones	Las etiquetas seleccionadas serán eliminadas del juego.
Escenarios principales	Escenario 1: Eliminar etiquetas de un juego.

	<ul style="list-style-type: none"> • El usuario accede a su biblioteca personal • El sistema muestra la lista de juegos en la biblioteca • El usuario selecciona un juego y elige la opción "Editar" • El sistema muestra los detalles del juego, incluyendo una lista de etiquetas asociadas • El usuario deselecta una o más etiquetas de la lista • El usuario selecciona "guardar cambios" • El sistema elimina las etiquetas seleccionadas del juego
<p>Escenarios alternativos (Si procede)</p>	<p>Escenario alternativo 1: Cancelar la eliminación de etiquetas de un juego.</p> <ul style="list-style-type: none"> • El usuario accede a su biblioteca personal. • El sistema muestra la lista de juegos en la biblioteca. • El usuario selecciona un juego y elige la opción "Editar". • El sistema muestra los detalles del juego, incluyendo una lista de etiquetas asociadas. • El usuario deselecta una o más etiquetas de la lista.

	<ul style="list-style-type: none"> • El usuario selecciona "cancelar edición" • El juego se mantiene intacto.
--	---

3.3.32. Caso de uso para RF.3.1: Acceder a todos los perfiles.

Título	Acceder a todos los perfiles.
Actor	Administrador.
Descripción	Este caso de uso describe cómo un administrador puede acceder a cualquier perfil en la plataforma, independientemente de si el perfil es público o privado, con el fin de gestionar el contenido o realizar acciones administrativas.
Precondiciones	El administrador debe estar autenticado en la plataforma con los permisos correspondientes. Deben existir usuarios con perfiles públicos y privados en la plataforma.
Postcondiciones	El administrador podrá ver toda la información de los perfiles de los usuarios, incluyendo aquellos que son privados.
Escenarios principales	<p>Escenario 1: Acceso a un perfil público</p> <ul style="list-style-type: none"> • El administrador accede a la plataforma con su cuenta de administrador. • El sistema muestra la lista de usuarios en la plataforma. • El administrador selecciona un usuario con perfil público.

	<ul style="list-style-type: none"> • El sistema muestra el perfil completo del usuario, incluyendo toda la información disponible. <p>Escenario 2: Acceso a un perfil privado</p> <ul style="list-style-type: none"> • El administrador accede a la plataforma con su cuenta administrativa. • El sistema muestra la lista de usuarios en la plataforma. • El administrador selecciona un usuario con perfil privado. • El sistema muestra el perfil completo del usuario, incluyendo la información que normalmente estaría oculta para otros usuarios.
<p>Escenarios alternativos (Si procede)</p>	<p>No procede.</p>

3.3.33. Caso de uso para RF.3.2: Eliminar permanentemente cuentas de usuarios.

Título	Eliminar permanentemente cuentas de usuarios.
Actor	Administrador.
Descripción	Este caso de uso describe cómo un administrador puede eliminar permanentemente la cuenta de cualquier usuario en la plataforma, borrando todos los datos asociados al mismo.

Precondiciones	<p>El administrador debe estar autenticado en la plataforma con los permisos correspondientes.</p> <p>El usuario cuya cuenta se eliminará debe existir en la plataforma.</p>
Postcondiciones	<p>La cuenta del usuario será eliminada permanentemente, incluyendo todos los datos asociados (juegos, amigos, biblioteca, etc.).</p>
Escenarios principales	<p>Escenario 1: Eliminar la cuenta de un usuario.</p> <ul style="list-style-type: none"> • El administrador accede a la plataforma con su cuenta administrativa. • El sistema muestra la opción "Admin" en el menú principal. • El administrador selecciona la opción "Admin" y es redirigido al panel de administración. • El panel de administración muestra una lista de usuarios, incluyendo nombre de usuario y su identificador, ID. • El administrador busca al usuario que desea eliminar mediante el campo de búsqueda o selecciona al usuario de la lista. • El administrador selecciona la opción "Eliminar" junto al nombre del usuario.

	<ul style="list-style-type: none"> • El sistema solicita confirmación para eliminar permanentemente la cuenta del usuario. • El administrador confirma la acción. • El sistema elimina la cuenta del usuario y todos los datos asociados al mismo.
<p>Escenarios alternativos (Si procede)</p>	<p>Escenario alternativo 1: Cancelar eliminar la cuenta de un usuario.</p> <ul style="list-style-type: none"> • El administrador accede a la plataforma con su cuenta administrativa. • El sistema muestra la opción "Admin" en el menú principal. • El administrador selecciona la opción "Admin" y es redirigido al panel de administración. • El panel de administración muestra una lista de usuarios, incluyendo nombre de usuario y su identificador, ID. • El administrador busca al usuario que desea eliminar mediante el campo de búsqueda o selecciona al usuario de la lista. • El administrador selecciona la opción "Eliminar" junto al nombre del usuario.

	<ul style="list-style-type: none"> • El sistema solicita confirmación para eliminar permanentemente la cuenta del usuario. • El administrador cancela la acción. • El usuario mantiene su cuenta intacta.
--	--

3.3.34. Caso de uso para RF.3.3: Crear etiquetas.

Título	Crear etiquetas.
Actor	Administrador.
Descripción	Este caso de uso describe cómo un administrador puede crear nuevas etiquetas, utilizando el panel de administración, para ser utilizadas en los juegos y perfiles dentro de la plataforma.
Precondiciones	El administrador debe estar autenticado en la plataforma con los permisos correspondientes.
Postcondiciones	La nueva etiqueta estará disponible para los usuarios y podrá ser utilizada en sus perfiles y en los juegos de sus bibliotecas.
Escenarios principales	<p>Escenario 1: Crear una nueva etiqueta.</p> <ul style="list-style-type: none"> • El administrador accede a la plataforma con su cuenta administrativa. • El sistema muestra la opción "Admin" en el menú principal.

	<ul style="list-style-type: none"> • El administrador selecciona la opción "Admin" y es redirigido al panel de administración. • En el panel de administración, el administrador localiza la sección de "Etiquetas". • El administrador introduce el nombre de la nueva etiqueta en el campo de texto. • El administrador selecciona "Añadir". • El sistema guarda la nueva etiqueta y la añade a la lista de etiquetas disponibles.
Escenarios alternativos (Si procede)	No procede.

3.3.35. Caso de uso para RF.3.4: Eliminar etiquetas.

Título	Eliminar etiquetas.
Actor	Administrador.
Descripción	Este caso de uso describe cómo un administrador puede eliminar etiquetas ya existentes en la plataforma, utilizando el panel de administración.
Precondiciones	El administrador debe estar autenticado en la plataforma con los permisos correspondientes. Deben existir etiquetas previamente creadas.
Postcondiciones	Las etiquetas seleccionadas serán eliminadas de la lista de etiquetas disponibles en la plataforma.

<p>Escenarios principales</p>	<p>Escenario 1: Eliminar una etiqueta</p> <ul style="list-style-type: none"> • El administrador accede a la plataforma con su cuenta administrativa • El sistema muestra la opción "Admin" en el menú principal • El administrador selecciona la opción "Admin" y es redirigido al panel de administración • En el panel de administración, el administrador localiza la sección de "Etiquetas". • El sistema muestra una lista de etiquetas con la opción de "Eliminar" junto a cada una. • El administrador selecciona la opción "Eliminar" junto a la etiqueta que desea eliminar. • El sistema solicita confirmación de la eliminación. • El administrador confirma la eliminación. • El sistema elimina la etiqueta de la lista de etiquetas disponibles.
<p>Escenarios alternativos (Si procede)</p>	<p>Escenario alternativo 1: Eliminar una etiqueta</p> <ul style="list-style-type: none"> • El administrador accede a la plataforma con su cuenta administrativa

	<ul style="list-style-type: none">• El sistema muestra la opción "Admin" en el menú principal• El administrador selecciona la opción "Admin" y es redirigido al panel de administración• En el panel de administración, el administrador localiza la sección de "Etiquetas".• El sistema muestra una lista de etiquetas con la opción de "Eliminar" junto a cada una.• El administrador selecciona la opción "Eliminar" junto a la etiqueta que desea eliminar.• El sistema solicita confirmación de la eliminación.• El administrador cancela la eliminación.• las etiquetas permanecen intactas.
--	---

4

Diseño del sistema

4.1 Estructura general

La aplicación GamingHub está diseñada siguiendo una arquitectura *RESTful*, lo que permite una comunicación eficiente entre el cliente y el servidor a través de peticiones HTTP, permitiendo las operaciones CRUD (Crear, Leer, Actualizar y Eliminar). La aplicación sigue el modelo *MVC* (Modelo-Vista-Controlador), lo que garantiza una clara separación de responsabilidades y facilita el mantenimiento y la escalabilidad del sistema.

El frontend está desarrollado utilizando *React*, mientras que el backend está construido con *Express*, siendo ambos frameworks de JavaScript.

Los datos de la aplicación están almacenados en una base de datos *MySQL*, que proporciona una estructura relacional para gestionar información.

4.2 Diagrama de arquitectura

Un diagrama de arquitectura es una representación visual que muestra la estructura y organización de los elementos de un sistema de software. Este tipo de diagrama es muy recomendable en el proceso de desarrollo de software porque ofrece una visión clara y estructurada de cómo interactúan las diferentes partes de un sistema, desde el frontend, que es la parte visible y con la que interactúan los usuarios, hasta el backend, que maneja la lógica de negocio y el almacenamiento de datos.

El diagrama de la Figura 2, muestra la arquitectura general del proyecto, dividido en dos secciones principales: el *frontend* y el *backend*. Estas secciones están claramente diferenciadas mediante el uso de colores distintos para facilitar la comprensión.

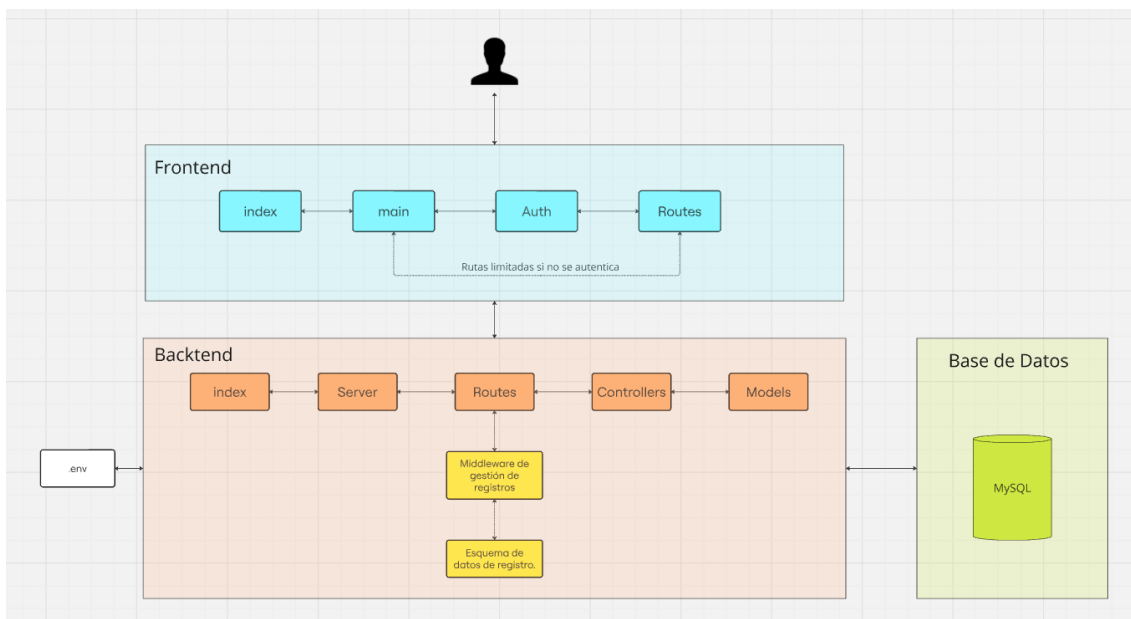


Figura 2: Diagrama de arquitectura de GamingHub [23].

Frontend

El frontend es la parte de la aplicación con la que interactúan directamente los usuarios.

En este caso, incluye los siguientes componentes:

Index y Main: Estas dos partes del frontend se encargan de inicializar y manejar la estructura principal de la aplicación.

Auth: Es el componente que maneja la autenticación de los usuarios. En caso de no autenticados, ciertas rutas o funcionalidades estarán limitadas.

Routes: Este componente gestiona la navegación de la aplicación. Dependiendo de las rutas definidas, un usuario puede acceder a diferentes vistas de la aplicación. Las rutas se dividen entre las disponibles para usuarios no autenticados, las que requieren inicio de sesión y las que solo tienen acceso los administradores.

Backend

El backend es la parte de la aplicación que implementa la lógica del servidor, las peticiones y la interacción con la base de datos. Incluye los siguientes componentes:

Index: Este componente es el punto de entrada del backend. Inicializa la aplicación cargando las variables de entorno definidas en el archivo ".env", como el puerto en el que el servidor escucha y las credenciales de la base de datos.

Server: Este componente configura el servidor que manejará las solicitudes HTTP y configura los middlewares necesarios, también define en qué puerto escucha el servidor y se configuran las rutas que gestionan las distintas funcionalidades del sistema.

Routes: Este componente define las rutas que el backend expone para realizar operaciones. Estas rutas reciben las solicitudes del frontend.

Controllers: Este componente contienen la lógica de negocio de la aplicación. Los controladores procesan las peticiones recibidas de las rutas, interactúan con los modelos y devuelven una respuesta al frontend.

Models: Este componente se encarga de procesar las solicitudes del controlador, adaptar los datos e interactuar con la base de datos, para posteriormente, devolver los datos al controlador de nuevo con el formato deseado.

4.3 Diseño de la base de datos

La base de datos utilizada por la aplicación GamingHub es MySQL, una base de datos relacional que sigue el modelo entidad-relación, el cual permite mantener integridad y consistencia en los datos. Dado que MySQL es una base de datos relacional, el diseño de la base de datos sigue un diagrama entidad-relación (Figura 3), adaptado a las necesidades de la plataforma.

