



C.P.R. Liceo “La Paz”

Proyecto Fin de Ciclo

Desarrollo de Aplicaciones Web

PixelVault

Autor: Adrián Castro Beiro
Tutor: Jesús Ángel Roca

Resumen

PixelVault es una plataforma de comercio electrónico especializada en el sector de videojuegos, diseñada para ofrecer una experiencia premium a los entusiastas del gaming. La tienda virtual se distingue por su amplia selección de productos de alta calidad, incluyendo consolas de última generación de Nintendo, Xbox, PlayStation y PC gaming, así como videojuegos exclusivos, ediciones coleccionista y los últimos lanzamientos del mercado.

El proyecto integra tecnologías modernas como .NET 8 para el backend, Vue.js para el frontend y sistemas de pago seguros, proporcionando una solución robusta y escalable para el comercio electrónico especializado.

Abstract

PixelVault is an e-commerce platform specialized in video games that provides a premium shopping experience and offers high-quality gaming products for enthusiasts and players. The online store is distinguished by its comprehensive selection of gaming-related merchandise across all major platforms: Nintendo, Xbox, PlayStation, and PC gaming.

The store features the latest generation consoles, exclusive video game titles, collector's editions, premium peripherals, and special limited releases from the most recognized brands in the industry. PixelVault is committed to delivering exceptional service and carefully curated products to satisfy both casual and hardcore gamers, ensuring the highest quality standards throughout its entire catalog.

Palabras Clave

.NET 8: Marco de desarrollo creado por Microsoft que proporciona un entorno para construir aplicaciones web, móviles y de escritorio. Es la versión más reciente del framework con mejoras en rendimiento y características.

C#: Lenguaje de programación orientado a objetos desarrollado por Microsoft como parte de la plataforma .NET. Se utiliza para desarrollar el backend de PixelVault.

Vue.js: Framework progresivo de JavaScript para construir interfaces de usuario. Vue facilita la creación de componentes reutilizables y la gestión del estado de la aplicación en el frontend.

JavaScript: Lenguaje de programación dinámico, interpretado y de alto nivel que se ejecuta principalmente en navegadores web, aunque también en servidores (Node.js) y otros entornos.

Tailwind CSS: Framework de CSS utilitario que permite construir diseños personalizados sin salir del HTML, facilitando el desarrollo rápido de interfaces responsivas.

E-commerce: Sistema de compra y venta de productos y servicios a través de medios electrónicos como Internet. PixelVault es un e-commerce especializado en videojuegos y periféricos.

API RESTful: Interfaz que permite la comunicación entre el frontend y backend de la aplicación mediante HTTP, siguiendo principios de diseño que facilitan la escalabilidad y mantenimiento.

Gaming: Término que abarca la actividad de jugar videojuegos y la cultura que la rodea, siendo el mercado objetivo principal de PixelVault.

Sumario

| | |
|---|----|
| Resumen..... | 3 |
| Abstract..... | 4 |
| Palabras Clave..... | 5 |
| Introducción/motivación..... | 9 |
| Fundamentos Estratégicos..... | 9 |
| Especialización en el Nicho Gamer..... | 9 |
| Oportunidad de Mercado Identificada..... | 9 |
| Implementación Tecnológica Avanzada..... | 9 |
| Visión de Ecosistema Integral..... | 10 |
| Motivación Personal y Técnica..... | 10 |
| Propósito Final..... | 10 |
| Objetivos..... | 11 |
| Interfaz y Usabilidad..... | 11 |
| Navegación y Accesibilidad..... | 11 |
| Objetivos Técnicos y Arquitectura..... | 11 |
| Objetivos de Contenido y Organización..... | 12 |
| Objetivos de Negocio y Posicionamiento..... | 12 |
| Estado del arte..... | 13 |
| Análisis del Mercado Actual..... | 13 |
| Plataformas Líderes del Mercado..... | 13 |
| Comparación con PixelVault..... | 14 |
| Caso de estudio..... | 16 |
| Diagramas..... | 17 |
| Modelo entidad relación:..... | 17 |
| Casos de uso:..... | 18 |
| Uml:..... | 19 |
| Diagrama de componentes:..... | 20 |
| Diagrama de bases de datos:..... | 21 |
| Núcleo del Sistema: Productos..... | 22 |
| Gestión Comercial..... | 22 |
| Migración y Control..... | 22 |
| Estado Actual..... | 22 |
| Diagrama de flujo:..... | 23 |
| 1.1.Registro..... | 23 |
| 1.2.Inicio Sesión..... | 24 |

Proyecto Desarrollo de Aplicaciones Web

| | |
|---------------------------------------|----|
| 1.3.Proceso de compra..... | 25 |
| 1.4.Privilegios del Admin..... | 27 |
| 1.5.Flujo de Usuario por la web..... | 28 |
| Mapa de Navegacion..... | 30 |
| Diseño..... | 32 |
| Wireframes:..... | 32 |
| Mockups:..... | 36 |
| Guia de estilos:..... | 41 |
| Desarrollo del proyecto..... | 42 |
| Navegación y UI..... | 42 |
| Autenticación y Seguridad..... | 42 |
| Modelo de Dominio..... | 42 |
| Carrito de Compra y Sesiones..... | 42 |
| Backend y Servicios..... | 43 |
| Rendimiento y Optimización..... | 43 |
| Concurrencia y Consistencia..... | 43 |
| Extensibilidad y Personalización..... | 43 |
| Testing..... | 43 |
| Conclusión del desarrollo..... | 44 |
| Manual Administrador..... | 45 |
| Manual Usuario..... | 46 |
| 1-Registro..... | 46 |
| 3-Home..... | 48 |
| 4-Producto según la plataforma..... | 50 |
| 5-Producto detalle..... | 51 |
| 6-Carrito..... | 52 |
| 8-Sobre nosotros..... | 54 |
| Viabilidad tecno-económica..... | 57 |
| Trabajo futuro..... | 58 |
| Conclusiones..... | 59 |
| Lo que aprendí..... | 59 |
| Impacto en el mercado..... | 59 |
| Tecnologías que usé..... | 60 |
| Cómo trabajé..... | 60 |
| Planes para el futuro:..... | 60 |
| Mi conclusión personal..... | 61 |

Proyecto Desarrollo de Aplicaciones Web

| | |
|---|----|
| Biblioteca de recursos web y referencias..... | 62 |
| Paquetes NuGet .NET..... | 62 |
| Recursos de Imágenes..... | 63 |
| Anexos..... | 64 |

Introducción/motivación.

PixelVault surge como un proyecto innovador en el ámbito del comercio electrónico especializado en productos para gamers. En un mercado digital saturado de tiendas generalistas, se identificó la necesidad de crear una plataforma dedicada exclusivamente a la comunidad gamer, con una experiencia de usuario optimizada y un enfoque en la creación de una comunidad activa.

Fundamentos Estratégicos

La decisión de desarrollar PixelVault como plataforma de e-commerce especializada en productos gaming se sustenta en una combinación de factores estratégicos, técnicos y personales que convergen en una oportunidad única de mercado.

Especialización en el Nicho Gamer

A diferencia de las plataformas generalistas existentes, PixelVault se posiciona como una solución específicamente diseñada para la comunidad gamer. Esta especialización permite ofrecer una experiencia de usuario más personalizada y relevante, adaptada a las necesidades particulares de los jugadores, desde la presentación de productos hasta las funcionalidades de búsqueda y recomendación.

Oportunidad de Mercado Identificada

El sector de productos para gamers mantiene un crecimiento constante y sostenido, con consumidores que demuestran una alta disposición a invertir en equipamiento y accesorios de calidad. Esta tendencia, combinada con la percepción de que las plataformas actuales no satisfacen completamente las expectativas de esta comunidad específica, representa una oportunidad significativa para una solución más enfocada y especializada.

Implementación Tecnológica Avanzada

El desarrollo de PixelVault presenta la oportunidad de integrar múltiples tecnologías modernas en un ecosistema cohesivo:

- Autenticación segura mediante tokens JWT
- Procesamiento de pagos a través de múltiples pasarelas (PayPal y Stripe)
- Gestión de medios optimizada con almacenamiento en la nube (Cloudinary)
- Comunicación automatizada mediante sistemas de emails transaccionales
- Interfaz moderna desarrollada con Vue.js y Tailwind CSS

Visión de Ecosistema Integral

Más allá de funcionar como una simple tienda online, PixelVault aspira a convertirse en un ecosistema completo para la comunidad gamer, incorporando funcionalidades sociales y de participación que fomenten la interacción entre usuarios y la construcción de una verdadera comunidad.

Motivación Personal y Técnica

La motivación principal del proyecto surge de la convergencia entre la pasión personal por los videojuegos y la oportunidad de aplicar conocimientos técnicos avanzados para resolver un problema real identificado en el mercado. La experiencia personal como jugador ha permitido identificar las carencias existentes en las plataformas actuales y visualizar una solución que mejore sustancialmente la experiencia de compra.

El deseo de crear una alternativa superior a las opciones existentes, caracterizada por un diseño moderno y una experiencia de usuario excepcional, constituye el núcleo de la motivación que impulsa el desarrollo de PixelVault.

Propósito Final

PixelVault representa la materialización de la visión de combinar la pasión por los videojuegos con las capacidades del desarrollo web moderno, creando una plataforma que no solo facilite las transacciones comerciales, sino que eleve la experiencia completa de descubrimiento, evaluación y adquisición de productos gaming para toda la comunidad de jugadores.

Objetivos.

Interfaz y Usabilidad

- Desarrollar una interfaz moderna y minimalista que refleje los estándares actuales del diseño web, con especial énfasis en un navbar de altura reducida que maximice el espacio de contenido
- Mantener coherencia visual absoluta en toda la plataforma mediante la implementación de un sistema de diseño unificado
- Garantizar un diseño completamente responsivo que proporcione una experiencia óptima en dispositivos móviles, tablets y escritorio

Navegación y Accesibilidad

- Crear una experiencia de navegación intuitiva y fluida que minimice la curva de aprendizaje para nuevos usuarios
- Facilitar el acceso rápido y eficiente a productos, categorías y marcas mediante sistemas de búsqueda y filtrado avanzados
- Implementar una visualización clara y atractiva de productos que destaque sus características principales

Objetivos Técnicos y Arquitectura

Frontend Moderno:

- Construir una aplicación web robusta utilizando Vue.js como framework principal y Tailwind CSS para el sistema de estilos
- Implementar un sistema de enrutamiento eficiente que garantice navegación fluida y gestión adecuada del estado de la aplicación
- Desarrollar componentes reutilizables y modulares que faciliten el mantenimiento y la escalabilidad del código

Backend y Servicios:

- Desarrollar una API RESTful robusta con .NET 8 y C# 12 para gestionar usuarios, productos, carritos de compra y procesamiento de pedidos
- Implementar múltiples pasarelas de pago integradas (PayPal y Stripe) para ofrecer flexibilidad en las opciones de pago
- Establecer un sistema de autenticación segura mediante tokens JWT con diferenciación de roles y permisos
- Integrar servicios cloud especializados como Cloudinary para la gestión optimizada de imágenes y medios
- Crear un sistema de comunicaciones automatizadas mediante emails transaccionales personalizados

Objetivos de Contenido y Organización

Gestión de Información:

- Organizar productos, categorías y marcas de manera clara, lógica y fácilmente navegable
- Mantener información actualizada y precisa sobre productos, incluyendo especificaciones técnicas, precios y disponibilidad
- Implementar un sistema de gestión de contenido que facilite la actualización y mantenimiento de la información

Comunicación Corporativa:

- Desarrollar secciones informativas sobre la empresa, incluyendo valores, misión y información de contacto
- Crear canales de comunicación efectivos entre la plataforma y los usuarios

Objetivos de Negocio y Posicionamiento

Especialización del Mercado:

- Establecer PixelVault como una plataforma líder especializada exclusivamente en productos y servicios para la comunidad gamer
- Crear un ecosistema digital que no solo facilite transacciones comerciales, sino que construya una comunidad en torno a la cultura gaming

Escalabilidad y Crecimiento:

- Diseñar una arquitectura técnica escalable que permita el crecimiento orgánico de la plataforma sin comprometer el rendimiento
- Facilitar la gestión eficiente de productos, categorías y operaciones comerciales mediante herramientas administrativas intuitivas
- Establecer las bases tecnológicas para futuras expansiones de funcionalidad y alcance de mercado

Estado del arte.

Análisis del Mercado Actual

El ecosistema de distribución digital de videojuegos está dominado por varias plataformas establecidas, cada una con características distintivas que definen diferentes segmentos del mercado. A continuación se presenta un análisis comparativo de las principales alternativas existentes.

Plataformas Líderes del Mercado

Steam (Valve Corporation)

Fortalezas:

- Dominio absoluto del mercado con el catálogo más extenso de juegos PC disponible
- Ecosistema social robusto que incluye logros, rankings, foros comunitarios y sistema de intercambio
- Funcionalidades avanzadas como streaming, workshop de mods y sistema de tarjetas coleccionables
- Infraestructura de distribución global altamente optimizada

Debilidades:

- Interfaz de usuario que mantiene elementos de diseño obsoletos desde su lanzamiento en 2003
- Estructura de precios que no siempre resulta competitiva frente a otras plataformas
- Limitaciones en la personalización de la experiencia de usuario
- Dependencia excesiva del cliente de escritorio para funcionalidades completas

Stack Tecnológico: Arquitectura propietaria basada en C++, con componentes web utilizando HTML5 y JavaScript para la interfaz de tienda.

Epic Games Store

Fortalezas:

- Diseño visual moderno y minimalista que prioriza la experiencia de usuario
- Estrategia agresiva de adquisición de usuarios mediante juegos gratuitos semanales
- Integración nativa con el ecosistema Unreal Engine para desarrolladores
- Términos comerciales más favorables para desarrolladores (88% vs 70% de Steam)

Proyecto Desarrollo de Aplicaciones Web

Debilidades:

- Catálogo significativamente más limitado comparado con competidores establecidos
- Ausencia de características sociales y comunitarias desarrolladas
- Rendimiento de descarga inconsistente en diferentes regiones geográficas
- Falta de compatibilidad con sistemas operativos alternativos

Stack Tecnológico: Solución propietaria construida sobre C++ con componentes web modernos utilizando tecnologías HTML5.

Comparación con PixelVault

Ventajas:

Tecnologías Frontend:

- Vue.js (Framework principal)
- Tailwind CSS (Sistema de estilos)
- Axios (Para llamadas API)

Tecnologías Backend:

- .NET 8 (Framework principal)
- Entity Framework Core (ORM)
- ASP.NET Core Web API (API REST)
- JWT (JSON Web Tokens) para autenticación

Características de Seguridad:

- JWT para autenticación y autorización
- Tokens de acceso y refresh
- Validación de tokens en cada petición
- Protección contra CSRF
- Encriptación de contraseñas
- Gestión de sesiones segura

Con esta arquitectura, PixelVault combina:

- Una interfaz moderna con Vue.js
- Una API robusta con .NET 8
- Una base de datos eficiente con Entity Framework
- Un sistema de seguridad sólido con JWT

Proyecto Desarrollo de Aplicaciones Web

Posibles Alternativas Tecnológicas:

Frontend:

- React.js: Alternativa popular con más recursos
- Angular: Más robusto para aplicaciones empresariales
- Svelte: Más ligero y rápido

Estilos:

- Material UI: Sistema de diseño más completo
- Bootstrap: Alternativa más tradicional
- CSS Modules: Para mayor encapsulación

Backend:

- Node.js con Express
- Python con Django/Flask
- Java con Spring

La elección de .NET 8, Vue.js y Tailwind CSS para PixelVault representa un equilibrio entre modernidad, rendimiento y facilidad de desarrollo, permitiendo crear una experiencia de usuario superior a las plataformas más tradicionales mientras mantiene la flexibilidad para implementar características específicas de gaming.

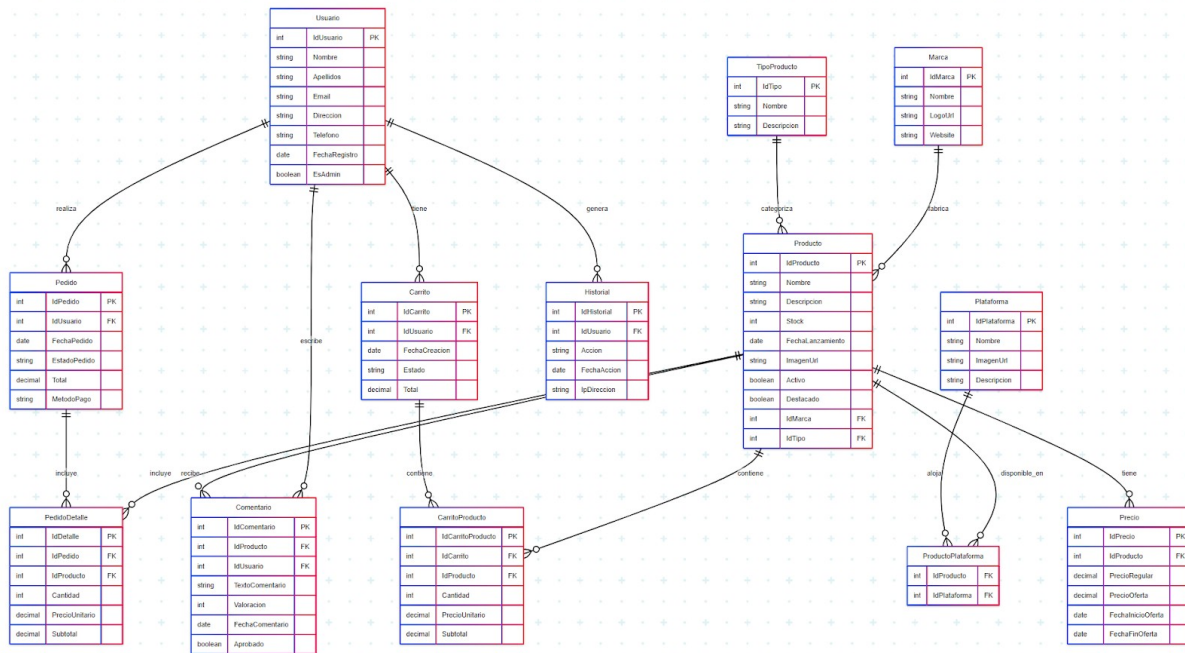
Caso de estudio.

