



Institut Puig Castellar
Santa Coloma de Gramenet



CHEST GAMES

(Projecte de desenvolupament)
CFGS Desenvolupament d'Aplicacions Web

**Autors: Jose Padilla Méndez
Christian Arias Lopez
Grup: A
Curs acadèmic: DAW 2**

A) Creative Commons:



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Resum del projecte

Temàtica del projecte

La temàtica del projecte es la creació de una plataforma de comerç electrònic especialitzada en la venda de videojuegos. El objectiu es oferir un espai segur i fiable on els usuaris puguin adquirir els seus videojuegos sense preocupar-se per perdre la llicència o l'accés al producte.

Objectiu del projecte

El objectiu principal es desenvolupar una tenda en línia que permeti la compra de videojuegos digitals mitjançant un sistema de propietat permanent. Això significa que, una vegada realitzat el pagament, l'usuari tindrà accés al joc de forma indefinida i sense necessitat d'utilitzar cap tipus de llicència o clau d'activació.

Metodologia seguida per aconseguir l'objectiu

Per aconseguir aquest objectiu, s'ha seguit una metodologia basada en el desenvolupament web àgil. En primer lloc, s'ha dissenyat l'arquitectura del sistema i la interfície d'usuari. A continuació, s'ha implementat el backend per gestionar les compres, els usuaris i l'accés als videojuegos. Finalment, s'han realitzat proves per garantir la seguretat de les transaccions i la correcta disponibilitat dels jocs després de la compra.

Resum de les conclusions

El projecte ha permès demostrar la viabilitat d'una tenda de videojuegos sense dependència de llicències externes. S'ha aconseguit crear un sistema estable, segur i fàcil d'utilitzar que garanteix als usuaris la possessió permanent dels jocs adquirits. Això representa una alternativa innovadora respecte als models tradicionals de distribució digital.

Paraules clau (entre 4 i 8):

- Tenda de videojuegos
- Plataforma digital de jocs
- Compra i descàrrega de videojuegos
- Marketplace gamer

- Abstract

Project Theme

The theme of the project is the creation of an e-commerce platform specialized in the sale of video games. The goal is to offer a safe and reliable space where users can purchase their video games without worrying about losing their license or access to the product.

Project Objective

The main objective is to develop an online store that allows the purchase of digital video games through a permanent ownership system. This means that once payment is completed, the user will have indefinite access to the game without the need for any type of license or activation key.

Methodology Followed to Achieve the Objective

To achieve this objective, a methodology based on agile web development was followed. First, the system architecture and user interface were designed. Then, the backend was implemented to manage purchases, users, and access to the video games. Finally, testing was carried out to ensure transaction security and the correct availability of the games after purchase.

Summary of Conclusions

The project has demonstrated the viability of a video game store that does not depend on external licenses. A stable, secure, and user-friendly system has been created, guaranteeing users permanent ownership of the games they purchase. This represents an innovative alternative to traditional digital distribution models.

Keywords (entre 4 i 8):

- **Video game store**
- **Digital gaming platform**
- **Purchase and download video games**
- **Gamer marketplace**

Índex

1 Introducció	5
1.1 Context	5
1.2 Justificació	5
1.3 Objectius	6
1.4 Estratègia i planificació del projecte	6
2 Descripció del projecte	8
2.1 Anàlisi de requisits	8
2.1 Previsió de tasques d'investigació	9
2.2 Tecnologies	9
2.3 Estructura del projecte	15
2.4 Descripció dels components	15
2.5 Definició de les tasques	17
2.5.1 Funcionalidades	18
3. Desplegament i Monitorització	39
4 Conclusions	40
4.1 Conclusions generals del projecte	40
4.2 Consecució dels objectius	40
4.2.1 Objetivos no logrados	42
4.3 Valoració de la metodologia i planificació	43
4.4 Visió de futur	43
5. Glossari	44
6. Bibliografia	45

Llista de figures

1 Introducció

El projecte trata sobre la creació de una tienda online de videojuegos digitales. La idea principal es ofrecer a los usuarios un lugar seguro y fiable donde comprar juegos, evitando problemas comunes de otras plataformas, como la pérdida de licencias, las restricciones de uso o la dependencia de servicios externos. El objetivo es que, una vez comprado un juego, el usuario lo tenga para siempre, sin condiciones ni limitaciones.

El propósito principal es desarrollar una tienda que permita comprar y descargar videojuegos de forma sencilla, garantizando la propiedad permanente del producto. Es decir, el usuario paga una vez y puede acceder al juego siempre, sin necesidad de claves o plataformas intermedias. Esto hace que la experiencia sea más clara, cómoda y directa.

Para desarrollar el proyecto, se ha seguido una metodología ágil, trabajando por fases. Primero se analizaron las necesidades de los usuarios y los requisitos del sistema. Después se diseñó la estructura de la plataforma, incluyendo la base de datos y las funciones principales. También se creó una interfaz fácil de usar, pensada para que la navegación sea intuitiva.

En la parte técnica, se desarrolló el sistema interno (backend) para gestionar compras, pagos y accesos a los juegos, y la parte visual (frontend), que es la que utiliza el usuario. Una vez terminado, se realizaron pruebas para asegurar que todo funciona correctamente y que los datos y las transacciones son seguras.

En conclusión, el proyecto demuestra que es posible crear una tienda de videojuegos digitales sin depender de licencias tradicionales. Se trata de una plataforma funcional, segura y fácil de usar, que ofrece una alternativa más directa y transparente. Además, propone un cambio en la forma de comprar videojuegos, dando más control al usuario y asegurando que realmente es dueño de lo que compra.

1.1 Context

El mercado actual de videojuegos digitales está dominado por un modelo de distribución centralizado, en el que grandes plataformas como Steam, Epic Games Store, GOG y las tiendas de consolas (Microsoft Store, PlayStation Store y Nintendo eShop) actúan como intermediarias casi obligatorias entre los desarrolladores y los consumidores.

1.2 Justificació

El proyecto consiste en crear una tienda online de videojuegos, algo que tiene sentido porque cada vez más personas compran y consumen contenido digital. Esta plataforma permite que los usuarios puedan acceder a los juegos de forma rápida desde cualquier lugar, sin las limitaciones de las tiendas físicas.

Además, incluye funciones que mejoran la experiencia del usuario, como recomendaciones personalizadas y herramientas para facilitar el uso de la web.

En general, el proyecto se adapta a cómo funciona hoy en día el mercado digital y aporta una forma más moderna de comprar y disfrutar videojuegos.

1.3 Objectius

Els objectius del projecte descriuen què es pretén aconseguir un cop finalitzada la creació de la botiga online de videojocs digitals. Aquests es divideixen en objectiu general i objectius específics.

1.3.1 Objectiu general

Un e-commerce de videojuegos con una estética amigable y fácil de usar, con pagos seguros y accesibilidad desde varias plataformas.

1.3.2 Objectius específics

- Diseñar una interfaz gráfica atractiva, clara y fácil de navegar para todo tipo de usuarios.
- Implementar un sistema de pagos seguro que garantice la protección de los datos personales y financieros de los clientes.
- Asegurar la compatibilidad y el correcto funcionamiento del sitio web en diferentes dispositivos (ordenadores, tabletas y teléfonos móviles).
- Desarrollar una base de datos eficiente para la gestión de productos, usuarios y transacciones.
- Incorporar funcionalidades de búsqueda y filtrado para mejorar la experiencia de compra y facilitar la localización de productos.
- Permitir la creación de cuentas de usuario para gestionar compras, preferencias e historial.
- Optimizar el rendimiento y la velocidad de carga del sitio web para garantizar una navegación fluida.
- Implementar medidas de accesibilidad que permitan el uso de la plataforma por parte de personas con diferentes necesidades o discapacidades.
- Integrar opciones de comunicación y soporte al cliente (chat, correo o formulario de contacto).
- Realizar pruebas de calidad (testing) para asegurar la fiabilidad y el funcionamiento global del sistema antes de su puesta en producción.

1.4 Estratègia i planificació del projecte

Para el desarrollo de este proyecto se ha optado por la estrategia de creación de un producto nuevo, diseñado e implementado completamente desde cero. Esta decisión responde a la necesidad de ofrecer una plataforma totalmente adaptada a los requisitos específicos del mercado de los videojuegos y a las

necesidades de los usuarios actuales, que valoran especialmente la usabilidad, la seguridad y la accesibilidad multiplataforma.

La elección de esta estrategia se considera la más adecuada para alcanzar los objetivos establecidos, ya que permite definir una arquitectura técnica propia, flexible y escalable. A diferencia de la adaptación de un producto existente, el desarrollo de un sistema nuevo garantiza un mayor control sobre la interfaz, el diseño visual, la base de datos y los módulos funcionales (compras, catálogo, autenticación, etc.).

Además, esta estrategia facilita la integración de tecnologías actuales y buenas prácticas de desarrollo web, asegurando un producto competitivo y de calidad. El estudio de viabilidad indica que, con una inversión inicial moderada, el proyecto es sostenible a largo plazo gracias a los bajos costes de mantenimiento y a la posibilidad de incorporar mejoras progresivas según la evolución del mercado.

1.5 Metodologia de treball

Especificar quin tipus de metodologia de treball se seguirà al llarg del desenvolupament del projecte (metodologia tradicional estil "waterfall", com per exemple Mètrica 3.0; o metodologia àgil, com per exemple Scrum). Argumentar per què es considera la metodologia més apropiada per al projecte i indicar quines eines es faran servir per al seu seguiment (diagrames de Gantt, aplicacions com Trello o Taiga.io, etc.).

1.6 Estudi econòmic i pressupostari

Concepto	Descripción	Coste
<i>Ordenador</i>	<i>Equipo necesario para el desarrollo del proyecto</i>	465 €
<i>Alquiler de oficina</i>	<i>Espacio de trabajo para el equipo durante el proyecto</i>	100 €
<i>Sueldos</i>	<i>Remuneración de 2 personas del equipo (1.500 € por persona)</i>	3.000 €
<i>Internet</i>	<i>Conexión necesaria para desarrollo, comunicación y despliegue</i>	75 €
<i>Consumo eléctrico</i>	<i>Gasto estimado de energía durante el periodo del proyecto</i>	50 €
<i>Dominio</i>	<i>Registro del nombre del proyecto en internet</i>	0 €
<i>Hosting</i>	<i>Servicio de alojamiento web</i>	0 €

<i>Promoción</i>	<i>Difusión del proyecto (redes sociales, marketing, etc.)</i>	<i>0 €</i>
TOTAL		3.690 €

2 Descripció del projecte

2.1 Anàlisi de requisits

Para garantizar que el proyecto de e-commerce de videojuegos cumpla los objetivos establecidos, se ha realizado una identificación de los requisitos que debe satisfacer el sistema. Estos requisitos aseguran que, una vez finalizado el desarrollo, el producto sea funcional, seguro y accesible para los usuarios.

2.1.1 Requisites funcionals

Los requisitos funcionales definen las funcionalidades concretas que debe permitir el sistema:

- Permitir la creación, modificación y eliminación de cuentas de usuario.
- Ofrecer un catálogo de videojuegos con información detallada de cada producto (descripción, imágenes, precio, disponibilidad).
- Implementar un sistema de búsqueda y filtrado de productos por categoría, plataforma y otros criterios.
- Facilitar el proceso de compra con un carrito de compra y resumen del pedido.
- Integrar pasarelas de pago seguras que acepten diferentes formas de pago.
- Generar confirmaciones de compra y facturación automática por correo electrónico.
- Permitir la gestión del inventario por parte del administrador (añadir, eliminar o modificar productos).
- Incorporar un sistema de valoraciones y reseñas de los usuarios para cada videojuego.
- Proporcionar un panel de administración para monitorizar ventas, usuarios y estadísticas.
- Implementar notificaciones y comunicación con el usuario sobre promociones o novedades.

2.1.2 Requisites no funcionals

Los requisitos no funcionales establecen las cualidades y restricciones del sistema, centradas en la experiencia del usuario y la seguridad:

- **Seguridad:** Protección de los datos personales y financieros mediante cifrado y protocolos seguros (SSL/TLS).
- **Usabilidad:** Interfaz intuitiva y fácil de usar, con diseño responsivo adaptable a ordenadores, tabletas y móviles.
- **Robustez:** Gestión de errores y fallos sin comprometer la continuidad del servicio.
- **Rendimiento:** Tiempos de carga rápidos y optimización del rendimiento web bajo diferentes niveles de uso.
- **Mantenibilidad:** Código estructurado y documentado para facilitar futuras actualizaciones y modificaciones.
- **Compatibilidad:** Funcionamiento correcto en los principales navegadores web y sistemas operativos.

2.1 Previsió de tasques d'investigació

2.2 Technologies

- Una vez definidos los requisitos, ya conocemos las partes que debe incluir nuestro proyecto. Por ello, podemos comparar las diferentes tecnologías que se pueden utilizar para las distintas tareas, haciendo un pequeño resumen con sus ventajas e inconvenientes. Después, se explicarán con más detalle aquellas que se han elegido.