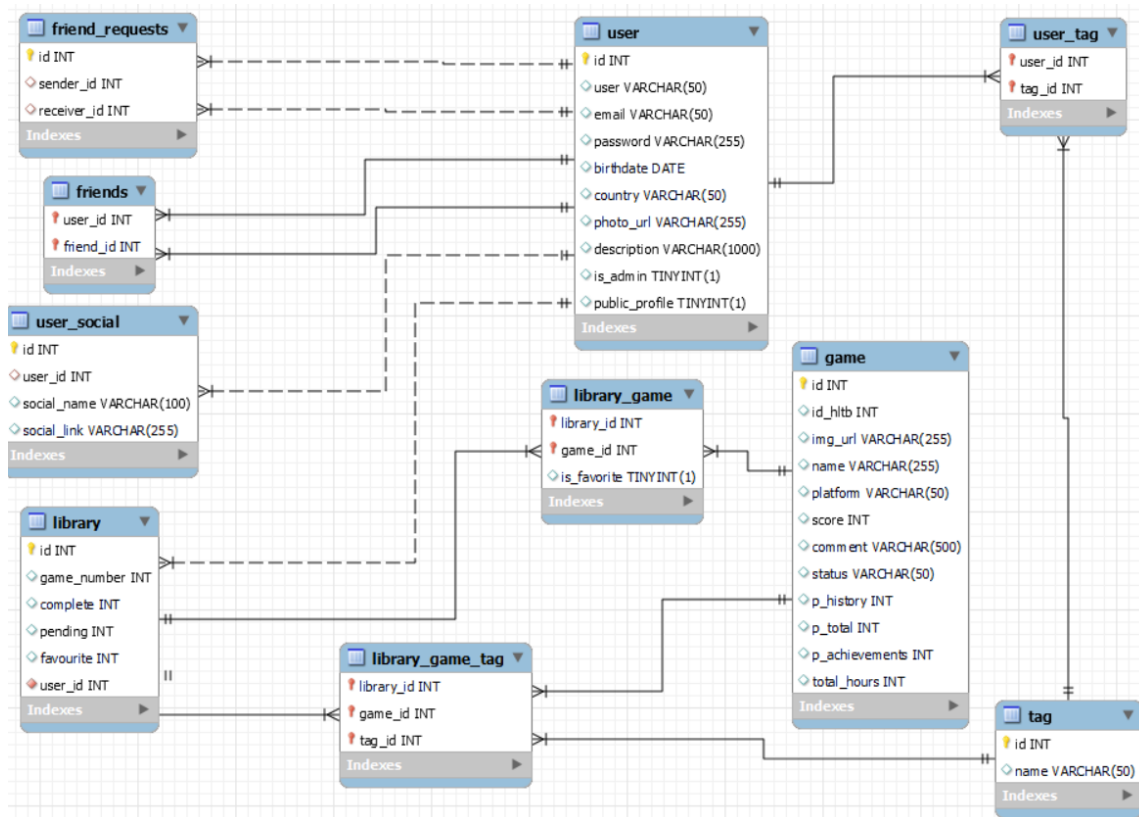


Figura 3: Modelo entidad relación

A continuación, se describen las principales entidades y sus relaciones:

4.3.1 Entidad User (Usuario)

La tabla User gestiona la información de los usuarios registrados en la aplicación. Un usuario puede ser un usuario normal o un administrador. Los atributos de la entidad User son:

id: Identificador único para cada usuario.

user: Nombre de usuario, único en la base de datos.

email: Dirección de correo electrónico del usuario, también única.

password: Contraseña almacenada de manera encriptada.

birthdate: Fecha de nacimiento del usuario.

country: País de residencia del usuario.

photo_url: URL de la imagen de perfil del usuario, por defecto se asigna una imagen genérica.

description: Descripción del perfil del usuario.

is_admin: Booleano que indica si el usuario es administrador.

public_profile: Indica si el perfil es público o privado.

4.3.2 Entidad Library (Biblioteca)

La tabla Library almacena la biblioteca de juegos de cada usuario. Cada usuario tiene una biblioteca única que contiene varios juegos. Los atributos de la entidad Library son:

id: Identificador único de la biblioteca.

game_number: Número de juegos en la biblioteca.

complete: Número de juegos completados.

pending: Número de juegos pendientes.

favourite: Número de juegos marcados como favoritos.

user_id: Relación con la tabla de usuarios, cada biblioteca pertenece a un único usuario.

4.3.3 Entidad Game (Juego)

La tabla Game representa los juegos individuales dentro de la plataforma, almacenando detalles sobre cada uno. Los atributos de la entidad Game son:

id: Identificador único del juego.

name: Nombre del juego.

platform: Plataforma en la que se juega.

score: Puntuación asignada por el usuario al juego, de 0 a 10.

comment: Comentario adicional sobre el juego.

status: Estado del juego.

p_history: Porcentaje de historia completada.

p_total: Porcentaje total completado.

p_achievements: Porcentaje de logros completados.

total_hours: Número total de horas jugadas.

4.3.4 Entidad Tags (Etiquetas)

La tabla Tags representa etiquetas que se pueden asignar tanto a perfiles como a juegos. Los atributos de la entidad Tag son los siguientes:

id: Identificador único de la etiqueta.

name: Nombre de la etiqueta.

4.3.5 Relaciones entre Entidades

La base de datos implementa relaciones de uno a muchos y muchos a muchos a través de tablas auxiliares:

User Tag: Relaciona usuarios con etiquetas de perfil, permitiendo que un usuario tenga múltiples etiquetas.

Library Game: Relaciona bibliotecas con juegos, permitiendo que una biblioteca contenga múltiples juegos.

Library Game Tag: Relaciona juegos con etiquetas en el contexto de una biblioteca, lo que permite categorizar los juegos con varias etiquetas.

4.3.6 Relaciones de Amistad

La aplicación permite a los usuarios enviar y aceptar solicitudes de amistad. Las entidades clave relacionadas con esta funcionalidad son:

Friends: Define la relación de amistad entre dos usuarios.

Friend Requests: Almacena las solicitudes de amistad pendientes entre usuarios.

4.3.7 Redes Sociales

La entidad User Social gestiona los enlaces a redes sociales de los usuarios. Cada usuario puede agregar varias redes sociales, con los siguientes atributos:

- `user_id`: Relación con el usuario.
- `social_name`: Nombre de la red social.
- `social_link`: URL del perfil del usuario en esa red social.

5

Implementación y pruebas

5.1 Primeros pasos

Para la realización del proyecto, se utilizaron diversas tecnologías y herramientas de apoyo. Antes de empezar a desarrollar el código, fue necesario realizar una serie de pasos e instalaciones previas para asegurar que el entorno de desarrollo estuviera correctamente configurado.

5.1.1 Instalación de Node.js y NPM

Para descargar Node.js, se descarga desde su web oficial y se instala siguiendo el asistente de instalación, el cuál instalará automáticamente NPM con él.

La aplicación, tanto en su parte frontend como backend, depende de Node.js y NPM. Se utiliza Node.js en su versión 20.13.0 para garantizar la compatibilidad con las últimas librerías y herramientas. NPM es necesario para la instalación y

gestión de dependencias del proyecto. Tras la instalación, se verifican las versiones con los comandos `node -v` y `npm -v`.

5.1.2 Instalación de MySQL y MySQL Workbench

Para la gestión de la base de datos del proyecto, se utiliza MySQL.

Su descarga se realiza desde la web oficial [24] y se instala siguiendo el asistente de instalación.

El uso de MySQL permite configurar una base de datos relacional para almacenar la información.

Además, se utiliza MySQL Workbench, una herramienta de interfaz gráfica que facilita la interacción con la base de datos, la cual se descarga también desde su web oficial [25].

En MySQL Workbench, se crea la base de datos `gaminghub` y se configura el esquema con sus respectivas tablas, relaciones y restricciones para el correcto funcionamiento de la aplicación.

5.1.3 Instalación de Postman

Postman se utiliza para probar la API RESTful del backend.

Su descarga se realiza desde su web oficial [21] y se instala siguiendo el asistente de instalación.

Se configuran las rutas del backend en Postman para realizar pruebas. Esto permite verificar que las respuestas del servidor fueran correctas.

5.1.4 Instalación Git

El control de versiones fue gestionado con git, una herramienta esencial para llevar un seguimiento de los cambios en el código.

Se debe descargar desde su web oficial [27] y seguir los pasos del instalador.

Se verifica la versión con el comando `git --version`.

5.1.5 Creación del repositorio de GitHub

Se ha creado un repositorio en GitHub para almacenar una copia de seguridad online del proyecto y para gestionar las distintas versiones del código de forma remota. Esto permite trabajar en diferentes ramas, realizar cambios en el código y subirlos mediante Git al repositorio, manteniendo un historial de los avances realizados y pudiendo revertir cualquier cambio si fuera necesario.

El repositorio de GamingHub está alojado en GitHub, pero es privado por motivos de privacidad, por ello se entrega una copia del proyecto.

5.1.6 Registro en Cloudinary

Cloudinary es un servicio en la nube especializado en la gestión de imágenes y videos, que facilita su almacenamiento. Ofrece una API para subir y gestionar archivos, ofreciendo un plan gratuito que es suficiente para el objetivo del proyecto.

En GamingHub se utiliza Cloudinary para almacenar las imágenes en la nube, lo que permite mantener las fotos de perfil de los usuarios online, agilizando considerablemente el peso de la aplicación web. La URL de la imagen generada por Cloudinary se almacena en la base de datos, permitiendo acceder a ella fácilmente. Existe una imagen por defecto que nunca se borra, mientras que las imágenes que suben los usuarios se sobrescriben al actualizar su perfil, asegurando que solo haya una imagen por usuario.

5.2 Estructura

5.2.1 Estructura general

El proyecto se estructura en dos partes principales: la interfaz de usuario (frontend) y la lógica del servidor (backend), los cuales se encuentran explicados en el capítulo 4 de este documento.

El backend ha sido desarrollado con Express.js. De esta librería se ha hecho uso principalmente de dos de sus características: la creación de rutas y el uso de middlewares. Haciendo uso de Express, podemos crear distintas rutas o “endpoints” las cuales pueden recibir peticiones del frontend. Express también permite añadir funciones que serán ejecutadas antes de procesar una petición, estas funciones son llamadas middlewares y pueden aplicarse a nivel global (para todas las rutas) o en rutas específicas. En este proyecto se ha hecho uso de distintos middlewares:

- express.json (global): Permite recuperar el cuerpo de las peticiones en formato json, sin necesidad de procesar nosotros manualmente los cuerpos de las peticiones.
- checkJWT (a nivel de ruta): Es un middleware creado en el transcurso del proyecto para la gestión de los JWT.
- zodUser (aplicado a la ruta para el registro del usuario): Lee el cuerpo de las peticiones y hace una serie de comprobaciones a los datos recibidos. Esto se hace mediante la librería Zod.
- cors (global): Gestiona el acceso al servidor desde distintos dominios.
- express-formidable (a nivel de ruta): Facilita la lectura de imágenes contenidas en las peticiones.

La configuración de la base de datos se realiza en el archivo "backend/mysql/.config.js". En este archivo se hace una comprobación para saber si la aplicación se está lanzando en modo testing o no. Este modo es usado para realizar las pruebas de integración con postman, asegurando que los tests no se hagan con los datos de los usuarios finales de la aplicación. En caso de estar en modo "testing" creo una conexión a la base de datos "gaminghubtest", en otro caso hago uso de la base de datos real "gaminghub".

```
“const database = process.env.NODE_ENV === 'testing' ?  
'gaminghubtest' : process.env.BD;”
```

Una vez decidida la base de datos deseada, se extrae la información necesaria para crear la conexión desde las variables de entorno y se finalmente se crea la conexión.

```
“const config = {  
  
  host: process.env.BDHOST,  
  
  user: process.env.BDUSER,  
  
  port: process.env.BDPORT,  
  
  password: process.env.BDPASS,  
  
  database  
  
};”
```

El frontend está desarrollado con la librería de React y se ha añadido un sistema de enrutado haciendo uso de la librería react-router. React-router desmonta y monta un nuevo componente cada vez que navegamos a otra ruta, esto permite no estar recargando la página por cada cambio de ruta que hagamos en la web, evitando así ese pequeño refresco de las páginas que hay en otras aplicaciones cada vez que se navega de una ruta a otra. Por tanto, como lo que cambia es el componente renderizado en la página y no página en sí, se podría decir que la web tiene una única página. Esto es común mente conocido como una SPA (Single Page Application).

Además, en el frontend se ha hecho uso de Axios para lanzar y gestionar las peticiones realizadas al servidor. Esta librería facilita el envío de solicitudes y el manejo de las respuestas HTTP. En el proyecto se crea una instancia personalizada con axios en la cual le decimos que la url base que debe usarse para hacer peticiones al servidor es “http://localhost:4000/”

```
“api = axios.create({ baseURL: 'http://localhost:4000/' });”
```

También se crean dos interceptores:

- Interceptor de las peticiones: Cada vez que hacemos una petición al servidor, axios intercepta la petición y le añade el token JWT del usuario a la cabecera “Authorization”.
- Interceptor de las respuestas: Cada vez que tenemos una respuesta del servidor, axios intercepta la respuesta antes de que llegue al cliente y comprueba el código de la respuesta. En caso de ser un código 401 (el usuario intentó realizar una acción para la cual no tiene permisos), el usuario será enviado a la pantalla de login (si no había iniciado sesión) o a su perfil (si el usuario había iniciado sesión).

Finalmente, se utilizó Vite como empaquetador de código. Vite ofrece un servidor rápido y eficiente, que realiza la construcción de los módulos de manera optimizada, mejorando los tiempos de carga y el rendimiento general durante el desarrollo. Actualmente es el método más popular para crear proyectos con React.

5.2.2 Seguridad

La seguridad es un aspecto esencial en el desarrollo de cualquier aplicación web, ya que los usuarios confían en la plataforma al ingresar sus datos personales y, por lo tanto, esperan que su privacidad y seguridad sean garantizadas. Proteger la información sensible y los procesos de autenticación es clave para mantener la confianza de los usuarios y la integridad del sistema. A lo largo del desarrollo de este proyecto, se han implementado diversas estrategias para asegurar tanto los datos como la interacción de los usuarios con la plataforma.

En este proyecto, se han aplicado varias medidas de seguridad para proteger los datos de los usuarios y garantizar la integridad de la base de datos. Una de estas medidas es la creación de un endpoint (Figura 4 y Figura 5) que reinicia la base de datos a un estado inicial con datos predefinidos. Este es inaccesible a menos que la aplicación esté en modo de prueba, lo que previene manipulaciones accidentales o maliciosas en entornos de producción (se puede lanzar en modo de pruebas si se utiliza el comando "npm run test").

```
const database = process.env.NODE_ENV === 'testing' ? 'gaminghubtest' : process.env.BD;
```

Figura 4: Comprobación de entorno de ejecución.

```
dbRouter.post('/reset', async (req, res) => {  
  if (process.env.NODE_ENV !== 'testing') {  
    return res.status(500).json({  
      ok: false,  
      msg: 'No se pueden realizar estas acciones en producción o desarrollo'  
    });  
  } else {
```

Figura 5: Protección de la ruta reset.

Otra capa de seguridad importante es el uso de bcrypt, una librería de encriptación utilizada para asegurar las contraseñas de los usuarios. Esta realiza un proceso en el que se aplica un hash y una sal en las contraseñas (Figura 6 y Figura 7), lo que las convierte en irreversibles, de modo que ni siquiera los administradores del sistema pueden conocer las contraseñas originales. Cuando un usuario se registra, la contraseña introducida se encripta mediante el uso de dicha librería. Posteriormente, durante el proceso de inicio de sesión, como la contraseña original no es accesible, la contraseña introducida se vuelve a encriptar y se compara con la versión previamente almacenada de la contraseña encriptada. Este proceso asegura que solo las contraseñas correctas, aún encriptadas, validen

el acceso, aumentando significativamente la protección contra intentos de violación de datos.

```
const salt = bcrypt.genSaltSync();
const hashPass = bcrypt.hashSync(password, salt);
await UserModel.create({ user, email, password: hashPass, birthdate, country });
```

Figura 6: Aplicación de hash y sal en el registro de usuario.

```
static async isCorrectPassword(password, hash) {
  return bcrypt.compareSync(password, hash);
}
```

Figura 7: Comprobación de contraseñas cifradas en el inicio de sesión.