Durante el desarrollo del proyecto se han identificado varias áreas que podrían mejorarse para aumentar la escalabilidad, mantenibilidad y robustez de la aplicación. Estas mejoras no fueron implementadas inicialmente por razones de tiempo o alcance, y se consideran actualmente como **deudas técnicas**:

1. **Internacionalización del sistema (i18n)**: Implementar soporte para múltiples idiomas mediante el almacenamiento de traducciones en la base de datos. Actualmente, la aplicación solo contempla un idioma, lo que limita su expansión internacional.
2. **Gestión de favoritos persistente**: Actualmente, la lista de favoritos se guarda en la sesión del usuario, lo cual no es persistente ni escalable. Se propone crear una nueva tabla en la base de datos para almacenar los favoritos de forma asociada al usuario.
3. **Validaciones adicionales**: Es necesario reforzar la validación de entradas en los formularios y en los endpoints, para asegurar la integridad de los datos y mejorar la seguridad de la aplicación.
4. **Códigos de estado HTTP más específicos**: Utilizar una mayor variedad de códigos HTTP para proporcionar información más detallada sobre los errores o estados de las peticiones. Esto facilitará la depuración y el manejo de errores en el futuro.
5. **Persistencia de noticias**: Las noticias actualmente no se almacenan en una base de datos, lo que limita su gestión y visualización futura. Se propone crear una tabla específica para almacenar estas noticias de forma estructurada.
6. **Sistema de códigos de descuento**: Aunque existe un apartado para introducir códigos de descuento, actualmente no está funcional. Se propone almacenar los códigos en base de datos y desarrollar la lógica de validación y aplicación de los mismos.
7. **Endpoints para gestión de plataformas**: Crear endpoints específicos para gestionar plataformas (añadir, actualizar o eliminar), anticipándose a futuras necesidades del sistema y mejorando la modularidad.

Diagramas.

Modelo entidad relación:



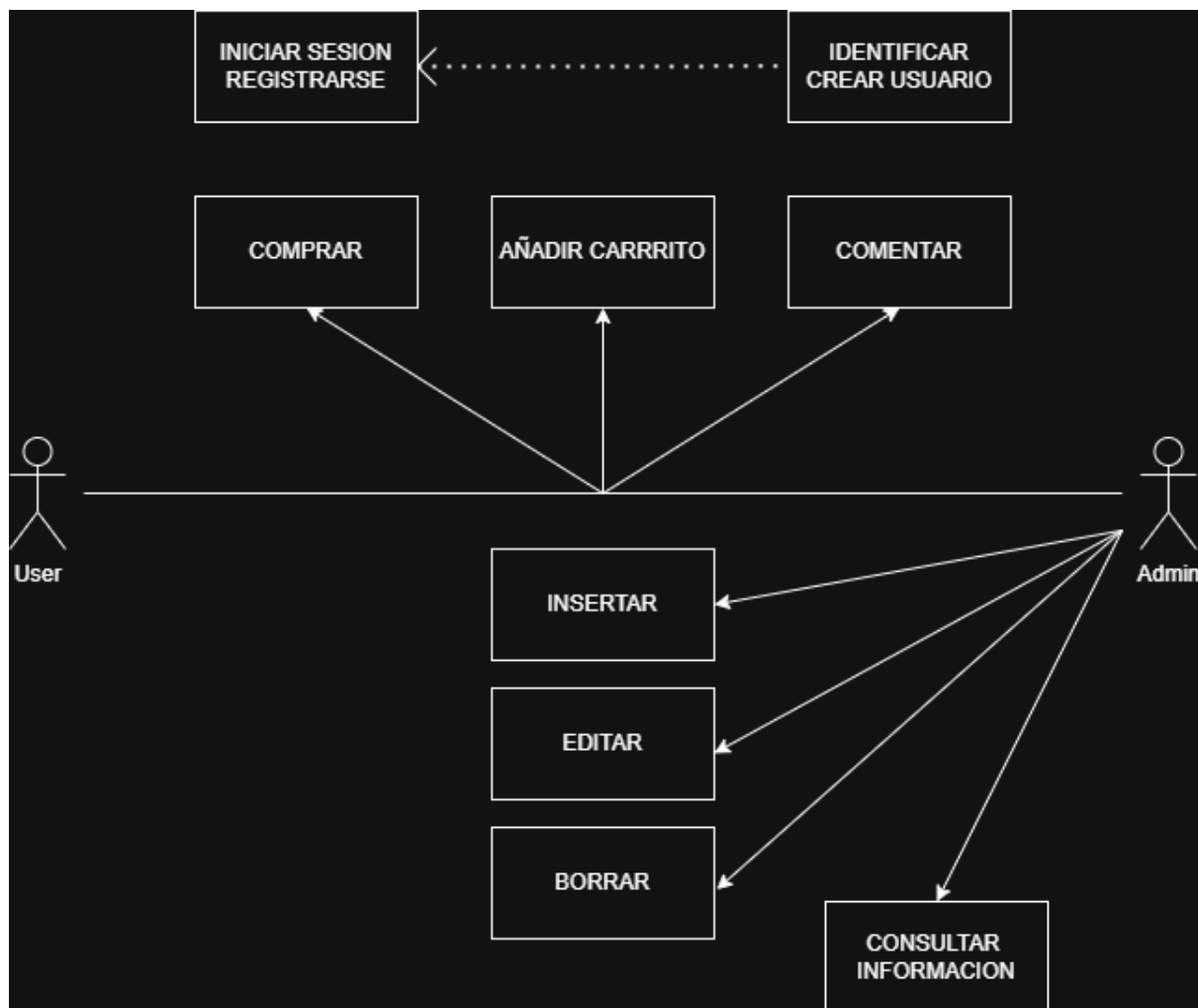
Este diagrama representa la base de datos de PixelVault. Los usuarios pueden hacer pedidos, tener carritos, dejar comentarios y generar historial de acciones.

Los productos se clasifican por marca y tipo, y pueden estar disponibles en varias plataformas. Además, tienen precios normales y en oferta con fechas específicas.

Los carritos almacenan productos antes de comprar, y al finalizar se convierten en pedidos, que registran detalles como cantidades, precios y método de pago.

Los comentarios permiten a los usuarios opinar sobre los productos, y pueden ser aprobados por un administrador. El historial guarda acciones hechas por cada usuario.

Casos de uso:



Este diagrama de casos de uso muestra las acciones de un usuario y un administrador. El usuario puede registrarse, iniciar sesión, comprar, comentar y añadir productos al carrito. El administrador puede insertar, editar, borrar y consultar información. Ambos interactúan con el sistema según sus permisos.

Los **productos** tienen atributos como nombre, precio, stock y descripción, y pueden estar relacionados con comentarios de los usuarios. Todo el sistema está organizado para permitir el flujo natural de una tienda en línea: desde la navegación y selección de productos hasta el pago y registro del pedido.

Diagrama de componentes:

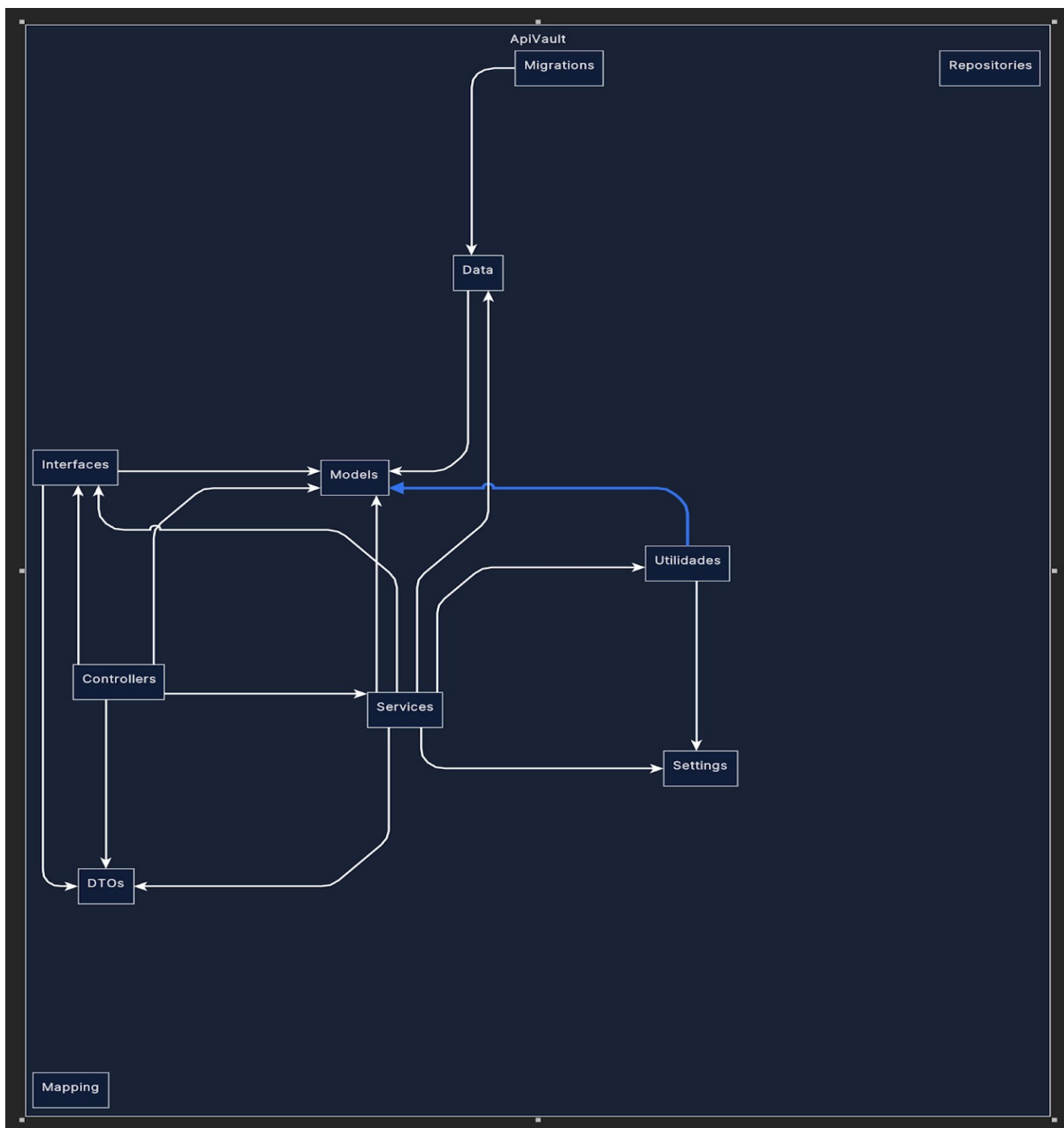
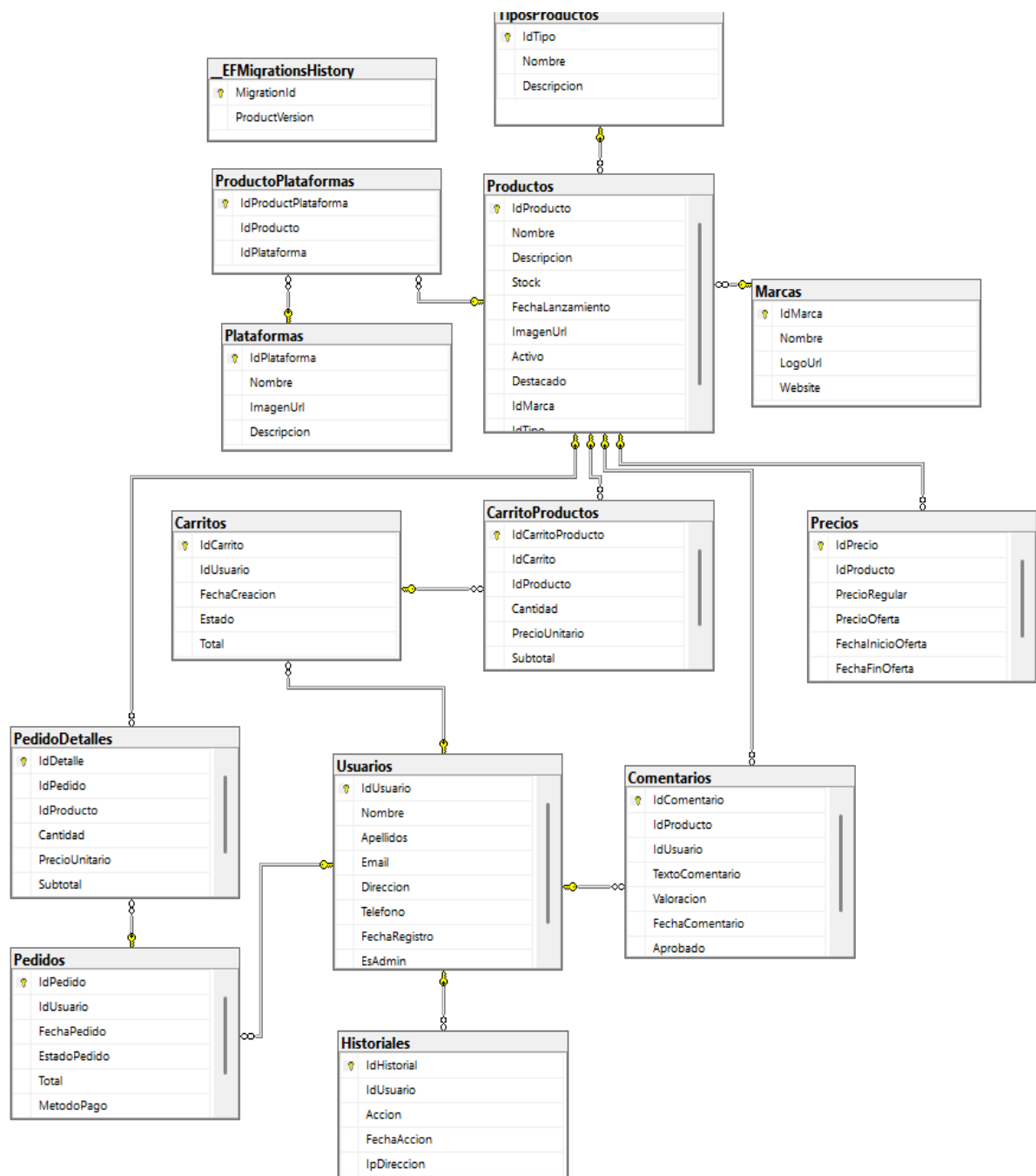


Diagrama que ilustra la arquitectura de una aplicación siguiendo el patrón **MVC (Model-View-Controller)** o una arquitectura en capas.

Diagrama de bases de datos:



Este diagrama representa la estructura de datos de PixelVault, un e-commerce especializado en productos gaming y tecnológicos. El diseño se centra en la gestión completa del ciclo de compra, desde el catálogo hasta la facturación.

Núcleo del Sistema: Productos

La tabla Productos constituye el corazón del sistema, almacenando información básica como nombre, descripción, stock y estado. Se relaciona estratégicamente con tres tablas clasificadoras fundamentales:

- TiposProductos: Define la categoría del producto (mandos, periféricos o videojuegos)
- Marcas: Registra la empresa desarrolladora o fabricante (ej: Rockstar Games para GTA 6)
- Plataformas: Especifica la plataforma de destino (PlayStation, Xbox, PC, etc.)

Esta arquitectura permite clasificar productos de manera precisa. Por ejemplo, un videojuego como GTA 6 tendría tipo "videojuego", marca "Rockstar Games" y plataforma "PlayStation".

Gestión Comercial

El sistema maneja el flujo de ventas a través de:

- Usuarios: Información completa de clientes registrados
- Carritos: Gestión del carrito de compras temporal
- Pedidos y PedidoDetalles: Procesamiento de órdenes con desglose detallado
- Precios: Control de precios regulares y ofertas especiales

Migración y Control

Las tablas EFMigrationsHistory y ProductoPlataformas gestionan el versionado de la base de datos y las relaciones muchos-a-muchos entre productos y plataformas respectivamente.

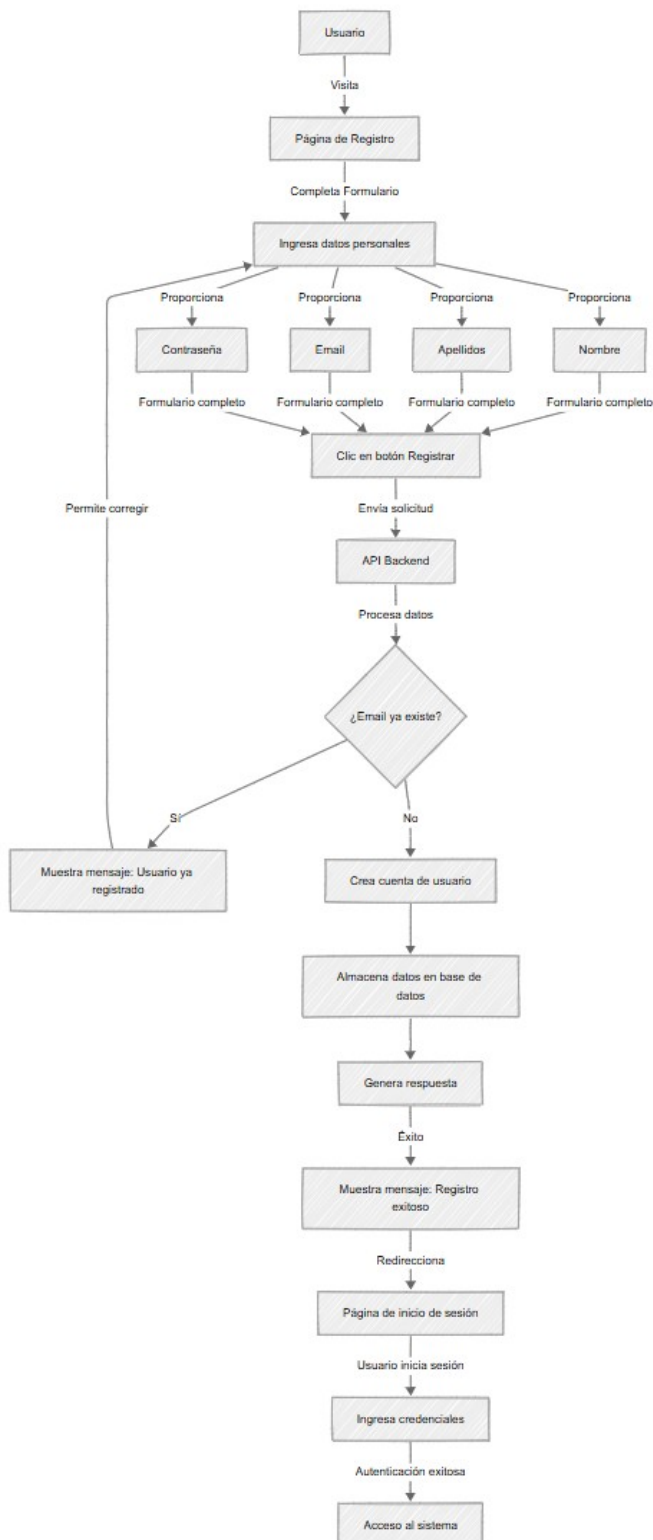
Estado Actual

Las tablas Historiales y Comentarios están implementadas estructuralmente pero no se utilizan operativamente, quedando como deuda técnica para futuras funcionalidades de seguimiento de acciones y sistema de reseñas.

El diseño garantiza escalabilidad y flexibilidad para el crecimiento del catálogo y la incorporación de nuevas características comerciales.

