




Videogames-Library. Plataforma de interacción de usuarios con juegos y otros usuarios.



VIDEOGAME LIBRARY


GENERAL

- Home
- Mis Juegos
- Wishlist

Configuración


Juan

Logout




Elden Ring
A fantasy action RPG.

59.99



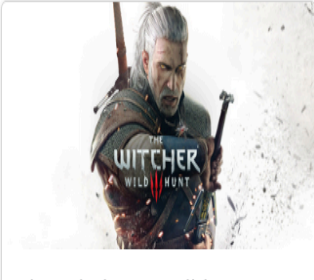
Diablo IV
An action RPG.

34.99




DOTA 2
A multiplayer online battle arena game.


Free to play



The Witcher 3: Wild Hunt
An open-world action RPG.



Cyberpunk 2077
A futuristic open-world RPG.



Hades
A roguelike dungeon crawler.



ÍNDICE:

- [1. Introducción](#)
- [2. Descripción del Proyecto](#)
- [3. Motivación que te lleva a realizar este proyecto:](#)
- [4. Stack tecnológico utilizado:](#)
 - [Backend](#)
 - [Base de Datos](#)
 - [Frontend](#)
- [5. Lista de requisitos funcionales del proyecto:](#)
- [6. Breve análisis/comparativa con las alternativas del mercado](#)
 - [Análisis de herramientas similares](#)
- [7. Justificación del proyecto](#)
- [8. Definición API REST publicación servicios](#)
- [9. Manual de Despliegue](#)
- [10. Postmortem y conclusiones del proyecto](#)
 - [Viabilidad de Puesta en Marcha Real](#)
 - [Posibles Mejoras](#)



1. Introducción

Este proyecto tiene como objetivo desarrollar una plataforma web que facilite la interacción de los usuarios con los juegos y entre ellos mismos. La plataforma permitirá a los usuarios explorar diversos juegos, acceder a información detallada sobre ellos, dejar comentarios y calificaciones, y compartir sus experiencias con otros usuarios. La creación de esta plataforma surge de la necesidad de proporcionar una experiencia de usuario más rica y envolvente en el ámbito de los videojuegos, donde la comunidad y la interacción social juegan un papel fundamental.

El objetivo principal de esta aplicación es mejorar la experiencia del usuario al interactuar con juegos, ofreciendo una interfaz intuitiva y funcional. Esto se logrará mediante el uso de tecnologías modernas y prácticas de desarrollo de software robustas. Además, este proyecto sirve como una excelente oportunidad para aplicar y consolidar los conocimientos adquiridos en el grado superior de desarrollo de aplicaciones web, abarcando desde el diseño de bases de datos hasta la implementación de la lógica del servidor y la creación de interfaces de usuario atractivas.

2. Descripción del Proyecto

Este proyecto tiene como objetivo desarrollar una plataforma web que facilite la interacción de los usuarios con los juegos y entre ellos mismos. La plataforma permitirá a los usuarios explorar diversos juegos, acceder a información detallada sobre ellos, dejar comentarios y calificaciones, y compartir sus experiencias con otros usuarios. La creación de esta plataforma surge de la necesidad de proporcionar una experiencia de usuario más rica y envolvente en el ámbito de los videojuegos, donde la comunidad y la interacción social juegan un papel fundamental.

El objetivo principal de esta aplicación es mejorar la experiencia del usuario al interactuar con juegos, ofreciendo una interfaz intuitiva y funcional. Esto se logrará mediante el uso de tecnologías modernas y prácticas de desarrollo de software robustas. Además, este proyecto sirve como una excelente oportunidad para aplicar y consolidar los conocimientos adquiridos en el grado superior de desarrollo de aplicaciones web, abarcando desde el diseño de bases de datos hasta la implementación de la lógica del servidor y la creación de interfaces de usuario atractivas.



3. Motivación que te lleva a realizar este proyecto:

La motivación principal detrás de este proyecto es la pasión por los videojuegos y el deseo de mejorar la experiencia de los jugadores a través de una plataforma innovadora y fácil de usar. Este proyecto ofrece la oportunidad de aplicar conocimientos teóricos en un contexto práctico, abarcando tecnologías y metodologías modernas en el desarrollo de software. La creación de una plataforma que mejore la interacción entre jugadores y la exploración de juegos contribuirá significativamente a la comunidad de jugadores y proporcionará una valiosa experiencia de aprendizaje y desarrollo personal y profesional.