2.2.1 Comparativa de les technologies valorades

Frontend:

Framework	Ventajas	Desventajas
Node.js (Express)	<ul style="list-style-type: none">-Gran desempeño gracias al motor V8 de Chrome.-Muy adecuado para aplicaciones en tiempo real (por ejemplo, monitorización de vehículos).-Ecosistema muy completo con numerosos paquetes NPM disponibles.-Estructura no bloqueante y con gran capacidad de escalado.	<ul style="list-style-type: none">-Resulta más complejo de gestionar al trabajar con bases de datos relacionales.-Puede consumir una cantidad elevada de memoria en procesos exigentes.-Requiere una configuración extra para asegurar la protección del sistema.

<p>React/Ionic</p>	<ul style="list-style-type: none"> - Permite desarrollar aplicaciones multiplataforma con una sola base de código. - Cuenta con una comunidad amplia y abundante documentación disponible. - Facilita la integración con servicios externos y APIs. - Es una buena opción para aplicaciones de gestión, reservas y paneles administrativos. Ofrece compatibilidad sencilla con bases de datos SQL y otros sistemas backend. 	<ul style="list-style-type: none"> - El rendimiento puede ser inferior al de aplicaciones nativas en procesos muy exigentes. - No es la opción más eficiente para aplicaciones con comunicación en tiempo real intensiva. - Algunas funcionalidades avanzadas del dispositivo pueden requerir plugins adicionales. - Puede presentar tiempos de carga mayores cuando la aplicación tiene mucha complejidad o alta demanda.
<p>Python (Django)</p>	<ul style="list-style-type: none"> -Alto grado de seguridad. -Arquitectura bien definida y escalable. -Adecuado para el desarrollo ágil y la gestión de usuarios. -Perfecto para aplicaciones web con análisis avanzados. 	<ul style="list-style-type: none"> -Puede volverse más lento en aplicaciones de gran escala. -Mayor consumo de memoria en comparación con Laravel.
<p>Ruby (Ruby on Rails)</p>	<ul style="list-style-type: none"> - Desenvolupament ràpid i sintaxi elegant. - Facilita la implementació d'APIs. - Comunitat activa i moltes eines preconstruïdes. 	<ul style="list-style-type: none"> - Pot ser més lent en aplicacions molt grans. - Menys popular en comparació amb Django o Laravel.

Backend:

Framework	Ventajas	Desventajas
------------------	-----------------	--------------------

Vue.js	<ul style="list-style-type: none"> -Fácil de aprender, con una sintaxis sencilla y clara. -Ligero y de alto rendimiento. -Compatible con integraciones progresivas. -Permite la reutilización de componentes. 	<ul style="list-style-type: none"> -Más reciente y con menor adopción en grandes empresas. -Comunidad más reducida en comparación con React.
Springboot	<ul style="list-style-type: none"> -Muy popular y flexible. -Amplio ecosistema con gran variedad de bibliotecas. -Alto rendimiento gracias a un renderizado eficiente. 	<ul style="list-style-type: none"> -Mayor complejidad en la gestión del estado. -Alta dependencia de bibliotecas externas.
Angular	<ul style="list-style-type: none"> -Bien estructurado e ideal para aplicaciones empresariales. -Incluye arquitectura MVVM para una mejor organización del código. -Cuenta con un sólido respaldo de Google. 	<ul style="list-style-type: none"> -Más pesado y con una curva de aprendizaje más pronunciada. -Necesita más recursos para su ejecución.
Base de datos	Ventajas	Desventajas
MySQL	<ul style="list-style-type: none"> -Popular y sencillo de utilizar. -Adecuado para pequeñas y medianas empresas. -Rendimiento adecuado en la mayoría de los casos. 	<ul style="list-style-type: none"> -Menor capacidad de escalado en comparación con PostgreSQL. -Menos eficiente en operaciones complejas.
PostgreSQL	<ul style="list-style-type: none"> Soporta JSON y operaciones avanzadas. -Mayor escalabilidad y seguridad. -Adecuado para aplicaciones con grandes volúmenes de datos. 	<ul style="list-style-type: none"> -Complicado de configurar y gestionar.
MongoDB	<ul style="list-style-type: none"> -Base de datos NoSQL, adecuada para datos 	<ul style="list-style-type: none"> -No es relacional, por lo que puede no ser ideal

	<p>dinámicos.</p> <p>-Alta capacidad de escalado.</p> <p>-Buena opción para aplicaciones en tiempo real.</p>	<p>para informes financieros.</p> <p>-Puede ofrecer menor seguridad en transacciones críticas.</p>
--	--	--

Pasarelas de Pago	Ventajas	Desventajas
Stripe	<p>-API moderna y muy bien documentada.</p> <p>-Fácil integración con aplicaciones web y móviles.</p> <p>-Compatible con pagos recurrentes y suscripciones.</p> <p>-Soporta múltiples métodos de pago como tarjetas, Apple Pay y Google Pay.</p> <p>-Entorno de pruebas (sandbox) muy completo.</p>	<p>-Comisiones por transacción relativamente elevadas.</p> <p>-Dependencia de un servicio externo.</p> <p>-Algunas configuraciones avanzadas pueden resultar complejas.</p>
Paypal	<p>-Muy conocida y confiable para los usuarios.</p> <p>-Integración sencilla en ecommerce.</p> <p>-Permite pagos sin compartir datos bancarios.</p> <p>-Amplia presencia internacional.</p>	<p>-Comisiones más elevadas que otras plataformas.</p> <p>-Experiencia de pago menos integrada debido a redirecciones.</p> <p>-Personalización limitada del checkout.</p>
Redsys	<p>-Muy utilizada en España y compatible con bancos nacionales.</p> <p>-Comisiones generalmente bajas.</p> <p>-Compatible con Bizum y pagos con tarjeta.</p>	<p>-Documentación técnica menos intuitiva.</p> <p>-Integración más compleja que Stripe o PayPal.</p> <p>-Menor soporte para funcionalidades modernas avanzadas.</p>

	<i>-Alta seguridad en transacciones bancarias.</i>	
--	--	--

2.2.2 Tecnologías escollides

Elección final: Spring boot + Ionic/React + MySQL

¿Por qué Spring Boot?

Framework basado en Java muy robusto y ampliamente utilizado en entornos profesionales.

Permite crear APIs REST de forma rápida y estructurada.

Gran integración con herramientas como Spring Data y Spring Security.

Mejor rendimiento y escalabilidad en comparación con otras opciones como Node.js en aplicaciones grandes.

¿Por qué React/Ionic?

Permite crear interfaces de usuario modernas e interactivas.

Basado en componentes reutilizables, lo que facilita el mantenimiento del código.

Virtual DOM que mejora el rendimiento de la aplicación.

Gran comunidad y mucha documentación disponible.

Más flexible que Angular y muy demandado en el mercado laboral.

¿Por qué MySQL?

Sistema de gestión de bases de datos relacional muy consolidado.

Fácil de usar y ampliamente compatible con diferentes tecnologías.

Buen rendimiento en aplicaciones con estructuras de datos definidas.

Integración sencilla con Spring Boot.

Alternativa más simple de gestionar en comparación con PostgreSQL para este proyecto.

¿Por qué Stripe?

Pasarela de pago moderna y ampliamente utilizada en aplicaciones web actuales.

Permite integrar pagos online de forma rápida y segura mediante APIs bien documentadas.

Compatible con múltiples métodos de pago como tarjetas, Apple Pay y Google Pay.

Ofrece un entorno de pruebas muy completo y una gran escalabilidad para futuros desarrollos.

Conclusión:

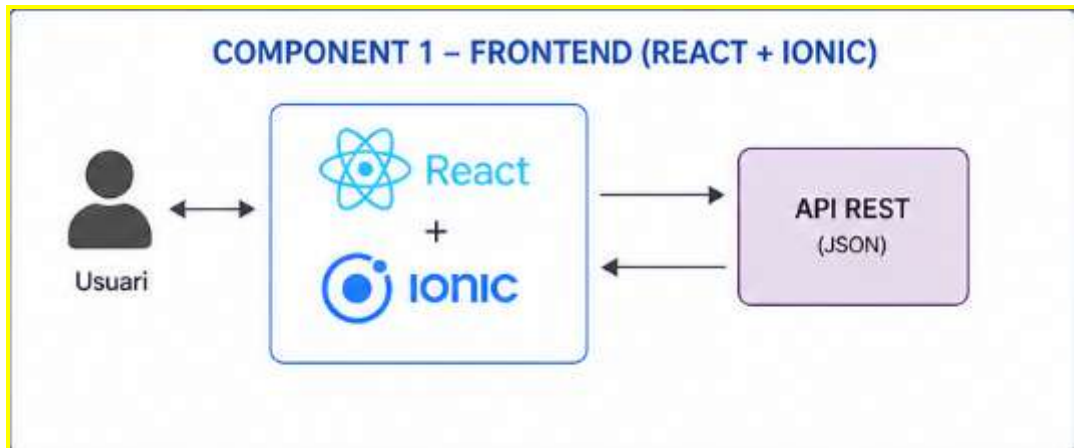
Spring Boot, React, MySQL y Stripe forman una combinación muy potente para el desarrollo de aplicaciones web modernas. Esta arquitectura permite separar frontend y backend, facilitando la escalabilidad, el mantenimiento y el desarrollo de una aplicación eficiente, segura y adaptada a las necesidades del proyecto, incorporando además un sistema de pagos online fiable y sencillo de integrar.

2.3 Estructura del proyecto



2.4 Descripció dels components

2.4.1 Component 1 - Frontend (React + Ionic)



Descripció:

El frontend es la interfaz con la que interactúa el usuario. Está desarrollado con React e Ionic para ofrecer una aplicación web y móvil moderna, responsive y fácil de usar.

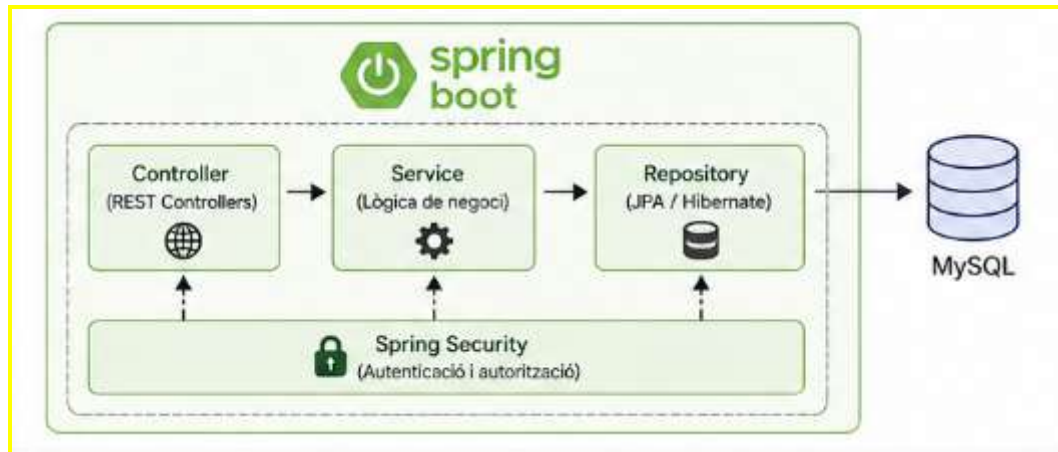
Tecnologías:

- React: biblioteca de JavaScript para construir interfaces de usuario mediante componentes.
- Ionic: framework para crear aplicaciones móviles híbridas.
- JavaScript / TypeScript, HTML y CSS.

Funcionalidades principales:

- Mostrar y gestionar la información.
- Interacción con el usuario.
- Gestión del estado y navegación.
- Consumo de la API REST del backend mediante peticiones HTTP/HTTPS.

2.4.2 Component 2 - Backend (Springboot + Java)



Descripción:

El backend se encarga de procesar las peticiones del frontend, aplicar la lógica de negocio y gestionar el acceso a la base de datos.