Diagrama de flujo:

1.1.Registro



Este diagrama muestra la experiencia del usuario durante el proceso de registro:

- 1.El usuario visita la página de registro
- 2.Completa el formulario con sus datos personales:

- Nombre
- Apellidos
- Email
- Contraseña

- 3.Al hacer clic en el botón "Registrar", los datos se envían al backend

- 4.El sistema verifica si el email ya está registrado:

- Si ya existe, muestra un mensaje de error y permite corregir

- Si no existe, crea la cuenta del usuario

- 5.Si el registro es exitoso, se muestra un mensaje de confirmación

- 6.El usuario es redirigido a la página de inicio de sesión

- 7.El usuario ingresa sus credenciales recién creadas

- 8.Tras una autenticación exitosa, obtiene acceso al sistema

1.2.Inicio Sesión

Este diagrama representa la experiencia del usuario durante el proceso de inicio de sesión:

1.El usuario visita la página de inicio de sesión

2.Completa el formulario con sus credenciales:

- Email

- Contraseña

3.Al hacer clic en "Iniciar Sesión", los datos se envían al backend

4.El sistema verifica si el usuario existe:

- Si no existe, muestra un mensaje de error

- Si existe, continúa con la validación de la contraseña

5.El sistema verifica si la contraseña es correcta:

- Si es incorrecta, muestra un mensaje de error

- Si es correcta, genera un token JWT

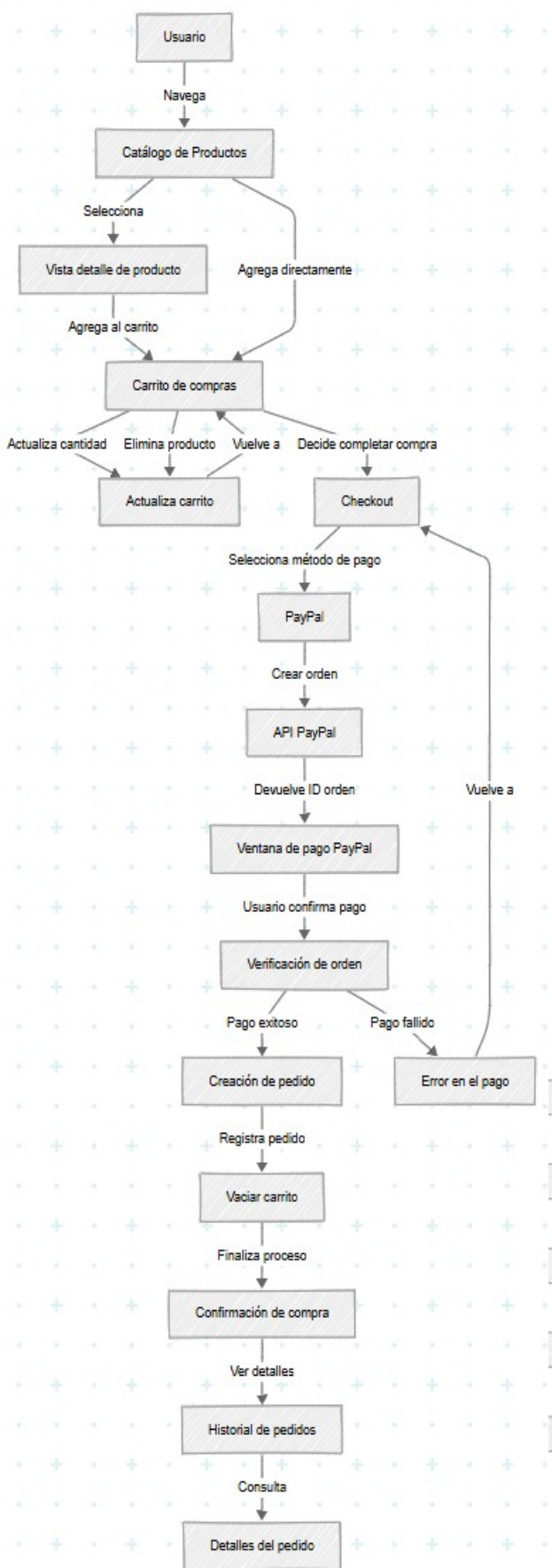
6.El token se devuelve al cliente y se almacena en sessionStorage

7.Se establece la sesión del usuario y se redirige a la página principal

8.El usuario puede navegar por la aplicación con acceso a las funcionalidades autorizadas

9.Para cada solicitud a recursos protegidos, el token JWT se incluye en las cabeceras de la petición

10.Cuando el token expira (según la configuración en appsettings.json), la sesión finaliza y se requiere un nuevo inicio de sesión



1.3.Proceso de compra

1.Exploración de productos:

- El usuario navega por el catálogo de productos
- Puede ver detalles de un producto específico
- Puede agregar productos al carrito desde el catálogo o desde la vista de detalle

2.Gestión del carrito:

- El usuario puede actualizar las cantidades de los productos
- Eliminar productos del carrito
- Visualizar el total a pagar

3.Proceso de checkout:

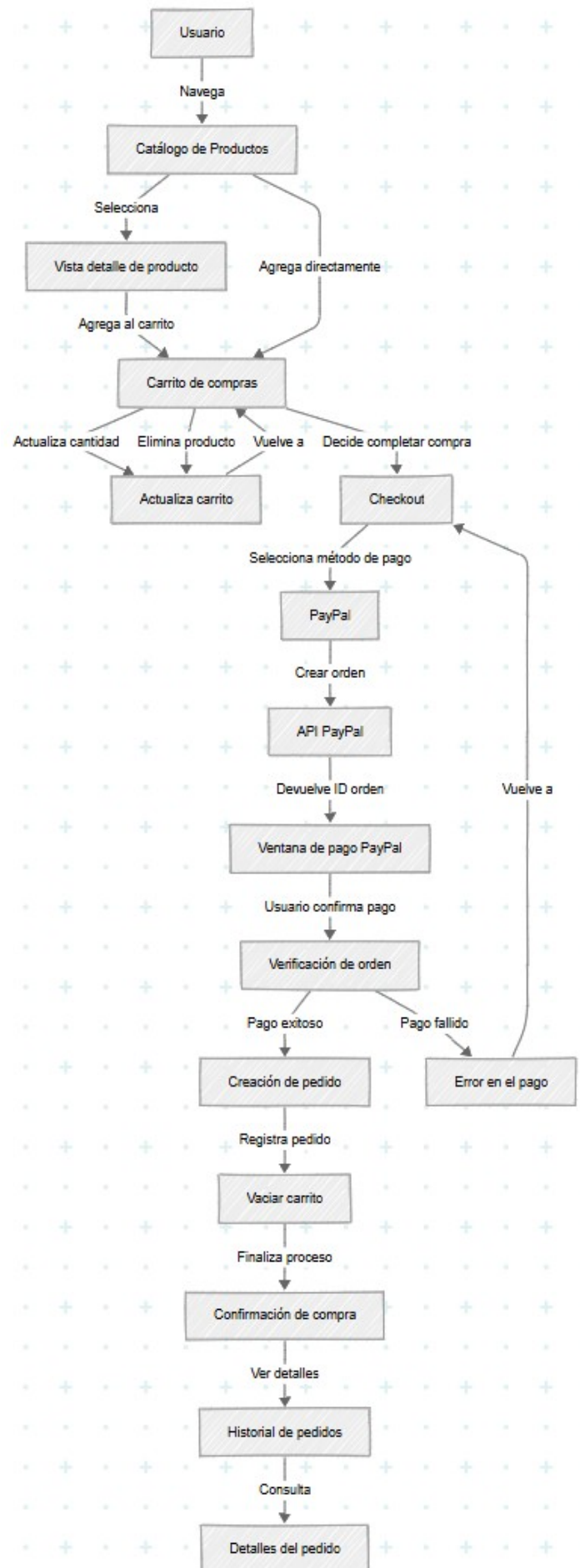
- El usuario inicia el proceso de pago
- Selecciona PayPal como método de pago

4.Proceso de pago con PayPal:

- El sistema crea una orden en la API de PayPal
- PayPal devuelve un ID de orden
- Se muestra la ventana de pago de PayPal al usuario
- El usuario confirma el pago en PayPal
- El sistema verifica el estado de la orden con PayPal

5.Finalización de la compra:

- Si el pago es exitoso, se crea un pedido en el sistema
- Se registra el pedido con el detalle de los productos
- Se vacía el carrito del usuario
- Se muestra una confirmación de compra



6. Seguimiento de pedidos:

- El usuario puede acceder al historial de pedidos
- Consultar los detalles de un pedido específico

1.4.Privilegios del Admin



Este diagrama muestra las principales funcionalidades disponibles para un administrador, divididas en cuatro áreas principales:

1.Gestión de usuarios:

- Ver la lista completa de usuarios del sistema
- Consultar los detalles de un usuario específico
- Actualizar la información personal de cualquier usuario
- Eliminar usuarios (con la restricción de no poder eliminar el único administrador)
- Modificar los roles de usuarios (promocionarlos a administradores o quitarles ese rol)

2.Gestión de productos:

- Ver el catálogo completo de productos
- Crear nuevos productos, incluyendo la carga de imágenes a Cloudinary
- Actualizar productos existentes, pudiendo modificar cualquier atributo
- Eliminar productos del sistema

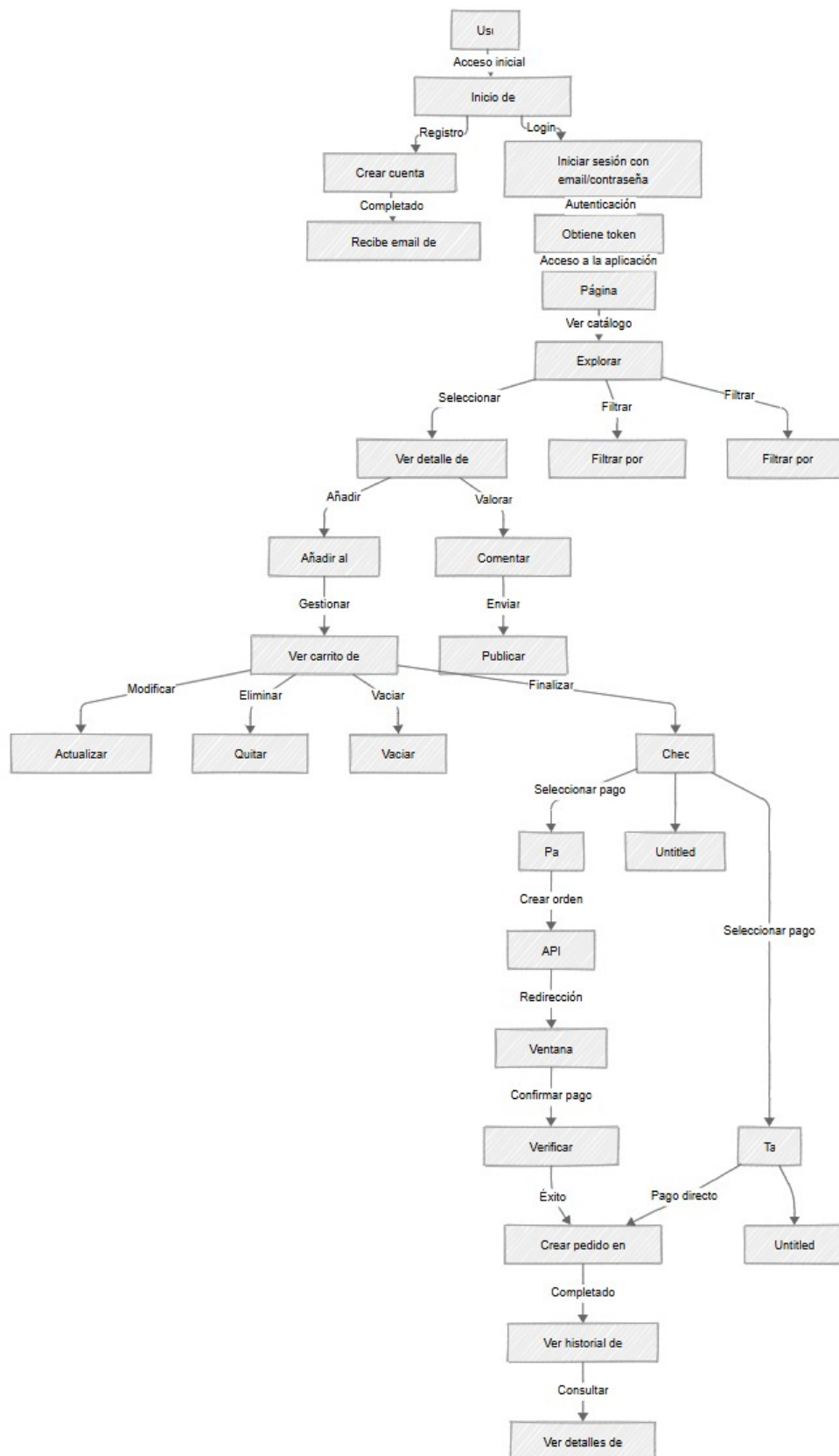
3.Gestión de pedidos:

- Ver la lista completa de todos los pedidos en el sistema
- Consultar los detalles específicos de cualquier pedido
- Acceder a los pedidos de cualquier usuario

4.Acceso a carritos:

- Ver el carrito de cualquier usuario
- Modificar el contenido de cualquier carrito
- Actualizar cantidades de productos en los carritos
- Vaciar carritos completos

1.5.Flujo de Usuario por la web



Este diagrama muestra la experiencia completa que un usuario normal puede tener en ApiVault, dividida en varias secciones:

1.Gestión de Cuenta:

- Registro como nuevo usuario
- Inicio de sesión con email y contraseña
- Recepción de email de bienvenida tras el registro
- Obtención del token JWT para autenticación

2.Navegación y Exploración:

- Ver el catálogo completo de productos
- Filtrar productos por categorías o marcas
- Ver detalles específicos de un producto

3.Gestión del Carrito:

- Añadir productos al carrito desde el detalle del producto
- Ver el contenido del carrito de compras
- Actualizar las cantidades de productos en el carrito
- Eliminar productos específicos del carrito
- Vaciar completamente el carrito

4.Proceso de Compra:

- Iniciar el proceso de checkout
- Seleccionar método de pago (tarjeta o PayPal)

5.Para PayPal:

- Crear orden en la API de PayPal
- Redirección a la ventana de pago de PayPal
- Confirmación del pago
- Verificación de la transacción
- Crear el pedido en el sistema una vez confirmado el pago

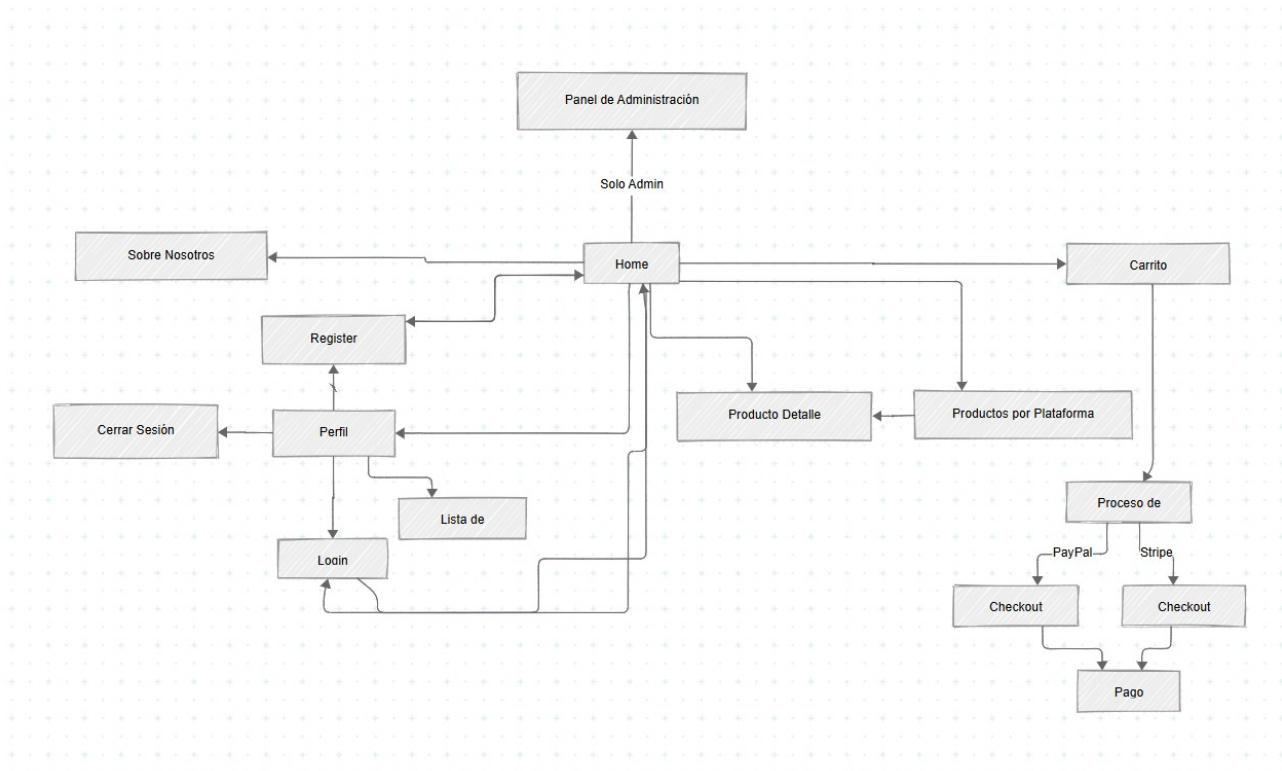
6.Gestión de Pedidos:

- Ver el historial completo de pedidos realizados
- Consultar los detalles específicos de un pedido

7.Interacción:

- Comentar y valorar productos

Mapa de Navegación



Tipos de Páginas

Página Principal: Punto de entrada a la aplicación, muestra productos destacados

Sobre Nosotros: Información sobre la empresa/tienda

Detalle de Producto: Información detallada de un producto específico

Productos por Plataforma: Listado de productos filtrados por plataforma (PC, PlayStation, Xbox, etc.)