Además de la encriptación de contraseñas, todas las rutas sensibles de la aplicación están protegidas mediante JSON Web Tokens (JWT). Un JWT es un estándar abierto utilizado para la autenticación y la creación de sesiones seguras. Funciona firmando cierta información encriptada que se transmite de manera segura entre el servidor y el cliente. Para ello, utilizamos una palabra clave secreta, la cual se encuentra en el archivo de variables de entorno ".env" para que no sea conocida, que asegura el proceso de firma, permitiendo generar un token que se divide en tres partes principales (Figura 8):

1. Header: Contiene información sobre el tipo de cifrado utilizado, como HS256, por ejemplo.
2. Payload: Incluye los datos que se desean transmitir de manera segura, en el caso del proyecto, el ID del usuario, correo electrónico, tiempo de expiración y la fecha de creación del token. Es importante mencionar que no se deben incluir datos demasiado sensibles en esta sección, ya que podría ser decodificada desde el frontend.

3. Signature: La firma generada al encriptar el header y el payload usando la clave secreta.

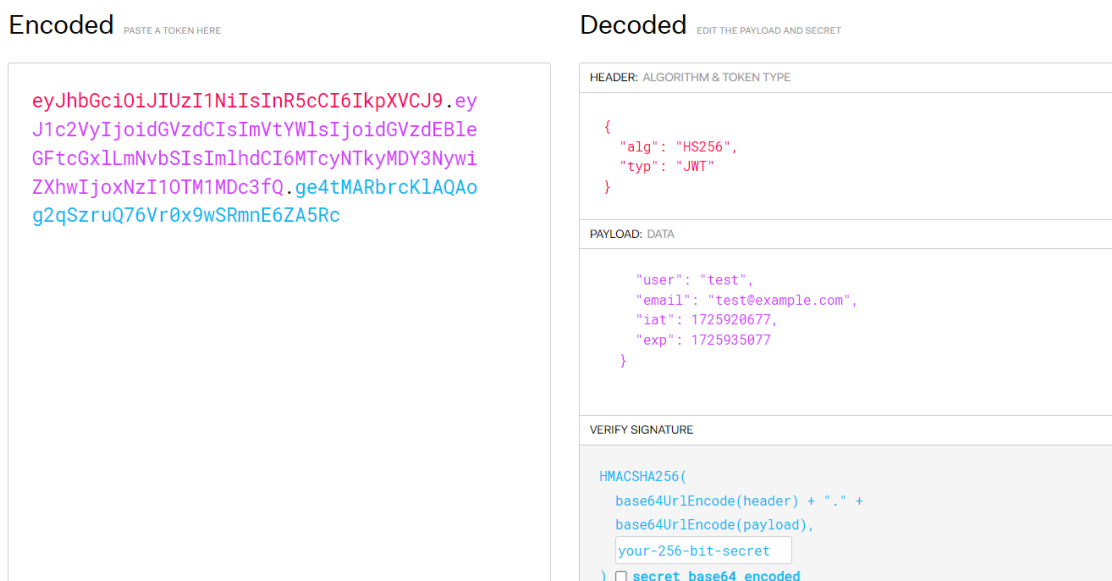


Figura 8: Decodificación de un token JWT.

El JWT se genera tanto en el proceso de registro como en el inicio de sesión, y el token resultante se envía al frontend como parte de la respuesta. Este token tiene un tiempo de expiración determinado (en el caso del proyecto se ha asignado una duración de 4 horas), lo que añade una capa adicional de seguridad al limitar la duración de cada sesión del usuario.

Gracias a estas medidas de seguridad, se asegura que solo usuarios autenticados puedan acceder a las rutas protegidas, se evitan accesos no autorizados y se garantiza la integridad de los datos durante las interacciones entre el cliente y el servidor.

5.3 Metodología de trabajo

Para el desarrollo de GamingHub, se siguió una metodología ágil basada en iteraciones cortas. El proceso se estructuró en sprints de aproximadamente dos semanas, donde al inicio de cada sprint se planifican las tareas a desarrollar, priorizando las más importantes para el avance del proyecto. En cada sprint, se

abordan diferentes componentes del sistema, y se realizan revisiones frecuentes para ajustar los requerimientos según sea necesario.

Inicialmente, se realizan configuraciones clave del entorno de desarrollo, como la instalación de dependencias y la configuración del control de versiones. Posteriormente, el enfoque se centra en el desarrollo del backend y la base de datos, asegurando que las API y la estructura relacional de la base de datos funcionen correctamente. A lo largo de los sprints, se alterna entre el desarrollo del frontend y la corrección de problemas o ajustes en el backend y la base de datos, a medida que se integran nuevas funcionalidades.

Al final de cada sprint, se revisa el progreso y se ajusta la planificación del siguiente ciclo, permitiendo que el proyecto evolucione de manera flexible. Este enfoque ágil permite avanzar de forma estructurada, garantizando que el sistema sea integrado de manera continua.

5.4 Pruebas

La fase de pruebas es fundamental en cualquier proyecto de desarrollo de software, ya que permite asegurar que los diferentes componentes de la aplicación funcionan como se espera, detectando posibles fallos y garantizando la calidad del producto final. Las pruebas ayudan a identificar errores en el código, inconsistencias en la lógica del negocio y problemas de integración entre los diferentes módulos de la aplicación, como el frontend, el backend y la base de datos.

5.4.1 Importancia de las pruebas

El motivo principal por el que se realizan pruebas es asegurar que la aplicación cumple con los requisitos funcionales y no funcionales, garantizando que los usuarios finales tengan una experiencia fluida y libre de errores. Las pruebas permiten:

- Verificar la funcionalidad de cada parte del sistema.
- Asegurar la fiabilidad de la aplicación en diferentes entornos.
- Detectar y corregir errores antes del despliegue en producción.
- Validar la interacción entre los diferentes módulos, como frontend, backend y base de datos.

5.4.2 Tipos de pruebas

Existen diversos tipos de pruebas (Figura 9) que se aplican en el desarrollo de software, entre las más comunes se encuentran:

- Pruebas unitarias: Se enfocan en probar partes individuales del código (como funciones, componentes o clases) para asegurar que cada unidad de trabajo funcione correctamente de manera aislada.
- Pruebas de integración: Estas pruebas comprueban que diferentes módulos del sistema, cuando se combinan, funcionan correctamente. Se evalúa la interacción entre varios componentes, como la comunicación entre el backend y la base de datos.
- Pruebas funcionales: Verifican que la aplicación cumpla con las especificaciones y los requisitos funcionales establecidos. Se evalúa si la aplicación hace lo que se supone que debe hacer.
- Pruebas de aceptación: Este tipo de prueba se realiza para garantizar que el sistema satisface las necesidades y expectativas del cliente o usuario final.

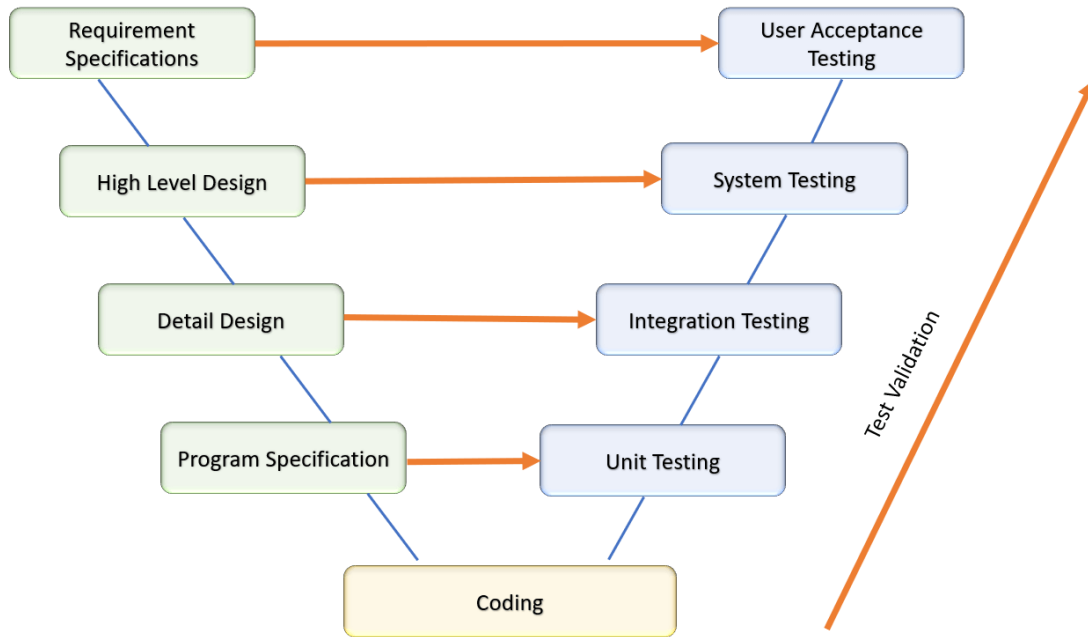


Figura 9: Relación entre pruebas del desarrollo software [28].

5.4.3 Pruebas del servidor y la base de datos

En el backend se han llevado a cabo pruebas de integración con el objetivo de verificar la interacción entre el servidor y la base de datos. Estas pruebas aseguran que, ante entradas correctas, el servidor responda adecuadamente y que los datos se almacenen y recuperen correctamente desde la base de datos.

Las pruebas se han realizado utilizando Postman, enviando peticiones a todos los endpoints del servidor para comprobar que las rutas están correctamente configuradas y que el sistema responde como se espera. Para garantizar la fiabilidad de las pruebas, se ha utilizado una base de datos alternativa llamada "gaminghubtest", separada de la base de datos de producción. Esta base de datos se restablece a un estado inicial antes de cada prueba, asegurando que las pruebas sean deterministas (Figura 10).

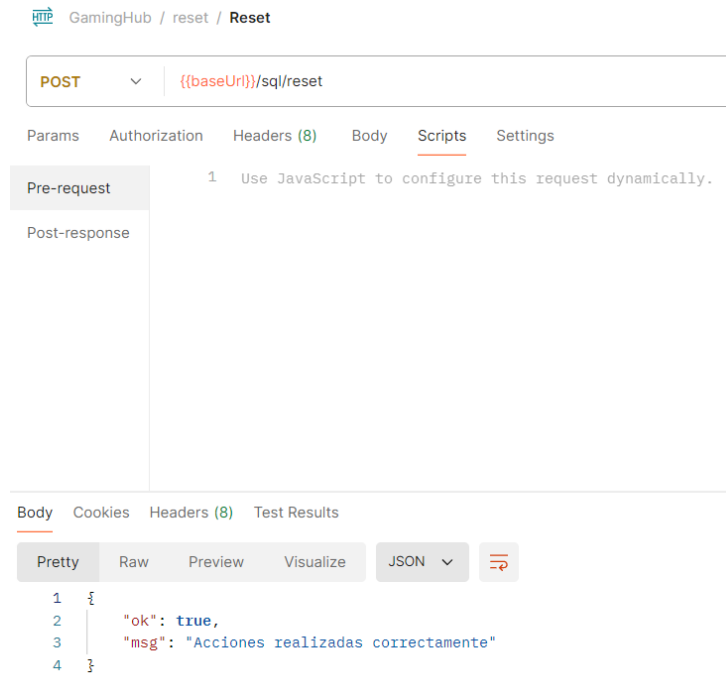


Figura 10: Establecimiento de la base de datos de prueba a un estado inicial.

Además, se han implementado medidas de seguridad para algunas rutas, donde es necesario autenticarse mediante tokens JWT. Estos tokens se generan al iniciar sesión o registrarse en la aplicación. En el contexto de las pruebas, antes de ejecutar todas las solicitudes en Postman, se realiza una petición de inicio de sesión que genera un token JWT válido (Figura 11), el cual se guarda automáticamente en una variable de entorno para su uso en las siguientes peticiones.

```
POST {{baseUrl}}/user/login

Params Authorization Headers (9) Body ● Scripts ● Settings

Pre-request •
Post-response •

1 pm.test("Status code is 200", function () {
2   | pm.response.to.have.status(200);
3   | });
4
5 pm.test("Check response", function () {
6   | var jsonData = pm.response.json();
7   | pm.expect(jsonData.ok).to.eql(true);
8   | pm.expect(jsonData.msg).to.eql("Login correcto");
9   | pm.expect(jsonData.user).to.eql("test");
10  | pm.expect(jsonData.id).to.eql(2);
11  | });
12
13 let response = pm.response.json();
14 if (response.token) {
15   | pm.environment.set("authToken", response.token);
16   | }
17
```

Figura 11: Generación del token en Postman para la realización de pruebas.

Para facilitar la ejecución de pruebas, se ha expuesto una ruta especial en la API que permite reiniciar la base de datos de pruebas a su estado inicial. Esta ruta está protegida mediante autenticación JWT y solo es accesible cuando la aplicación se ejecuta en modo de prueba. La distinción entre el modo de desarrollo y el modo de prueba se gestiona mediante una variable de entorno, que se inicializa dependiendo del comando utilizado al iniciar la aplicación. Para asegurar la compatibilidad en todos los sistemas operativos, se ha utilizado la librería cross-env, que soluciona problemas específicos de Windows.

Finalmente, en la carpeta mysql del backend se encuentra una subcarpeta llamada "Test Postman" donde se encuentran los archivos necesarios para importar y ejecutar todas las pruebas, que incluyen las pruebas de precondiciones y postcondiciones utilizando scripts en JavaScript.

Se ha añadido una funcionalidad en Postman que permite ejecutar todas las pruebas dentro de una misma sesión de conexión, como se observa en la Figura 12.

Source	Duration	All tests	Passed	Failed	Skipped	Avg. Resp. Time
Runner	25s 871ms	56	56	0	0	66 ms

· GamingHub · 1 completed

Figura 12: Resultado de las pruebas de integración en Postman.

Pasos para ejecutar las pruebas:

1. En primer lugar, se deben importar los archivos anteriormente mencionados.
2. A continuación, se debe lanzar el servidor en modo de pruebas, es decir, utilizando el comando "npm run test".
3. Una vez configurado lo anterior, se establece la base de datos al estado inicial de pruebas como en la Figura 10.
4. Ahora que se tiene la base de datos en el estado inicial, se hace la llamada al inicio de sesión como en la figura 11.
5. Por último, se ejecutan todas las pruebas de la colección GamingHub como en las Figura 13 y Figura 14.