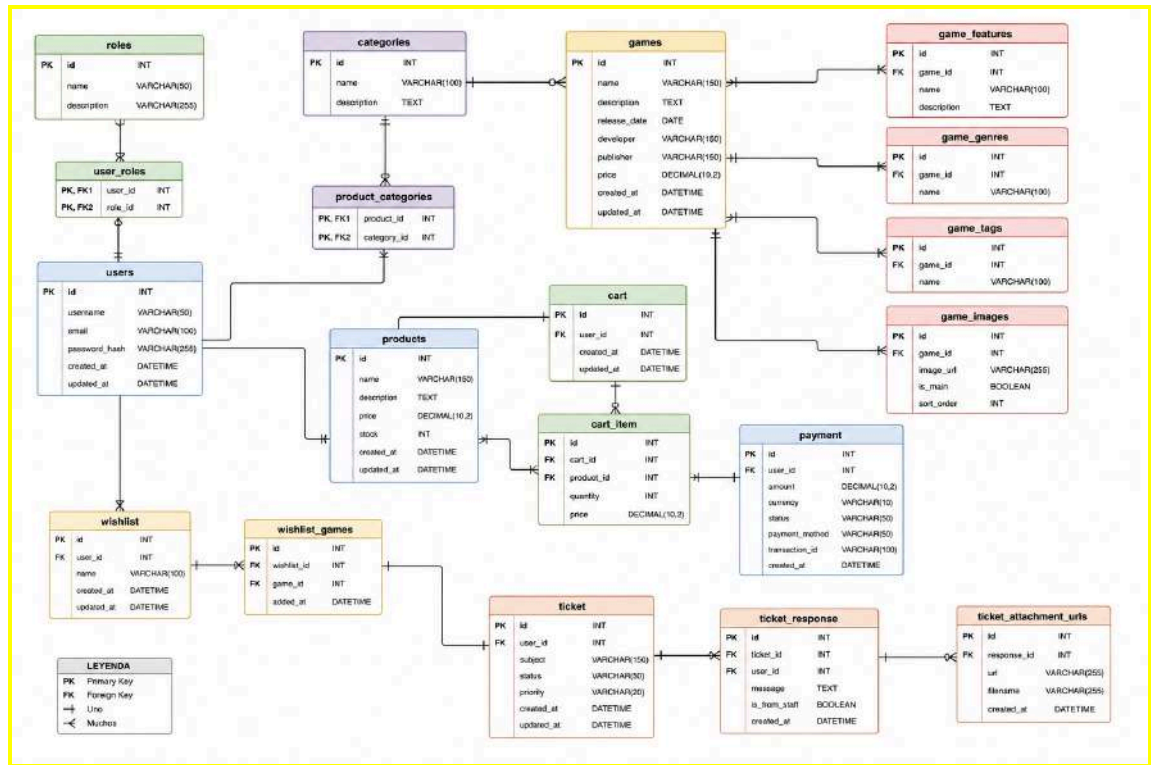
Tecnologías:

- Spring Boot: framework de Java para crear aplicaciones robustas y escalables.
- Spring Data JPA / Hibernate: acceso a datos de forma eficiente.
- Spring Security: gestión de autenticación y autorización.
- Java.

Funcionalidades principales:

- Exposición de una API REST.
- Lógica de negocio y validación de datos.
- Gestión de usuarios y permisos.
- Acceso seguro a la base de datos MySQL.

2.4.3 Component 3 - Base de datos (MySQL)



Descripción:

La base de datos se encarga de almacenar toda la información de la aplicación de forma persistente y estructurada.

Tecnologías:

- MySQL: sistema de gestión de bases de datos relacionales.

Funcionalidades principales:

- Almacenamiento de datos.
- Relaciones entre tablas.
- Consultas y transacciones.
- Integridad y consistencia de los datos.

2.5 Definición de las tareas

La aplicación ha sido desarrollada utilizando **Spring Boot** como tecnología principal en el backend, integrando distintos componentes para la autenticación, gestión de usuarios, procesamiento de pagos, envío de correos electrónicos y administración del sistema.

2.5.1 Funcionalidades

Sistema de autenticación y gestión de usuarios

Uno de los aspectos más importantes del proyecto ha sido la implementación de un sistema de autenticación seguro y flexible, capaz de diferenciar distintos tipos de usuarios y proteger la información almacenada dentro de la plataforma.

Gestión de usuarios y roles

La aplicación dispone de dos roles principales:

- Administrador
- Cliente

Cada rol dispone de permisos y funcionalidades diferentes dentro del sistema.

Los usuarios clientes pueden navegar por la tienda, realizar compras, gestionar su perfil y utilizar el sistema de soporte. Por otro lado, los administradores tienen acceso a un panel interno desde el cual pueden supervisar el funcionamiento general de la plataforma.

Para el panel de administración se ha optado por una arquitectura server-side integrada directamente con Spring Boot, evitando el uso de una API REST pública para esta parte del sistema. Esta decisión se tomó con el objetivo de aumentar la seguridad y centralizar el control de acceso, ya que el panel administrativo únicamente será utilizado de forma interna.

El control de acceso se realiza mediante Spring Security, permitiendo restringir las rutas y funcionalidades según el rol autenticado.

Autenticación segura con JWT y Spring Security

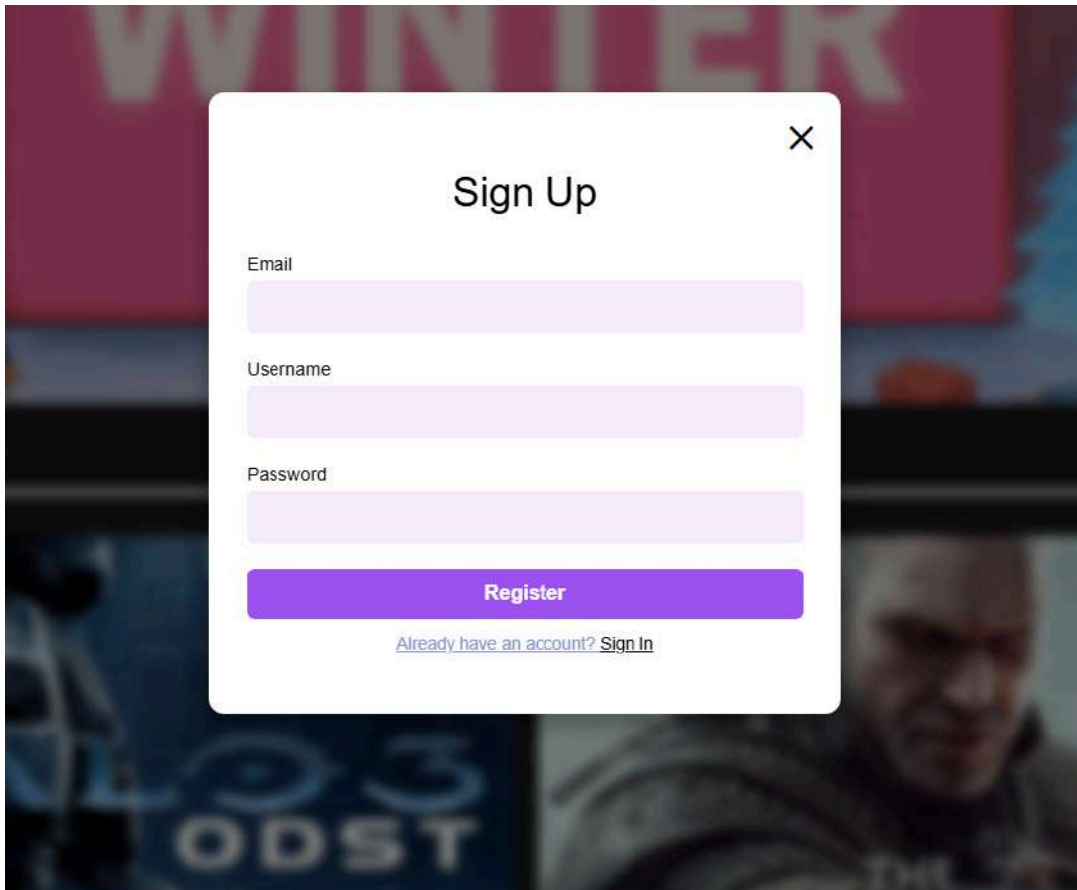
La autenticación tradicional se realiza mediante correo electrónico y contraseña.

Para ello, se ha implementado un sistema basado en:

- Spring Security
- JWT (JSON Web Token)

El uso de JWT permite gestionar sesiones seguras sin necesidad de almacenar el estado de sesión en el servidor. Esto mejora el rendimiento de la aplicación y facilita la comunicación entre frontend y backend.

Las contraseñas se almacenan cifradas utilizando algoritmos de hash seguros, evitando que la información sensible pueda verse comprometida en caso de acceso no autorizado a la base de datos.



Sistema de autenticación en dos pasos (2FA)

Con el objetivo de reforzar la seguridad de las cuentas de usuario, se ha implementado un sistema de autenticación en dos factores (2FA).

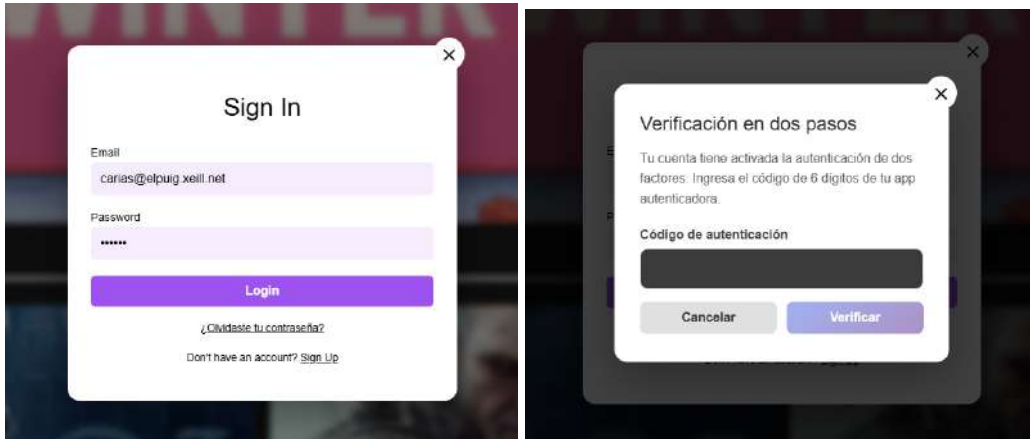
Después de introducir correctamente el correo y la contraseña, el usuario debe introducir un código temporal generado desde la aplicación Google Authenticator.

El funcionamiento del sistema sigue el estándar TOTP (Time-based One-Time Password), generando códigos temporales válidos únicamente durante unos segundos.

Esta funcionalidad añade una capa adicional de protección y reduce significativamente el riesgo de accesos no autorizados.

El sistema permite además:

- Activar o desactivar la autenticación 2FA.
- Verificar códigos temporales.
- Asociar el dispositivo del usuario mediante código QR.



Sistema de envío de correos electrónicos

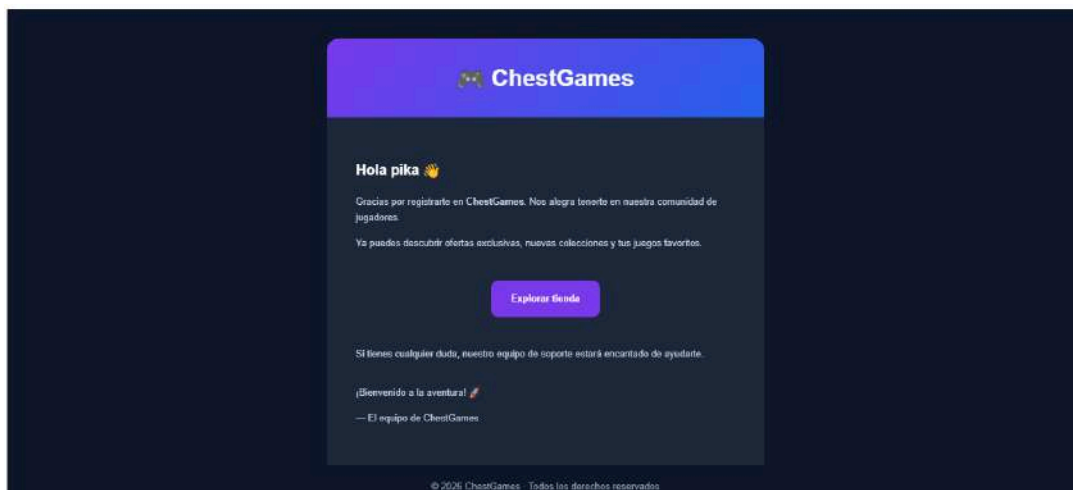
La plataforma incorpora un sistema automático de envío de correos electrónicos utilizando Spring Boot Mail.

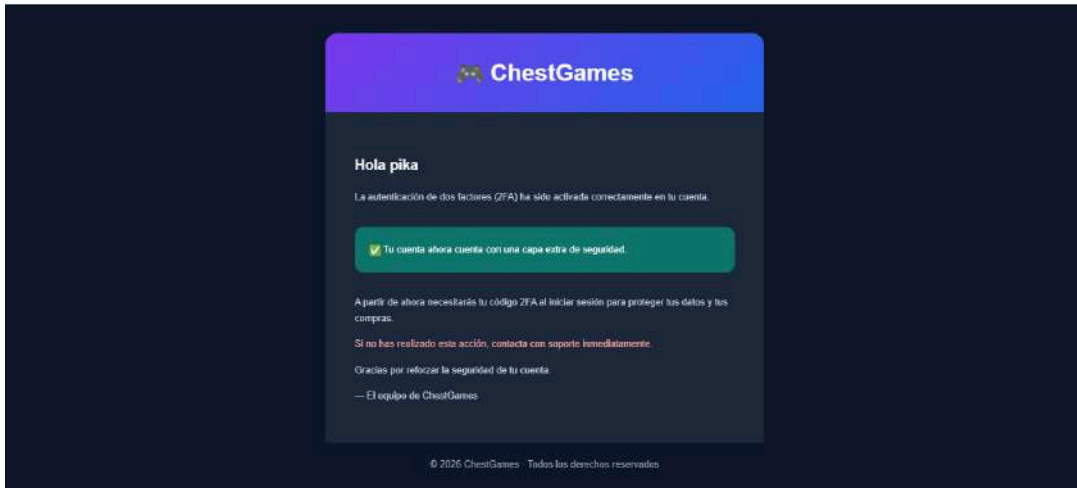
Para garantizar una comunicación profesional y fiable, se utiliza un servicio de correo corporativo mediante Zoho Mail y un dominio personalizado.