Iniciar Sesión: Página para que los usuarios accedan a sus cuentas

Registro: Página para que los nuevos usuarios creen una cuenta

Páginas que Requieren Autenticación

Carrito: Gestiona los productos añadidos para comprar

Lista de Favoritos: Muestra los productos guardados como favoritos por el usuario

Páginas Solo para Administradores

Panel de Administración: Panel que permite gestionar productos, usuarios y pedidos

Flujos de Navegación Principales

Navegación Básica:

Página Principal → Productos por Plataforma → Detalle de Producto

Proyecto Desarrollo de Aplicaciones Web

Proceso de Compra:

Detalle de Producto → Carrito → Proceso de Pago → Pago Exitoso

Acceso a la Cuenta:

Página Principal → Iniciar Sesión → Página Principal (con sesión iniciada)

Gestión de Favoritos:

Página Principal → Perfil → Lista de Favoritos

Controles de Acceso

El Carrito y la Lista de Favoritos solo son accesibles para usuarios autenticados

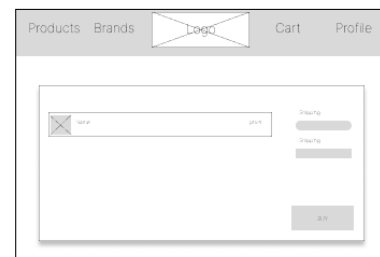
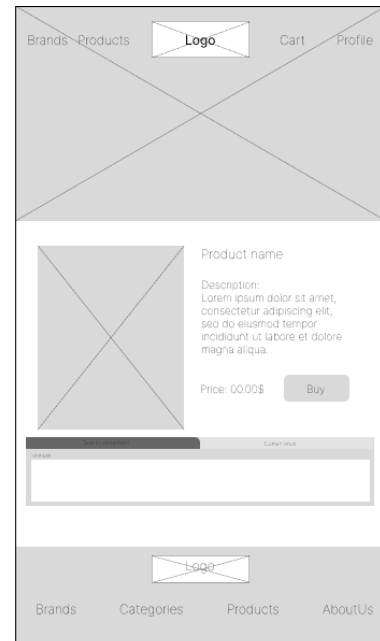
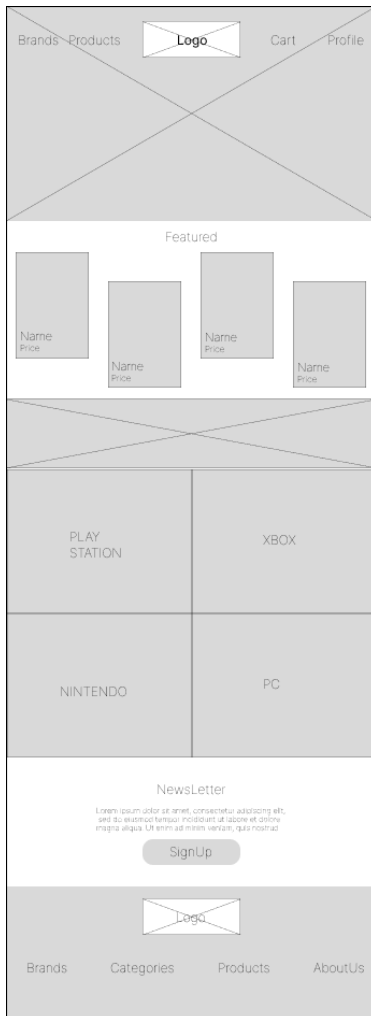
El Panel de Administración solo es accesible para usuarios con privilegios de administrador

Si un usuario no autenticado intenta acceder al Carrito, será redirigido a la página de Iniciar Sesión

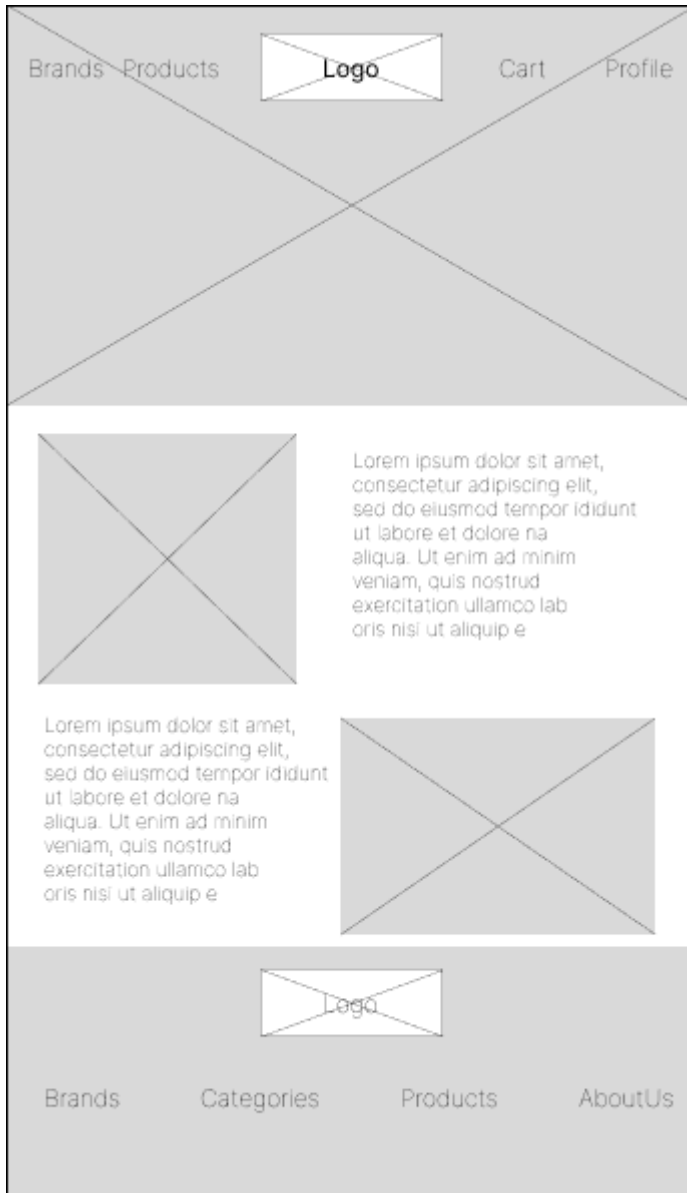
Diseño

Wireframes:

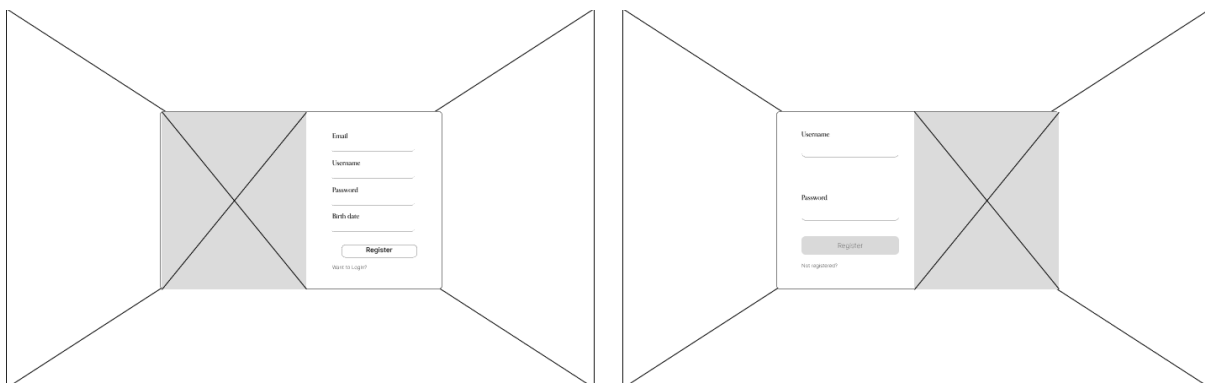
Wireframes del Home, Producto Plataforma y Producto Detalle, respectivamente



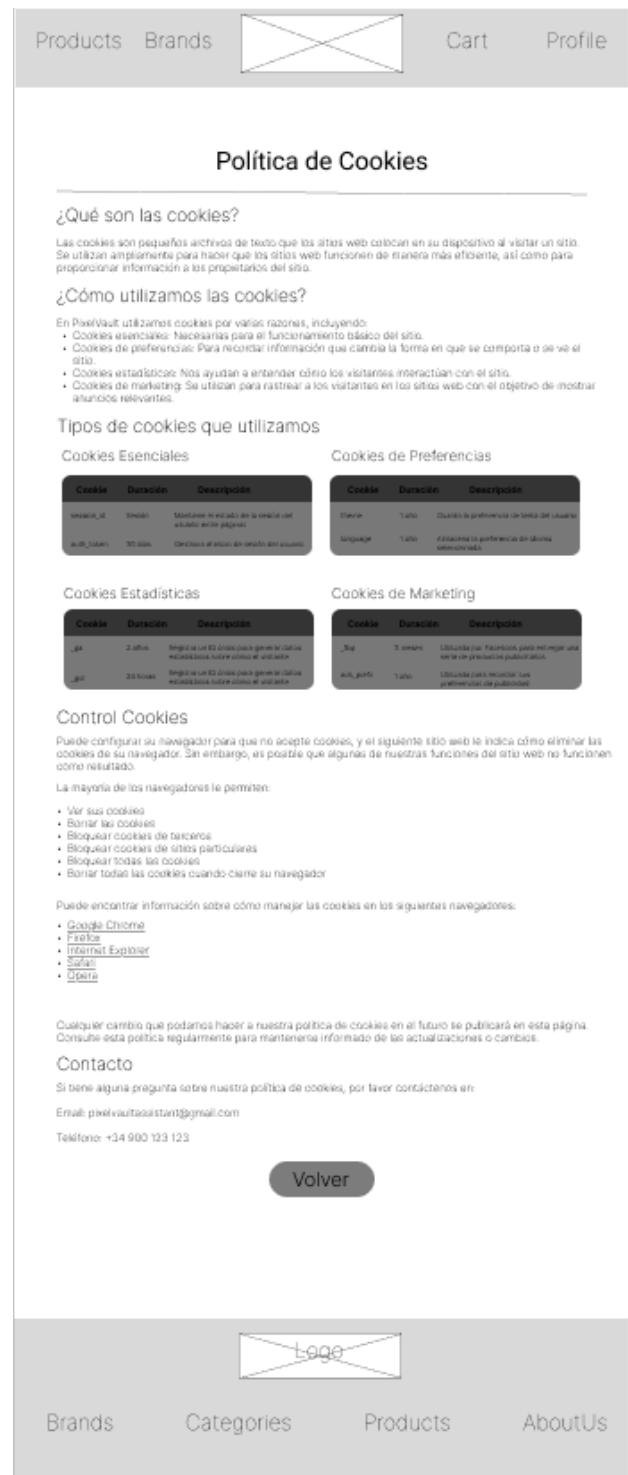
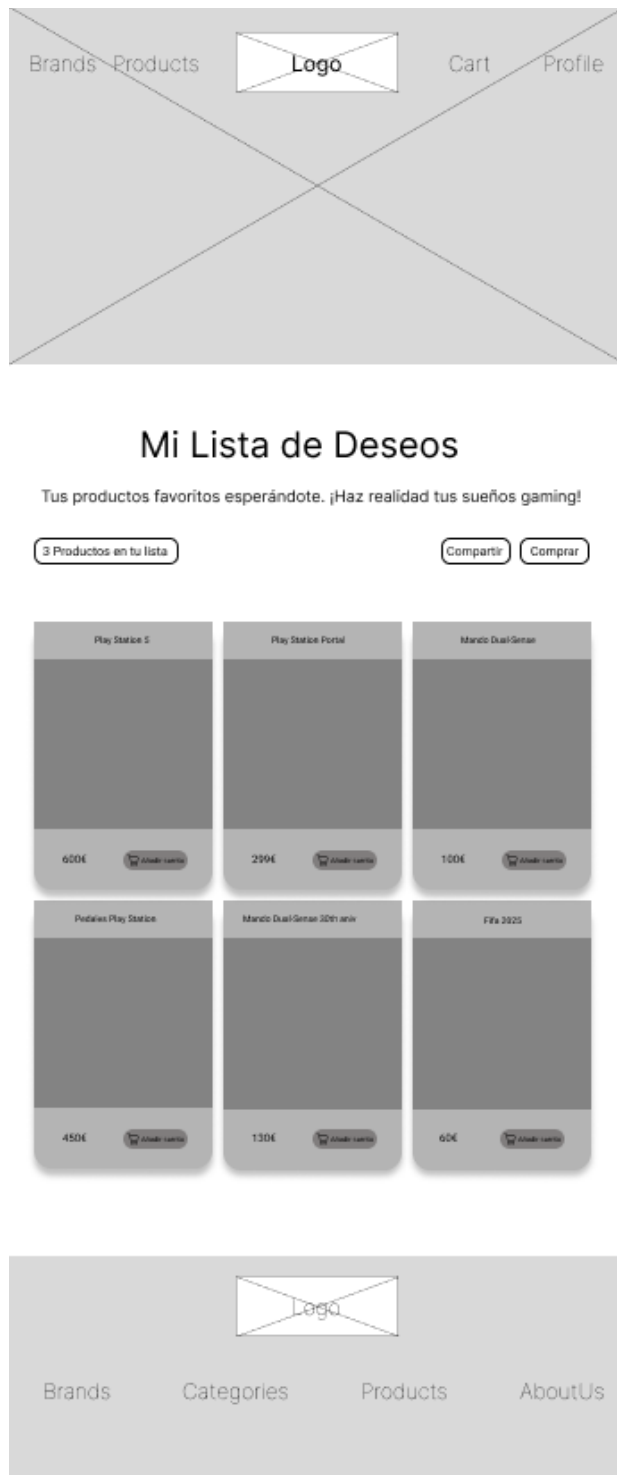
Wireframe de Sobre Nosotros



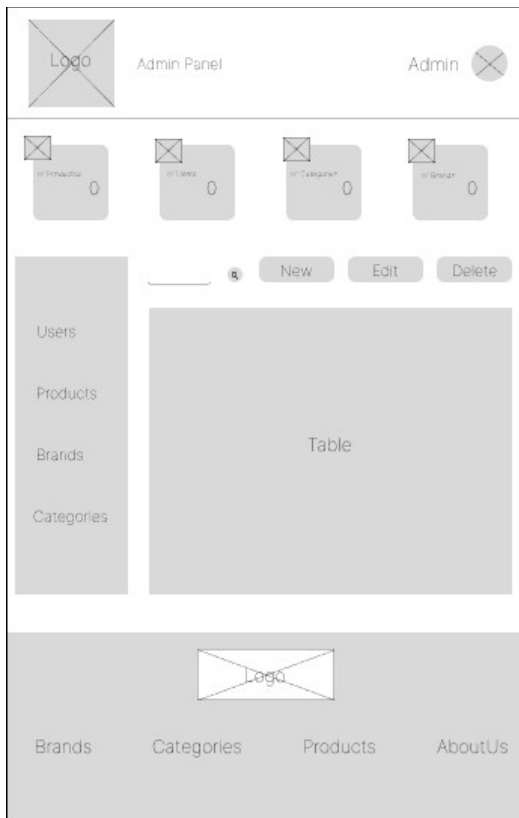
Wireframes de Inicio de sesión y Registro



Wireframe de las Cookies y Favoritos



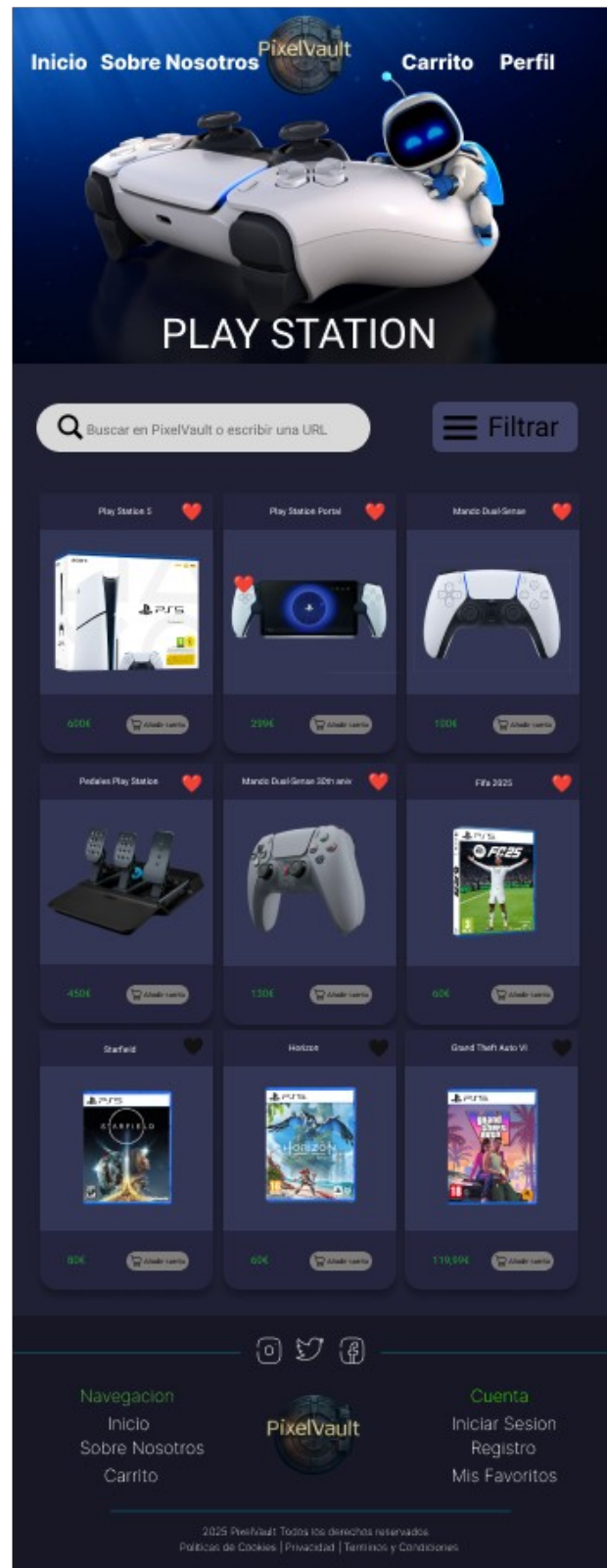
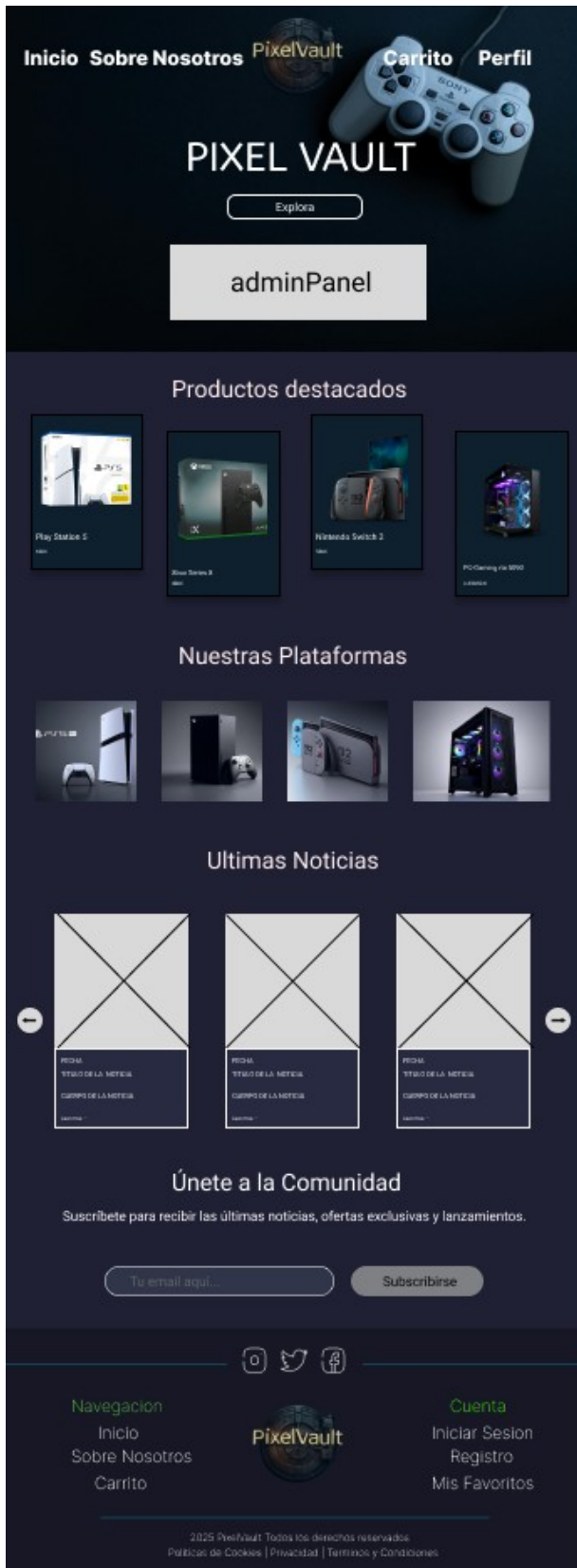
Wireframes del Panel de Administrador