4. Stack tecnológico utilizado:

Backend

- **Java con Spring Boot:** Se utiliza para la creación del backend de la aplicación debido a su robustez, seguridad y amplia adopción en la industria. Spring Boot facilita la configuración y el desarrollo de aplicaciones Java al proporcionar un conjunto de herramientas y funcionalidades preconfiguradas.

Base de Datos

- **MySQL:** Una base de datos relacional de alto rendimiento y escalabilidad, ideal para manejar grandes volúmenes de datos y asegurar la integridad y consistencia de los mismos. Se utilizará para almacenar datos de usuarios, juegos, comentarios y demás entidades de la aplicación.

Frontend

- **Angular:** Framework de desarrollo web moderno y eficiente que permite crear aplicaciones web de alto rendimiento. Angular facilita la creación de interfaces de usuario dinámicas y reactivas, utilizando componentes modulares.
- **Bootstrap:** Un framework CSS que facilita el diseño de interfaces de usuario atractivas y responsivas. Bootstrap se utiliza junto con Angular para asegurar una experiencia de usuario consistente y estilizada.
- **SASS:** Un preprocesador CSS que permite escribir estilos de manera más eficiente y organizada.
- **Typescript:** Un superset de JavaScript que aporta tipado estático y otras características avanzadas, mejorando la calidad y mantenibilidad del código.



5. Lista de requisitos funcionales del proyecto:

- RF01 - Sistema de registro e inicio de sesión:
 1. Los usuarios deben poder registrarse en la plataforma proporcionando un nombre de usuario, correo electrónico y contraseña.
 2. Los usuarios registrados deben poder iniciar sesión utilizando su nombre de usuario y contraseña.
- RF02 - Sistema de cierre de sesión:
 1. Los usuarios autenticados deben poder cerrar sesión en la plataforma.
- RF03 - Exploración de juegos:
 1. Los usuarios autenticados deben poder explorar una lista de juegos disponibles, obteniendo información detallada sobre cada juego, como título, descripción, género, calificación, etc.
- RF04 - Comentarios y calificaciones:
 1. Los usuarios autenticados deben poder dejar comentarios y calificaciones en los juegos.
 2. Los comentarios deben mostrar el nombre de usuario del autor, la fecha de publicación y el contenido del comentario.
- RF05 - Edición de perfil de usuario:
 1. Los usuarios autenticados deben poder editar su perfil, incluyendo información como nombre, correo electrónico, contraseña, etc.
- RF06 - Búsqueda de juegos:
 1. Los usuarios autenticados deben poder buscar juegos por título, género, calificación, etc.
- RF07 - Favoritos:
 1. Los usuarios autenticados deben poder marcar juegos como favoritos para acceder rápidamente a ellos en el futuro.
- RF08 - Socialización:
 1. Los usuarios autenticados deben poder seguir a otros usuarios y ver sus actividades, como los juegos que están jugando, los comentarios que han dejado, etc.
- RF09 - Integración con la Base de Datos:
 1. La plataforma debe integrarse con la base de datos para obtener información actualizada sobre los juegos, como imágenes, descripciones, etc.



VideogameLibrary - Adrián Tellado

- RF10 - Autenticación OAuth con Steam, SSL/TLS y JWT:
 1. Los usuarios deben poder iniciar sesión en la plataforma a través de OAuth.
 2. Se debe implementar SSL/TLS para garantizar la seguridad de la comunicación entre el cliente y el servidor. Se utilizará JWT para la transmisión segura de información de autenticación entre el cliente y el servidor.

- RF11 - Gestión de sesiones de usuario:
 1. La plataforma debe gestionar adecuadamente las sesiones de usuario, incluyendo la expiración de la sesión, la recuperación de contraseña, etc.

- RF12 - Gestión de roles de usuario:
 1. Se deberán de implementar los distintos roles, como usuario administrador y usuario básico, limitando el acceso a los usuarios básicos a algunas funcionalidades, etc.

- RF12 - Notificaciones:
 1. Los usuarios deben recibir notificaciones sobre nuevos comentarios en sus juegos favoritos, nuevas amistades, etc.

6. Breve análisis/comparativa con las alternativas del mercado

Análisis de herramientas similares

Existen varias plataformas en el mercado que ofrecen servicios similares, como Steam, IGN, y Metacritic. A continuación, se presenta una comparación de estas plataformas con la aplicación que se desarrollará:

- **Steam:**
 - **Puntos fuertes:** Gran biblioteca de juegos, comunidad activa, funcionalidades de comentarios y calificaciones, actualizaciones constantes.
 - **Puntos débiles:** Puede ser abrumador para nuevos usuarios, interfaz algo saturada.
 - **Nuestra propuesta:** Proveer una interfaz más limpia y accesible, centrada en la simplicidad y la usabilidad.