Los correos electrónicos se utilizan en diferentes procesos importantes del sistema:

- Confirmación de registro.
- Activación de la autenticación 2FA.
- Confirmación de compras.
- Recuperación de contraseña.
- Mensajes de bienvenida.
- Notificaciones relacionadas con pedidos.

El objetivo de esta funcionalidad es automatizar la comunicación con el usuario y mejorar la experiencia general dentro de la plataforma.



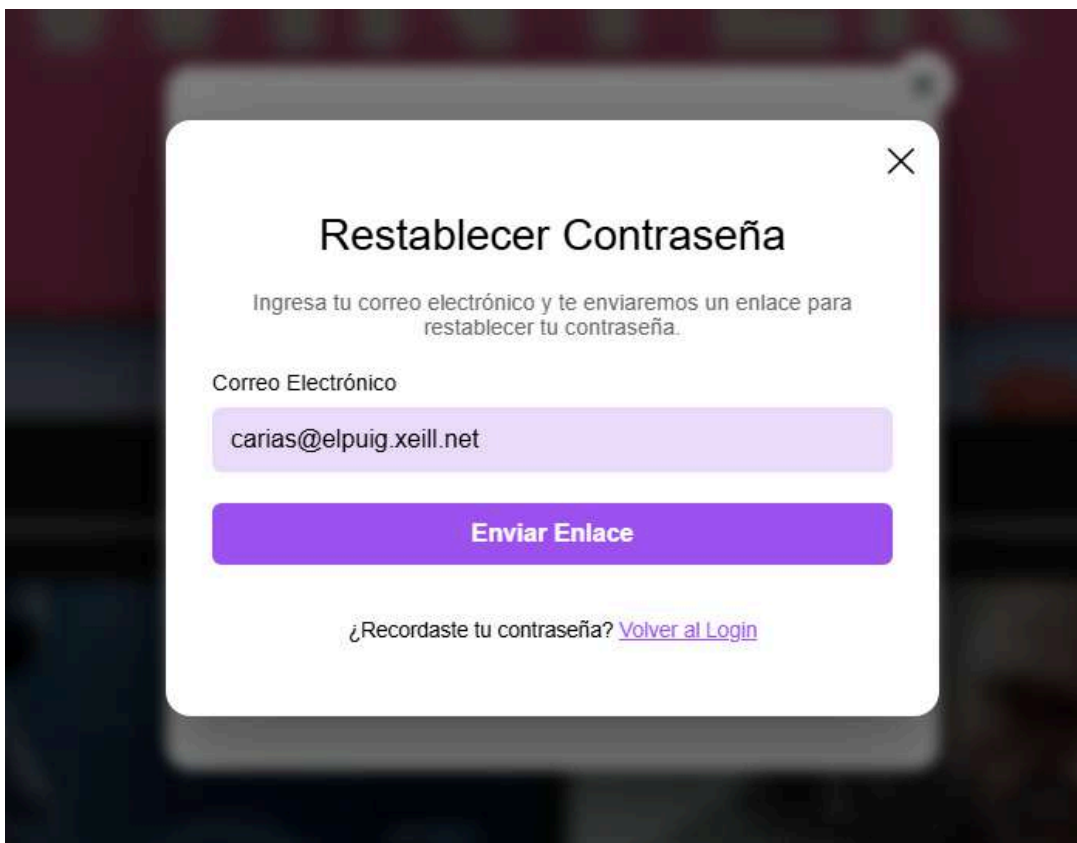


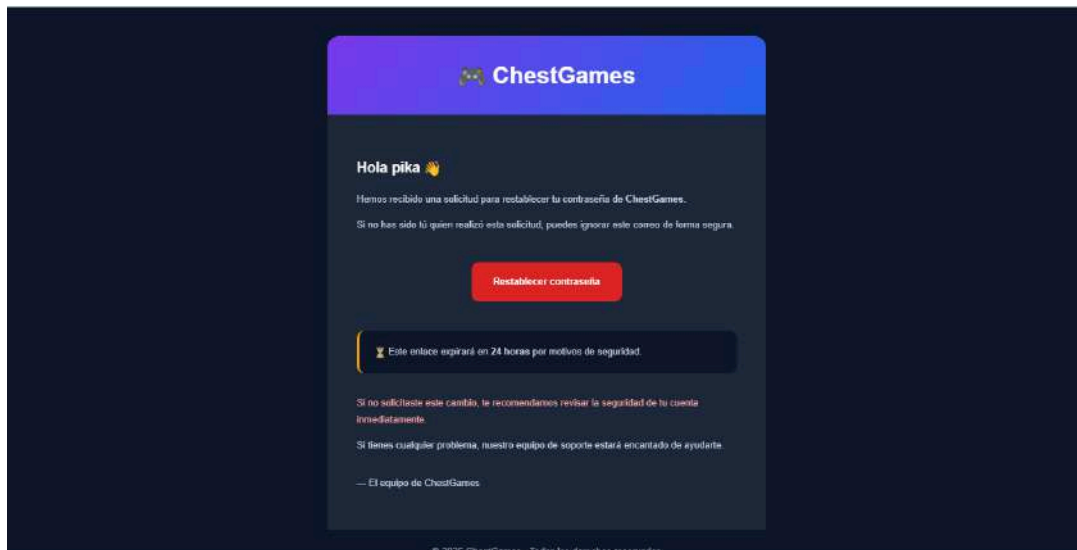
Recuperación de contraseña

La aplicación incluye un sistema seguro de recuperación de contraseña mediante correo electrónico.

Cuando un usuario solicita recuperar su cuenta, el sistema genera un enlace temporal y seguro que permite establecer una nueva contraseña.

Este mecanismo evita accesos indebidos y garantiza que únicamente el propietario del correo pueda modificar las credenciales de acceso.





Perfil y configuración del usuario

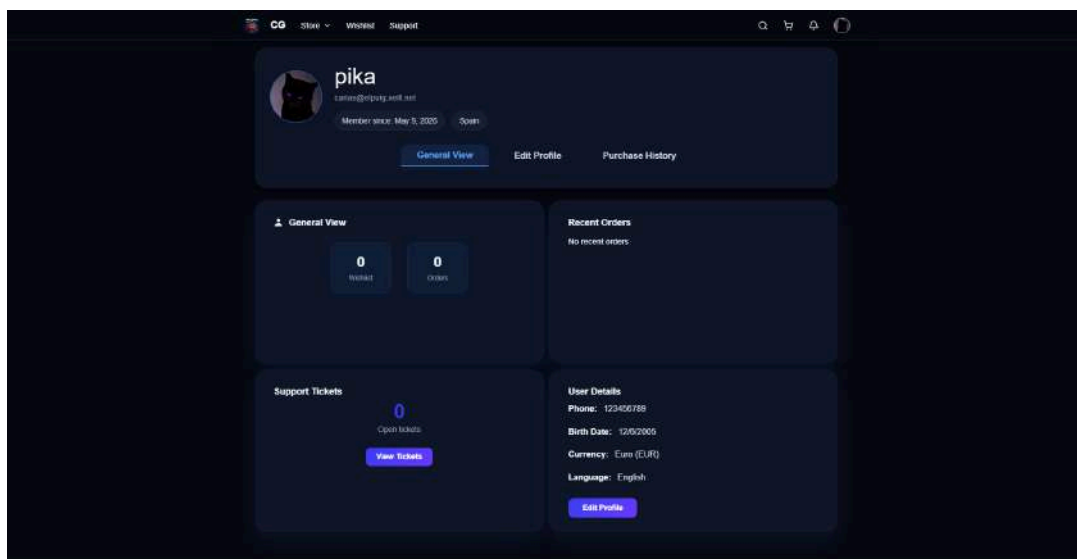
Cada usuario dispone de un perfil personalizado dentro de la plataforma.

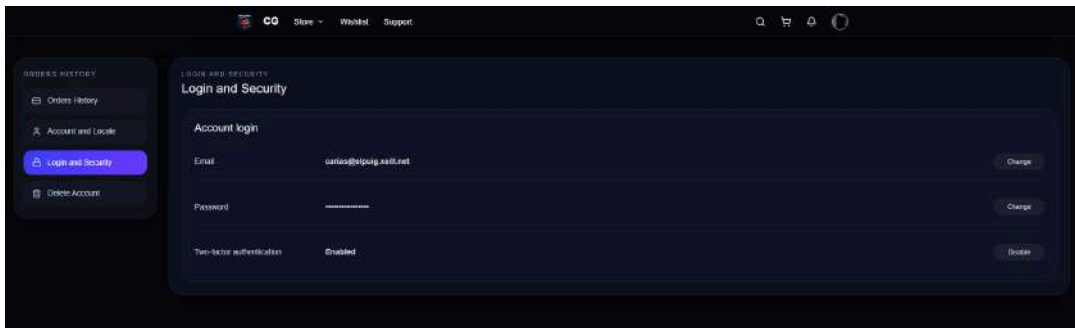
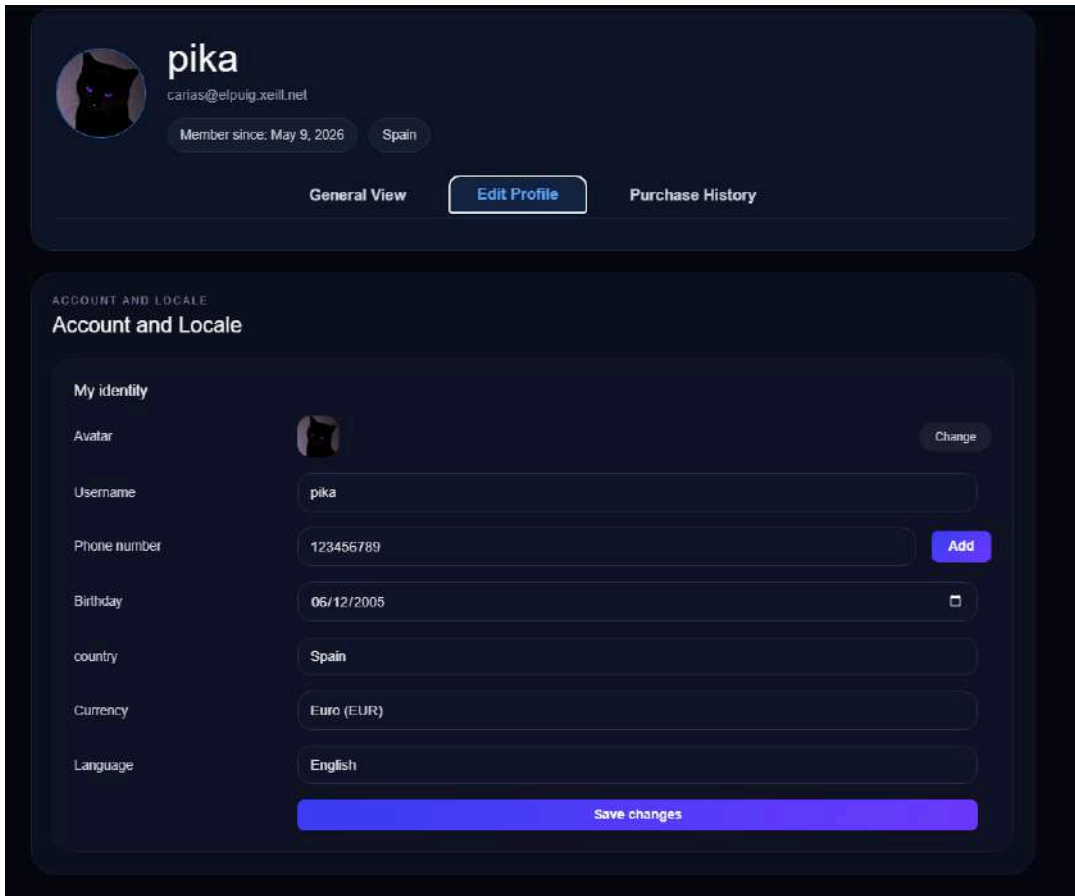
Desde este apartado puede gestionar diferentes elementos relacionados con su cuenta:

- Cambiar nombre de usuario.
- Modificar imagen de perfil.
- Añadir fecha de cumpleaños.
- Cambiar idioma de la plataforma.
- Seleccionar moneda.
- Modificar correo electrónico.
- Cambiar contraseña.
- Activar o desactivar 2FA.
- Eliminar la cuenta.

La aplicación adapta dinámicamente tanto el idioma como la moneda según las preferencias seleccionadas por el usuario, mejorando la experiencia de navegación.

Además, el usuario puede consultar su actividad reciente y acceder a información relacionada con sus compras.