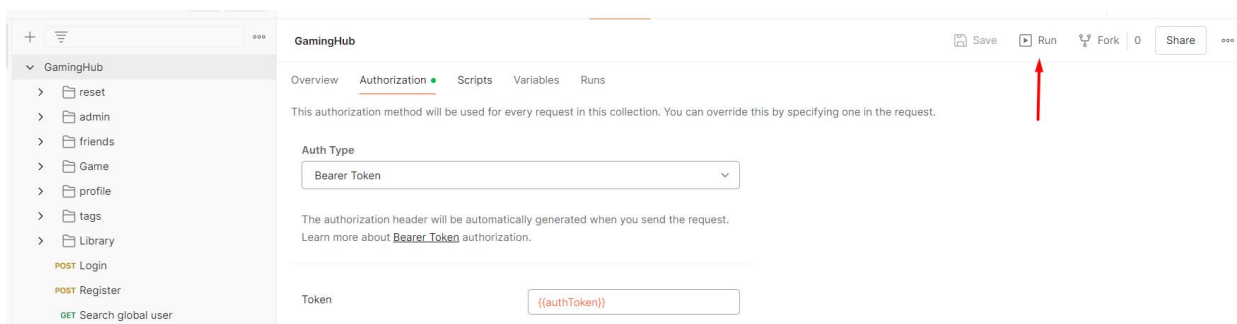


Figura 13: Ejecutar pruebas de la colección GamingHub.

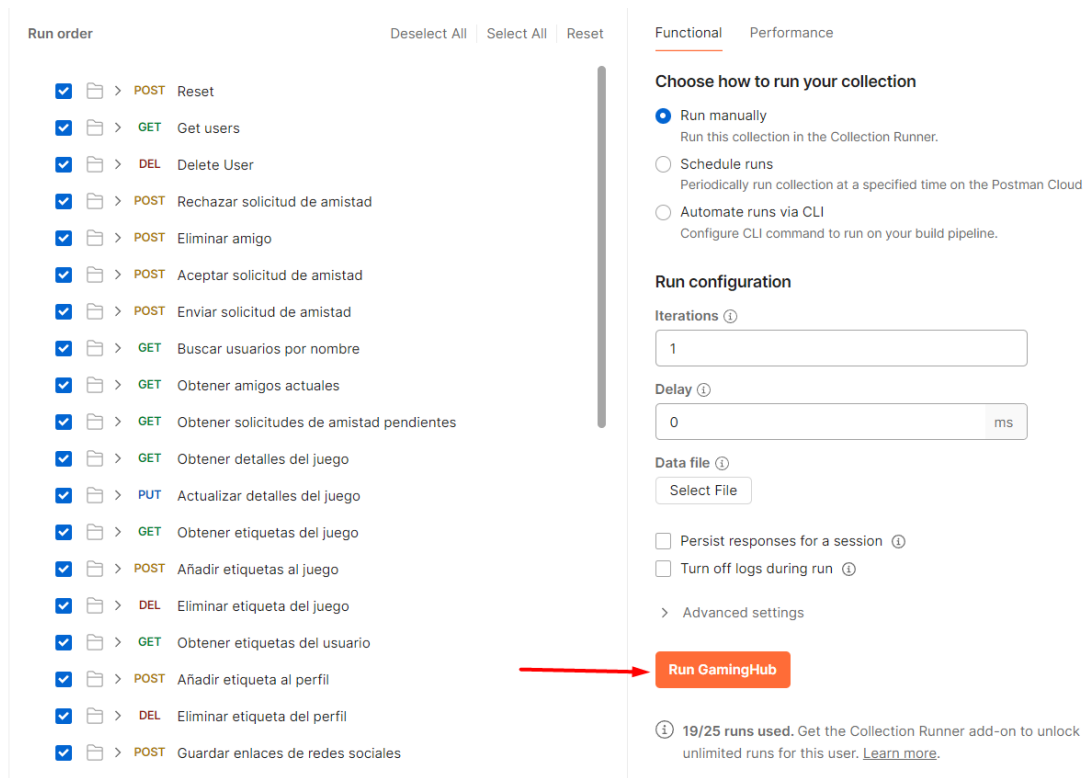


Figura 14: Selección y ejecución de pruebas Postman.

5.4.4 Pruebas del cliente

Se han implementado pruebas unitarias para los componentes de la interfaz de usuario utilizando las librerías Jest y React Testing Library. Estas pruebas permiten verificar que cada componente del frontend funcione correctamente de manera aislada, garantizando que las funcionalidades se implementen sin errores.

Para realizar las pruebas unitarias, se ha utilizado Jest, una de las librerías más populares en el entorno de JavaScript para pruebas de este tipo, junto con React Testing Library, que proporciona herramientas para renderizar componentes de React de manera aislada y facilitar la verificación de su comportamiento. Las pruebas se ejecutan mediante el comando "npm test".

Además, es posible generar un informe de cobertura de código utilizando el siguiente comando "npm test -- --coverage".

En algunos casos, ha sido necesario simular ciertas funcionalidades de la aplicación mediante "mocks". Esto implica crear versiones falsas o controladas de funciones

y módulos, permitiendo probar un componente sin depender de la implementación real. Por ejemplo, se ha simulado el Hook personalizado que gestiona la sesión del usuario (su nombre e ID), inicializando sus variables con valores de prueba.

También se ha simulado el funcionamiento de la librería Axios, utilizada para realizar peticiones al servidor. Esto permite evitar que se realicen peticiones reales durante las pruebas, a la vez que se verifica si se han enviado correctamente solicitudes POST a las URLs esperadas con la información adecuada, sin necesidad de ejecutar las peticiones reales.

En la Figura 15, se observa el resultado de las pruebas unitarias del frontend para un total de 72 pruebas.

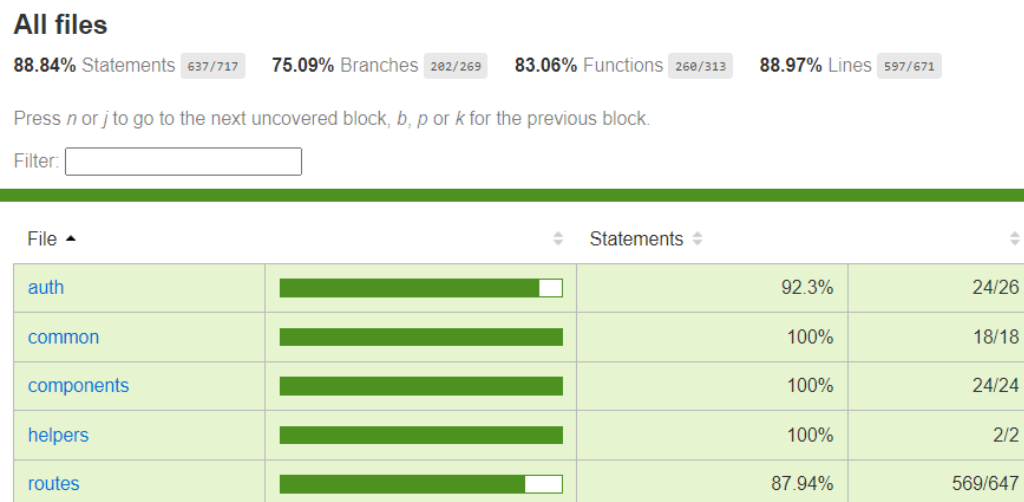


Figura 15: Resultado de las pruebas unitarias del frontend.

En la Figura 16, podemos observar una de las pruebas más sencillas, en la que se comprueba que se renderiza un componente y se muestra un texto "Perfil".

```
46 test('should render Perfil button when authenticated', () => {
47   // Simular que el usuario está autenticado
48   useAuth.mockReturnValue({ isAuthenticated: true, userId: '123' });
49
50   render(
51     <Router>
52       <Navbar />
53     </Router>
54   );
55
56   // Verificar que el botón de Perfil aparece
57   expect(screen.getByText(/Perfil/i)).toBeInTheDocument();
58 });
59
```

Figura 16: Test unitario que comprueba si se renderiza el componente y se muestra el botón de "Perfil".

6

Conclusiones y Trabajos Futuros

6.1 Conclusiones

El desarrollo de GamingHub surge como respuesta a la creciente necesidad de los jugadores de gestionar sus bibliotecas de videojuegos de manera unificada. Con el aumento exponencial del número de plataformas físicas y digitales, como consolas y tiendas en línea, los jugadores han enfrentado la problemática de mantener un control sobre los títulos que poseen, fragmentados entre diferentes servicios. Esta dispersión dificulta no solo la organización de los juegos, sino también el seguimiento de su progreso.

Este proyecto ha permitido crear una plataforma centralizada para gestionar bibliotecas de videojuegos dispersas entre múltiples plataformas físicas y digitales. A lo largo del proyecto, se han alcanzado los objetivos planteados al principio, como facilitar al usuario la gestión de sus juegos, el seguimiento de su progreso y la interacción social con otros usuarios. La plataforma proporciona una solución eficiente a la problemática actual de las bibliotecas fragmentadas de videojuegos,

ofreciendo una forma de organizarlas de manera sencilla y accesible desde un único lugar.

El uso de tecnologías como React para el frontend, Express en el backend, y MySQL para la gestión de la base de datos, ha sido fundamental para el éxito de este proyecto. Estas herramientas han permitido la creación de una aplicación web a medida, capaz de manejar datos individuales asociados a cada usuario y proporcionar una experiencia de usuario fluida, además de contar con un factor social.

El uso de tecnologías como MySQL ha proporcionado una base de datos relacional, asegurando la consistencia en la gestión de datos. Además, React permite construir una interfaz de usuario dinámica y adaptable, mejorando la experiencia de uso. Por su parte, Express ha sido fundamental para la construcción del servidor, garantizando tiempos de respuesta rápidos y una arquitectura modular que facilita la escalabilidad y el mantenimiento de la aplicación.

Uno de los desafíos más significativos ha sido integrar las funcionalidades de autenticación de usuarios manteniendo la seguridad, así como la correcta asociación entre usuarios, bibliotecas y juegos, respetando las relaciones entre ellos. Además, al tener diversas tecnologías, librerías e incluso APIs externas interactuando entre sí en tiempo real, es muy importante verificar en todo momento que el funcionamiento es correcto.

Aunque el despliegue de GamingHub en la web sería un paso fundamental para hacerlo accesible a un público más amplio, actualmente no se ha realizado, ya que se trata de un proyecto universitario. El proyecto ha sido desarrollado en un entorno de prueba local para cumplir con los requisitos académicos, sin la necesidad de ser alojado en un servidor en la nube o en una plataforma de hosting.

6.2 Líneas Futuras

Actualmente, GamingHub es totalmente funcional y cumple con las exigencias básicas planteadas para lograr su objetivo, no obstante, existen un sinnúmero de mejoras que podría impulsar la calidad y versatilidad de la aplicación web hasta el punto de llegar a ser competitiva y comercializable. A pesar de no ser el objetivo de este proyecto, se mencionan a continuación una serie de posibles pasos a futuro que podrían desarrollarse con el fin de tener una aplicación web más completa y accesible:

6.2.1 Despliegue en la web

El despliegue de GamingHub en la web sería un paso esencial para hacer que la plataforma esté disponible para un público más amplio.

Para llevar la plataforma a la producción, sería necesario desplegarla en un servidor en la nube o en un servicio de hosting adecuado.

Para garantizar un despliegue exitoso, sería necesario configurar correctamente tanto el backend como el frontend para funcionar de manera óptima en un entorno de producción.

6.2.2 Soporte para múltiples idiomas

En su versión actual, GamingHub está disponible únicamente en español. Sin embargo, para hacer la plataforma más accesible y atractiva para usuarios de diferentes partes del mundo, sería importante implementar soporte para múltiples idiomas, permitiendo un soporte para los idiomas más hablados, como el inglés, portugués, chino, ruso, francés y alemán, pudiendo incrementar así considerablemente el número de usuarios potenciales de la plataforma.

6.2.3 Integración multiplataforma

Actualmente, GamingHub es una aplicación web, pero el siguiente paso natural para mejorar la accesibilidad sería desarrollar una versión móvil

responsiva. Esto permitiría a los usuarios tener una interfaz más agradable y usable en tabletas y dispositivos móviles, lo que no solo mejoraría la comodidad, sino también el alcance de la plataforma.

6.2.4 Mejora en la búsqueda de juegos

Una funcionalidad clave que podría ser mejorada es el sistema de búsqueda y filtrado de juegos. Implementar un motor de búsqueda más avanzado que permita a los usuarios filtrar juegos no solo por nombre, sino también por género, plataforma, desarrollador o incluso por etiquetas personalizadas, facilitaría la gestión de bibliotecas grandes. Asimismo, añadir un sistema de recomendaciones basado en los juegos que los usuarios ya poseen o han jugado podría mejorar la experiencia general, permitiendo descubrir nuevos juegos de manera más intuitiva y personalizada.

6.2.5 Ampliación de funcionalidades sociales

Si bien la plataforma ya permite una cierta interacción entre los usuarios, una futura ampliación podría centrarse en mejorar las capacidades sociales dentro de la plataforma. Implementar funciones de chat en tiempo real, foros temáticos o la posibilidad de realizar comentarios en las bibliotecas de otros usuarios enriquecería la interacción social, fomentando una mayor participación y creando una comunidad más activa.

6.2.6 Tener una base de datos de juegos propia

Actualmente, la plataforma depende de fuentes externas para obtener datos de los juegos (HowLongToBeat API [4]). Una posible mejora significativa sería desarrollar y mantener una base de datos propia de juegos, lo que proporcionaría mayor control sobre la información almacenada y presentada a los usuarios. Con una base de datos propia, se podrían incluir características a medida, lo que enriquecería la experiencia del usuario. Además, al no depender

de APIs externas, se garantizaría una mayor autonomía y disponibilidad de los datos, reduciendo la posibilidad de problemas por caídas o restricciones en las fuentes externas.

6.2.7 Mejoras en seguridad

Un área clave para el futuro es mejorar la seguridad del sistema, especialmente en el proceso de registro y autenticación de usuarios. Una posible mejora sería la implementación de un sistema de verificación de correo electrónico para confirmar la identidad del usuario antes de que se complete el registro. Esto reduciría la creación de cuentas falsas o duplicadas. Adicionalmente, se podrían incluir mecanismos como la autenticación en dos pasos (2FA) para reforzar la protección de las cuentas de los usuarios. También sería importante implementar validaciones más estrictas y filtros de seguridad para evitar ataques de fuerza bruta.

6.2.8 Implementación de modo oscuro

Otra mejora que podría añadirse a la experiencia de usuario es el desarrollo de un modo oscuro. Hoy en día, muchos usuarios prefieren interfaces más suaves a la vista, especialmente durante las horas nocturnas o en ambientes con poca luz. El modo oscuro no solo mejora la experiencia visual, sino que también puede reducir la fatiga ocular en sesiones largas.

Referencias

[1] National Geographic, “¿Cuál fue el primer videojuego? Esta es la historia de un pasatiempo universal,” *historia.nationalgeographic.com.es*, Jan. 09, 2024.

https://historia.nationalgeographic.com.es/a/cual-fue-primer-videojuego-esta-es-historia-pasatiempo-universal_20123 (*Accedido en Sep. 01, 2024*).