- **IGN:**
 - **Puntos fuertes:** Contenido editorial de alta calidad, reseñas profesionales, comunidad activa.
 - **Puntos débiles:** Enfoque en noticias y reseñas puede dejar de lado la interacción directa entre usuarios.



VideogameLibrary - Adrián Tellado

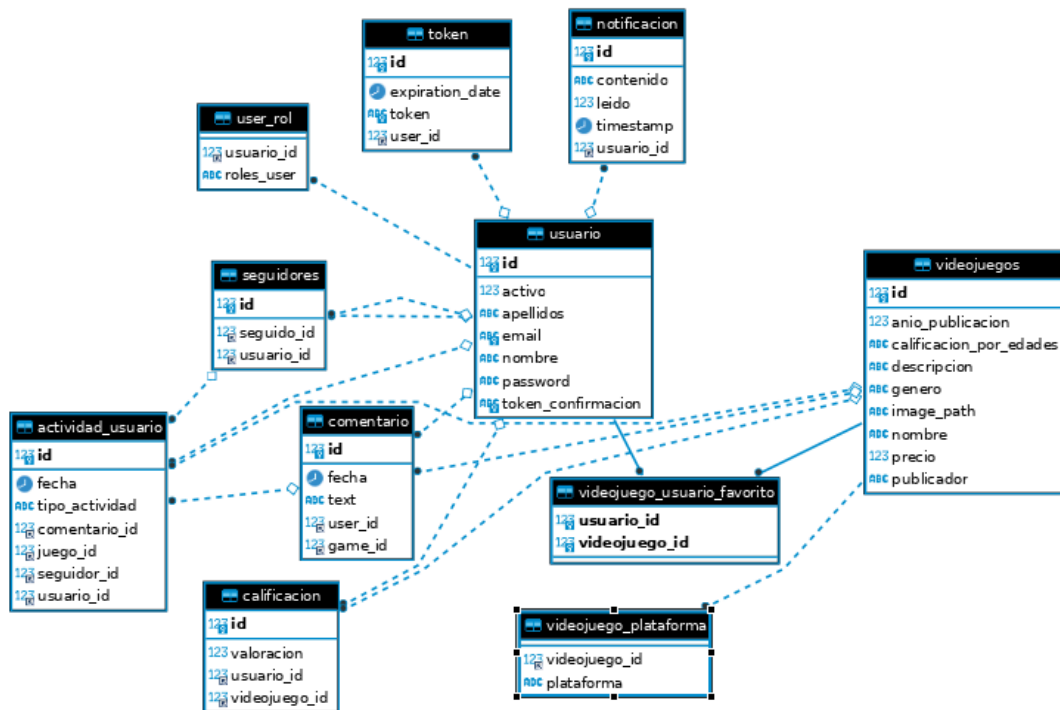
- **Nuestra propuesta:** Fomentar más la interacción directa entre usuarios, comentarios y calificaciones más prominentes.
- **Metacritic:**
 - **Puntos fuertes:** Sistema de calificaciones agregadas, amplio rango de juegos y reseñas.
 - **Puntos débiles:** Interfaz de usuario anticuada, falta de interacción directa entre usuarios.
 - **Nuestra propuesta:** Modernizar la interfaz de usuario y fomentar la interacción entre usuarios.

7. Justificación del proyecto

Este proyecto se justifica por la necesidad de una plataforma que combine lo mejor de las funcionalidades existentes en el mercado, mejorando la experiencia de usuario con un enfoque en la simplicidad, usabilidad, y una mayor interacción social. Además, permite aplicar de manera práctica los conocimientos adquiridos en el desarrollo de aplicaciones web.



8. Modelo Entidad-Relación



User: Esta entidad representa la información de un usuario en el sistema, incluyendo su nombre, apellidos, correo electrónico, contraseña y el campo booleano activo que indica si el correo electrónico ha sido confirmado o no. Además, mantiene relaciones con otras entidades, como comentarios realizados por el usuario, videojuegos marcados como favoritos, notificaciones recibidas, actividades realizadas en el sistema, seguidores, tokens de autenticación y calificaciones proporcionadas a videojuegos por el usuario.

Videojuegos: La entidad Videojuegos almacena información detallada sobre los videojuegos, como su nombre, género, descripción, año de publicación, calificación por edades, publicador y plataforma. Además, esta entidad está relacionada con otras entidades, como comentarios hechos sobre el videojuego y usuarios que lo han marcado como favorito y actividades realizadas por los usuarios relacionadas con el videojuego.