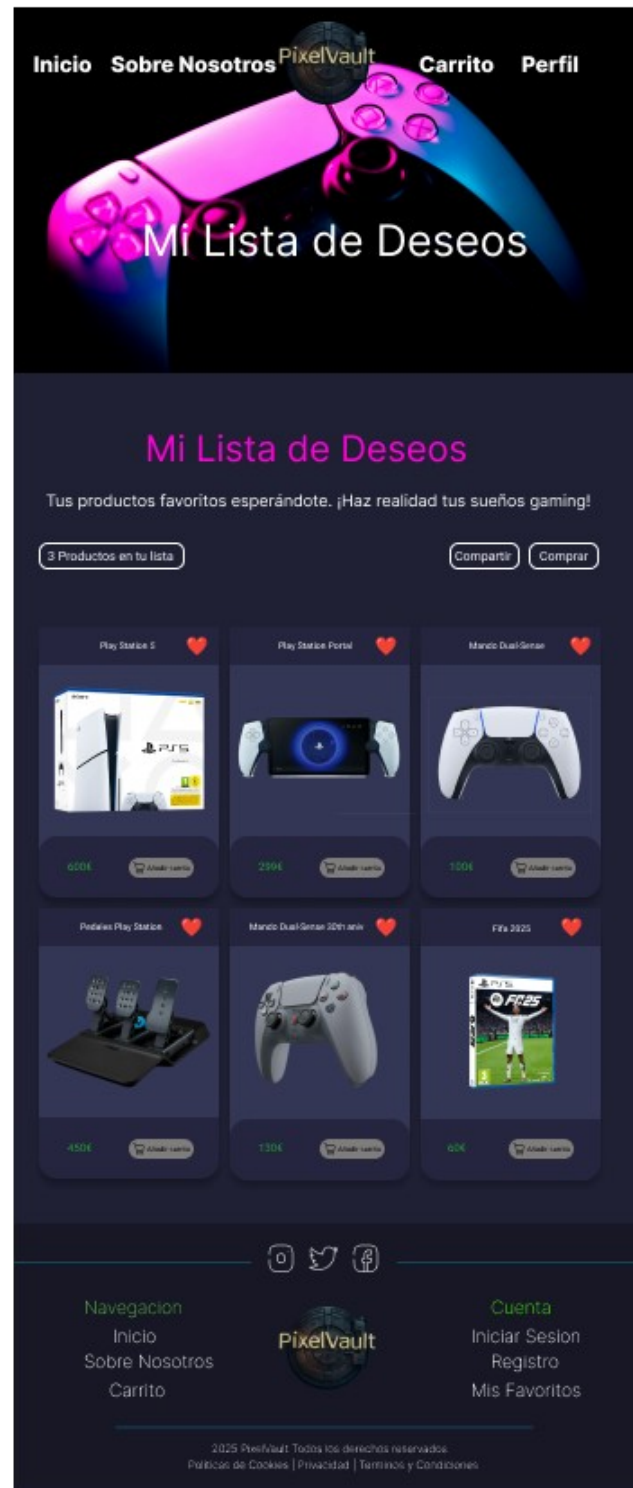
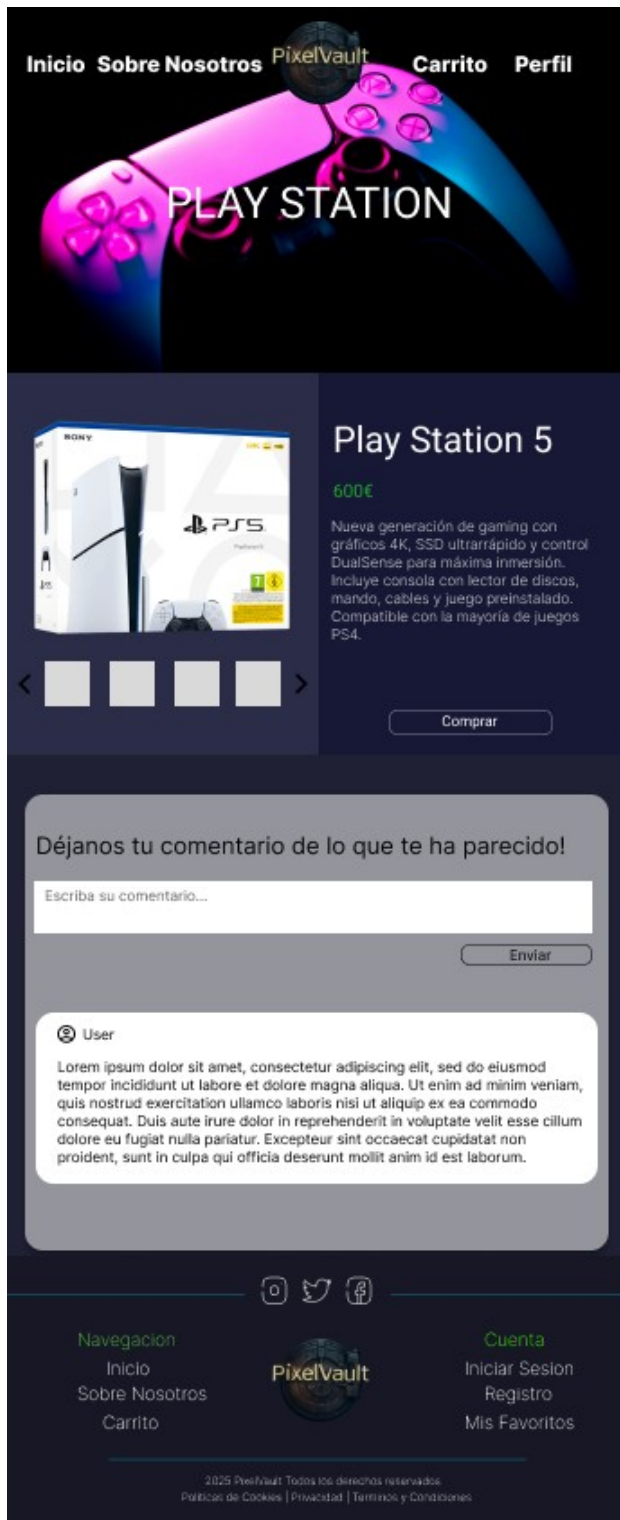
Los wireframe en diseño web es un esquema visual básico que representa la estructura de una página o aplicación web. Sirve para planificar la distribución de los elementos, como menús, botones, imágenes y textos, sin entrar en detalles de diseño gráfico. Es una herramienta clave en la etapa inicial del desarrollo, ya que facilita la comunicación entre diseñadores, desarrolladores y clientes. Normalmente se presenta en escala de grises y con formas simples. Su objetivo es enfocarse en la funcionalidad y usabilidad antes de diseñar la apariencia final.

Mockups:

Mockups del Home y de Productos Plataforma



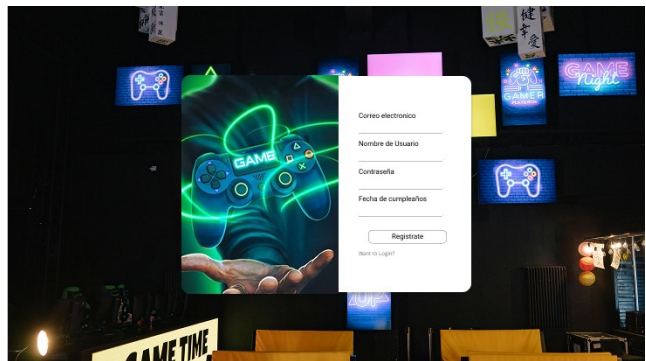
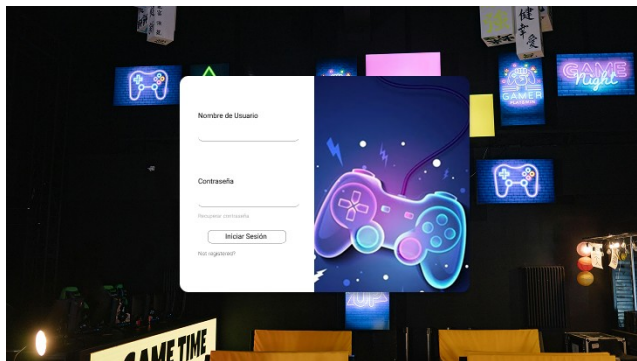
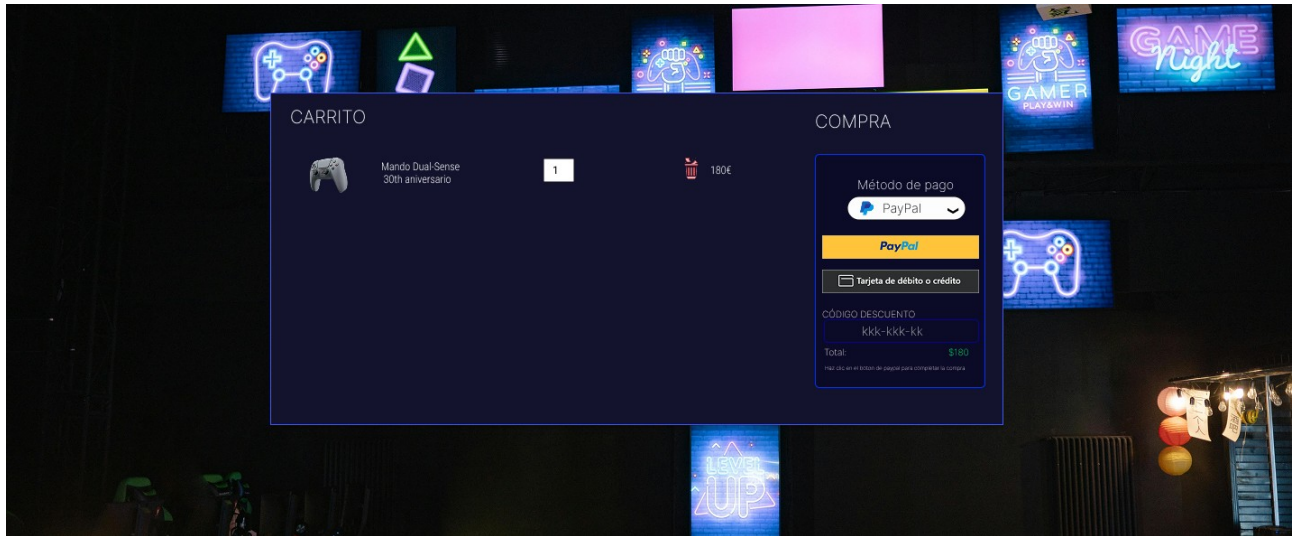
Mockups de Producto Detalle y Lista de Deseos



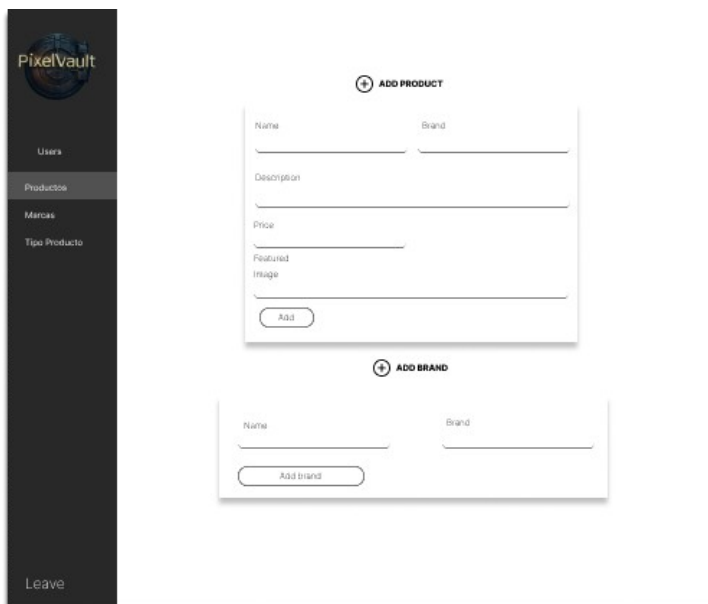
Mockups de Sobre Nosotros y Cookies



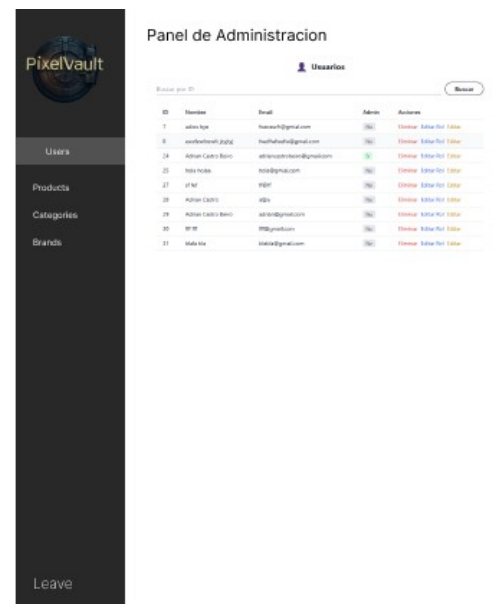
Mockups de Carrtiro, Inicio Sesión y Registro



Mockups del Panel de Administración



The mockup shows a sidebar on the left with the 'PixelVault' logo and navigation links: 'Users', 'Productos', 'Marcas', and 'Tipo Producto'. The main content area features a form titled '+ ADD PRODUCT'. The form has input fields for 'Name' and 'Brand', a 'Description' field, a 'Price' field, and a 'Featured Image' field. An 'Add' button is at the bottom of the form. Below the product form is a section titled '+ ADD BRAND' with input fields for 'Name' and 'Brand', and an 'Add brand' button. A 'Leave' button is located at the bottom left of the sidebar.



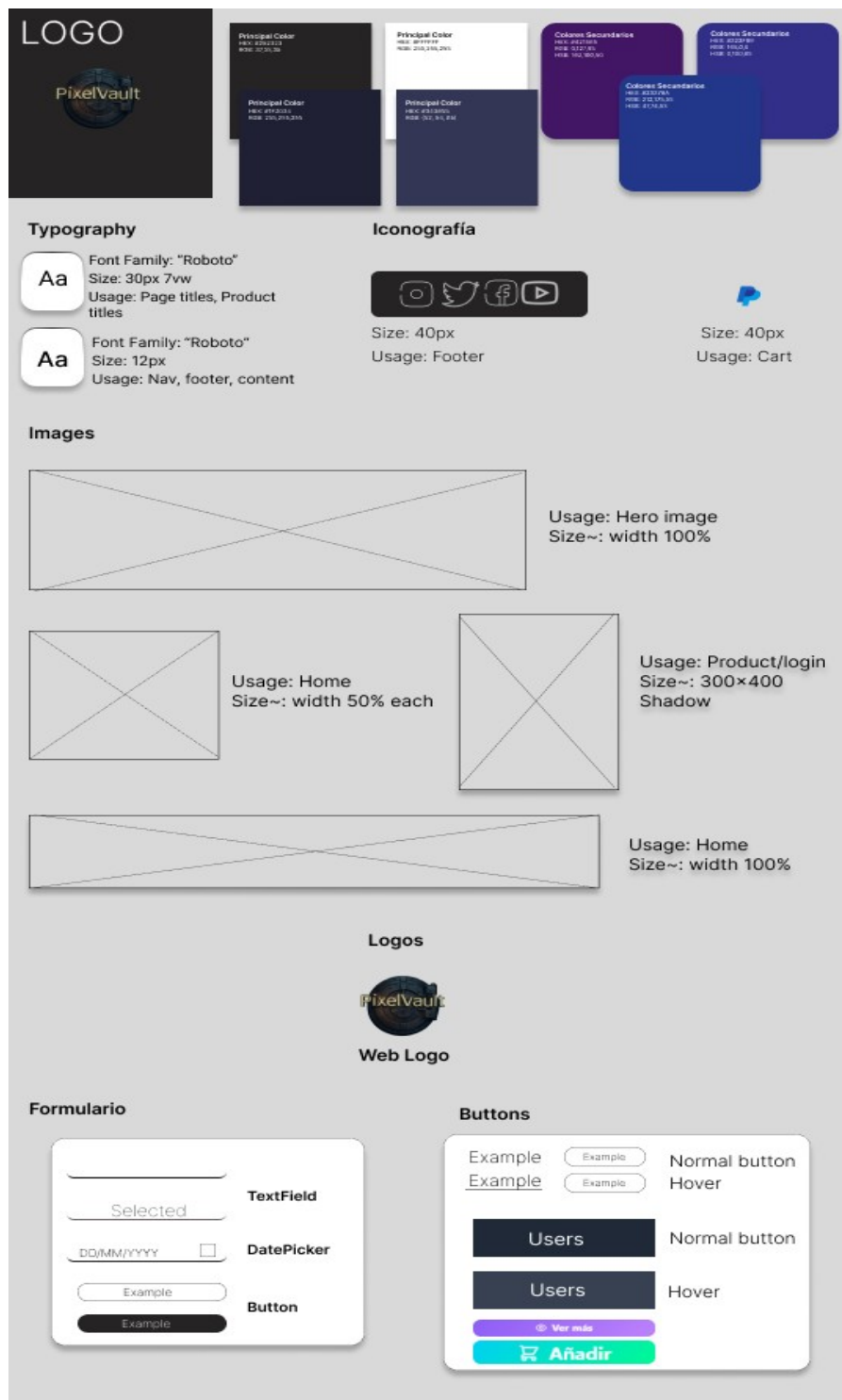
The mockup shows the 'Usuarios' (Users) section of the admin panel. It includes a sidebar with the 'PixelVault' logo and navigation links: 'Users', 'Productos', 'Categorías', and 'Brands'. The main content area displays a table of users with columns for 'ID', 'Nombre', 'Email', 'Admin', and 'Acciones'. The table contains 11 rows of user data. A 'Buscar por ID' search bar is located at the top left of the table, and a 'Nuevo' button is at the top right.

| ID | Nombre | Email | Admin | Acciones |
|----|--------------------|----------------------------|-------|---------------------|
| 7 | Admin User | admin@pixelvault.com | Yes | Eliminar Editar Ver |
| 8 | Admin User | admin@pixelvault.com | Yes | Eliminar Editar Ver |
| 14 | Admin Carlos Bello | carlosbello@pixelvault.com | Yes | Eliminar Editar Ver |
| 25 | Admin User | admin@pixelvault.com | Yes | Eliminar Editar Ver |
| 27 | Admin User | admin@pixelvault.com | Yes | Eliminar Editar Ver |
| 28 | Admin User | admin@pixelvault.com | Yes | Eliminar Editar Ver |
| 29 | Admin User | admin@pixelvault.com | Yes | Eliminar Editar Ver |
| 30 | Admin User | admin@pixelvault.com | Yes | Eliminar Editar Ver |
| 31 | Admin User | admin@pixelvault.com | Yes | Eliminar Editar Ver |

Los mockup en diseño web es una representación visual detallada de cómo se verá una página o aplicación una vez terminada. A diferencia del wireframe, incluye colores, tipografías, imágenes y estilos gráficos. Su propósito es mostrar el diseño final con precisión, ayudando a visualizar la experiencia del usuario. Se utiliza para presentar propuestas a clientes o validar decisiones estéticas. Aunque no es interactivo, refleja fielmente el aspecto del producto final.

Los diseños de los mockups de la página utilizan una estética gaming moderna con colores oscuros y acentos neón (azul, rosa, verde) que crean un ambiente nocturno atractivo para gamers. Los mandos funcionan como elementos decorativos centrales, mientras que la navegación mantiene claridad con menús intuitivos y tarjetas de productos consistentes. Los formularios usan fondos translúcidos para buena legibilidad y el carrito presenta una estructura limpia que facilita las compras. Esta combinación logra una experiencia visual cohesiva que transmite profesionalismo gaming y genera confianza en los usuarios.

Guía de estilos:



La página PixelVault utiliza una estética futurista con una paleta de colores predominantemente oscura (tonos azul-negro #161630, #10102a) combinada con destellos de azul cian (#00ccff) y verde neón (#00ff88) que crean un aspecto cyberpunk. Los elementos tienen bordes sutilmente iluminados y efectos de desenfoque (backdrop-blur) que dan profundidad, mientras que los gradientes y las sombras suaves refuerzan la temática tecnológica de esta tienda de videojuegos.

Desarrollo del proyecto

Arquitectura y Diseño General

El proyecto PixelVault se diseñó como una plataforma de comercio electrónico especializada en productos para gamers, priorizando la experiencia de usuario, seguridad, rendimiento y escalabilidad. Desde sus cimientos, se apostó por una arquitectura modular y orientada a servicios, utilizando tecnologías modernas como Vue.js en el frontend y .NET 8 con Entity Framework Core en el backend.

Navegación y UI

Se implementó un navbar global adaptativo para ofrecer una navegación consistente. Este cambia su estilo según la página: transparente en la portada (para destacar la imagen de fondo) y oscuro. La arquitectura de componentes en Vue.js permitió una interfaz mantenible y altamente reutilizable.

Autenticación y Seguridad

La seguridad fue prioridad desde el inicio:

- Se utilizó JWT con refresh tokens para una autenticación stateless y escalable
- Las contraseñas se almacenan de forma segura usando PBKDF2 con sal única
- Se aplicaron múltiples capas de seguridad: CORS, validación de inputs, protección contra SQL Injection y cifrado de datos sensibles
- El sistema de roles comenzó de forma simple con campo EsAdmin con miras a evolucionar a una estructura más granular

Modelo de Dominio

El sistema está centrado en la entidad Usuario, que se relaciona con:

- Historial de acciones (para trazabilidad y seguridad)
- Comentarios y valoraciones (para fomentar comunidad)
- Carritos de compra persistentes
- Pedidos y su seguimiento

Esto permite una experiencia personalizada y coherente.

Carrito de Compra y Sesiones

Uno de los mayores retos fue mantener la coherencia del carrito entre sesiones y dispositivos. La solución fue persistir el carrito en base de datos, permitiendo:

- Recuperarlo al iniciar sesión desde cualquier dispositivo
- Guardar productos aunque se cierre sesión

- Implementar recomendaciones personalizadas

Backend y Servicios

- Se utilizó Entity Framework Core por su integración con LINQ y el patrón Repository
- Cada entidad cuenta con su propio servicio especializado, lo que permite separar responsabilidades, facilitar el testing y mantener los controladores limpios
- Se integraron APIs externas como Cloudinary, PayPal y Stripe a través de wrappers personalizados, facilitando cambios futuros y pruebas mediante mocks

Rendimiento y Optimización

Se implementaron técnicas como:

- Lazy loading de imágenes
- Caché de datos
- Consultas optimizadas en EF Core

Estas medidas mejoraron notablemente el rendimiento de la plataforma.

Concurrencia y Consistencia

Para asegurar la integridad de los datos en un entorno concurrido, se aplicaron:

- Transacciones en operaciones críticas (como finalizar compras)
- Verificación de stock en tiempo real
- Bloqueos optimistas para evitar condiciones de carrera

Extensibilidad y Personalización

El sistema está preparado para crecer:

- Estructura relacional lista para un motor de recomendaciones
- Seguimiento de acciones para análisis de comportamiento
- Productos fácilmente categorizables y extensibles

Testing

Se aplicaron distintas estrategias de pruebas:

- Unitarias para reglas de negocio
- Integración para APIs externas
- Pruebas de carga para rendimiento
- Validaciones de seguridad

La arquitectura modular facilitó estas pruebas al permitir aislar componentes.

Conclusión del desarrollo

PixelVault es un proyecto sólido, escalable y seguro, con una estructura pensada para facilitar el mantenimiento, la personalización y el crecimiento a futuro. La combinación de buenas prácticas, tecnologías modernas y foco en el usuario garantiza una plataforma de alto nivel para el comercio electrónico enfocado al sector gamer.

Manual Administrador

Requisitos del Sistema

1. Ejecutar el script de la base de datos
2. Configuración de requisitos técnicos necesarios:
 - .NET 8 SDK
 - Node.js (versión 16 o superior)
 - SQL Server
 - Visual Studio y Visual Studio Code

Manual de instalación y ejecución

Requisitos necesarios:

- .NET 8
- Node.js
- SQL Server

Proceso de instalación:

1. Instalación de dependencias:

bash

`npm install`

2. Configuración de la base de datos:

- Configurar las credenciales de SQL Server en el archivo AppSettings.json

3. Ejecución de la aplicación:

- 1°. Para ejecutar el backend

`dotnet run`

- 2°. Para arrancar el frontend

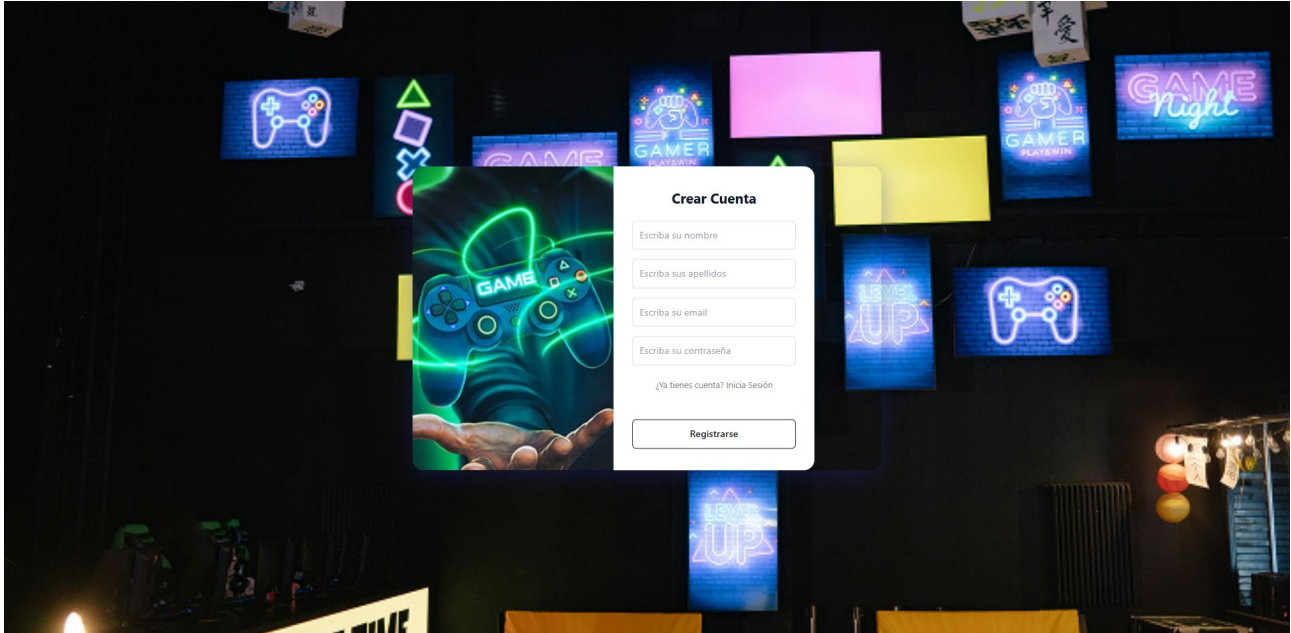
`npm run dev`

4. Acceso como administrador:

Utilizar usuario y contraseña proporcionados en el proceso de registro

Manual Usuario

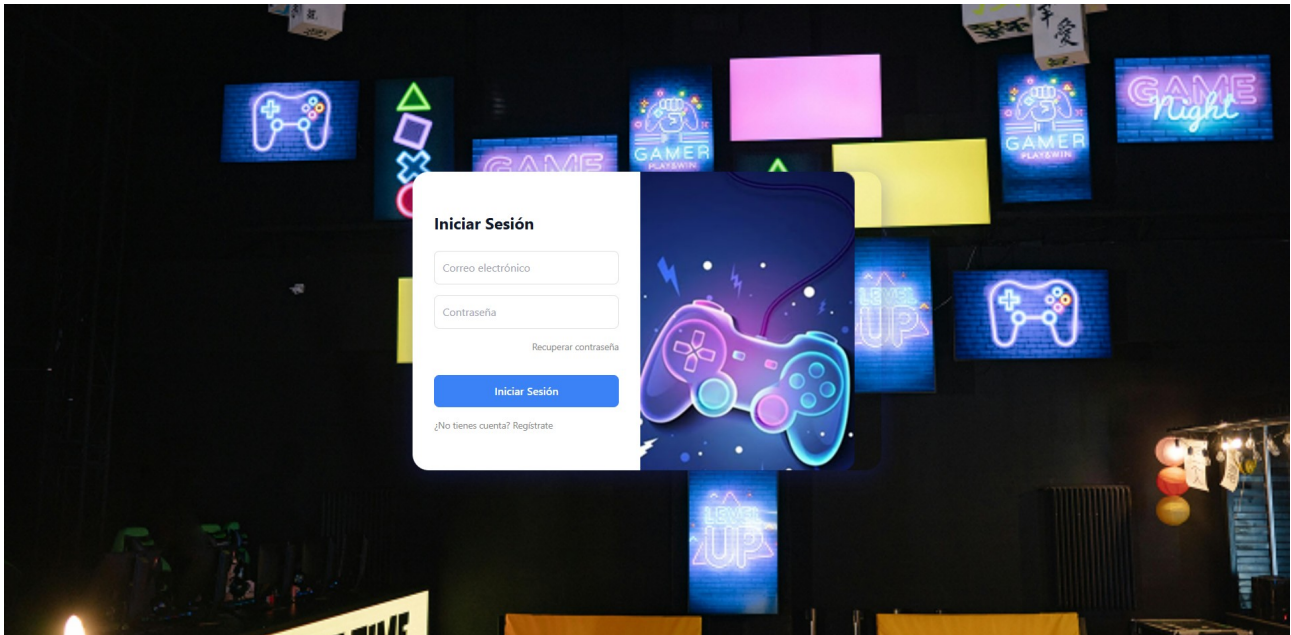
1-Registro



La vista de registro presenta un split layout donde los usuarios pueden crear su cuenta para acceder a todas las funcionalidades de la plataforma como lista de deseos, carrito y realizar compras. El diseño combina un fondo gaming inmersivo con elementos neón coloridos en tonos azules, verdes, rosas y amarillos que crean una atmósfera moderna y energética.

El formulario "Crear Cuenta" se posiciona centralmente sobre un fondo blanco limpio que contrasta perfectamente con el entorno gaming colorido. Esta división visual funciona muy bien: mientras el lado izquierdo muestra elementos decorativos neón distribuidos dinámicamente, el formulario central actúa como punto focal con campos organizados de forma clara y un botón "Registrarse" coherente con el diseño general. El resultado es una interfaz atractiva que balancea la estética gaming con la funcionalidad práctica del registro.

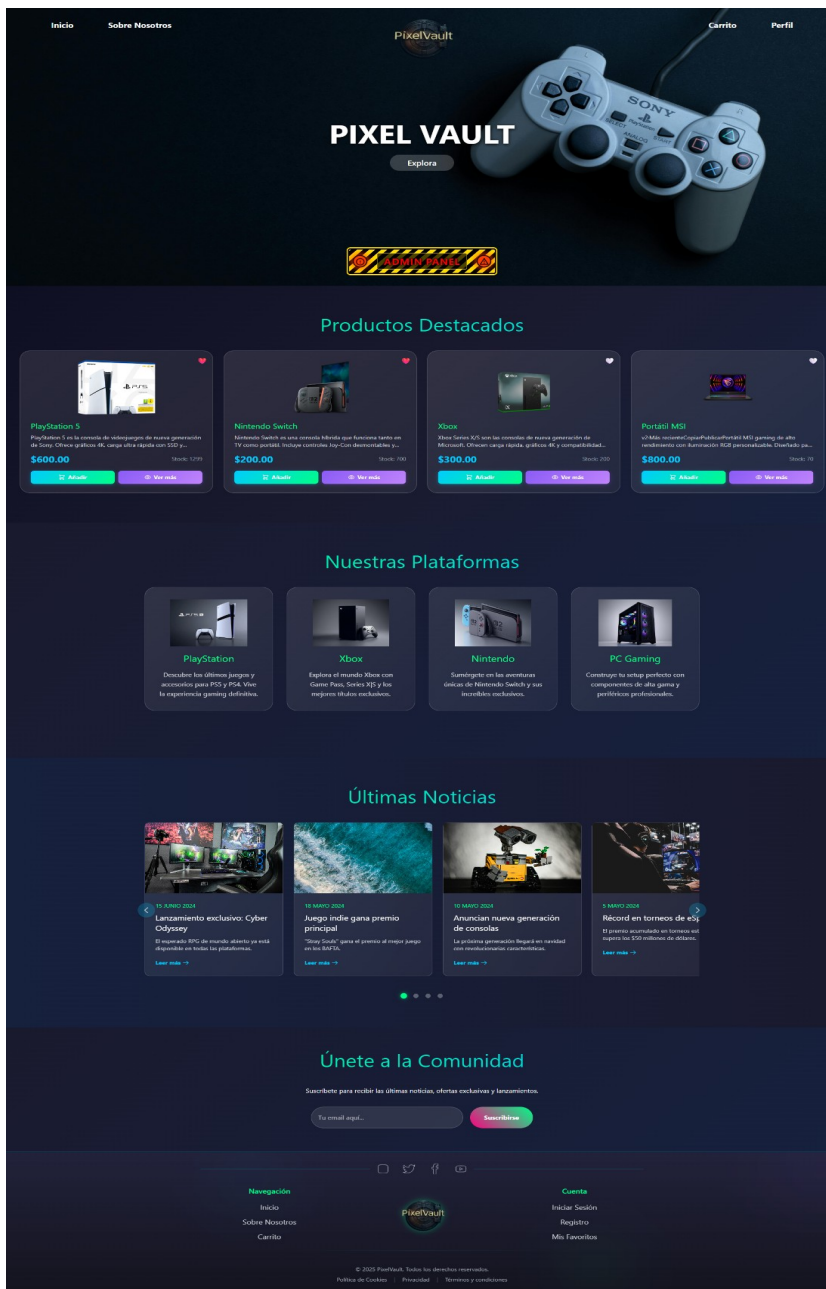
2-Inicio de sesion



La vista de inicio de sesión presenta un split layout donde los usuarios acceden después de registrarse o cuando necesitan autenticarse en la plataforma. El formulario del lado izquierdo incluye campos para correo y contraseña con un diseño limpio sobre fondo blanco, mientras que el lado derecho muestra un mando gaming con efectos neón coloridos que mantienen la estética visual de la marca.

Una vez que el usuario inicia sesión correctamente, es redirigido automáticamente al home de la web con un token de autenticación que dura 60 minutos. Este token se renueva automáticamente mientras el usuario permanezca activo en la página, garantizando una experiencia fluida sin interrupciones por caducidad de sesión.