[2] Statista GmbH, “Industria del videojuego: valor de mercado a nivel mundial 2021-2027,” *Statista*, Nov. 2023.

<https://es.statista.com/estadisticas/598622/valor-de-mercado-del-videojuego-en-el-mundo/> (*Accedido en Sep. 01, 2024*).

[3] The Game Awards, “The Game Awards,” *The Game Awards*, 2019.

<https://thegameawards.com/> (*Accedido en Sep. 01, 2024*).

[4] “How long does it take to beat your favorite games? -

HowLongToBeat.com,” *Howlongtobeat.com*, 2019. <https://howlongtobeat.com/> (*Accedido en Sep. 01, 2024*).

[5] *U. de Murcia*, “Introducción general a HTML y CSS. Desarrollo de Aplicaciones Web. Rafael Menéndez-Barzanallana Asensio. Universidad de Murcia (DIS),” *www.um.es*.

<https://www.um.es/docencia/barzana/DAWEB/Introduccion-a-html-y-css.html#:~:text=El%20origen%20de%20HTML%20se> (*Accedido en Sep. 01, 2024*).

[6] *ESIC*, “HTML5: qué es, características y cómo funciona | ESIC,”

www.esic.edu. <https://www.esic.edu/rethink/tecnologia/html5-que-es-caracteristicas-y-como-funciona->

[c#:~:text=El%20HTML5%20es%20un%20est%C3%A1ndar](https://www.esic.edu/rethink/tecnologia/html5-que-es-caracteristicas-y-como-funciona-#:~:text=El%20HTML5%20es%20un%20est%C3%A1ndar) (*Accedido en Sep.*

01, 2024).

[7] *tailwindcss*, “Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.,” *tailwindcss.com*, 2023. <https://tailwindcss.com/> (Accedido en Sep. 01, 2024).

[8] *diberry*, “Introducción a Node.js con JavaScript - Training,” *Microsoft.com*, 2024. <https://learn.microsoft.com/es-es/training/modules/intro-to-nodejs/> (Accedido en Sep. 01, 2024).

[9] *R. Dahl*, “Installing Node.js via package manager,” *Node.js*. <https://nodejs.org/en/download/package-manager> (Accedido en Sep. 01, 2024).

[10] *MySQL*, “MySQL :: MySQL Workbench,” *Mysql.com*, 2019. <https://www.mysql.com/products/workbench/> (Accedido en Sep. 01, 2024).

[11] *L. Llamas*, “Cómo usar una base de datos MySQL con Node.js,” *Luis Llamas*, Apr. 29, 2024. <https://www.luisllamas.es/como-usar-mysql-con-nodejs/> (Accedido en Sep. 01, 2024).

[12] *VPN Unlimited*, “Qué es Bcrypt - Términos y Definiciones de Ciberseguridad,” *Vpnunlimited.com*, 2024. <https://www.vpnunlimited.com/es/help/cybersecurity/bcrypt> (Accedido en Sep. 01, 2024).

[13] *Open Webinars*, “Qué es Json Web Token y cómo funciona,” *OpenWebinars.net*, Jan. 17, 2020. <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/> (Accedido en Sep. 01, 2024).

[14] *Microsoft*, “Visual Studio Code,” *Visualstudio.com*, 2024. <https://code.visualstudio.com/> (Accedido en Sep. 01, 2024).

[15] *E. You*, “Vite,” *es.vitejs.dev*. <https://es.vitejs.dev/guide/> (Accedido en Sep. 01, 2024).

- [16] Meta Open Source, “React,” *react.dev*, 2024. <https://react.dev/> (Accedido en Sep. 01, 2024).
- [17] Refine, “Understanding Virtual DOM in React | Refine,” *Refine.dev*, Nov. 23, 2023. <https://refine.dev/blog/react-virtual-dom/#what-is-the-dom> (Accedido en Sep. 01, 2024).
- [18] OpenJS Foundation, “Express - Node.js web application framework,” *Expressjs.com*, 2017. <https://expressjs.com/> (Accedido en Sep. 01, 2024).
- [19] Atlassian, “Qué es Git: conviértete en todo un experto en Git con esta guía,” *Atlassian*. <https://www.atlassian.com/es/git/tutorials/what-is-git> (Accedido en Sep. 01, 2024).
- [20] GitHub, “GitHub,” *GitHub*, 2024. <https://github.com/> (Accedido en Sep. 01, 2024).
- [21] Postman, “Postman | The Collaboration Platform for API Development,” *Postman*, 2021. <https://www.postman.com/> (Accedido en Sep. 01, 2024).
- [22] Meta, “Empezando · Jest,” *jestjs.io*. <https://jestjs.io/es-ES/docs/getting-started> (Accedido en Sep. 01, 2024).
- [23] Khusid, Andrey . “Miro | Plataforma de Pizarra Virtual Colaborativa Gratuita.” <https://Miro.com/>, miro.com/es/. (Accedido en 1 Sept. 2024).
- [24] “MySQL :: Download MySQL Installer,” *Mysql.com*, 2019. <https://dev.mysql.com/downloads/installer/> (Accedido en Sep. 01, 2024).
- [25] “MySQL :: Download MySQL Workbench,” *dev.mysql.com*. <https://dev.mysql.com/downloads/workbench/> (Accedido en Sep. 01, 2024).
- [26] R. Dahl, “Index of /dist/v20.13.0/,” *Nodejs.org*, 2024. <https://nodejs.org/dist/v20.13.0/> (Accedido en Sep. 01, 2024).
- [27] “Git - Downloading Package,” *git-scm.com*. <https://git->

scm.com/download/win (Accedido en Sep. 01, 2024).

[28] *Lambdatest, “Test Verification vs Validation in Website Testing,” LambdaTest, May 06, 2019. <https://www.lambdatest.com/blog/test-verification-vs-validation-in-website-testing/> (Accedido en Sep. 01, 2024).*

Apéndice A

Manual de Usuario

En este manual se pretende guiar al usuario en la navegación y uso de las funciones principales de la plataforma GamingHub, proporcionando instrucciones claras para interactuar con sus distintas herramientas y características.

A.1 Páginas accesibles por todos los usuarios

La página principal de GamingHub ofrece tres posibilidades: Iniciar sesión, registrar nuevo usuario y navegar como invitado mediante el buscador (Figura 17).

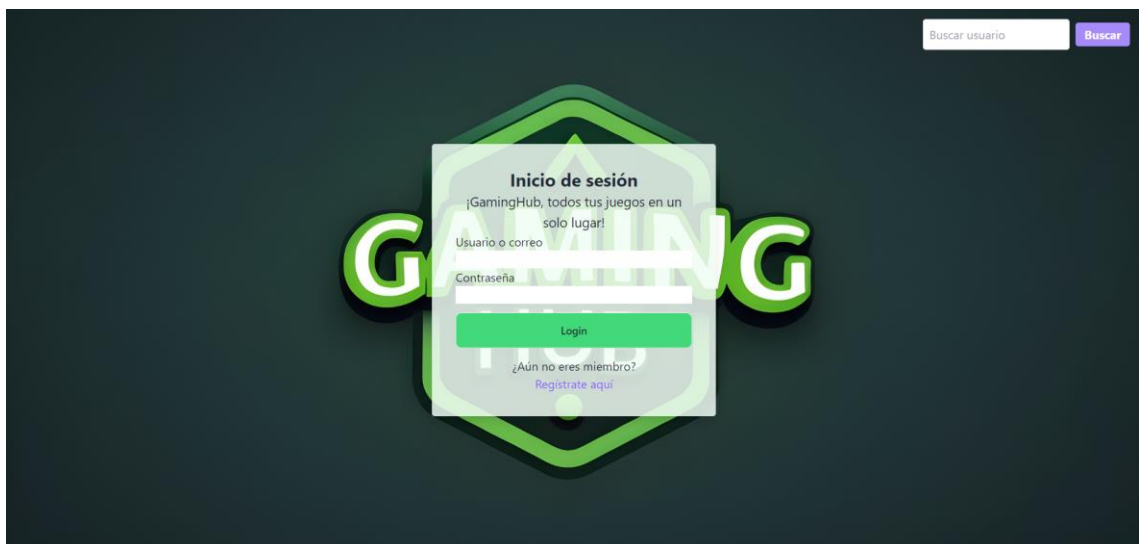


Figura 17: Página principal de GamingHub.

Si se desea navegar por la web como invitado, se deberá introducir el nombre del usuario que se desea observar en el buscador, pudiendo suceder tres casos distintos:

1. No se ha encontrado el usuario por lo que deberá introducir otro distinto.
2. El usuario tiene el perfil privado por lo que solo se podrá ver cierta información reducida como se muestra en la Figura 18.
3. El último caso es el más interesante ya que se trata de poder ver perfiles públicos como se muestra en la Figura 19. Además, se podrán ver sus redes sociales (Figura 20) seleccionando el botón “Redes Sociales” o ver su librería de juegos (Figura 21) seleccionando el botón “Ver Librería”. Si es posible acceder a su librería también se podrá ver la información específica de los juegos que tenga dicha librería, además de poder utilizar el filtro de búsqueda por título o etiqueta y poder ordenar los resultados por título, estado o plataforma (Figura 22).

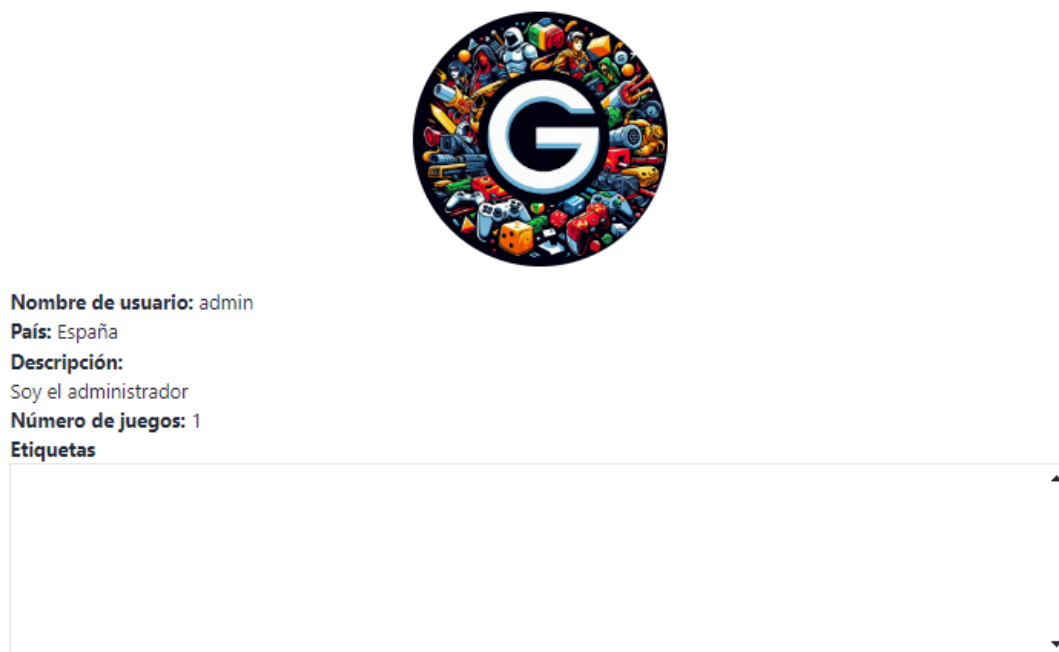


Figura 18: Vista de perfil privado.



Ver Librería

Redes Sociales

Nombre de usuario: test

País: México

Descripción:

Descripción de test

Número de juegos: 1

Juegos completados: 0

Juegos pendientes: 1

Juegos favoritos: 0

Etiquetas

RPG

Figura 19: Vista de un perfil público, de amigo o desde la vista de administrador.

Redes Sociales

Twitter

<https://twitter.com/test>

Figura 20: Vista de las redes sociales de un usuario.

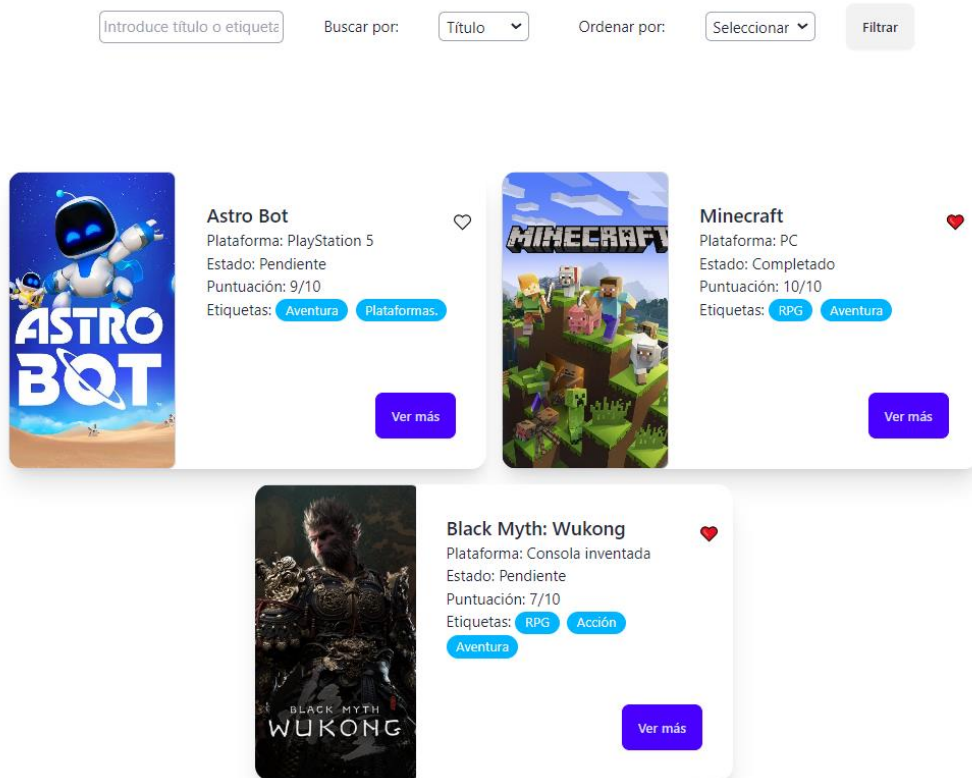


Figura 21: Vista de una librería de juegos ajena.



Minecraft

Plataforma: PC

Estado: Completado

Puntuación: 10/10

Comentarios: Construcción divertida.

Historia completada: 100%

Juego completado: 100%

Logros completados: 100%

Horas totales jugadas: 56

Etiquetas del juego

RPG Aventura

Figura 22: Vista de un juego de otro usuario.

Si por el contrario se decide registrar nuevo usuario se debemos pulsar sobre el texto "Regístrate aquí" y rellenar los campos que aparecen en la Figura 23, teniendo en cuenta que deben ser usuarios y correos no usados con anterioridad, que las contraseñas deben coincidir y que el correo tener el formato adecuado.