Calificacion: La entidad Calificacion registra la calificación numérica dada por un usuario a un videojuego específico. Está relacionada con la entidad Usuario, que representa al usuario que realizó la calificación, y la entidad Videojuegos, que representa el videojuego calificado.

Seguidores: Esta entidad modela la relación de seguimiento entre dos usuarios en el sistema. Es una relación reflexiva que afecta a dos usuarios: el seguidor y el usuario seguido. Además, la entidad Seguidores está relacionada con la entidad ActividadUsuario, que registra la actividad de seguir a otro usuario.



VideogameLibrary - Adrián Tellado

Notificacion: La entidad Notificacion representa una notificación enviada a un usuario. Contiene información sobre el contenido de la notificación, la fecha de envío y si ha sido leída por el usuario destinatario. Mantiene una relación con la entidad Usuario, que representa al usuario destinatario de la notificación.

Actividad_usuario: Esta entidad registra las actividades realizadas por los usuarios en el sistema. Almacena información sobre el tipo de actividad realizada y la fecha en que se llevó a cabo. Mantienen una relación con la entidad Usuario, que representa al usuario que realizó la actividad, y también está relacionada con otras entidades como Comentario, Calificacion, Seguidores y Videojuegos, para registrar actividades específicas de los usuarios en el sistema.

Token: La entidad Token guarda los tokens de autenticación asociados a los usuarios en el sistema, incluyendo el token de autenticación y la fecha de

9. Definición API REST publicación servicios

La API REST desarrollada en Spring Boot expone diversos endpoints para gestionar los recursos de la aplicación. A continuación se detallan los endpoints de los diferentes controladores:

Documentación completa:

-URL de Postman:

AuthenticationController

Todos los métodos indicados en este controlador se forman desde la URL base:

“ <http://localhost:8080/api/v1/auth> ”

ENDPOINT	DESCRIPCIÓN
POST /signup	
POST /signin	
POST /logout	



VideojuegosController

Todos los métodos indicados en este controlador se forman desde la URL base:

“ <http://localhost:8080/api/v1/videojuegos> ”

ENDPOINT	DESCRIPCIÓN
GET /	Método para mostrar todos los videojuegos paginados con el token de un usuario registrado (usuarios ADMIN y NO ADMIN).
GET /todos	Método para mostrar todos los videojuegos sin paginado con el token de un usuario registrado (usuarios ADMIN y NO ADMIN).
GET /{id}	Método para obtener un videojuego por su id con el token de un usuario registrado (usuarios ADMIN y NO ADMIN).
GET /search	Método para buscar videojuegos por su nombre con el token de un usuario registrado (usuarios admin y no admin). Devuelve una lista de todos los videojuegos que contengan los caracteres de búsqueda especificados.
GET /{id}/calificaciones	Método para obtener las calificaciones de un videojuego por id del juego con el token de un usuario registrado (usuarios ADMIN y NO ADMIN).
GET /{id}/comentarios	Método para obtener los comentarios de un videojuego por id del juego con el token de un usuario registrado (usuarios ADMIN y NO ADMIN).
GET /{id}/usuarios-favoritos	Método para traer los usuarios que han seleccionado un videojuego específico como favorito, basándose en el ID del



	videojuego con el token de un usuario registrado (usuarios ADMIN y NO ADMIN).
POST /	Método para añadir un videojuego. Sólo para usuarios ADMIN.
PUT /{id}	Método para editar un videojuego. Sólo para usuarios ADMIN.
DELETE /{id}	Método para eliminar un videojuego. Sólo para usuarios ADMIN.

UsuarioController

Todos los métodos indicados en este controlador se forman desde la URL base:
“ <http://localhost:8080/api/v1/usuario> ”

ENDPOINT	DESCRIPCIÓN
GET /	Método para obtener todos los usuarios. (usuarios ADMIN y NO ADMIN). Los usuarios NO ADMIN sólo obtienen información reducida de los demás usuarios como id, nombre y email, mientras que los ADMIN obtienen más información.
GET /{id}	Método para obtener los datos de un usuario (usuarios ADMIN y NO ADMIN). Los usuarios NO ADMIN sólo obtienen información reducida de los demás usuarios como id, nombre y email, mientras que los ADMIN obtienen más información.
GET /{id}/videojuegos-favoritos	Método para obtener los videojuegos favoritos de un usuario por el id del usuario.
POST /	Método para añadir un usuario. Sólo para usuarios ADMIN.
POST /{id}/videojuegos-favoritos/{idVideojuego}	Método para añadir un videojuego a la lista de favoritos de un usuario por el id



	del usuario e id del juego (usuarios ADMIN y NO ADMIN).
PUT /{id}	Método para editar un usuario. Sólo para usuarios ADMIN.
DELETE /{id}	Método para eliminar un usuario. Sólo para usuarios ADMIN.
DELETE /{id}/videojuegos-favoritos/{videoguego}	Método para eliminar un videojuego de la lista de favoritos de un usuario por el id del usuario e id del juego (usuarios ADMIN y NO ADMIN).