3-Home

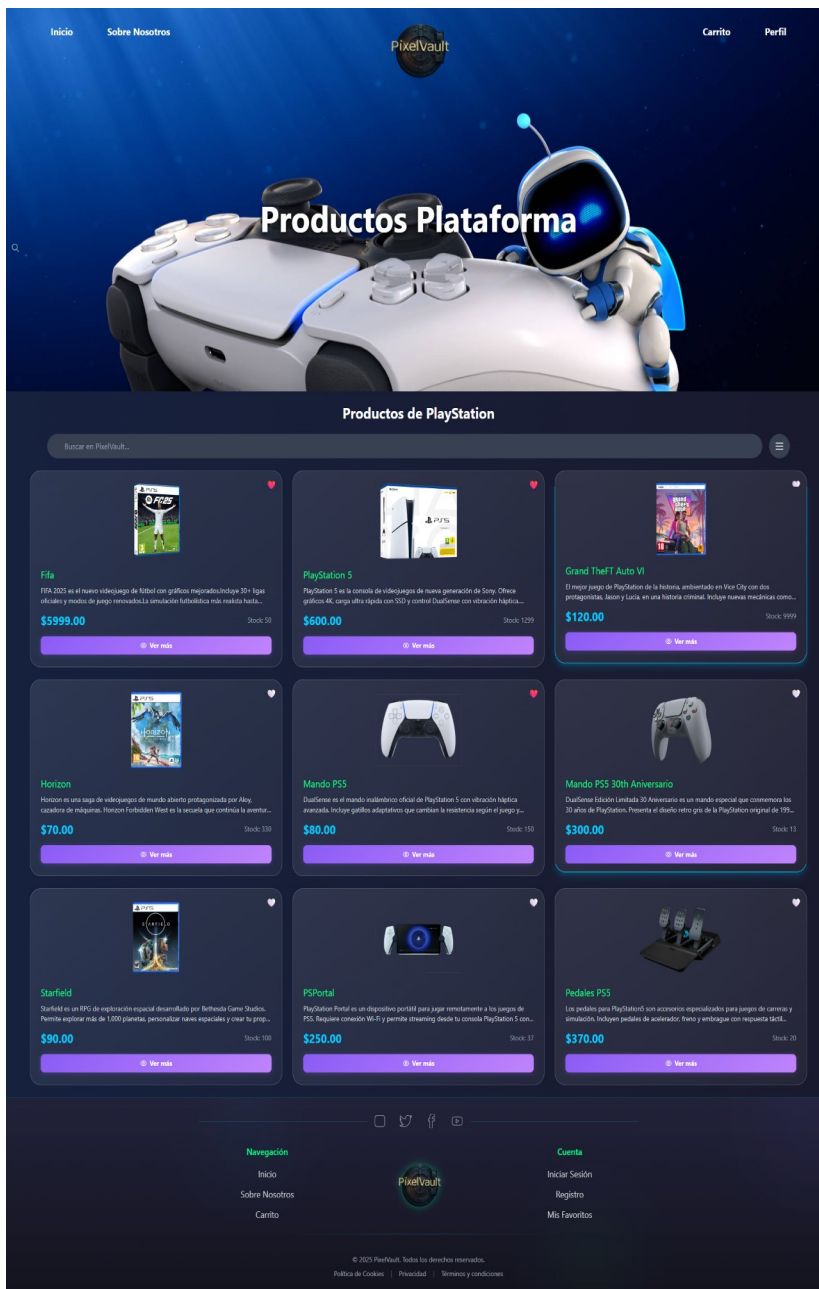


El home de PixelVault presenta una página completa donde los usuarios autenticados pueden acceder a todas las funcionalidades de la plataforma. La interfaz utiliza un hero layout en el header con navegación hacia "Sobre Nosotros", carrito de compras y un menú desplegable de perfil que incluye acceso a lista de deseos y opción de cerrar sesión. Cuando un usuario se autentica como administrador, aparece un botón adicional en el hero layout que permite acceso directo al panel de administración.

Proyecto Desarrollo de Aplicaciones Web

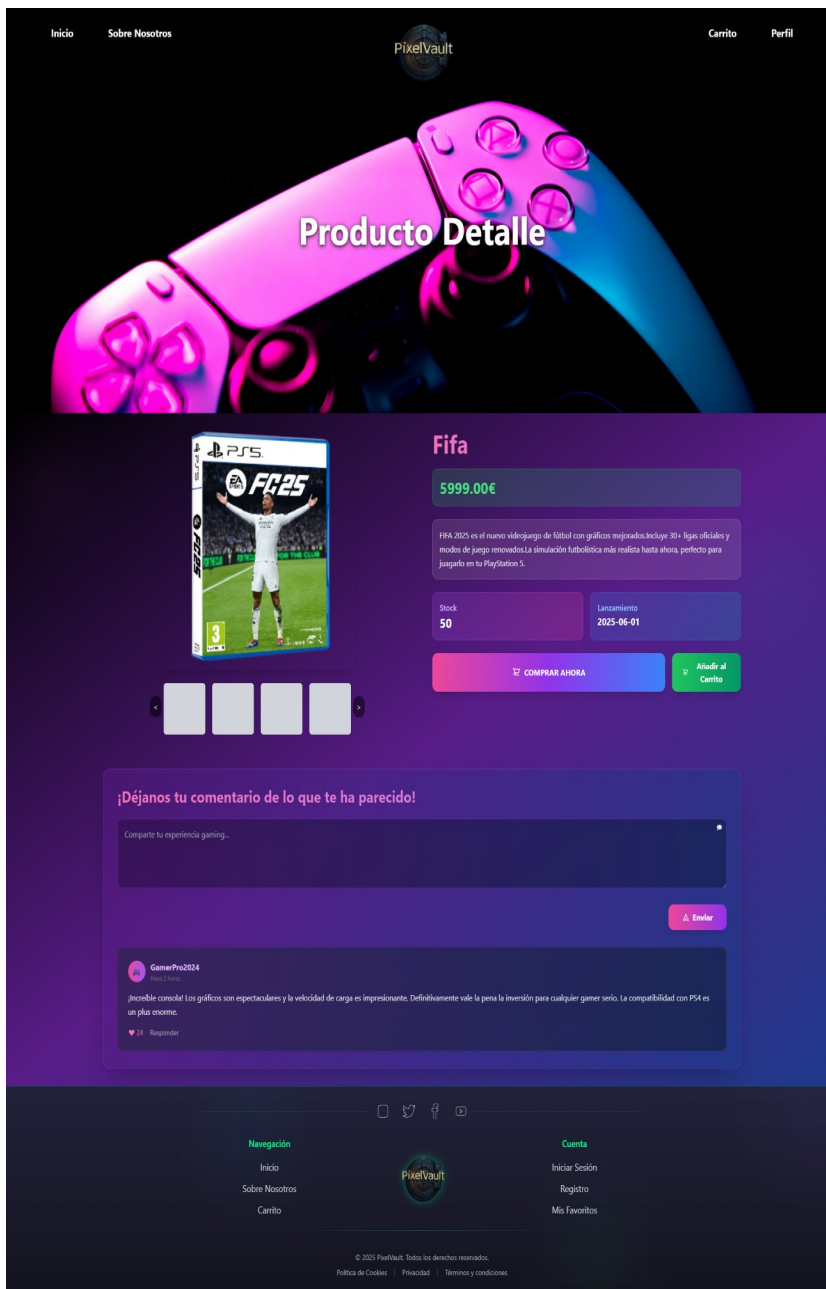
La página se estructura en varias secciones clave: "Productos Destacados" muestra una selección rotativa de ofertas semanales con opción de ver detalles de cada producto, "Nuestras Plataformas" permite filtrar productos por consola específica (PlayStation, Xbox, Nintendo, PC Gaming), "Últimas Noticias" mantiene a los usuarios informados sobre novedades del mundo gaming, y "Únete a la Comunidad" ofrece suscripción para recibir notificaciones sobre lanzamientos y ofertas. El footer complementa la navegación principal, permitiendo acceso directo a diferentes secciones de la web.

4-Producto según la plataforma



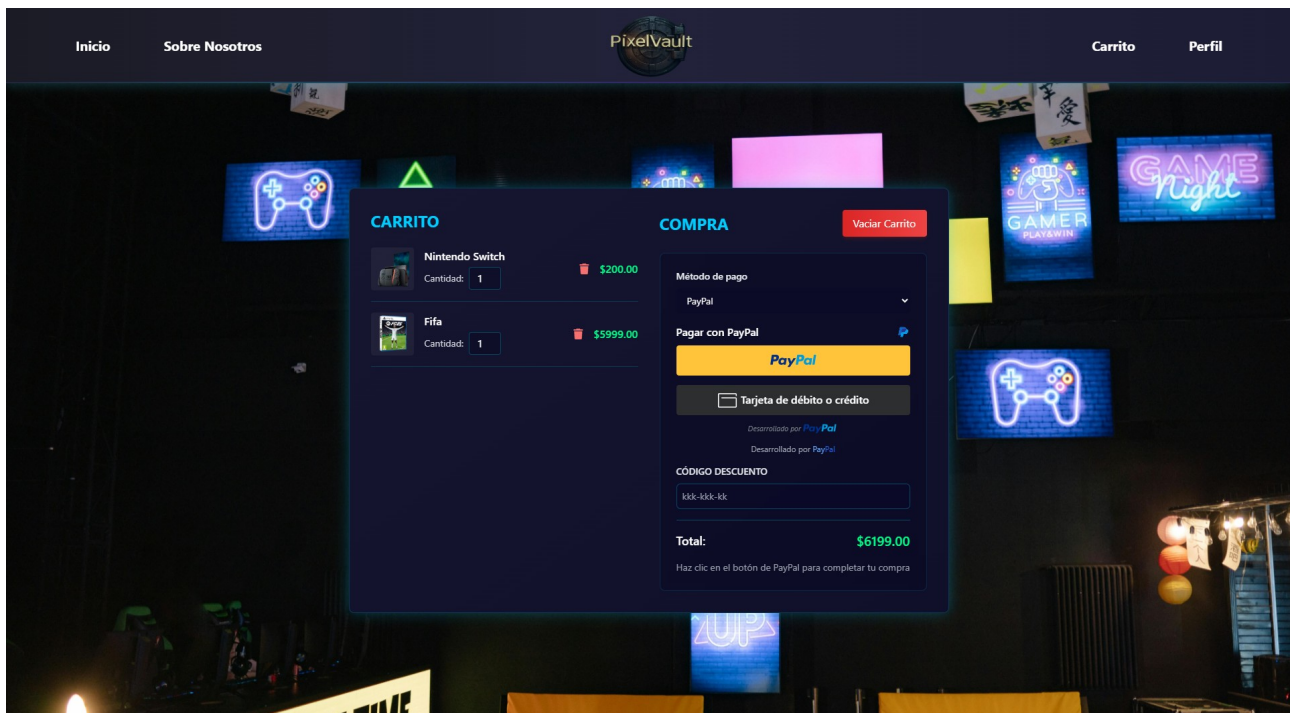
Después de que el usuario selecciona una plataforma en la página principal, es dirigido a la vista de productos de esa plataforma específica, donde puede ver todos los artículos disponibles como videojuegos, consolas y accesorios. Para facilitar la búsqueda, el usuario cuenta con una barra de búsqueda donde puede escribir exactamente lo que busca, y también tiene acceso a un menú hamburguesa de filtros que le permite filtrar los productos por categorías como videojuegos, periféricos o consolas. Cuando el usuario hace clic en el botón "Ver más" de cualquier producto, será redirigido a la página de detalles del producto donde podrá ver información completa y realizar la compra.

5-Producto detalle



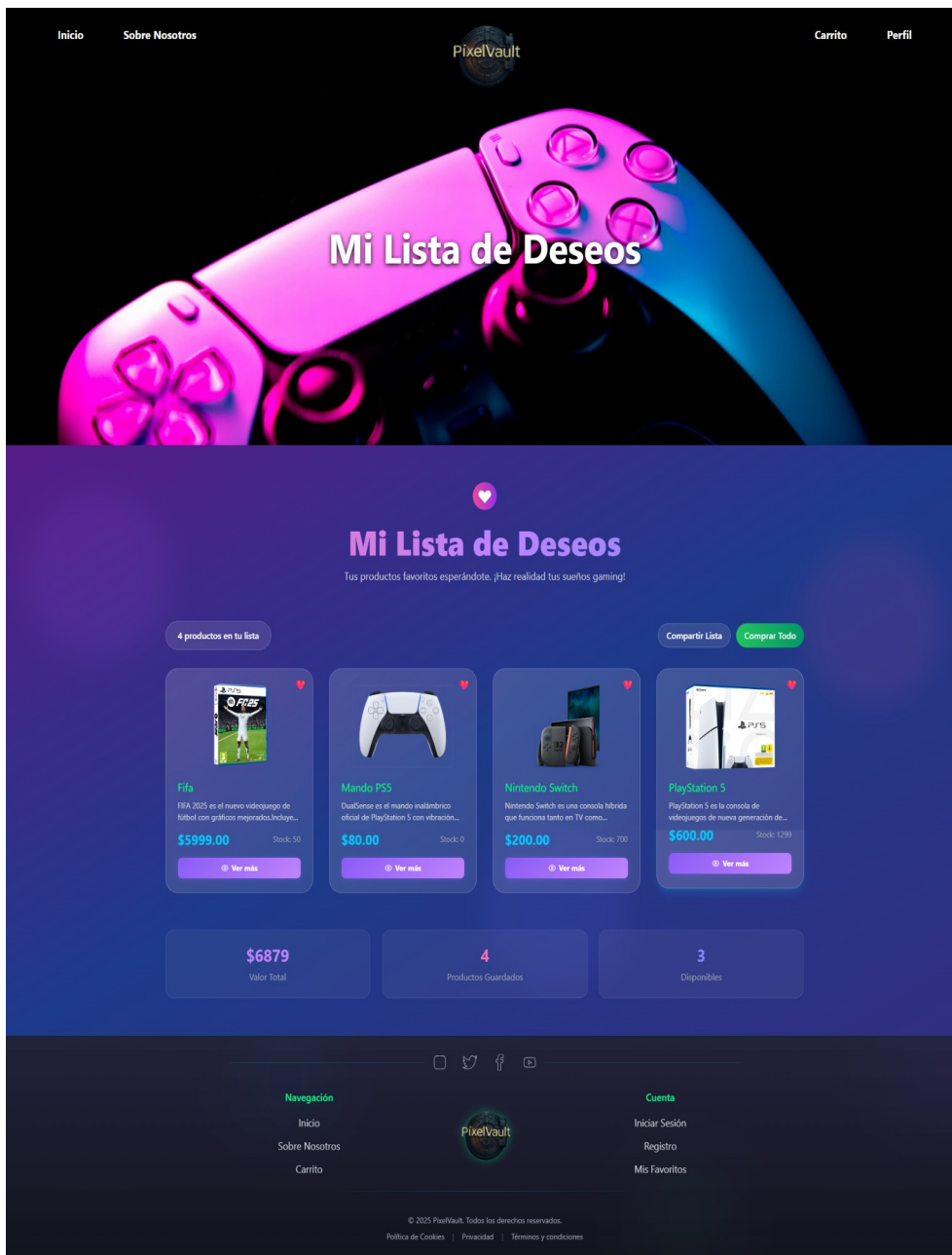
Cuando el usuario hace clic en "Ver más" desde la vista anterior, llega a la página de detalles del producto donde puede ver toda la información completa. Aquí encuentra un carrusel de imágenes que le permite ver el producto desde diferentes ángulos con fotos más grandes, además de leer una descripción detallada, conocer cuántas unidades quedan en stock y la fecha de lanzamiento del producto. El usuario puede elegir entre comprar el producto inmediatamente o añadirlo al carrito de compras. En la parte inferior de la página hay una sección de comentarios donde puede leer las opiniones y experiencias de otros usuarios que ya compraron el producto, y si él mismo lo ha comprado, también puede escribir su propia reseña e interactuar con los comentarios de otros compradores.

6-Carrito



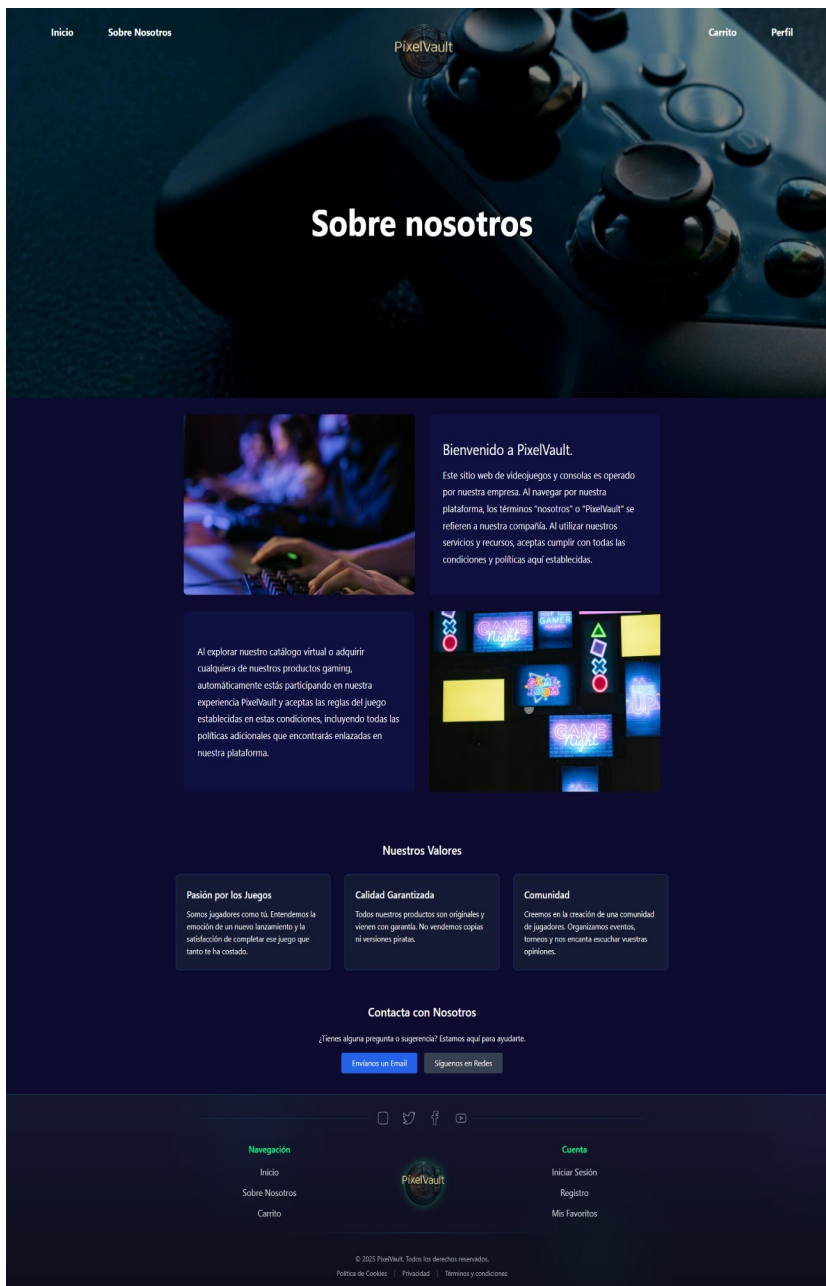
En la vista del carrito, el usuario puede ver todos los productos que ha añadido junto con la cantidad de cada uno. Desde aquí tiene varias opciones: puede eliminar productos individuales o vaciar todo el carrito de una vez, también puede aumentar o disminuir la cantidad de unidades de cada producto según sus necesidades. Para realizar el pago, puede elegir entre PayPal o tarjeta de crédito/débito a través de Stripe. Además, si está suscrito al newsletter de la página, puede aplicar códigos de descuento que le hayan sido enviados, y una vez que esté satisfecho con su selección, solo necesita hacer clic en el botón de pago para completar su compra.

7-Lista de deseos



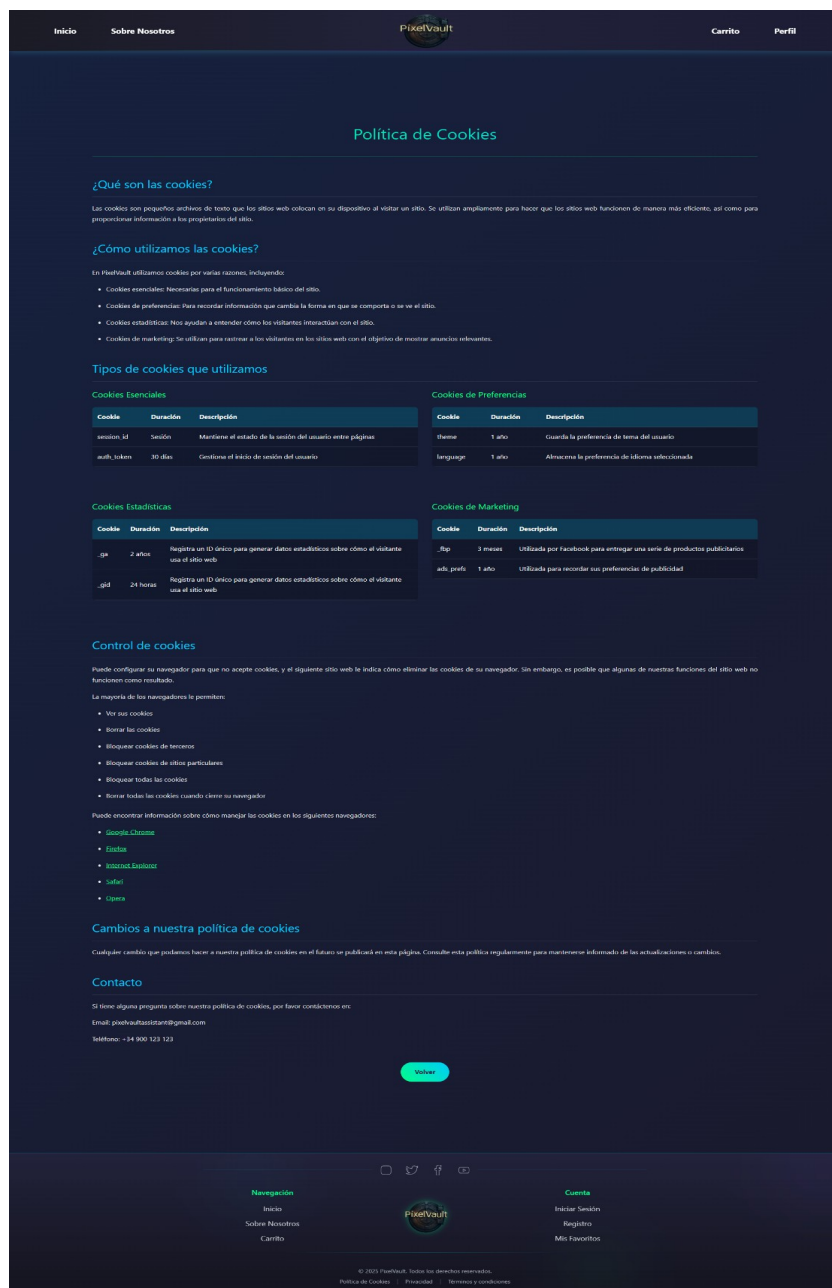
En la página "Mi Lista de Deseos", el usuario puede ver todos los productos que ha marcado como favoritos mientras navega por la tienda. Esta sección le permite guardar los artículos que le interesan para revisarlos más tarde y facilita la decisión de compra. Desde aquí puede ver el precio de cada producto, leer una breve descripción y acceder a más detalles haciendo clic en "Ver más". La página también incluye opciones para compartir su lista de deseos con otros usuarios o comprar todos los productos de la lista de una vez. En la parte inferior se muestra un resumen con el valor total de todos los productos guardados, cuántos productos tiene en su lista y cuántos están disponibles en stock, lo que le ayuda a planificar mejor sus futuras compras.

8-Sobre nosotros



La página "Sobre Nosotros" presenta a PixelVault como una tienda especializada en videojuegos y consolas, explicando que al usar la plataforma los usuarios aceptan automáticamente los términos y condiciones establecidos. La sección incluye información sobre la empresa, destacando que al explorar el catálogo o comprar productos, los clientes participan en la experiencia PixelVault bajo las reglas del juego y políticas establecidas. La página también muestra los valores principales de la empresa: pasión por los juegos, calidad garantizada en todos los productos originales, y el compromiso de crear una comunidad de jugadores a través de eventos y torneos. Al final, ofrece opciones de contacto donde los usuarios pueden enviar preguntas por email o seguir las redes sociales de la empresa para mantenerse actualizados.

9-Cookies



Política de Cookies

¿Qué son las cookies?

Las cookies son pequeños archivos de texto que los sitios web colocan en su dispositivo al visitar un sitio. Se utilizan ampliamente para hacer que los sitios web funcionen de manera más eficiente, así como para proporcionar información a los propietarios del sitio.

¿Cómo utilizamos las cookies?

En PixelVault utilizamos cookies por varias razones, incluyendo:

- Cookies esenciales: Necesarias para el funcionamiento básico del sitio.
- Cookies de preferencias: Para recordar información que cambia la forma en que se comporta o se ve el sitio.
- Cookies estadísticas: Nos ayudan a entender cómo los visitantes interactúan con el sitio.
- Cookies de marketing: Se utilizan para rastrear a los visitantes en los sitios web con el objetivo de mostrar anuncios relevantes.

Tipos de cookies que utilizamos

Cookies Esenciales

| Cookie | Duración | Descripción |
|------------|----------|--|
| session_id | Sesión | Mantiene el estado de la sesión del usuario entre páginas. |
| auth_token | 30 días | Continúa el inicio de sesión del usuario. |

Cookies de Preferencias

| Cookie | Duración | Descripción |
|----------|----------|---|
| theme | 1 año | Guarda la preferencia de tema del usuario. |
| language | 1 año | Almacena la preferencia de idioma seleccionada. |

Cookies Estadísticas

| Cookie | Duración | Descripción |
|--------|----------|--|
| _ga | 2 años | Registra un ID único para generar datos estadísticos sobre cómo el visitante usa el sitio web. |
| _gid | 24 horas | Registra un ID único para generar datos estadísticos sobre cómo el visitante usa el sitio web. |

Cookies de Marketing

| Cookie | Duración | Descripción |
|-----------|----------|--|
| _fbp | 3 meses | Utilizada por Facebook para entregar una serie de productos publicitarios. |
| ads_prefs | 1 año | Utilizada para recordar sus preferencias de publicidad. |

Control de cookies

Puede configurar su navegador para que no acepte cookies, y el siguiente sitio web le indica cómo eliminar las cookies de su navegador. Sin embargo, es posible que algunas de nuestras funciones del sitio web no funcionen como resultado.

La mayoría de los navegadores le permiten:

- Ver sus cookies
- Borrar las cookies
- Bloquear cookies de terceros
- Bloquear cookies de sitios particulares
- Bloquear todas las cookies
- Borrar todas las cookies cuando cierre su navegador

Puede encontrar información sobre cómo manejar las cookies en los siguientes navegadores:

- [Google Chrome](#)
- [Firefox](#)
- [Internet Explorer](#)
- [Safari](#)
- [Opera](#)

Cambios a nuestra política de cookies

Cualquier cambio que podamos hacer a nuestra política de cookies en el futuro se publicará en esta página. Consulte esta política regularmente para mantenerse informado de las actualizaciones o cambios.

Contacto

Si tiene alguna pregunta sobre nuestra política de cookies, por favor contáctenos en:

Email: pixelvaultassistant@gmail.com

Teléfono: +54 900 123 123

Inicio

Navegación

- Inicio
- Sobre Nosotros
- Carrito

Cuenta

- Iniciar Sesión
- Registro
- Mis Favoritos

© 2023 PixelVault. Todos los derechos reservados.
[Política de Cookies](#) | [Privacidad](#) | [Términos y condiciones](#)

La página de "Política de Cookies" explica de manera detallada qué son las cookies y cómo PixelVault las utiliza en su sitio web. Comienza definiendo las cookies como pequeños archivos de texto que se almacenan en el dispositivo del usuario para mejorar el funcionamiento del sitio y personalizar la experiencia de navegación. La página describe los diferentes tipos de cookies que utiliza la plataforma: cookies esenciales para el funcionamiento básico, cookies de preferencias para recordar configuraciones del usuario, cookies estadísticas para generar datos sobre el uso del sitio, y cookies de marketing para mostrar publicidad relevante. También incluye una sección de control donde explica cómo los usuarios pueden gestionar sus preferencias de cookies a través del navegador, con opciones para ver, borrar o bloquear diferentes tipos de cookies. Al final, proporciona

Proyecto Desarrollo de Aplicaciones Web

información de contacto para consultas relacionadas con la política de cookies y un botón "Volver" para regresar al Home.

Viabilidad tecno-económica.

Coste de implementación

Para poner en marcha PixelVault se estima una inversión inicial aproximada de 5.000€ a 10.000€, si se parte de una configuración básica con recursos gratuitos y servicios cloud económicos. Este coste incluye:

- **Hardware/Infraestructura:** Hosting en Azure o AWS (aproximadamente 600-800€/año)
- **Software:** Herramientas de desarrollo gratuitas (Visual Studio, Vue CLI, Git), dominio personalizado (aproximadamente 12€/año), certificado SSL (incluido en hosting o aproximadamente 10€/año)
- **Servicios externos:** Cloudinary, Stripe, PayPal, SendGrid (mayoría con planes gratuitos)
- **Desarrollo:** Puede realizarse internamente, pero si se contrata, el coste podría elevarse a 15.000-25.000€

Rentabilidad

PixelVault presenta viabilidad de rentabilidad si se gestiona adecuadamente.

- Con un ingreso mensual estimado de 5.000-8.000€ (ventas, suscripciones, publicidad), el proyecto cubriría los costes en 4 a 6 meses
- A partir de ese punto, los beneficios mensuales podrían rondar los 2.000-3.000€, incluso más al escalar

Conclusión

El proyecto presenta una excelente viabilidad económica: bajo coste inicial, costes mensuales controlables y alta capacidad de crecimiento. Además, al estar centrado en un nicho como el gaming, tiene un público activo con potencial de fidelización y compras recurrentes.

Trabajo futuro.

Funcionalidades No Implementadas por Falta de Tiempo

Durante el desarrollo del proyecto, se priorizó alcanzar un Producto Mínimo Viable (MVP) funcional. Sin embargo, por limitaciones de tiempo, quedaron varias funcionalidades importantes sin implementar. A continuación, se describen estas funcionalidades pendientes, con el objetivo de tenerlas en cuenta para futuras iteraciones del proyecto:

1. **Redirección por plataforma a productos específicos:** Actualmente solo se ha implementado la plataforma de **Sony**, y al acceder a ella se muestran los productos correspondientes. Sin embargo, la lógica para redirigir a los productos de otras plataformas aún no está desarrollada. El objetivo final es que, al seleccionar una plataforma (por ejemplo, Microsoft, Nintendo, etc.), el usuario sea llevado directamente a una vista filtrada con los productos de esa plataforma.
2. **Sistema de comentarios:** Aunque se contempla la posibilidad de que los usuarios dejen comentarios en los productos, esta funcionalidad no ha sido desarrollada todavía. La estructura de base de datos ya está preparada para soportarla.
3. **Historial de pedidos y pagos:** Es fundamental para cualquier sistema de e-commerce ofrecer al usuario un historial de pedidos y pagos. Aunque las tablas necesarias ya están diseñadas y relacionadas en la base de datos, aún no se ha implementado la lógica ni las vistas que permiten su consulta.
4. **Registro de acciones de administradores:** Para fines de auditoría y control, se planea almacenar todas las acciones realizadas por los administradores del sistema (por ejemplo, creación o eliminación de productos). Esta funcionalidad también cuenta con su diseño en base de datos, pero no se ha puesto en funcionamiento aún.
5. **Gestión del carrusel de imágenes:** Se tiene previsto incluir un carrusel en la página principal con imágenes destacadas de productos. Sin embargo, no se han incorporado suficientes imágenes debido a limitaciones de tiempo, por lo que esta funcionalidad aún no se muestra correctamente.

Conclusiones.

Conseguí crear una tienda online completa para vender videojuegos y consolas. La plataforma funciona bien y es fácil de usar para los compradores.

Mis logros principales:

- Desarrollé una tienda online funcional y fácil de navegar
- Implementé un sistema de pagos seguro con PayPal y Stripe
- Creé un panel de administración para gestionar productos y pedidos
- Agregué funciones sociales como listas de deseos y reseñas
- Diseñé una interfaz atractiva pensada para gamers

Lo que aprendí

Durante el desarrollo de este proyecto aprendí muchas tecnologías y conceptos nuevos:

Vue.js: Fue mi primera vez trabajando con este framework y me sorprendió lo intuitivo que es. Aprendí a crear componentes reutilizables, manejar el estado de la aplicación y hacer que la interfaz sea reactiva. La Composition API me ayudó mucho a organizar el código de manera más limpia.

C# y .NET: Nunca había trabajado con estos lenguajes antes del proyecto. Aprendí a crear APIs, manejar bases de datos con Entity Framework, y estructurar el backend de manera profesional. Me gustó mucho la sintaxis de C# y lo robusto que es .NET 8.

Seguridad: Este fue uno de los temas más importantes que aprendí. Implementé:

- Autenticación con JWT tokens
- Sistemas de roles y permisos
- Validación de datos en frontend y backend
- Protección contra ataques comunes
- Manejo seguro de contraseñas
- Validaciones de transacciones financieras

Sistemas de pago: Integrar PayPal y Stripe fue complicado, pero aprendí a manejar transacciones seguras, gestionar errores y validar pagos correctamente.

Optimización: Aprendí técnicas para hacer que la página cargue rápido, incluso con muchos productos en el catálogo.

Impacto en el mercado

Proyecto Desarrollo de Aplicaciones Web

Mi proyecto PixelVault se diferencia de otras tiendas de videojuegos porque:

- Tiene un diseño muy visual que gusta a los gamers
- Vende productos para todas las consolas (PlayStation, Xbox, Nintendo, PC)
- Permite a los usuarios interactuar entre ellos con comentarios y recomendaciones
- Es más fácil de usar que otras tiendas similares

Tecnologías que usé

Elegí bien las herramientas de desarrollo y aprendí a usarlas:

- **Vue.js:** Para crear toda la interfaz de usuario (primera vez que lo uso)
- **C# y .NET 8:** Para el servidor y la lógica del negocio (completamente nuevo para mí)
- **TailwindCSS:** Para el diseño visual
- **SQL Server:** Para guardar todos los datos
- **Stripe/PayPal:** Para procesar los pagos de forma segura

Todas estas tecnologías funcionaron bien juntas y me permitieron crear una plataforma robusta mientras aprendía cada una de ellas.

Cómo trabajé

Usé metodologías ágiles (Scrum) que me ayudaron a:

- Entregar funcionalidades de forma regular
- Adaptarme a cambios rápidamente
- Mantener la calidad del código
- Organizar mejor mi tiempo

Los mayores desafíos fueron aprender Vue.js desde cero, implementar los sistemas de pago seguros, y entender todos los aspectos de seguridad web.

Planes para el futuro:

Mejoras que quiero hacer:

- Sistema de recomendaciones personalizadas
- Suscripción premium con beneficios exclusivos
- Aplicación móvil

- Soporte para más idiomas y países

Funciones avanzadas que me gustaría agregar:

- Marketplace para juegos de segunda mano
- Integración con plataformas de streaming
- Realidad aumentada para ver productos
- Foros de la comunidad más completos

Mi conclusión personal

PixelVault es un proyecto del que me siento muy orgulloso porque cumplí todos mis objetivos. Creé una tienda online moderna, segura y fácil de usar que se adapta bien a lo que buscan los gamers.

Lo más valioso que me llevo:

- Aprendí Vue.js desde cero y ahora me siento cómodo usándolo
- Dominé C# y .NET 8, que son tecnologías muy demandadas
- Entendí la importancia de la seguridad web y cómo implementarla correctamente
- Gané experiencia con sistemas de pago reales
- Desarrollé una aplicación completa de principio a fin

Este proyecto me enseñó que puedo aprender tecnologías nuevas y aplicarlas en proyectos reales. La base técnica que construí es sólida y me permitirá agregar nuevas funciones en el futuro.

Con todo lo que aprendí, especialmente Vue.js, C#, .NET y seguridad web, me siento preparado para enfrentar proyectos más complejos y trabajar como desarrollador full-stack.

Biblioteca de recursos web y referencias.

Paquetes NuGet .NET

1.Asp.Versioning.Mvc (8.1.0): Biblioteca que permite la gestión de versiones de APIs en aplicaciones ASP.NET Core MVC, facilitando el mantenimiento de múltiples versiones de una API.

2.Asp.Versioning.Mvc.ApiExplorer (8.1.0): Extensión de Asp.Versioning para integrar la documentación de Swagger con APIs versionadas.

3.AutoMapper (14.0.0): Biblioteca para mapeo de objetos que simplifica la transformación entre DTOs y modelos de dominio, reduciendo código repetitivo.

4.CloudinaryDotNet (1.27.5): Cliente oficial de Cloudinary para .NET que permite la gestión de imágenes y vídeos en la nube, incluyendo carga, transformación y optimización.

5.Microsoft.AspNetCore.Identity.Core (2.2.4): Núcleo del sistema de identidad de ASP.NET, proporcionando funcionalidades básicas de gestión de usuarios y roles.

6.Microsoft.AspNetCore.Authentication.JwtBearer (8.0.0): Middleware para la autenticación basada en tokens JWT (JSON Web Tokens) en aplicaciones ASP.NET Core.

7.Microsoft.AspNetCore.Identity.EntityFrameworkCore (8.0.0): Implementación del sistema de identidad de ASP.NET Core utilizando Entity Framework Core para almacenamiento de datos.

8.Microsoft.EntityFrameworkCore (9.0.4): ORM (Object-Relational Mapper) de Microsoft que permite trabajar con bases de datos relacionales utilizando objetos .NET.

9.Microsoft.EntityFrameworkCore.Design (9.0.4): Herramientas de diseño en tiempo de desarrollo para Entity Framework Core, necesarias para las migraciones y scaffolding.

10.Microsoft.EntityFrameworkCore.SqlServer (9.0.4): Proveedor de Entity Framework Core para SQL Server, permitiendo la conexión y operaciones con bases de datos SQL Server.

11.Microsoft.EntityFrameworkCore.Tools (9.0.4): Herramientas adicionales para Entity Framework Core, incluyendo comandos de consola para migraciones.

11.Stripe.net (48.1.0): Cliente oficial de Stripe para .NET que facilita la integración con la API de pagos de Stripe.

12,Swashbuckle.AspNetCore (8.1.1): Framework para generación automática de documentación API utilizando Swagger/OpenAPI, facilitando las pruebas y la documentación.

13.XAct.Core.PCL (0.0.5014): Biblioteca que proporciona funcionalidades criptográficas y de seguridad para aplicaciones .NET.

Proyecto Desarrollo de Aplicaciones Web

Recursos de Imágenes

Las imágenes utilizadas en el proyecto fueron generadas íntegramente mediante Inteligencia Artificial utilizando las siguientes plataformas:

- Grok (X.com): <https://x.com/i/grok>
- ChatGPT (OpenAI): <https://chat.openai.com/>

Anexos.

Las imágenes que se puedan llegar a ver mal, ya sea por dimensiones o por pérdida de calidad, estas estarán subidas en la carpeta Resources/ .