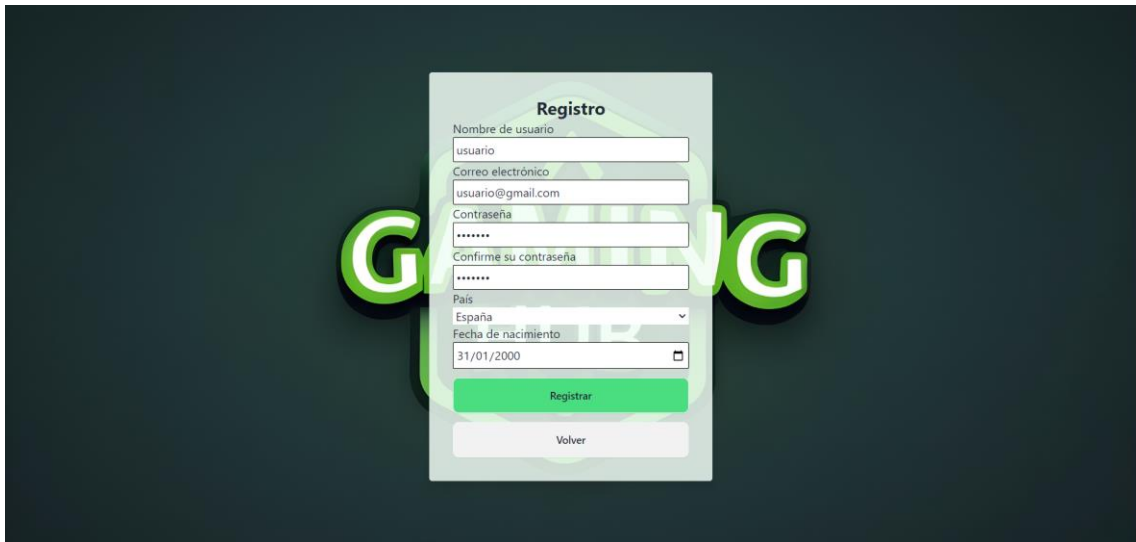


Figura 23: Página de registro de nuevo usuario.

A.2 Perfil de usuario registrado

Tanto si se decide iniciar sesión con un usuario existente o realizando un nuevo registro, se accederá automáticamente al perfil del usuario como se muestra en la Figura 24.

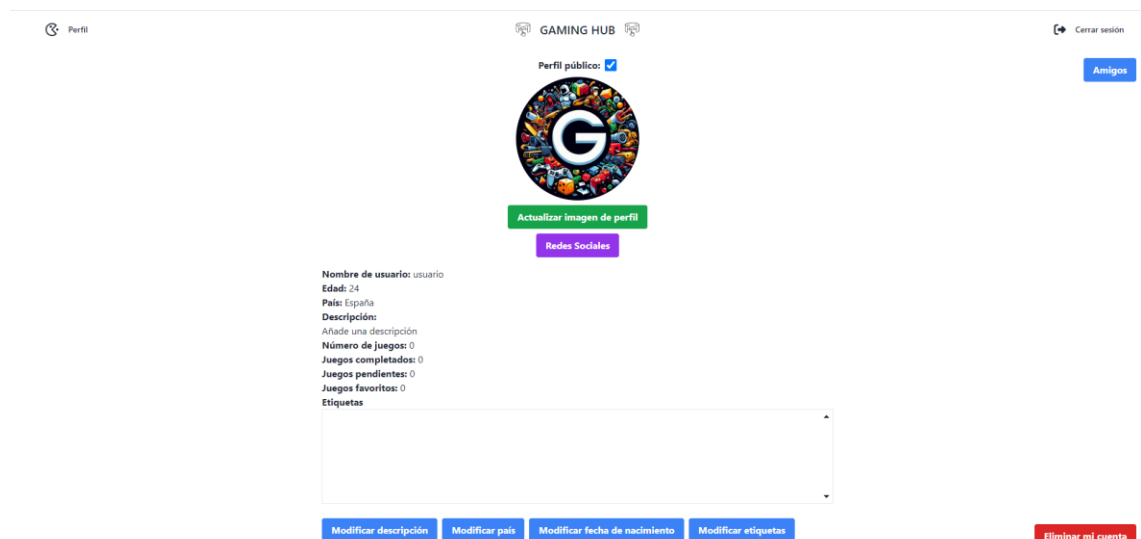



Figura 24: Perfil de usuario.

Tanto la imagen de perfil (Figura 25), como la fecha de nacimiento (Figura 26), país (Figura 27), descripción (Figura 28), etiquetas (Figura 29) y redes sociales (Figura 30), son modificables seleccionando el botón correspondiente.




Figura 25: Menú de selección de imagen.

Nombre de usuario: usuario
Edad: 24
País: España
Fecha de nacimiento:
 

Guardar **Cancelar**

Descripción:
Añade una descripción
Número de juegos: 0
Juegos completados: 0
Juegos pendientes: 0
Juegos favoritos: 0
Etiquetas



Modificar descripción **Modificar país** **Modificar fecha de nacimiento** **Modificar etiquetas**

Figura 26: Menú de edición de fecha de nacimiento.

Nombre de usuario: usuario
Edad: 24
País:
España

Guardar **Cancelar**

Descripción:
Añade una descripción
Número de juegos: 0
Juegos completados: 0
Juegos pendientes: 0
Juegos favoritos: 0
Etiquetas

Modificar descripción **Modificar país** Modificar fecha de nacimiento Modificar etiquetas

Figura 27: Menú de edición de país.

Nombre de usuario: usuario
Edad: 24
País: España
Descripción:
Añade una descripción

Guardar **Cancelar**

Número de juegos: 0
Juegos completados: 0
Juegos pendientes: 0
Juegos favoritos: 0
Etiquetas

Modificar descripción Modificar país Modificar fecha de nacimiento Modificar etiquetas

Figura 28: Menú de edición de descripción.

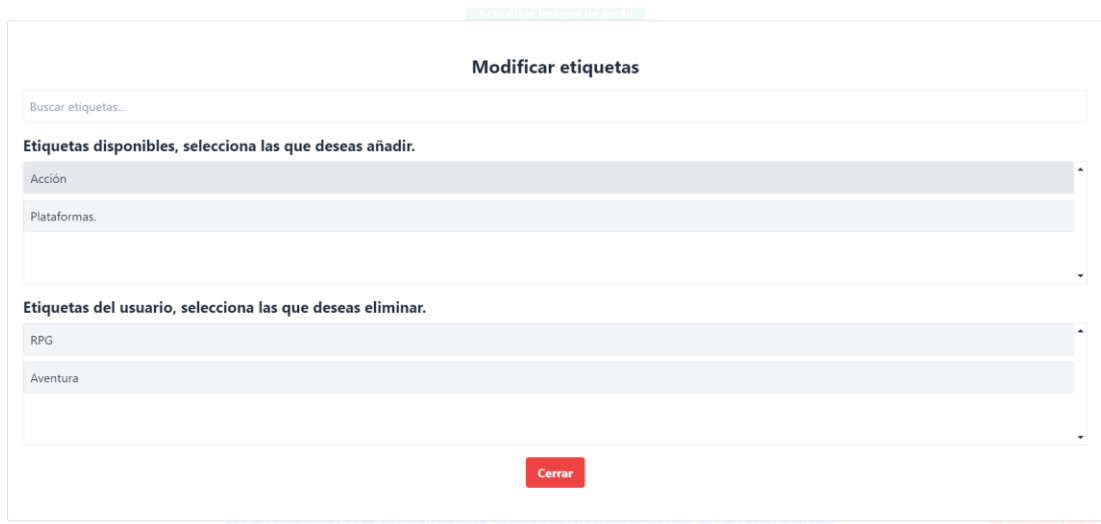


Figura 29: Menú de edición de etiquetas.

Para editar las etiquetas basta con seleccionar las que se desean añadir de las disponibles y hacer clic sobre las que se quieran eliminar.



Figura 30: Edición de redes sociales.

Las redes sociales se editan introduciendo el texto deseado junto a su enlace, si se desean añadir más pulsando sobre “Añadir otra red social” aparecerá un nuevo campo rellenable y para eliminar basta con pulsa la “X roja” que aparece a su derecha. Para confirmar los cambios se debe hacer seleccionar “Guardar redes sociales”.

Si se desea eliminar toda la información relativa al usuario se puede seleccionar el botón rojo “Eliminar mi cuenta” y confirmar que se desea eliminar permanentemente (Figura 31).

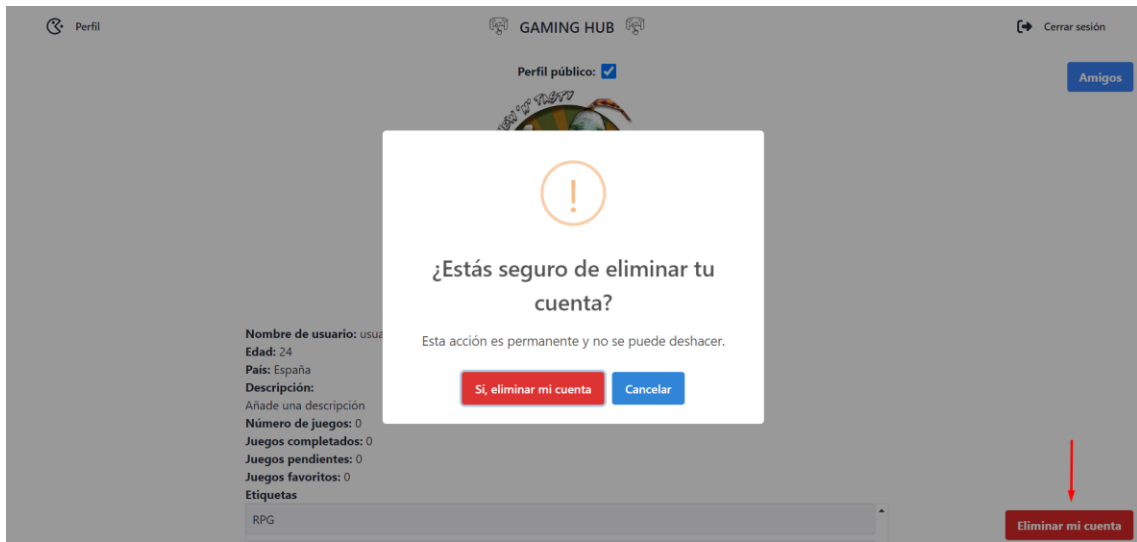


Figura 31: Eliminar permanentemente cuenta del usuario.

Otra opción que brinda el perfil es el menú de amigos (Figura32), accesible desde el botón “Amigos”, aquí se puede aceptar o rechazar solicitudes, ver el listado de amigos y acceder a sus perfiles haciendo clic encima de sus nombres, agregar nuevos usuarios (Figura 33) e incluso, buscar a otros usuarios para ver sus perfiles del mismo modo que un invitado.

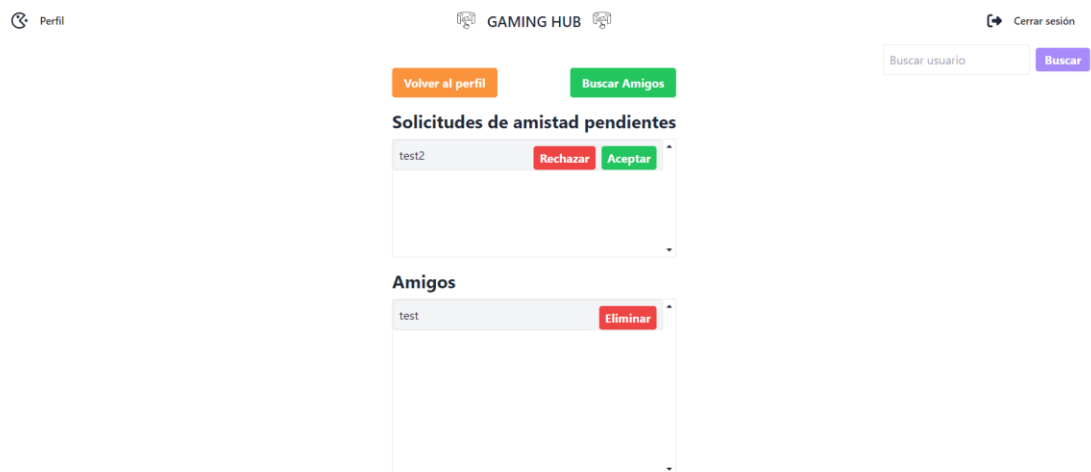


Figura 32: Menú de amigos de un usuario.

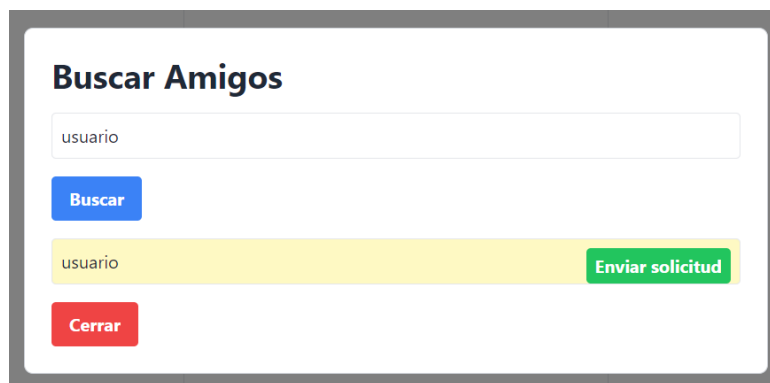


Figura 33: Agregar nuevos usuarios.

Para acceder a la biblioteca se debe seleccionar el botón “GAMING HUB” de la barra de navegación (Figura 34).

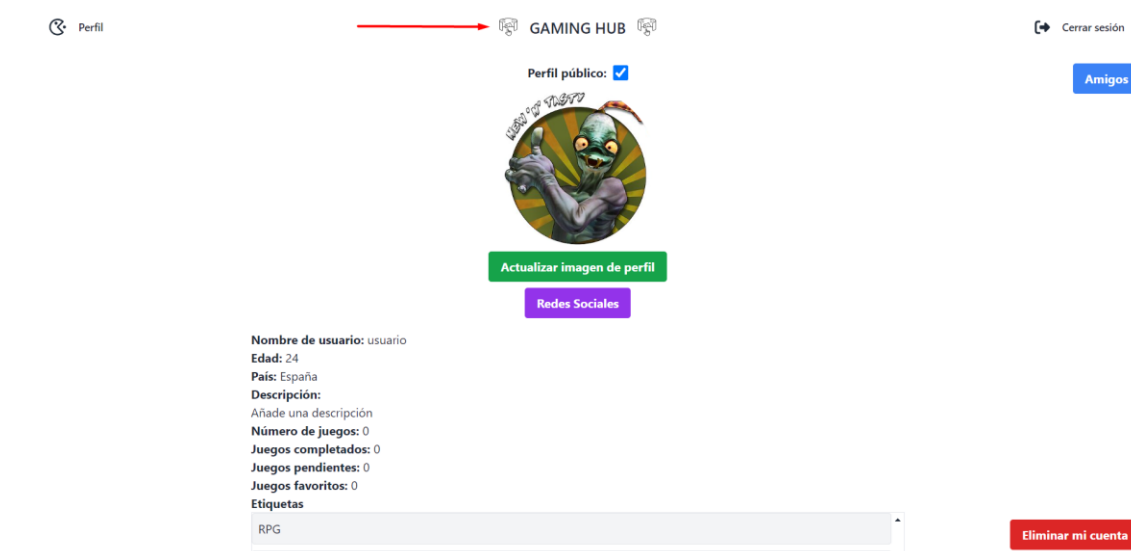


Figura 34: Acceso a librería.