ComentarioController

Todos los métodos indicados en este controlador se forman desde la URL base:

“ <http://localhost:8080/api/v1/comentarios> ”

ENDPOINT	DESCRIPCIÓN
GET /	Método para obtener todos los comentarios (usuarios ADMIN y NO ADMIN).
GET /{id}	Método para obtener un comentario por el id del comentario (usuarios ADMIN y NO ADMIN).
POST /	Método para añadir un comentario (usuarios ADMIN y NO ADMIN).
PUT /{id}	Método para editar un comentario. Los usuarios ADMIN pueden editar cualquier comentario, pero los usuarios NO ADMIN sólo pueden editar los suyos propios.
DELETE /{id}	Método para eliminar un comentario. Los usuarios ADMIN pueden eliminar



cualquier comentario, pero los usuarios NO ADMIN sólo pueden eliminar los suyos propios.

10. Manual de Despliegue

Para ejecutar el proyecto, se debe seguir los siguientes pasos:

0. Liberar los puertos usados para los servicios en el DockerCompose o cambiar los puertos que usan (Opcional)

*Si se quiere parar el servidor mysql por tenerlo en el puerto 3306, ejecutar el comando:

```
sudo systemctl stop mysql
```

Para este punto, en Ubuntu podemos usar el comando: `sudo fuser -k 8080/tcp` (sustituir el 8080 por el puerto que se quiera liberar).

```
usuario@ubuntu:~$ sudo fuser -k 8080/tcp
[sudo] contraseña para usuario:
usuario@ubuntu:~$ sudo fuser -k 3306/tcp
3306/tcp:          1149
usuario@ubuntu:~$ sudo fuser -k 8080/tcp
usuario@ubuntu:~$ sudo fuser -k 4200/tcp
usuario@ubuntu:~$
```

1. Hacer un git clone/Descargar el repositorio.
2. Abrir la terminal y dirigirse al directorio raíz del repositorio en la terminal (donde se encuentra el archivo docker-compose.yml).

3. Ejecutar el comando:

```
docker-compose build
```

4. Ejecutar el comando:

```
docker-compose up
```



5. Ya debería de estar creado el contenedor.

*Extra. Para acceder al servicio mysql del contenedor y ejecutar el script de inserción de datos de videojuegos.

- Ejecutar, una vez arrancado el contenedor mysql con el servicio, el comando:

```
docker exec -it <nombre_del_contenedor_mysql> mysql -uroot -proot
```

- Meter los datos del script ubicado en la carpeta resources de la api en la carpeta Springboot

11. Postmortem y conclusiones del proyecto

Viabilidad de Puesta en Marcha Real

La plataforma desarrollada tiene un gran potencial para ser lanzada en un entorno de producción real. Sin embargo, antes de su lanzamiento, sería importante realizar pruebas exhaustivas, implementar mejoras de seguridad y considerar la escalabilidad para manejar un gran número de usuarios simultáneos.

Posibles Mejoras

- **Funcionalidades adicionales:** Se podrían agregar características no implementadas como recomendaciones personalizadas y notificaciones, seguimiento de usuarios, etc.
- **Optimización de rendimiento:** Mejorar el rendimiento de la aplicación mediante técnicas de caching y optimización de consultas a la base de datos.
- **Mejorar la seguridad:** Implementar medidas de seguridad adicionales, como autenticación de dos factores y monitoreo de actividades sospechosas.

Conclusión:



VideogameLibrary - Adrián Tellado

En conclusión, el desarrollo de esta plataforma no solo ha permitido aplicar y consolidar conocimientos técnicos, sino que también ha proporcionado una experiencia valiosa en la gestión de proyectos y desarrollo de software. Pese a la dificultad añadida de ser mi primer proyecto con el framework de angular y también el primer contacto con typescript, he realizado un gran trabajo. Con algunas mejoras y ajustes, la plataforma tiene el potencial de convertirse en una herramienta útil y apreciada por la comunidad de jugadores.