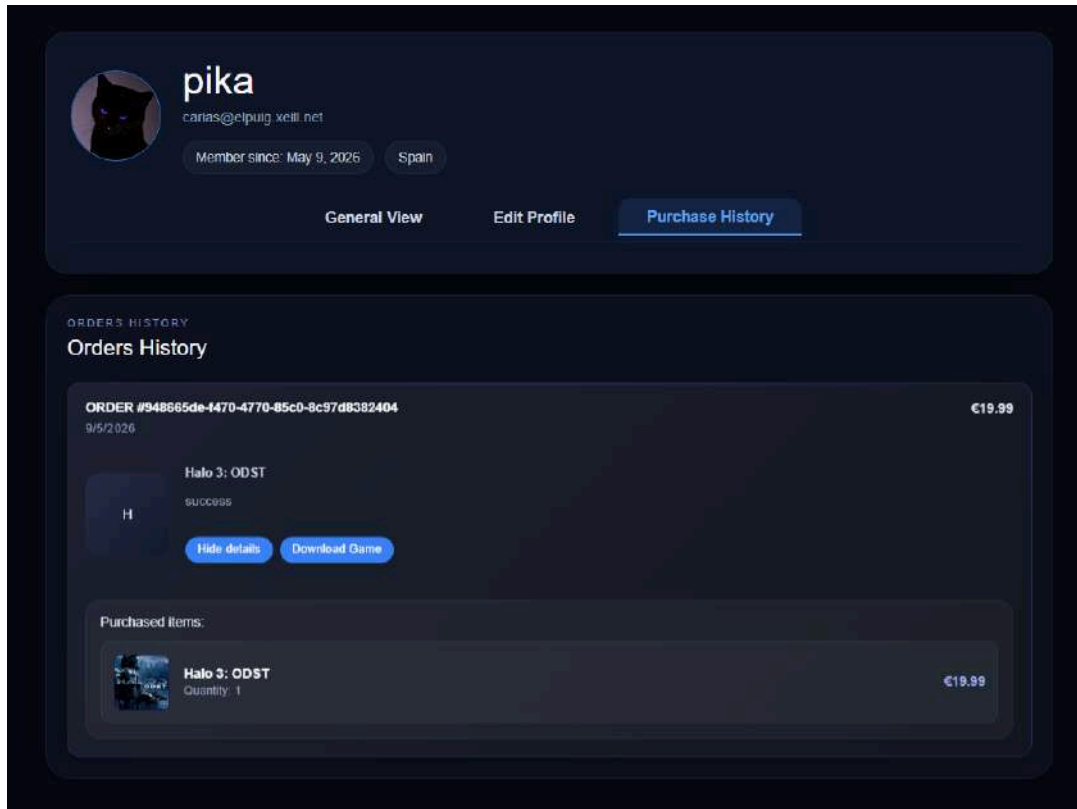
7. Historial de pedidos y pagos

El sistema registra automáticamente todas las compras realizadas por cada usuario.

Desde el perfil personal es posible consultar:

- Pedidos realizados.
- Productos comprados.
- Fecha de compra.
- Estado del pedido.
- Pagos recientes.

Esta funcionalidad proporciona trazabilidad y transparencia dentro de la plataforma.



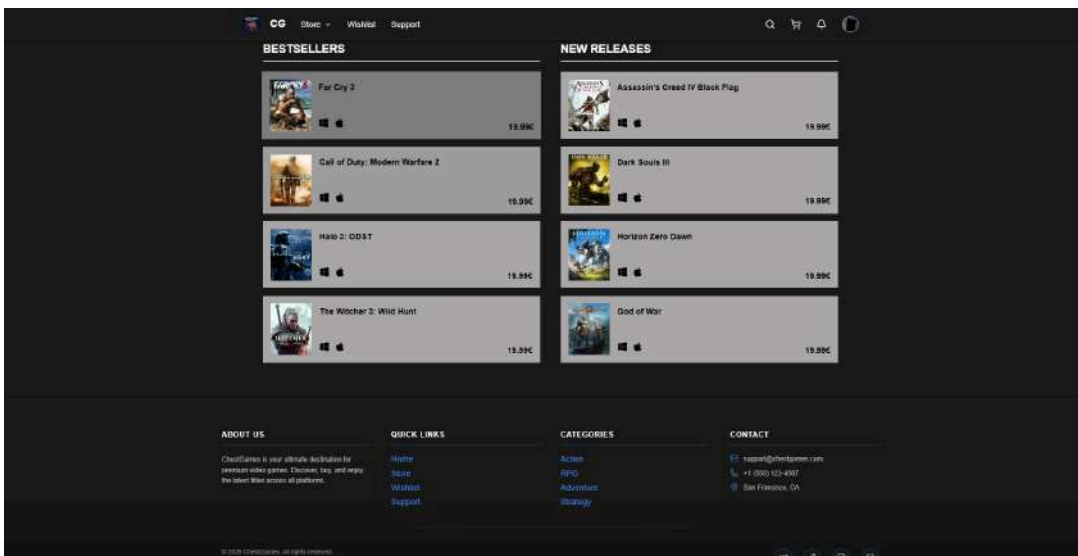
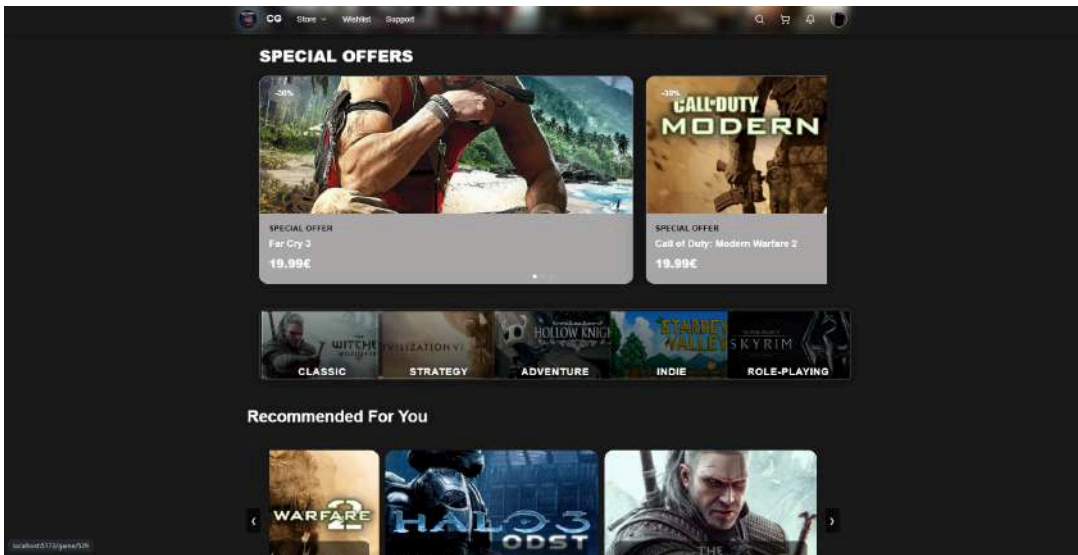
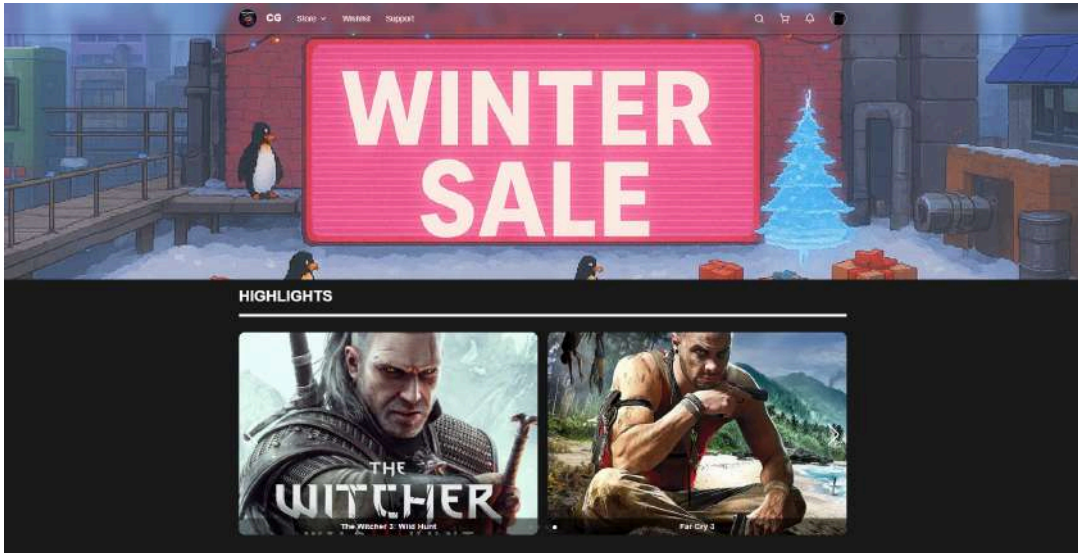
Página principal (Home)

La plataforma dispone de una página principal que actúa como punto de entrada para el usuario.

En esta sección se muestran videojuegos destacados y contenido relevante del catálogo.

El diseño de esta página está orientado a:

- Mejorar la experiencia visual.
- Facilitar la navegación.
- Mostrar productos destacados.
- Incentivar la exploración de la tienda.



Catálogo de videojuegos

La aplicación incluye una página dedicada al catálogo completo de videojuegos.

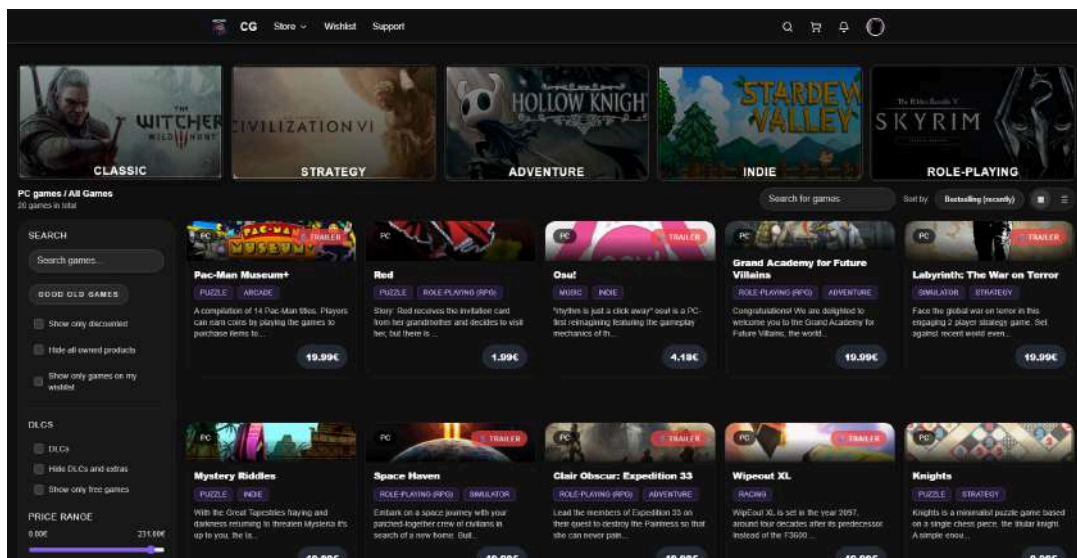
Cada producto contiene información detallada como:

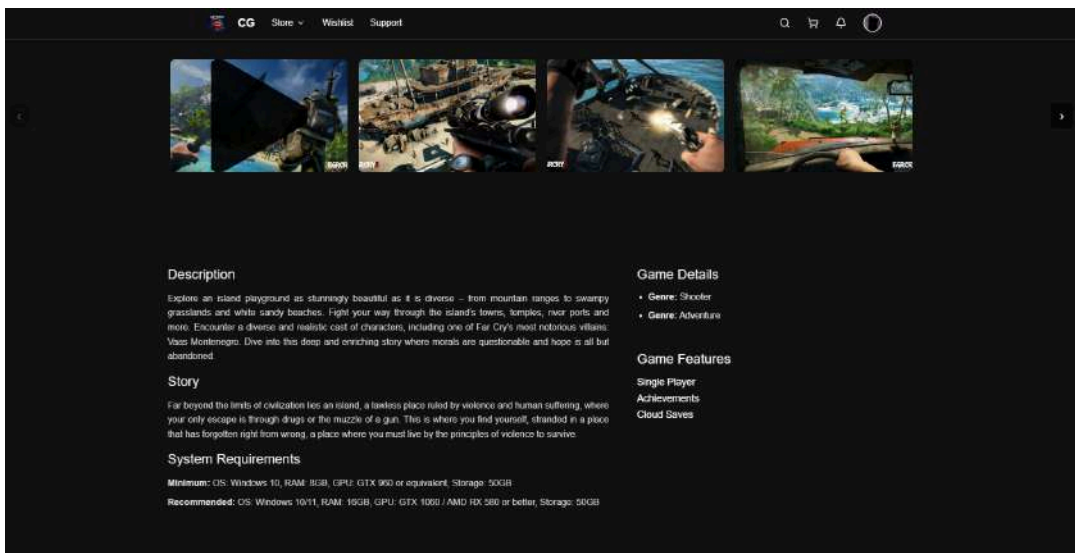
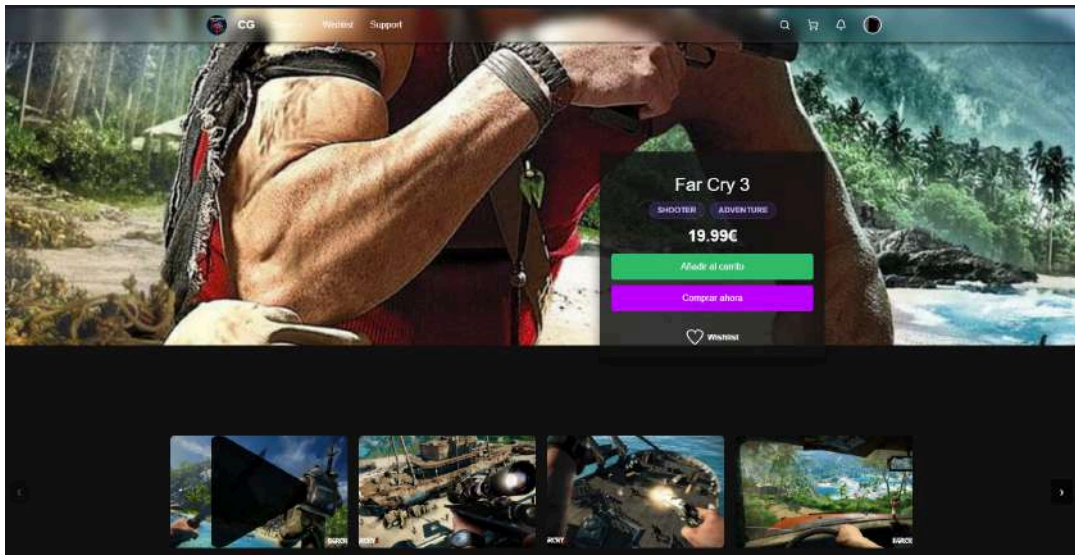
- Nombre.
- Descripción.
- Precio.
- Trailer
- Imágenes.
- Categoría.
- Requisitos del sistema.

El catálogo incorpora además un sistema de filtros dinámicos que permite al usuario buscar videojuegos según distintos criterios.

Los filtros se actualizan en tiempo real y facilitan la localización de productos sin necesidad de recorrer manualmente todo el catálogo.

Esto mejora considerablemente la usabilidad y la experiencia de usuario.





Gestión de compras y pagos

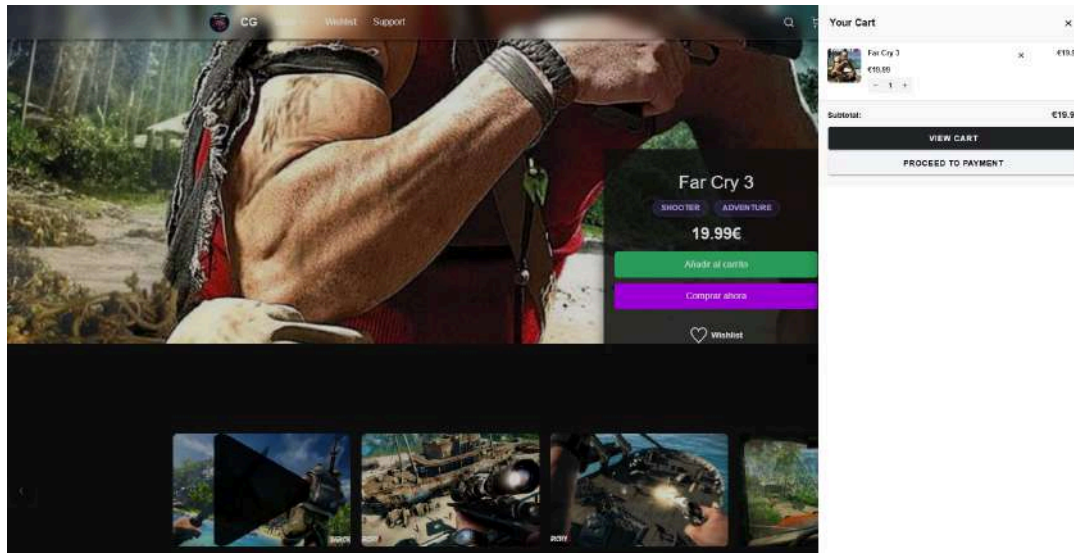
El proceso de compra permite seleccionar videojuegos y realizar pagos directamente desde la plataforma.

Para el procesamiento de pagos se ha integrado la plataforma Stripe, permitiendo realizar transacciones seguras y fiables.

Durante el proceso de compra, el sistema:

1. Valida los productos seleccionados.
2. Gestiona el pago mediante Stripe.
3. Registra el pedido en la base de datos.
4. Asocia el pedido al usuario.
5. Envía una confirmación por correo electrónico.

Esta integración permite automatizar completamente el flujo de compra.



Lista de deseos (Wishlist)

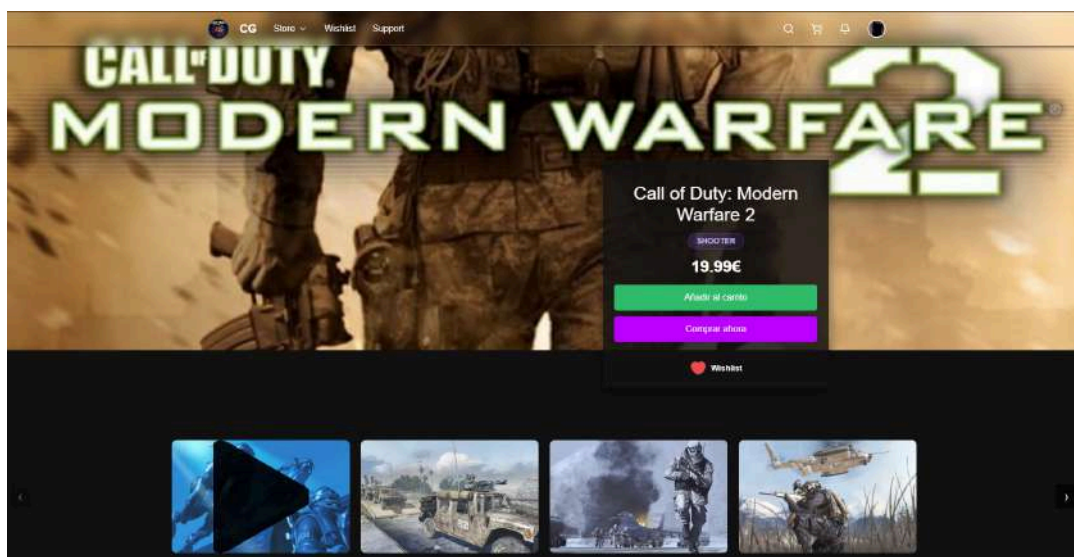
La aplicación incorpora una funcionalidad de lista de deseos.

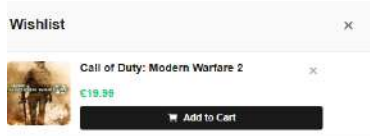
Cada usuario puede guardar videojuegos que le interesen para futuras compras.

La wishlist permite:

- Añadir videojuegos.
- Eliminar videojuegos.
- Mantener un seguimiento personalizado de productos.

Esta funcionalidad mejora la experiencia de usuario y facilita futuras decisiones de compra.





Sistema de soporte y tickets

La plataforma dispone de un sistema de soporte basado en tickets.

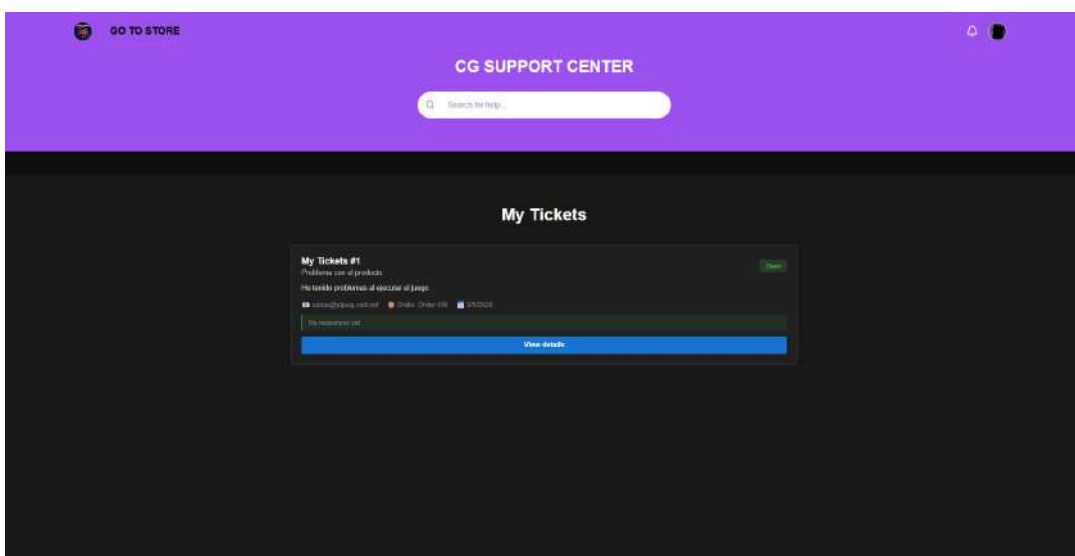
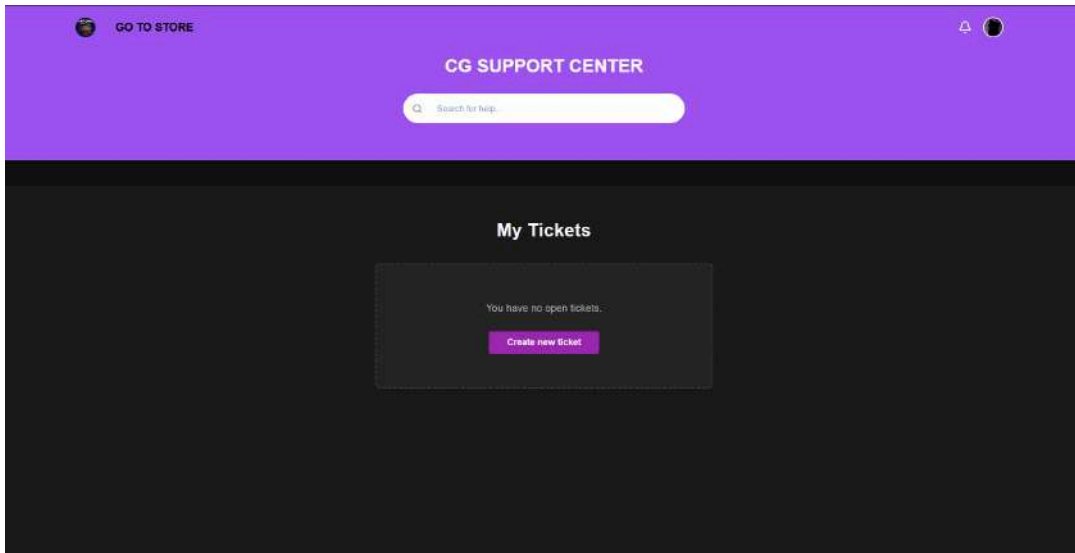
Los usuarios pueden crear incidencias, dudas o solicitudes directamente desde su cuenta.

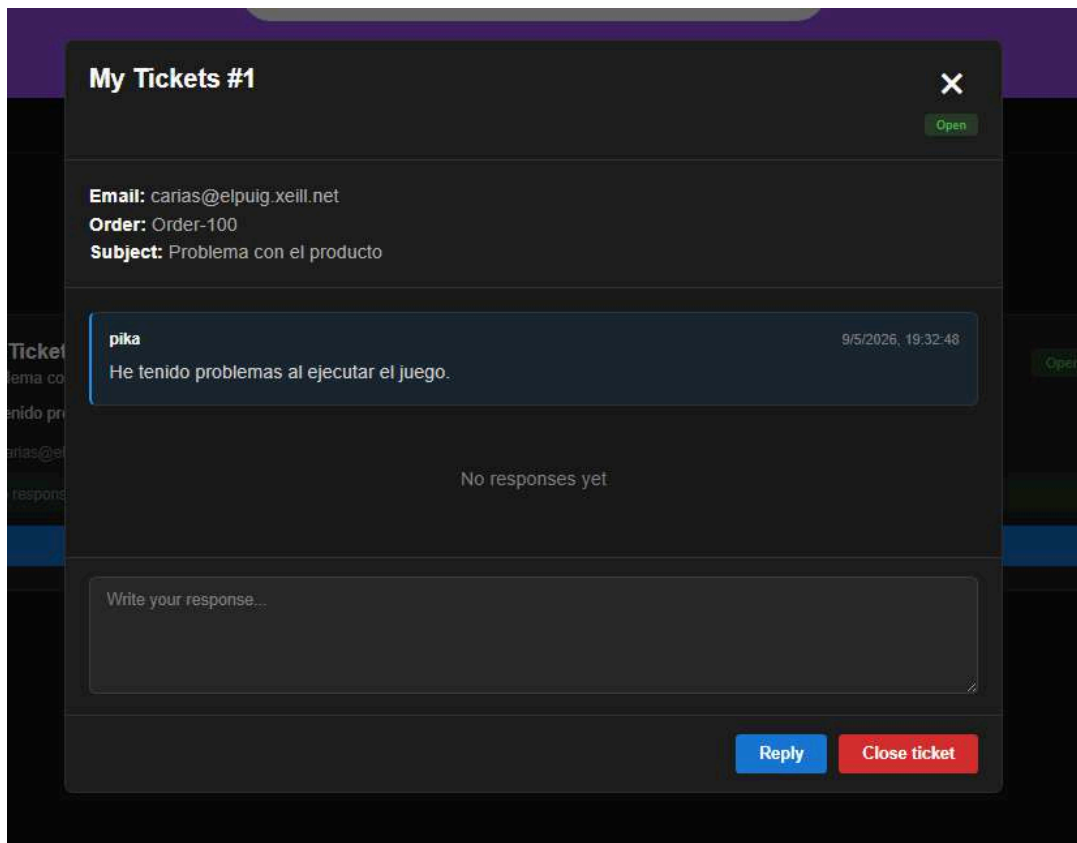
Cada ticket queda registrado en el sistema y puede ser gestionado por el administrador.

El sistema permite:

- Crear tickets.
- Consultar el estado.
- Responder incidencias.
- Mantener conversaciones entre usuario y administrador.

Esto proporciona un canal de comunicación estructurado y organizado.





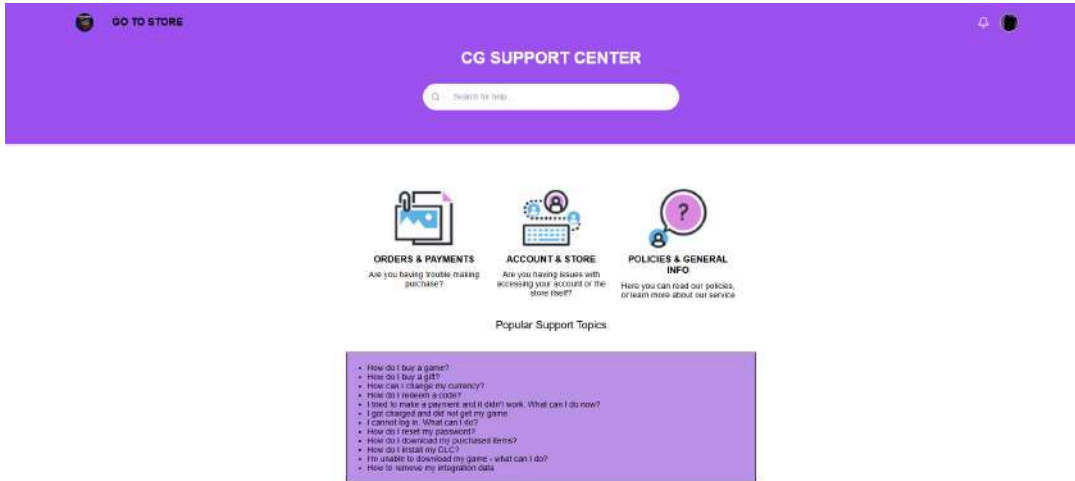
Preguntas frecuentes (FAQ)

Dentro del apartado de soporte se ha implementado una sección de preguntas frecuentes.

El objetivo de esta funcionalidad es resolver dudas comunes sin necesidad de abrir un ticket.

Esto permite:

- Reducir la carga del sistema de soporte.
- Resolver problemas rápidamente.
- Mejorar la experiencia del usuario.



Panel de administración

La plataforma incorpora un entorno administrativo independiente destinado a la gestión interna del sistema.

Este panel permite supervisar el funcionamiento general del e-commerce y administrar diferentes aspectos de la aplicación.

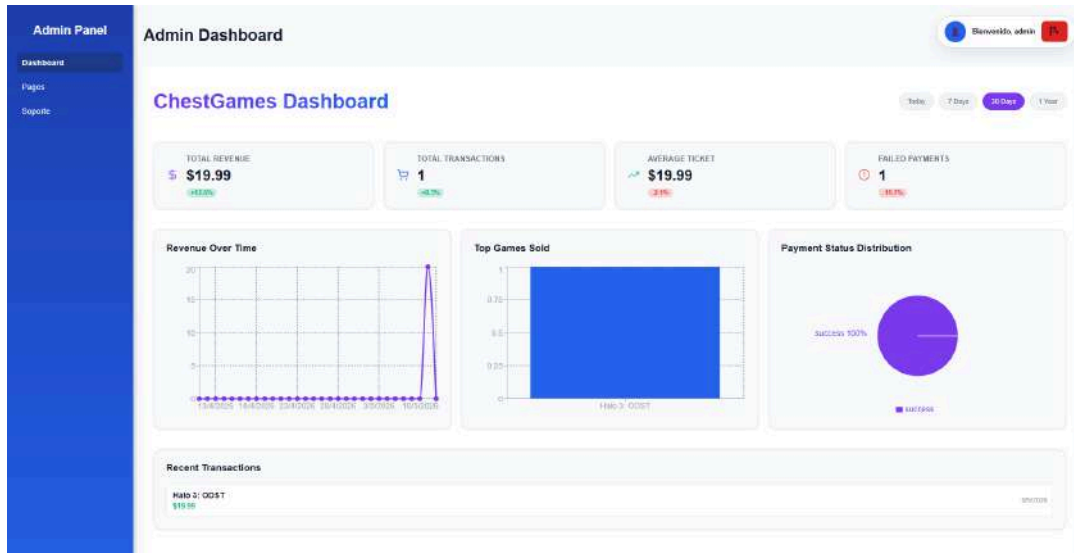
Dashboard y estadísticas

El administrador dispone de un dashboard principal donde se muestran estadísticas relevantes del sistema.

Entre la información disponible se encuentran:

- Actividad de usuarios.
- Volumen de compras.
- Número de pedidos.
- Estado general de la plataforma.

Estas estadísticas permiten realizar un seguimiento global del funcionamiento del e-commerce.



Gestión de pagos y pedidos

El administrador puede visualizar todas las transacciones realizadas en la plataforma.

El sistema permite consultar:

- Pagos realizados.
- Pedidos generados.
- Información de compras.
- Estado de transacciones.

Esta funcionalidad facilita el control administrativo y económico del sistema.

Gestión de incidencias

Desde el panel de administración, el administrador puede acceder a todos los tickets enviados por los usuarios.

El sistema permite:

- Revisar incidencias.
- Responder tickets.
- Gestionar solicitudes de soporte.

Esto facilita una comunicación directa y eficiente con los clientes.

Conclusión

El desarrollo de este proyecto ha permitido implementar una plataforma e-commerce completa orientada a la venta de videojuegos, integrando funcionalidades avanzadas de seguridad, gestión de usuarios y administración.

La utilización de tecnologías como Spring Boot, Spring Security, JWT, Stripe, Google Authenticator y Spring Mail ha permitido construir una aplicación segura, escalable y preparada para entornos reales.

Además, la separación entre usuarios clientes y administradores, junto con el sistema de soporte y gestión interna, proporciona una estructura sólida y organizada para el correcto funcionamiento de la plataforma.

El resultado final es una aplicación moderna que combina seguridad, usabilidad y automatización de procesos, ofreciendo una experiencia completa tanto para los usuarios como para la administración del sistema.

Flux de funcionament (Usuari):

1. **Acceso a la plataforma**

El usuario accede a la página principal, donde puede visualizar videojuegos destacados y navegar por las distintas secciones de la aplicación.

2. **Registro / Inicio de sesión**

El usuario puede registrarse en la plataforma o iniciar sesión con sus credenciales.

Una vez autenticado, obtiene acceso a todas las funcionalidades personalizadas.

3. **Exploración del catálogo**

El usuario accede al catálogo de videojuegos, donde puede visualizar todos los productos disponibles.

Puede utilizar el sistema de filtrado para encontrar videojuegos según sus preferencias.

4. **Consulta de producto**

El usuario selecciona un videojuego y accede a su ficha detallada, donde puede consultar información relevante como descripción, precio e imágenes.

5. **Acciones sobre el producto**

Desde la ficha del producto, el usuario puede:

- *Comprar el videojuego*
- *Añadirlo a la lista de deseos (wishlist)*

6. **Proceso de compra**

En caso de realizar una compra, el sistema procesa la operación y registra el pedido en la base de datos, asociándolo al usuario.

7. **Gestión del perfil**

El usuario puede acceder a su perfil para:

- *Modificar sus datos personales*
- *Consultar su historial de pedidos*
- *Gestionar su lista de deseos*

8. **Uso del sistema de soporte**

En caso de duda o incidencia, el usuario puede:

- *Consultar la sección de preguntas frecuentes (FAQ)*
- *Crear un ticket de soporte*

9. **Seguimiento de tickets**

El usuario puede consultar el estado de sus tickets y visualizar las respuestas proporcionadas por el administrador.

Flux de funcionament (Admin):

1. **Acceso al panel de administración**

El administrador accede al sistema mediante autenticación y entra en el panel de control.

2. **Visualización del dashboard**

Se muestran estadísticas generales de la plataforma, permitiendo obtener una visión global del sistema.

3. **Gestión de tickets**

El administrador revisa los tickets enviados por los usuarios y responde a las incidencias o consultas.

4. **Supervisión de pedidos y pagos**

El administrador puede visualizar los pedidos realizados y los pagos asociados, permitiendo el control de la actividad económica del sistema.

Arquitectura i Patrons de Disseny

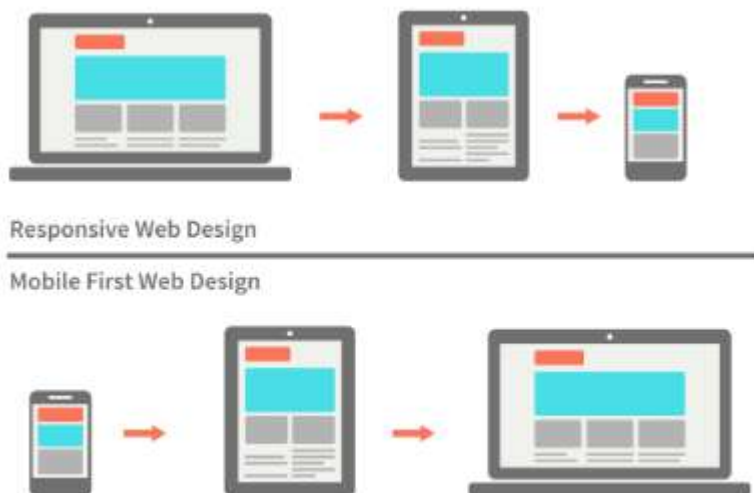
Les tecnologies i eines utilitzades en el nostre projecte, agrupades per categories lògiques que reflecteixen el flux global i l'arquitectura del projecte. Aquestes eines i tecnologies s'utilitzen per gestionar la lògica del servidor, la gestió de dades i les API.

- **API (Interfície de Programació d'Aplicacions):** L'API serveix per a la comunicació entre el frontend i el backend, permetent que es gestionin les dades de forma centralitzada. A través de les API es poden realitzar operacions de lectura, creació, actualització i eliminació de dades



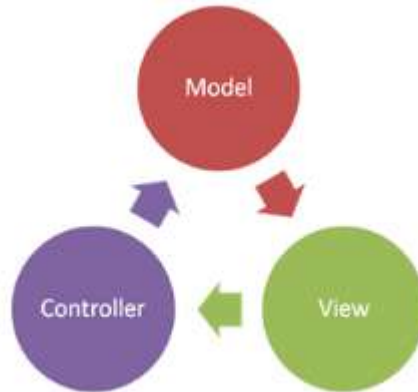
Aquestes tecnologies s'utilitzen per crear la interfície d'usuari (UI) i gestionar la interacció entre l'usuari i l'aplicació.

- **Disseny Responsiu:** El disseny responsiu garanteix que una pàgina web sigui compatible i es vegi de manera correcta a diferents dispositius (mòbils, tauletes, ordinadors de sobretaula). Permet adaptar el disseny de la pàgina en funció de la mida de la pantalla.



Aquestes eines i conceptes ajuden a estructurar l'aplicació de manera lògica i a garantir la seva mantenibilitat.

- **MVC (Model-View-Controller):** Patró de disseny que divideix l'aplicació en tres components: Model (gestiona les dades i la lògica de negoci), Vista (gestiona la presentació i la interfície d'usuari) i Controlador (gestiona la interacció entre el model i la vista). Aquest patró ajuda a organitzar el codi i a mantenir-lo net i escalable.



Aquestes eines i conceptes ajuden a estructurar l'aplicació de manera lògica i a garantir la seva mantenibilitat.

- **MVC (Model-View-Controller):** Patró de disseny que divideix l'aplicació en tres components: Model (gestiona les dades i la lògica de negoci), Vista (gestiona la presentació i la interfície d'usuari) i Controlador (gestiona la interacció entre el model i la vista). Aquest patró ajuda a organitzar el codi i a mantenir-lo net i escalable.

- **Aplicació Web de Pàgina Única (SPA):** Les aplicacions SPA carreguen només una sola pàgina HTML i actualitzen dinàmicament el contingut sense necessitat de recarregar la pàgina. Això permet una interacció més ràpida i una millor experiència d'usuari.

- **Router:** És el mecanisme utilitzat en les aplicacions SPA per gestionar la navegació entre diferents vistes o seccions de l'aplicació sense recarregar la pàgina. Permet que les rutes de l'URL canviïn mentre es mantenen les mateixes dades de l'aplicació.

Eines per al Desenvolupament i Col·laboració

Visual Studio Code (VS Code): Editor de código ligero pero muy potente que incluye funciones como IntelliSense, sugerencias de código, depuración, extensiones e integración con Git. Es una de



las herramientas más populares en el desarrollo de software moderno.



Figma: Se ha utilizado Figma como herramienta de diseño de interfaces para crear los prototipos de la plataforma antes de comenzar el desarrollo. Permite definir de forma visual la estructura de la web, comprobar su usabilidad y compartir los diseños con el equipo para recibir retroalimentación de manera rápida.

GitHub Copilot: Se ha utilizado GitHub Copilot como herramienta de inteligencia artificial integrada en VS Code que proporciona sugerencias automáticas de código. Es capaz de generar líneas de código e incluso funciones completas, lo que permite ahorrar tiempo y automatizar tareas repetitivas.



Swagger: Se ha utilizado esta herramienta para probar y depurar APIs. Permite enviar peticiones HTTP a los endpoints, visualizar las respuestas y detectar posibles errores de comunicación entre el frontend y el backend.

Chat GPT: Se trata de un modelo de inteligencia artificial desarrollado por OpenAI que asiste en la generación de código, la resolución de dudas de programación y la depuración de errores, facilitando así el trabajo del desarrollador.



Visualització de Dades

Aquestes eines permeten crear representacions visuals de les dades.

- **Chart.js:** Llibreria per crear gràfics interactius que permet visualitzar dades d'una manera atractiva i fàcil d'entendre. Ideal per mostrar estadístiques, informes o altres dades que requereixin representació gràfica

3. Desplegament i Monitorització

Aquestes eines són importants per gestionar l'aplicació un cop està en

producció.

- *pgAdmin*: Aquesta eina gràfica s'utilitza per gestionar la base de dades PostgreSQL. Permet als desenvolupadors realitzar consultes SQL, gestionar les taules i els índexs, així com controlar el rendiment de la base de dades, ajudant a assegurar-se que les dades s'emmagatzemen de manera eficient i segura.

4 Conclusions

4.1 Conclusions generals del projecte

El desenvolupament d'aquesta plataforma digital de gestió i reserva de vehicles de lloguer ha estat una experiència molt enriquidora tant acadèmicament com professionalment. A nivell acadèmic, ha permès aplicar de manera pràctica els coneixements adquirits al cicle formatiu de Desenvolupament d'Aplicacions Web, treballant amb tecnologies modernes com Laravel, Vue.js i PostgreSQL. A nivell professional, ha servit per simular un entorn real de desenvolupament col·laboratiu, gestionant tasques, dissenyant una arquitectura robusta i resolent problemes tècnics de forma autònoma i eficient. També ha potenciat habilitats clau com la planificació, la comunicació tècnica i la presa de decisions.

4.2 Consecució dels objectius

Crear un ecommerce funcional de videojuegos:

Objetivo logrado. *Se ha desarrollado una plataforma completa de comercio electrónico donde los usuarios pueden comprar videojuegos, navegar por el catálogo y realizar acciones básicas de tienda online.*

Sistema de compra de videojuegos:

Objetivo logrado. *Se ha implementado un sistema de compra funcional que permite a los usuarios adquirir videojuegos de forma segura dentro de la plataforma.*

Gestión de favoritos:

Objetivo logrado. Los usuarios pueden añadir videojuegos a su lista de favoritos para guardarlos y acceder a ellos rápidamente.

Sistema de soporte al usuario:

Objetivo logrado. Se ha implementado un sistema mediante el cual los usuarios pueden enviar solicitudes o mensajes al equipo de soporte.

Autenticación de usuarios:

Objetivo logrado. Se ha desarrollado un sistema de autenticación que permite iniciar sesión de forma segura en la plataforma.

Registro y cierre de sesión:

Objetivo logrado. Los usuarios pueden registrarse en la plataforma, iniciar sesión y cerrar sesión correctamente.

Gestión de perfil de usuario:

Objetivo logrado. Cada usuario dispone de un perfil propio donde puede visualizar y modificar sus datos personales.

Edición de datos del perfil:

Objetivo logrado. Se permite a los usuarios actualizar información de su cuenta como nombre, correo electrónico o contraseña.

Sistema de categorías de videojuegos:

Objetivo logrado. Los videojuegos están organizados por categorías para facilitar la navegación y búsqueda dentro del catálogo.

Carrito de compra persistente:

Objetivo logrado. Se ha implementado un carrito de compra que se mantiene guardado incluso cuando el usuario cierra sesión, permitiendo conservar los productos añadidos.

Panel de administración:

Objetivo logrado. Se ha desarrollado un panel de administración para gestionar videojuegos, usuarios y pedidos dentro de la plataforma.

Lista de deseos:

Objetivo logrado. Los usuarios pueden añadir videojuegos a una lista de deseos para guardarlos y consultarlos posteriormente.

Integración de pagos reales (Stripe / PayPal):

Objetivo logrado. Se ha integrado un sistema de pagos reales mediante plataformas como Stripe y/o PayPal, permitiendo realizar compras de forma segura.

Sección de “juegos más vendidos” o “tendencias”:

Objetivo logrado. Se ha implementado una sección que muestra los videojuegos más vendidos y las tendencias actuales, facilitando la visualización de los títulos más populares.

Sistema de filtros de búsqueda:

Objetivo logrado. Se han implementado filtros para buscar videojuegos según distintos criterios como género, precio o popularidad.

4.2.1 Objetivos no logrados

Durante el desarrollo del proyecto de e-commerce de videojuegos se han conseguido implementar la mayoría de las funcionalidades planteadas inicialmente. Sin embargo, algunos objetivos no han podido completarse dentro del tiempo previsto o han quedado pendientes para futuras versiones del sistema.

Sistema de recomendaciones personalizadas:

No logrado. Aunque la plataforma permite la navegación y búsqueda de videojuegos, no se ha implementado un sistema de recomendación basado en inteligencia artificial o en el comportamiento del usuario. Esta funcionalidad queda planteada como mejora futura.

Modo invitado o compra rápida:

No logrado. El sistema requiere autenticación obligatoria para realizar compras, por lo que no se ha desarrollado la opción de realizar compras como usuario invitado o mediante un proceso de registro simplificado.

Sistema de cupones o códigos de descuento:

No logrado. Actualmente la plataforma no dispone de un sistema de promociones, cupones o descuentos aplicables al carrito o al proceso de compra.

Sistema de valoraciones de videojuegos:

No logrado. Aunque los usuarios pueden comprar videojuegos y gestionarlos en su perfil, no se ha implementado la funcionalidad que permita dejar reseñas, puntuaciones o comentarios sobre los productos adquiridos.

Optimización avanzada de rendimiento y escalabilidad:

Parcialmente logrado. La aplicación funciona correctamente en condiciones normales, pero no se han realizado optimizaciones avanzadas orientadas a grandes volúmenes de usuarios simultáneos ni pruebas de carga exhaustivas.

Estos puntos representan oportunidades de mejora que podrían implementarse en futuras versiones del proyecto para aumentar la competitividad, la experiencia de usuario y la profesionalización de la plataforma.

4.3 Valoració de la metodologia i planificació

Tot i que inicialment es va contemplar la utilització de Jira com a eina de gestió de projectes, degut al nombre reduït de membres de l'equip (dues persones), es va optar per una gestió més pràctica i directa mitjançant una llista de tasques a Google Sheets. Aquesta solució va permetre un seguiment clar i estructurat, facilitant tant la planificació com l'assignació de responsabilitats. Tot i que no es va fer un ús estricte de Jira, l'enfocament adoptat va ser prou flexible per adaptar-se als canvis sorgits durant el desenvolupament. En definitiva, l'estratègia de gestió implementada va resultar adequada per a les necessitats del projecte i va contribuir positivament al seu bon desenvolupament.

4.4 Visió de futur

El proyecto de e-commerce de videojuegos ya cuenta con las funcionalidades principales de una tienda online completa, incluyendo sistema de login, carrito de compra, panel de administración, soporte al usuario, historial de compras y perfil de usuario. A pesar de ello, la arquitectura del sistema permite su ampliación y mejora en futuras versiones.

Las posibles líneas de evolución del proyecto se centran en la optimización, escalabilidad y mejora de la experiencia del usuario:

- **Sistema de recomendaciones personalizadas**, basado en inteligencia artificial o análisis del comportamiento del usuario dentro de la plataforma.
- **Mejora del sistema de soporte**, incorporando un chatbot automatizado o asistencia en tiempo real más avanzada.
- **Optimización del rendimiento de la plataforma**, mejorando tiempos de carga, consultas a base de datos y escalabilidad para un mayor número de usuarios simultáneos.

- **Aplicación móvil nativa**, que permita una experiencia de compra más rápida y accesible desde dispositivos móviles.
- **Modo invitado o compra rápida**, permitiendo comprar sin registro completo (o con registro simplificado).
- **Sistema de cupones o códigos de descuento**, que permita aplicar promociones en el carrito o durante la compra.
- **Sistema de valoraciones de videojuegos**, donde los usuarios puedan puntuar y opinar sobre los juegos adquiridos, ayudando a otros usuarios en sus decisiones de compra.

Estas mejoras permitirían transformar la plataforma en un entorno más profesional, competitivo y adaptado a los estándares actuales del comercio electrónico en el sector de los videojuegos.

5. Glossari

API (Interfaz de Programación de Aplicaciones): Es una interfaz que permite la comunicación entre diferentes programas o componentes de software. En este proyecto se utiliza para intercambiar datos entre el frontend (Vue.js) y el backend (Laravel).

CRUD (Crear, Leer, Actualizar, Eliminar): Acrónimo que hace referencia a las cuatro operaciones básicas que se pueden realizar sobre una base de datos: crear registros, leerlos, modificarlos y eliminarlos.

Frontend: Es la parte visual de una aplicación web con la que interactúa el usuario. Este proyecto se ha desarrollado con Vue.js y Bootstrap.

Backend: Es la parte interna de una aplicación que gestiona la lógica del sistema, la seguridad y la conexión con la base de datos. En este proyecto se ha implementado con Laravel (PHP).

SPA (Single Page Application): Aplicación web que carga una única página HTML y actualiza el contenido de forma dinámica sin necesidad de recargar completamente la página.

MVC (Modelo-Vista-Controlador): Patrón de diseño de software que divide una aplicación en tres componentes: Modelo (datos), Vista (interfaz de usuario) y Controlador (lógica que conecta ambos).

Spring Boot: Framework de código abierto basado en Java que facilita el desarrollo de aplicaciones y APIs REST. Proporciona herramientas para la gestión de rutas, seguridad, autenticación y acceso a bases de datos de forma rápida y escalable.

React: Biblioteca de JavaScript utilizada para crear interfaces de usuario interactivas y dinámicas. Se emplea principalmente para el desarrollo del frontend mediante componentes reutilizables.

Ionic: Framework multiplataforma que permite desarrollar aplicaciones móviles y web utilizando tecnologías como HTML, CSS y JavaScript junto con React.

MySQL: Sistema de gestión de bases de datos relacional de código abierto, conocido por su rendimiento, fiabilidad y capacidad de escalabilidad.

Spring Security: Framework de seguridad para Spring utilizado para proteger aplicaciones y APIs mediante autenticación y control de acceso basado en roles y permisos.

JWT (JSON Web Token): Estándar de autenticación basado en tokens utilizado para proteger APIs y gestionar sesiones de usuario de forma segura en aplicaciones web y móviles.

Axios: Librería de JavaScript basada en promesas que permite realizar peticiones HTTP de forma sencilla. Se utiliza para conectar el frontend con la API desarrollada en Spring Boot.

Chart.js: Librería de JavaScript utilizada para generar gráficos visuales como barras, líneas o sectores, permitiendo representar estadísticas e información de manera dinámica.

Bootstrap: Framework de CSS utilizado para diseñar interfaces modernas, adaptativas y responsivas de forma rápida y eficiente.

Context API: Herramienta integrada en React para la gestión global del estado de la aplicación, permitiendo compartir información entre componentes sin necesidad de pasar propiedades manualmente.

6. Bibliografía

- *Ionic Framework*
Ionic Team. (2026). *Ionic Framework Documentation*. Consultado en: [Ionic Framework Official Site](#)
- *Supabase*
Supabase Inc. (2026). *Supabase Documentation*. Consultado en: [Supabase Official Site](#)
- *Spring Boot*
VMware / Spring Team. (2026). *Spring Boot Reference Documentation*. Consultado en: [Spring Boot Documentation](#)
- *React*
Meta Platforms, Inc. (2026). *React Documentation*. Consultado en: [React Official Site](#)

- *MySQL*
Oracle Corporation. (2026). *MySQL Documentation*. Consultado en: [MySQL Official Site](#)
- *CSS*
W3C. (2026). *CSS: Cascading Style Sheets*. Consultado en: [W3C CSS Specification](#)
- *HTML*
W3C. (2026). *HTML Standard*. Consultado en: [W3C HTML Specification](#)
- *TypeScript*
Microsoft Corporation. (2026). *TypeScript Documentation*. Consultado en: [TypeScript Official Site](#)