A.3 Biblioteca de un usuario

Un usuario puede añadir, editar y eliminar juegos a su biblioteca, además de utilizar el filtro de búsqueda y ordenación, funcionando este del mismo modo que para un invitado explicado en el apartado A.1 de este manual.

Para añadir juego se selecciona el botón “Añadir juego” y en el buscador se introduce el título o parte de este para encontrarlo (Figura 35).

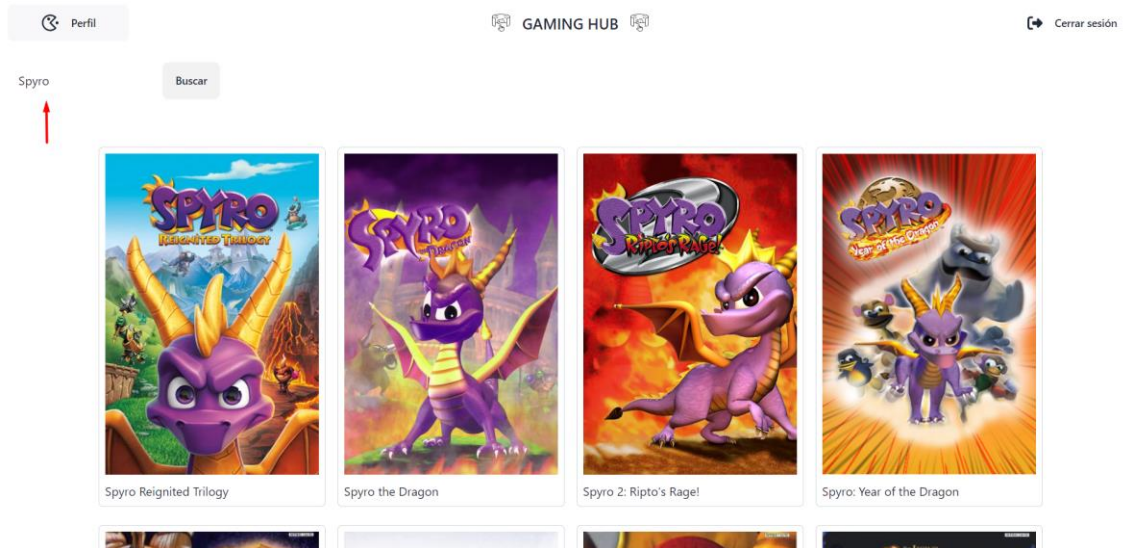


Figura 35: Búsqueda de un juego.

Una vez encontrado el juego deseado, se debe seleccionar y elegir la plataforma (Figura 36) pudiendo elegirse la opción “Otro” en la que se podrá introducir la plataforma a mano (Figura 37).

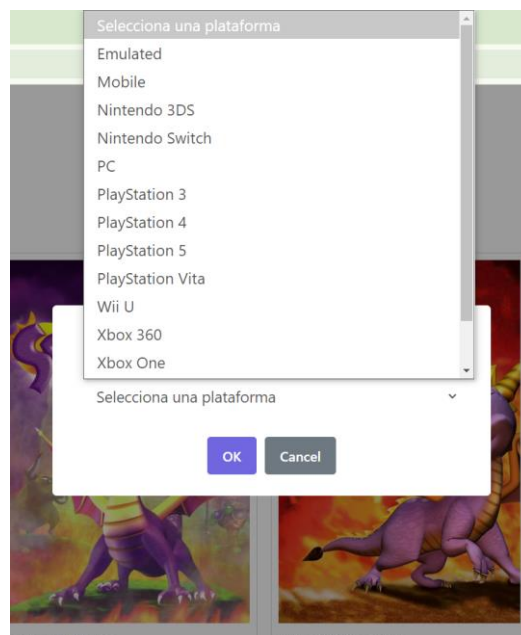


Figura 36: Seleccionar plataforma.

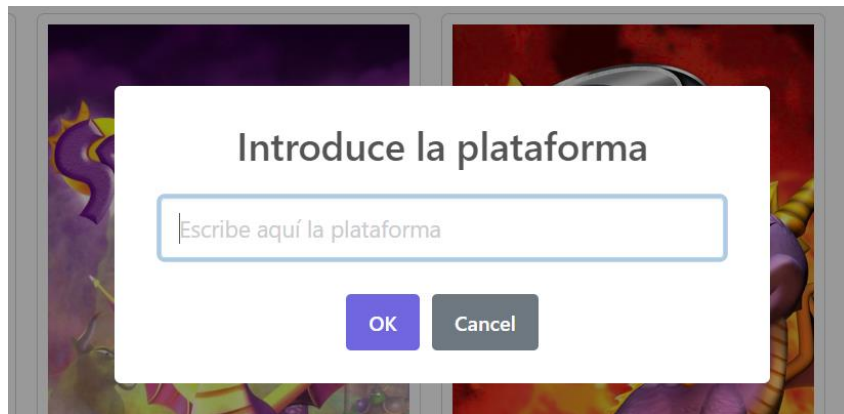


Figura 37: Plataforma personalizada.

Ahora que el juego se encuentra en la biblioteca, es posible eliminarlo seleccionando el botón con forma de basura (Figura 38), ver si es de los juegos favoritos si el corazón aparece en rojo o acceder al juego para ver y editar los detalles.



Figura 38: Vista de un juego en la librería del propio usuario.

Si se quiere descubrir a otros usuarios con ciertos gustos, ya sea por los títulos que poseen o por las etiquetas que han asignado a los mismos, puede hacerse desde el botón “Descubrir” (Figura 39), pudiendo hacer clic sobre sus nombres para acceder a sus perfiles.



Figura 39: Menú descubrir.

Si por algún casual se baja demasiado en la página, es posible pulsar el botón “Volver arriba” (Figura 40) para no tener que hacerlo manualmente.

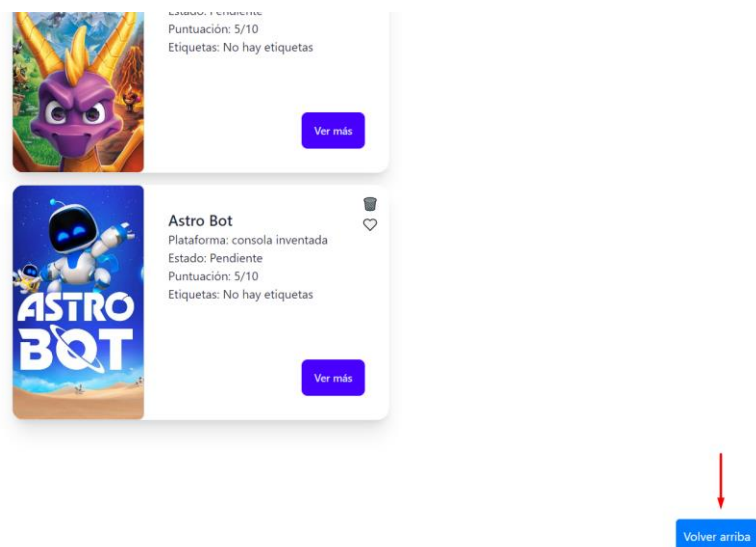


Figura 40: Botón volver arriba.

Si se pulsa el botón “Ver más” sobre un juego, se accede a la página del juego.

A.4 Juego de un usuario

La vista general del juego es similar a la mencionada en el apartado A.1 de este manual, con la diferencia de que aparece un botón con forma de corazón, el cual,

si se pulsa marca como favoritos el juego, asimismo, si es favorito y se pulsa se desmarca como favorito (Figura 41).



Figura 41: Vista de un juego propio de un usuario.

Además del botón de favorito también aparece un botón llamado “Editar” el cuál abre el menú de edición del juego, este menú permite editar todo menos la foto y el título (Figura 42).

Spyro Reignited Trilogy

Plataforma: Nintendo Switch

Pendiente

5

Añade un comentario

Historia completada: 0%

Juego completado: 0%

Logros completados: 0%

Horas totales jugadas: 0

Añadir etiquetas al juego

Select...

Guardar cambios

Cancelar edición

Figura 42: Edición de un juego.

Una vez editado, se debe pulsar en “Guardar cambios” para hacerlos efectivos.

En la Figura 43 podemos observar los cambios realizados.

Spyro Reignited Trilogy

Plataforma: Nintendo Switch

Estado: En proceso

Puntuación: 10/10

Comentarios: Plataformas divertido, continuar la semana siguiente.

Historia completada: 71%

Juego completado: 35%

Logros completados: 23%

Horas totales jugadas: 14

Etiquetas del juego

Acción Aventura Plataformas.

Editar

Figura 43: Juego editado.

Si se desean eliminar etiquetas, basta con hacer clic sobre las cruces rojas que aparecen a la izquierda de cada etiqueta.

A.5 Panel de administrador

Un administrador podrá acceder a todos los perfiles, incluso si son privados.

Además de esta cualidad, también dispone de un botón en el perfil llamado “Admin” (Figura 44) que al pulsarlo aparecen ciertas funciones especiales (Figura 45).

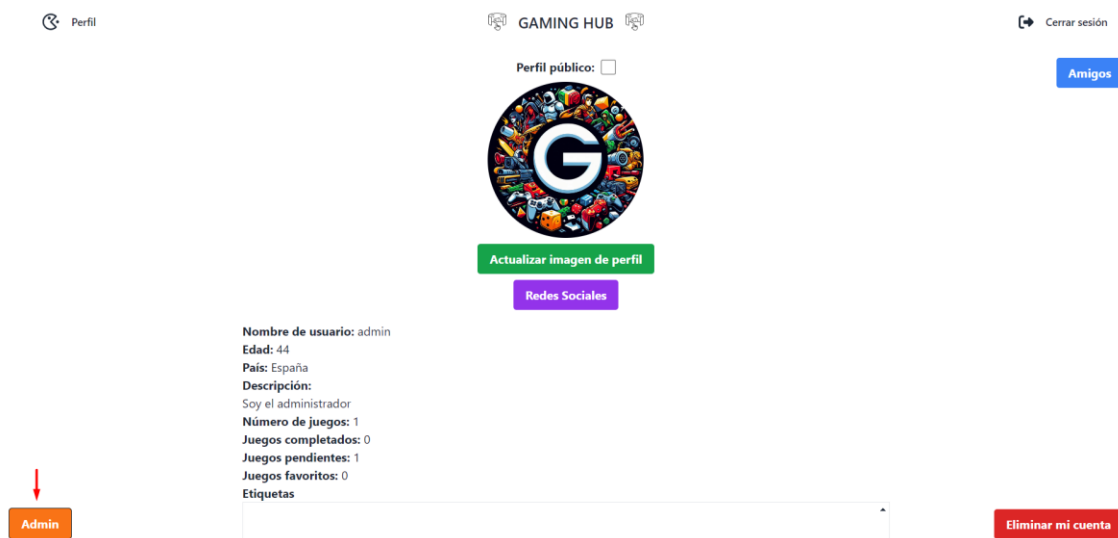


Figura 44: Botón de administrador.



Figura 45: Panel de administración.

Dentro de este panel, el administrador puede ver y buscar a los usuarios registrados utilizando el buscador habilitado para esta función. Si selecciona a un usuario podrá eliminar su cuenta para siempre, no sin antes confirmar que desea hacerlo (Figura 46).

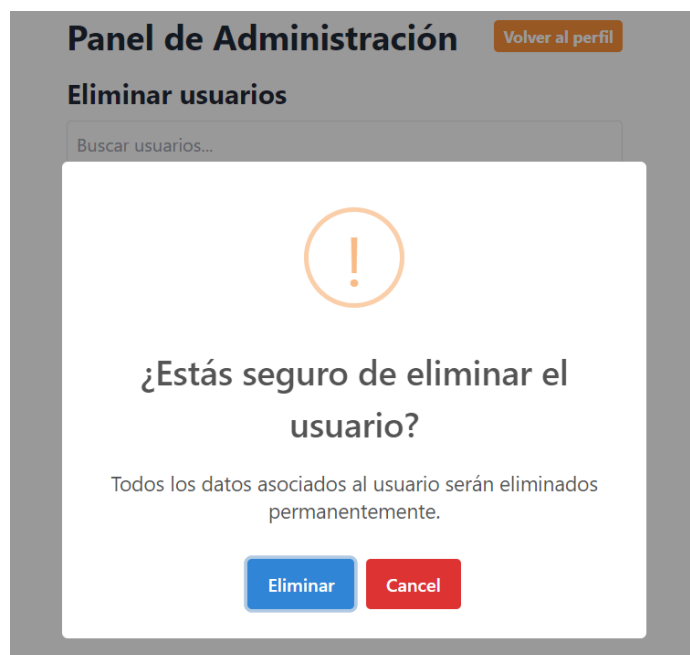


Figura 46: Eliminar usuario desde panel de administrador.

Además de las funciones de usuarios, también dispone de la posibilidad de buscar, añadir y eliminar etiquetas de uso común, pues las etiquetas que pueden asignar los usuarios a sus perfiles y juegos están predefinidas por los administradores.

Para añadir una etiqueta debe escribirlo en la casilla habilitada junto al botón “Añadir” y pulsar este.

Si se desea eliminar una etiqueta, puede buscarla en el buscador de etiquetas o buscar manualmente en la lista la etiqueta, una vez encontrada solo deberá pulsar sobre el botón “Eliminar” posicionado junto a ella. Esto invocará un aviso para confirmar que desea eliminarla (Figura 47) y al confirmar, esta etiqueta desaparecerá de la lista y de todos los lugares de la aplicación donde se utilizaba.

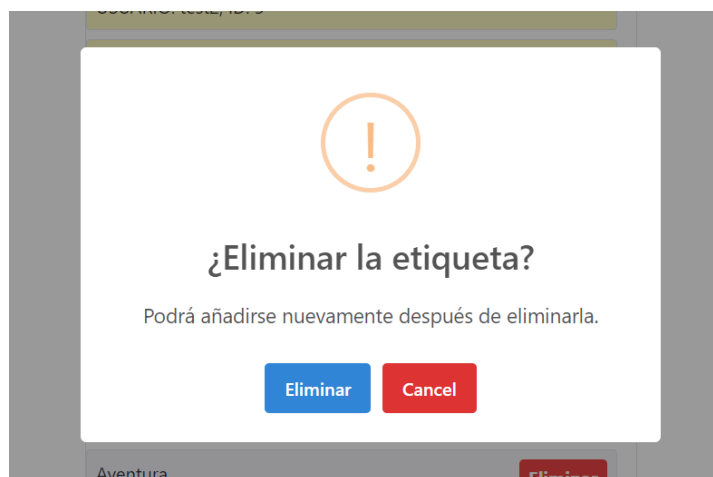


Figura 47: Aviso de eliminación de etiqueta.

Apéndice B

Manual de Instalación

Para poder ejecutar GamingHub de forma local es necesario realizar varias instalaciones antes de poder ejecutarse, a continuación, podrá ver una guía detallada de los pasos a seguir.

1. Instalar Node.js (v20.13.0 preferiblemente)

Se debe acceder a la página web oficial de Node [5] como se muestra en la Figura 48 y seguir las indicaciones de instalación.

Si se prefiere, se puede acceder directamente a la lista de ejecutables de la versión v20.13.0 [23].

Download Node.js®

Download Node.js the way you want.

[Package Manager](#) [Prebuilt Installer](#) [Prebuilt Binaries](#) [Source Code](#)

Install Node.js on using

```
1 # installs fnm (Fast Node Manager)
2 winget install Schniz.fnm
3
4 # configure fnm environment
5 fnm env --use-on-cd | Out-String | Invoke-Expression
6
7 # download and install Node.js
8 fnm use --install-if-missing 20
9
10 # verifies the right Node.js version is in the environment
11 node -v # should print `v20.17.0`
12
13 # verifies the right npm version is in the environment
14 npm -v # should print `10.8.2`
```

PowerShell


 Copy to clipboard

Figura 48: Página oficial de Node.js.

2. Instalación de la aplicación GamingHub

- En primer lugar, se debe descomprimir el proyecto.
- Una vez dentro de la carpeta, se debe escribir "cmd" en la barra de búsqueda (Figura 49) y pulsar para buscar, esto ejecutara la consola de comandos desde la ubicación del proyecto.

También es posible abrir una consola y desplazarnos a la raíz del proyecto mediante el comando `cd "ruta"`.

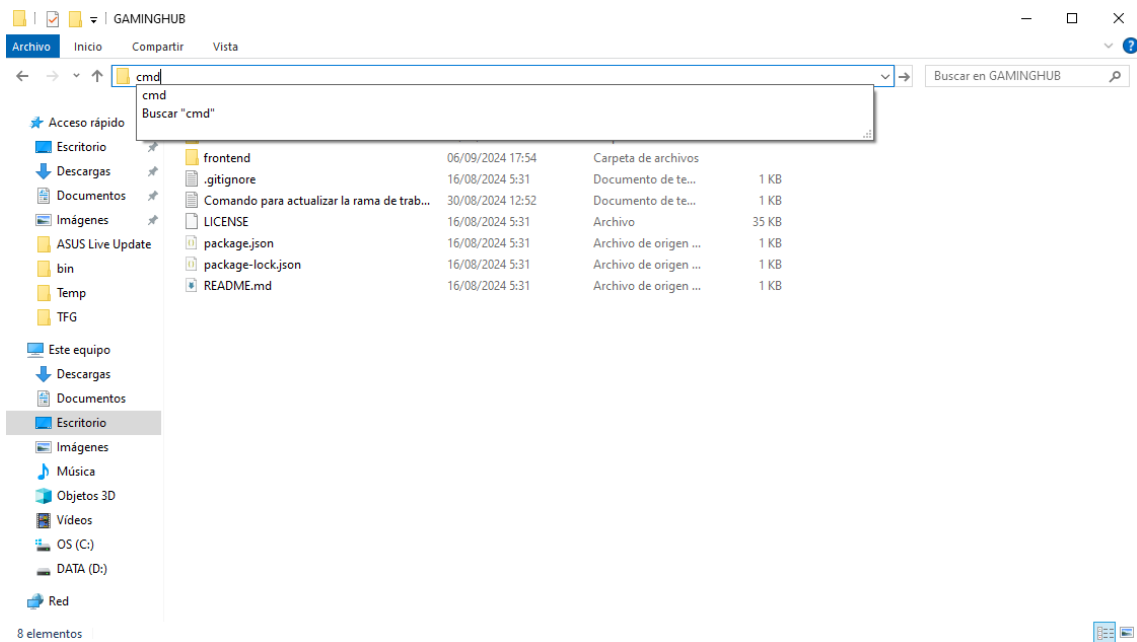


Figura 49: Acceder a la consola de comandos desde el proyecto.

El proyecto está configurado para instalar todas las dependencias necesarias tanto para backend como para frontend, es por ello que solo habrá que ir a dichas carpetas y solicitar la instalación.

- c. Escribimos "cd backend" y posteriormente "npm install", esto instalará las dependencias necesarias para ejecutar el backend. Una vez finalizado, será posible ejecutar el backend escribiendo "npm run dev". Si no hubo errores, debería ver algo similar a la Figura 50.

```
C:\Users\Nacho\Desktop\Universidad\TFG\GAMINGHUB\backend>npm run dev
> backend@1.0.0 dev
> nodemon index.js

[nodemon] 3.1.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
Server running on port http://localhost:4000
```

Figura 50: Backend funcionando correctamente.

- d. Para el frontend se deben realizar los anteriores pasos con la única diferencia de que escribiremos "cd frontend", el resto del es idéntico. Si no hubo errores, debería ver algo similar a la Figura 51.

```
C:\Users\Nacho\Desktop\Universidad\TFG\GAMINGHUB\frontend>npm run dev
> frontend@0.0.0 dev
> vite

VITE v5.2.13 ready in 822 ms

→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

Figura 51: Frontend funcionando correctamente.

- e. La URL que aparece en "Local:" será la que se debe introducir en el navegador para acceder a GamingHub. (En la Figura 51 la URL sería "http://localhost:5173/")

3. Instalación de la base de datos.

Para la instalación de la base de datos, puede hacerse de diversas formas, lo importante es el script que la genera. Sin embargo, a continuación, se proporciona una sugerencia de instalación:

- Descargar e instalar MySQL Installer [21].
- Descargar e instalar MySQL Workbench [22].
- Crear una nueva conexión similar a la Figura 52.
- Crear un nuevo esquema (Figura 53).
- Ejecutar el script que se encuentra en:
"GAMINGHUB\backend\mysql\Scripts" (Figura 54).

Welcome to MySQL Workbench

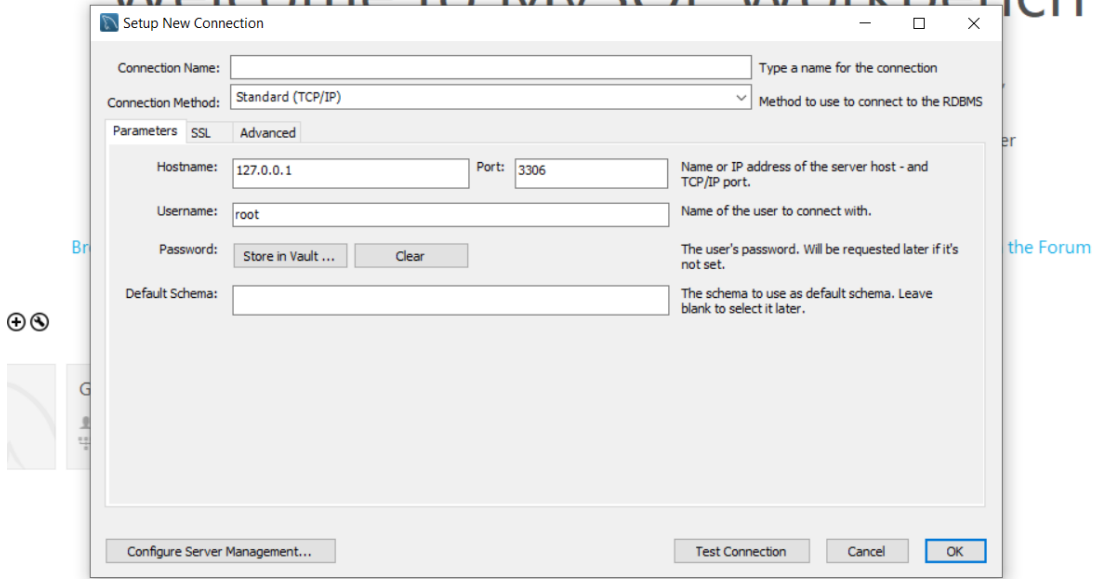


Figura 52: Crear nueva conexión.

Apply SQL Script to Database

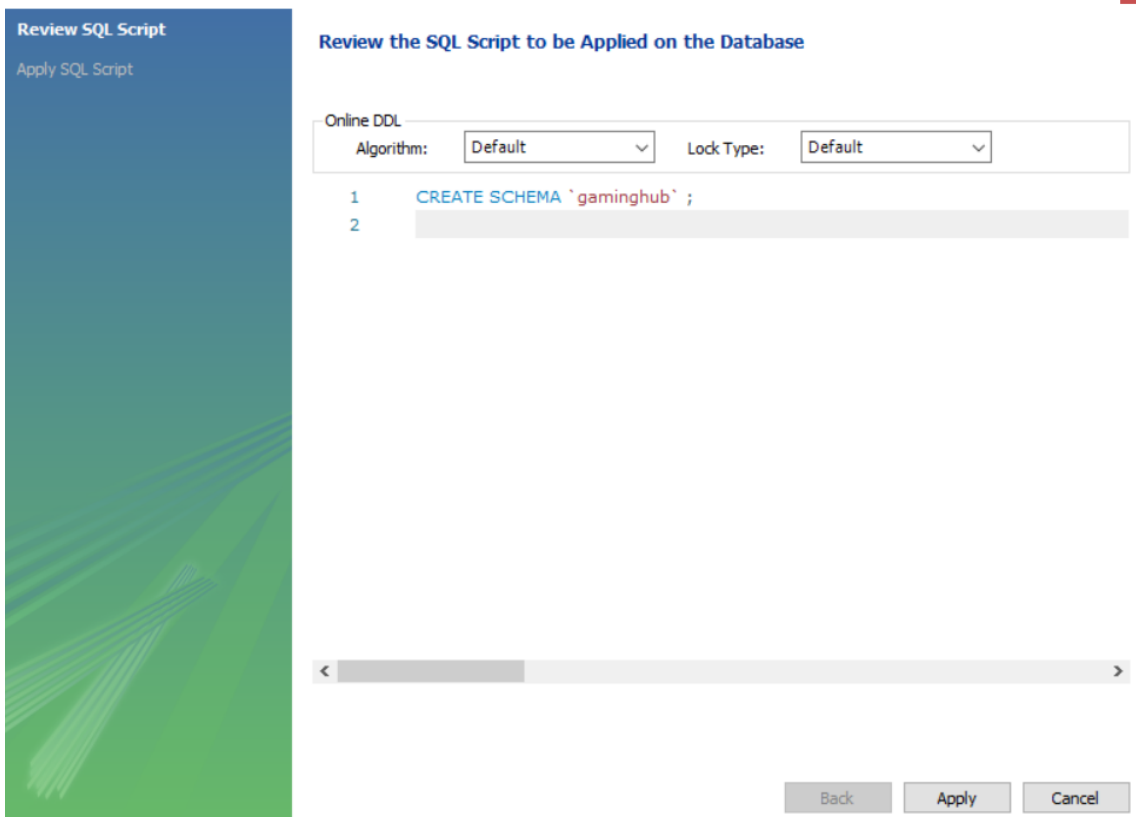
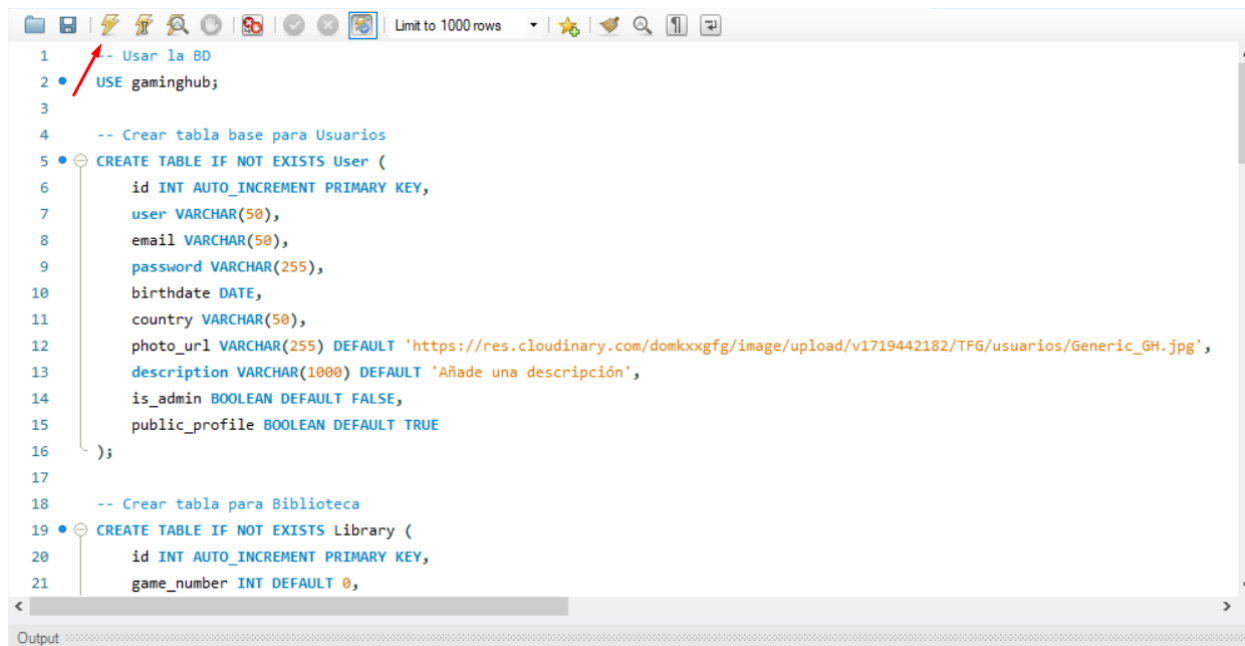


Figura 53: Crear nuevo esquema.



```
1 -- Usar la BD
2 • USE gaminghub;
3
4 -- Crear tabla base para Usuarios
5 • CREATE TABLE IF NOT EXISTS User (
6     id INT AUTO_INCREMENT PRIMARY KEY,
7     user VARCHAR(50),
8     email VARCHAR(50),
9     password VARCHAR(255),
10    birthdate DATE,
11    country VARCHAR(50),
12    photo_url VARCHAR(255) DEFAULT 'https://res.cloudinary.com/domkxxgfg/image/upload/v1719442182/TFG/usuarios/Generic_GH.jpg',
13    description VARCHAR(1000) DEFAULT 'Añade una descripción',
14    is_admin BOOLEAN DEFAULT FALSE,
15    public_profile BOOLEAN DEFAULT TRUE
16 );
17
18 -- Crear tabla para Biblioteca
19 • CREATE TABLE IF NOT EXISTS Library (
20     id INT AUTO_INCREMENT PRIMARY KEY,
21     game_number INT DEFAULT 0,
```

Output

Figura 54: Crear las tablas y relaciones con el Script.



UNIVERSIDAD
DE MÁLAGA

| **uma.es**

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA