

PROYECTO FIN DE CICLO



IES JOSÉ LUIS MARTÍNEZ PALOMO

Departamento de Informática y Comunicaciones
Técnico Superior en Desarrollo de Aplicaciones Multiplataforma

Título del proyecto

PÍXEL SHOP

Autor: Juan Antonio Peñaranda Madrid

Tutor: Jesús Ibáñez López

Alquerías, 4 de diciembre de 2025

INDICE

APP PÍXEL SHOP

1.INFORMACION DEL PROYECTO

1.1 Descripción del proyecto

El proyecto consiste en desarrollar una **aplicación de escritorio** de tienda digital de videojuegos, similar a plataformas como Steam, Epic Games o GOG. Permitirá la compra y gestión de juegos indies¹ y triple A², ofreciendo una experiencia sencilla, intuitiva y de alto rendimiento para jugadores y desarrolladores.

1.2 Utilidad del proyecto

Que beneficios aporta:

Los beneficios de Pixel Shop se centran en tres pilares fundamentales: **simplicidad, economía y accesibilidad**. A diferencia de las plataformas existentes, Pixel Shop no intenta competir en características avanzadas, sino en ofrecer lo esencial de forma eficiente.

1. Beneficios para jugadores

- **Interfaz simple y rápida:**
 - Frente a plataformas como Steam que han acumulado funciones durante años, Pixel Shop ofrece una experiencia más limpia y directa
- **Mejor descubrimiento de juegos indie:**
 - Menor saturación de contenido facilita encontrar juegos de calidad
- **Transparencia total:**
 - Muestra claramente como se distribuye el precio
 - Sin letra pequeña ni condiciones confusas
 - Políticas claras y justas

2. Beneficios para desarrolladores

- **Comisión más justa**

Ejemplo comparativo de ganancia por cada juego vendido a 20€

Plataforma	Comisión	Gana desarrollador	Ganancia Plataforma
Steam	30%	14.00€	6€
GOG	30%	14.00€	6€
Epic games	12%	17.60€	2.40€

¹ Juegos indies: videojuego desarrollado por un estudio independiente, sin el apoyo financiero de una gran editorial de videojuegos

² Triple A: juegos de gran presupuesto, producidos por grandes editoriales con enormes costos de desarrollo y marketing

Pixel Shop	15%	17.00€	3€
------------	-----	--------	----

*Epic 12% solo después del primer millón de dólares

- **Cuota de entrada reducida (25€)**

Steam/Epic Games : 100\$ (93€)

Ahorro: 69€ por juego publicado

Esto para un estudiante o desarrollador indie ,68€ de ahorro pueden destinarse:

- Marketing en redes sociales
- Assets para el juego
- Hosting de página web

- **Visibilidad garantizada**

- *Problema en steam o epic games:* Tu juego puede pasar desapercibido
- *Solución en Pixel Shop:* Debido al catálogo pequeño tu juego no se pierde entre miles

Quien puede estar interesados:

Usuarios finales:

- Edad: 12-50 años (rango amplio del gaming moderno)
- Jugadores que buscan alternativas a Steam
- Usuarios interesados en apoyar desarrolladores independientes
- Estudiantes con presupuestos limitados que valoran mejores precios
- Early adopters³ dispuestos a probar nuevas plataformas

Desarrolladores:

- Estudios indie con presupuestos limitados
- Desarrolladores noveles que quieren minimizar costos de entrada
- Estudios que buscan diversificar sus canales de distribución
- Desarrolladores insatisfechos con las comisiones de plataformas establecidas

Estudio de mercado:

Steam
Comisión: <ul style="list-style-type: none">- 30% (estándar), con sistema escalonado
Cuota de subida de juego a la plataforma: <ul style="list-style-type: none">- 100 USD por juego (reembolsable tras alcanzar 1.000\$ en ventas)

³ Early adopters: consumidores pioneros que adoptan rápidamente productos o servicios nuevos e innovadores, incluso antes que la mayoría del público

Ventajas:

- Líder absoluto del mercado con aproximadamente 75% de cuota de mercado
- Base de usuarios enorme (más de 24 millones de usuarios simultáneos en pico)
- Ecosistema completo con comunidad, talleres, logros, etc.

Desventajas:

- Alta competencia (más de 40.000 juegos disponibles)
- Interfaz compleja y sobrecargada
- Solo el 6% de desarrolladores considera justificada la comisión del 30%⁴

Epic Games Store

Comisión:

- 0% en el primer millón de dólares por aplicación al año (desde junio 2025)
- 12% después del primer millón (88% para el desarrollador)

Cuota de subida de juego a la plataforma:

- 100 USD por juego

Ventajas:

- Modelo de reparto de ingresos más generoso de la industria: 100%/0% primer millón, luego 88%/12%
- Exención de royalties del 5% de Unreal Engine para juegos vendidos en Epic Games Store
- Regalos semanales de juegos gratuitos que atraen usuarios

Desventajas:

- Catálogo menos extenso
- Funcionalidades comunitarias limitadas comparadas con Steam
- Menor ecosistema de mods y contenido generado por usuarios

GOG

Comisión:

- 30%

Cuota de subida de juego a la plataforma:

- Gratuita (pero proceso de curación selectivo)

Ventajas:

- Política sin DRM (los juegos se pueden descargar e instalar offline)
- Proceso de curación que asegura calidad
- Propiedad real de los juegos (no solo licencias revocables)

Desventajas:

- Menor visibilidad que Steam
- Catálogo más limitado por política de curación estricta

⁴ Encuesta *State of the Industry 2020* de la GDC

- Público nicho enfocado en jugadores "old school" y opositores al DRM

Itch.io

Comisión:

- Variable, decidida por el desarrollador (0-100%), con 10% como valor sugerido por defecto

Cuota de subida de juego a la plataforma:

- Gratuita - sin barreras de entrada

Ventajas:

- Modelo "paga lo que quieras" tanto para clientes como para desarrolladores
- Los desarrolladores controlan completamente el porcentaje que va a la plataforma, incluso 0%
- Barrera de entrada nula: cualquiera puede publicar

Desventajas:

- Menor tráfico y visibilidad que plataformas principales
- Base de usuarios más pequeña
- Percibida como plataforma para desarrolladores principiantes
- Control de calidad variable debido a la apertura total

Microsoft Store

Comisión:

- 12% para juegos (15% para aplicaciones no-gaming)

Cuota de subida de juego a la plataforma:

- Aproximadamente \$99 USD para cuentas de empresa (una sola vez para toda la cuenta, no por juego)

Ventajas:

- Comisión competitiva del 12% para juegos, igualando a Epic Games Store
- Acceso a más de 1 mil millones de usuarios de Windows en más de 200 mercados
- Opción de usar plataforma de comercio propia y mantener 100% de ingresos para apps no-gaming
- Potencial de cross-play⁵ con Xbox

Desventajas:

- Histórico de problemas técnicos y de interfaz
- Funcionalidades comunitarias limitadas
- Catálogo de juegos significativamente menor que Steam

⁵ Cross-play: capacidad de los jugadores de un videojuego multijugador para jugar juntos sin importar la plataforma que utilicen

Pixel Shop
Comisión: <ul style="list-style-type: none"> - 15%
Cuota de subida de juego a la plataforma: <ul style="list-style-type: none"> - 25€
Ventajas: <ul style="list-style-type: none"> - Comisión intermedia entre Steam (30%) y Epic (12% post-millón) - Interfaz limpia y moderna diseñada desde cero - Enfoque en visibilidad para desarrolladores indies - Menor saturación de contenido
Desventajas: <ul style="list-style-type: none"> - Nueva plataforma sin base de usuarios establecida - Necesidad de inversión significativa en marketing - Competencia directa con gigantes establecidos - Construcción de confianza con desarrolladores y jugadores

Resumen				
Plataforma	Comisión	Cuota subida	Ventajas	Desventajas
Steam	30%	100\$	Líder mercado, gran visibilidad	Interfaz compleja, alta comisión
Epic Games	12%	100\$	Baja comisión post-millón	Menos catálogo
GOG	30%	Gratuita	Sin DRM ⁶ , propiedad real	Catálogo limitado
Itch.io	Variable (0-100%), sugerido 10%	Gratuita	Desarrollador elige comisión	Menor visibilidad
Microsoft Store	12%	\$99 (cuenta)	Integración Windows/Xbox	Menor adopción en PC
Píxel shop	15%	25€	Equilibrio comisión/accesibilidad	Nueva en el mercado

2.PLANIFICACION DEL PROYECTO

2.1 Fases

Diagrama de Gantt:

Este va desde el comienzo de curso 15 septiembre hasta el 4 diciembre

⁶ DRM: Digital Rights Management o Gestión de Derechos Digitales

Planificación temporal del proyecto de 10 semanas, distribuidas en diferentes fases.

*Aclaración: La S significa semana

FASE	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
1. <i>Análisis y documentación inicial</i>										
2. <i>Diseño de la arquitectura</i>										
3. <i>Configuración del entorno</i>										
4. <i>Desarrollo del Backend</i>										
5. <i>Desarrollo del Frontend</i>										
6. <i>Integración y pruebas</i>										
7. <i>Optimización y correcciones</i>										
8. <i>Documentación final</i>										
9. <i>Despliegue y Lanzamiento</i>										

Análisis y fases posteriores:

1. Preparar la presentación

- Hacer el PowerPoint
- Preparar qué voy a explicar y practicar
- Decidir qué partes del código/app mostrar

2. Ajustes finales antes de la defensa

- Probar que todo funcione bien el día de la presentación
- Corregir bugs de última hora
- He de asegurarme de que el .exe se ejecuta sin problemas

3. Defensa del proyecto

- Presentación ante los profesores
- Demo en vivo de la aplicación
- Responder preguntas de los profesores

4. Correcciones después de la evaluación

- Arreglar cosas que me digan los profesores
- Mejorar documentación si falta algo
- Actualizar el PDF final si hay cambios

2.2 Coste económico

Cuánto costaría llevar a cabo publicarlo al público general:

Concepto	Descripción	Cantidad	Precio Unit.	Total
COSTES DE DESARROLLO				
Desarrollador	Desarrollar de la página web	386 h	7€	2.702€
LICENCIAS Y SOFTWARE				
Spring boot	Framework backend	-	Gratis	0€
Microsoft Azure (Student)	Servidor online (gratis)	3 meses	0€	0€
Microsoft Azure (Estandar)	Servidor online (Si hubiese pagado)	3 meses	14€	(42€)
GitHub	Control de versiones	-	Gratis	0€
Visual Studio	IDE desarrollo	-	Gratis	0€
INFRAESTRUCTURA				
Equipo de desarrollo	Portátil	1	1.000€	1.000€
Dispositivos de prueba	Móvil	2	400€	800€
MARKETING				
Diseño de logo (Canva)	Suscripción Pro	1	12€	12€
Campaña de lanzamiento	Anuncios redes sociales	1	150€	150€
TOTAL, INVERSION INICIAL:				4.706€

*Sobre las horas me he tenido en cuenta a mi: Trabajo 173 h al mes y como son 2 meses y 1 semana, serian un total de 386

*Sobre el sueldo por hora he hecho un aproximado de lo que cobra una persona normal en un trabajo normal

Planificación amortización costes:

Fuentes de ingreso:

1. **Cuota publicación:** 25€ por cada juego que un desarrollador publica en la plataforma
2. **Comisión por venta:** 15% de cada transacción realizada

Análisis beneficios cantidad y tiempo:

Escenario	Usuarios	Juegos Subidos	Ingresos por subida	Usuarios compra n	Compra promedio	Comisión 15%	Ingres. Totales mes
Pesimista	300	8	200€	10% = 30	25€	113€	313€

Realista	1.500	25	625€	20% = 225	27€	911€	1.536€
Optimista	5.000	60	1.500€	30% = 1.000	28€	4.200€	5.700€

*Cada juego subido/publicado cuesta 50€

*Cálculo de la comisión ejemplo o= 30 usuarios compran x 25 promedio compra x 0,15 comisión, que daría un total de 133€

Tiempo de recuperación de la inversión:

Inversión Inicial: 4.706€

Escenario	Ingresos/Mes	Costes/Mes	Neto/Mes	Recuperación	Beneficio Año1
Pesimista	313€	14€	299€	51,7 meses	-1.118€
Realista	1.536€	14€	1.522€	3,1 meses	13.558€
Optimista	5.700€	14€	5.686€	0,83 meses	63.526€

*Los 14€ son del servidor de Microsoft Azure

Calculo	
Calculo Pesimista:	<i>Recuperación</i> 4.706€ / 299€ = 15,7 meses
	Beneficio año1 Ingresos: 313€ × 12 = 3.756€ Costes: 14€ × 12 = 168€ Neto: 3.756€ - 168€ = 3.588€ Final: 3.588€ - 4.706€ = -1.118€
Calculo Realista:	<i>Recuperación</i> 4.706€ / 1.522€ = 3,1 meses
	Beneficio año1 Ingresos: 1.536€ × 12 = 18.432€ Costes: 14€ × 12 = 168€ Neto: 18.432€ - 168€ = 18.264€ Final: 18.264€ - 4.706€ = 13.558€
Calculo Optimista:	<i>Recuperación</i> 4.706€ / 5.686€ = 0,83 meses (~25 días)
	Beneficio año1 Ingresos: 5.700€ × 12 = 68.400€ Costes: 14€ × 12 = 168€

	Neto: 68.400€ - 168€ = 68.232€ Final: 68.232€ - 4.706€ = 63.526€
--	---

Evolución beneficios primer año (Escenario Realista):

Trimestre	Ingresos	Costes	Beneficio	Acumulado
Q1 (Mes 1-3)	4.608€	42€	4.566€	4.566€
Q2 (Mes 4-6)	4.608€	42€	4.566€	9.132€
Q3 (Mes 7-9)	4.608€	42€	4.566€	13.698€
Q4 (Mes 10-12)	4.608€	42€	4.566€	18.264€
Total, año	18.432€	168€	18.264€	

Calculo:

- Ingresos trimestre: 1.536€/mes × 3 = 4.608€
- Costes trimestre: 14€/mes × 3 = 42€
- Beneficio trimestre: 4.608€ - 42€ = 4.566€

Beneficio neto final: 18.264€ - 4706€ inversión = 13.558€

ROI⁷: (13.558€ / 4.706€) × 100 = 288%

Conclusión: Por cada euro invertido, recupero 2,88€ en el primer año

Beneficios No económicos:

- **Experiencia técnica:** Dominio completo de Spring Boot, JavaFX, Azure MySQL y desarrollo full-stack
- **Portfolio profesional:** Proyecto funcional completo para demostrar habilidades en entrevistas
- **Impacto social:** Plataforma 73% más barata que Steam que ayuda a desarrolladores indie españoles
- **Aprendizaje empresarial:** Gestión de proyectos, análisis de mercado y viabilidad económica

2.3 Reflexión

Viabilidad temporal del proyecto:

El proyecto de 10 semanas (386 horas) es viable por estos motivos:

- Tecnologías ya conocidas (Java, Spring Boot, MySQL)
- Alcance limitado y claro
- Planificación equilibrada en el Gantt

Puede haber algunos riesgos como:

⁷ ROI: Retorno de inversión

- Integración JavaFX + Spring Boot (poca documentación)
- Primera vez usando Azure en producción
- Tiempo justo si surgen imprevistos

Por lo tanto, mi estrategia es:

- Priorizar funcionalidades básicas
- Dejar algunas funciones como foro de la comunidad para versión futura si hay retrasos
- Reuniones semanales con tutor

2.4 Alcance y limitaciones del sistema

Alcance:

Versión 1.0 de la aplicación

- **Usuarios:** Registro, login, catálogo, compras, biblioteca, perfil
- **Administradores:** Publicar juegos, gestionar catálogo
- **Técnico:** App escritorio Windows (.exe), servidor online Azure, interfaz JavaFX

Limitaciones:

Aspecto	Limitación
Plataforma	Solo Windows
Funcional	Sin foros, logros, mods
Juegos	No se descargan (voy a intentar que si se descarguen)

Mejoras futuras:

Aspecto	Mejoras futuras
Técnicas	Versión web y móvil Multiplataforma (Linux/Mac)
Funcionales	Descarga de juegos con DRM Foros y comunidad Sistema de amigos
Negocio	Panel estadísticas para desarrolladores Sistema de afiliados

3. INFORMACIÓN Y DESARROLLO

3.1 Información

Cuando se obtiene la información:

Información	Fase	Momento específico
-------------	------	--------------------

Requisitos funcionales	Análisis y documentación inicial	Semana 1
Análisis competencia y mercado	Análisis inicial	Semana 2
Especificaciones técnicas de arquitectura	Diseño de la arquitectura	Semana 2-3
Configuración de herramientas	Configuración del entorno	Semana 3
Validación de requisitos con tutor	Durante todo el proyecto	Reuniones Semanales
Feedback de usuarios y pruebas	Integración y pruebas	Semana 8

Donde se obtiene o quien la ofrece:

Información	Fuente	Contacto/Enlace
Requisitos funcionales iniciales	Tutor	Reuniones con tutor, guía de TFG
Análisis competencia y mercado	Steam, Epic Games, GOG	https://store.steampowered.com/?l=spanish https://www.epicgames.com/site/es-ES/home https://www.gog.com/
Documentación Spring Boot	Sitio oficial de Spring	https://docs.spring.io/spring-boot/documentation.html
Documentación MySQL	Sitio oficial de MySQL	https://dev.mysql.com/doc/
Documentación Java	Sitio oficial de java	https://docs.oracle.com/en/java/

De qué forma se obtiene:

Método	Descripción
Encuestas Online	Cuestionarios con Google Forms
Análisis Competitivo	De webs similares (Steam /Epic Games)
Cursos y Tutoriales Online	Plataformas educativas como Udemy, YouTube, etc.
Consulta a expertos	Contacto con desarrolladores con experiencia real
Pruebas de usabilidad	Testing con usuarios reales
Análisis de código abierto	Revisión de repositorios en GitHub de proyectos similares

3.2 Lista de apartados a desarrollar

Priorización: Método MoSCow

- **MUST:** Obligatorio para que funcione
- **SHOULD:** Importante pero no crítico
- **COULD:** Deseable si hay tiempo
- **WON'T:** Fuera de alcance v1.0

Definición MoSCow: Técnica de priorización que clasifica tareas o requisitos de un proyecto en cuatro categorías para asegurar el éxito del producto, incluso con recursos limitados

Prioridad 1: Must Have	
Configuración Inicial (Semana 1-3)	Configurar Visual Studio Code + Maven + Spring Boot Conectar proyecto a GitHub Crear base de datos MySQL en Azure Diseñar modelo entidad-relación
Backend (Semana 4-5)	Crear entidades JPA (Usuario, Artículo, Compra, Biblioteca) Implementar repositorios con Spring Data Crear controladores
Frontend (Semana 5-6)	Integrar JavaFX con WebView Crear vistas con Thymeleaf:
Funcionalidades Core (Semana 6-7)	Sistema de autenticación (login/logout) Biblioteca personal Panel admin (añadir/editar/eliminar juegos)
Testing y Ajustes (Semana 7-8)	Pruebas funcionales completas Corrección de bugs

Prioridad 2: Should Have	
Mejoras importantes (Semana 8)	Mejoras de diseño CSS Sistema de notificaciones (toast/alertas)

Prioridad 3: Could Have	
Funcionalidades extra (Semana 9-10)	Sistema de wishlist Simulación de descarga de juegos Dashboard con estadísticas (admin) Búsqueda avanzada con filtros combinados

Prioridad 4: Wont Have	
Versión futura	Sistema de amigos Foros de comunidad Sistema de logros Mods y workshop

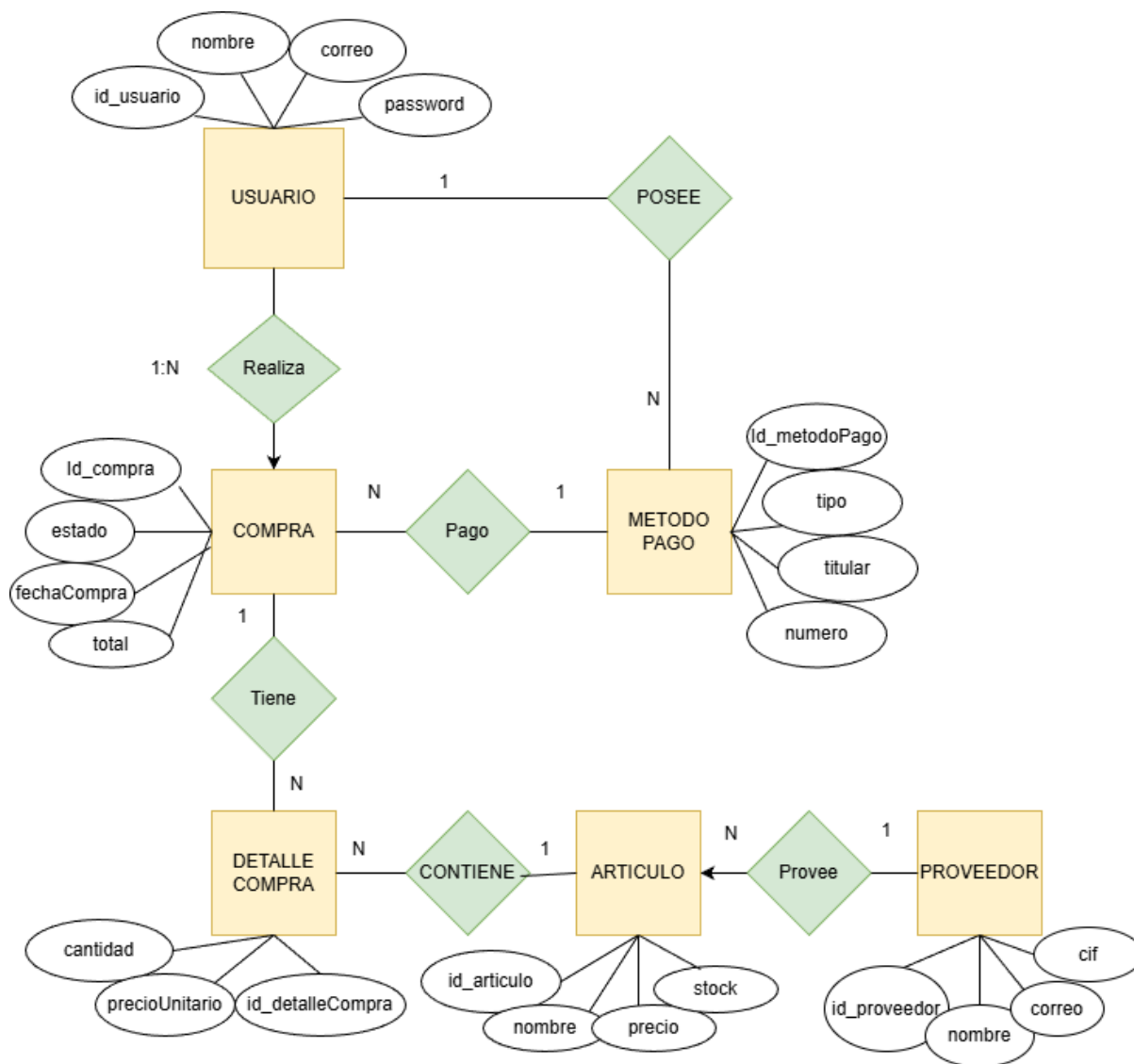
Resumen por semanas:

Semana	Prioridad	Tareas
1-3	MUST	Entorno, BD, diseño
4-5	MUST	Backend completo
5-6	MUST	Frontend base
6-7	MUST	Funcionalidades principales
7-8	MUST	Testing y correcciones
8	SHOULD	Mejoras importantes
9-10	COULD	Funcionalidades extra

4. DISEÑO

4.1 Desarrollo

Modelo Entidad Relación (modelo conceptual):



Modelo Estándar de datos:

- Usuario => (*id_usuario*, nombre, correo, localidad, provincia, calle, tipoUsuario, password, activo)
- Compra => (*id_compra*, *id_usuario*, fechaCompra, total, *id_metodoPago*, estado, detalleCompra)
- DetalleCompra => (*id_detalleCompra*, *id_compra*, *id_articulo*, cantidad, precioUnitario)
- Articulo => (*id*, nombre, descripcion, precio, desarrollador, genero, stock, videoUrl, creadoPor, *id_proveedor*)
- BibliotecaUsuario => (*id*, *id_usuario*, *id_articulo*, fechaAdquisicion)
- MetodoPago => (*id_metodoPago*, tipo, titular, numero, fecha_caducidad, cvv, *id_usuario*, activo)
- Proveedor => (*id_proveedor*, *id_articulo*, nombre, correo, telefono, direccion, cif, web)

*El método de pago y proveedor todavía no lo he desarrollado puede cambiar

Diseño de la interfaz:

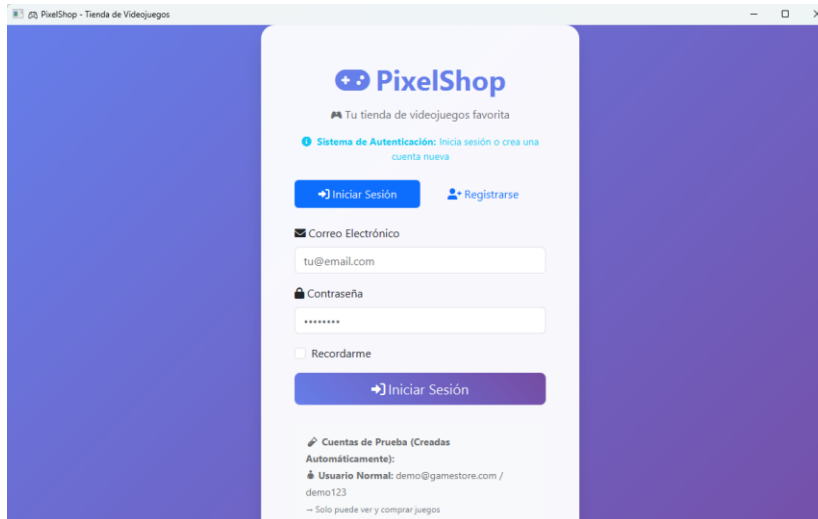
Juan Antonio
Peñaranda Madrid

Estas son algunas funciones que tendrá el programa

*(Por desarrollar, de momento solo tengo esto, pero obviamente no va a ser el resultado final)

1. Inicio de sesión

Hay 2 diferentes inicios de sesión uno para el usuario y otro para el admin, el admin tendrá más funciones



2. Inicio de sesión como usuario

El usuario normal solo podrá ver su perfil, el juego (artículo), su biblioteca y comprar

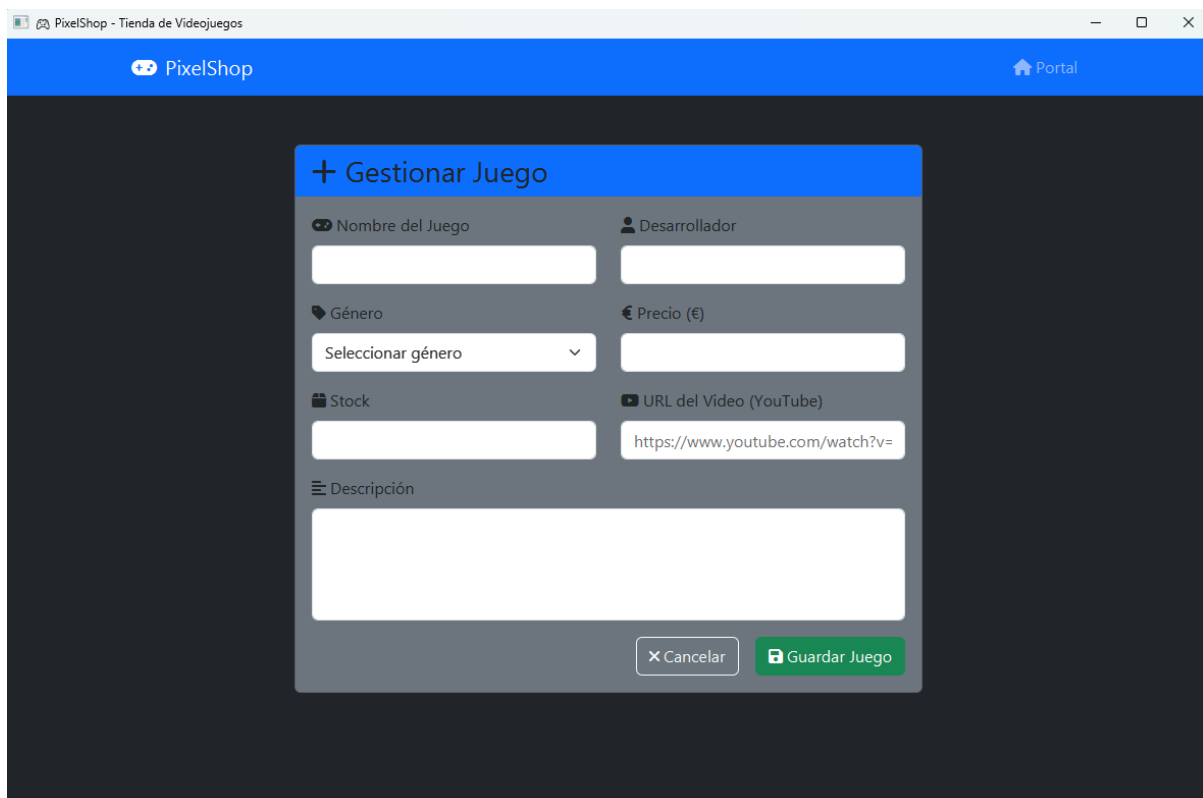


3. Inicio de sesión como admin

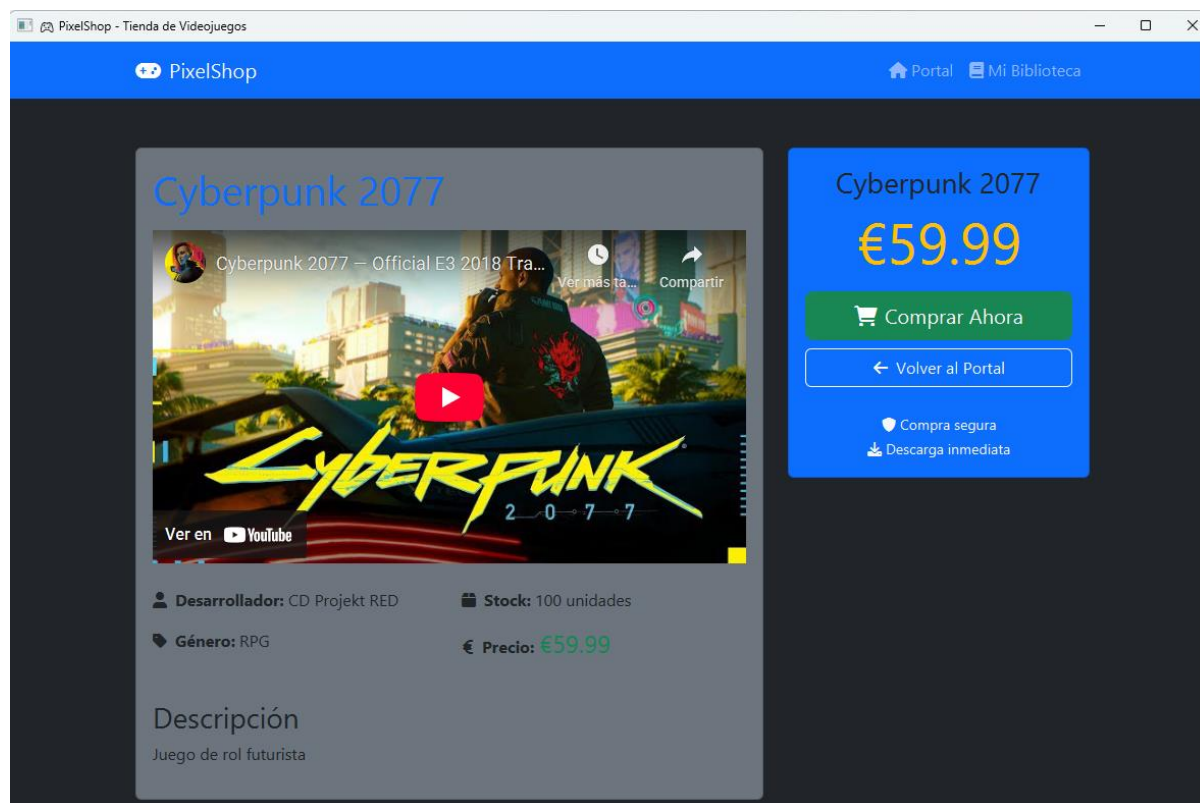
El usuario admin tendrá las mismas funciones que el usuario normal, pero este podrá añadir juego para ponerlo a la venta



4. Añadir Juego



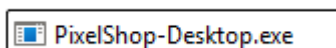
5. Ver juego



*(Obviamente este no es el resultado final , ni estos van a ser el diseño real es para que se vea mas o menos las funciones que tendrá el programa , tambien añadire mas cosas que a lo largo del tiempo pondre en el pdf)

*(Decir tambien que me he enfocado mas en el backend que en el frontend)

*El programa final quedara como un .exe por el cual cualquier persona pueda instalarlo en su pc y ejecutarlo y que funcione con una base datos a tiempo real



Tecnologías utilizadas y justificación:

1.Lenguaje

Tecnología: Java

Descripción: Lenguaje principal de desarrollo para backend y frontend

Justificación: Se ha elegido Java porque es el primer lenguaje aprendido en el ciclo DAM, permite desarrollar aplicaciones web robustas y se integra perfectamente con Spring Boot y otras herramientas necesarias para el proyecto.

2.Backend

Tecnología: Spring Boot

Descripción: Framework principal para lógica de negocio, servicios y arquitectura

Justificación: Simplifica el desarrollo al autoconfigurar todo lo necesario e incluir un servidor embebido. Además, ya lo había usado antes y me resulta familiar

Tecnología: Spring Data JPA

Descripción: Capa de persistencia, repositorios y acceso a base de datos

Justificación: Me ahorra escribir SQL manualmente para operaciones básicas y gestiona automáticamente las conexiones con la base de datos

Tecnología: Spring Web

Descripción: Framework para servicios REST y controladores

Justificación: Proporciona las anotaciones necesarias para crear los endpoints y manejar las peticiones HTTP de forma sencilla.

Tecnología: Thymeleaf

Descripción: Motor de plantillas para vistas HTML dinámicas

Justificación: Me permite crear páginas HTML que se actualizan con datos del servidor de forma natural, sin necesidad de un framework frontend complejo.

3.Frontend/Interfaz

Tecnología: JavaFX

Descripción: Framework para crear la interfaz gráfica de escritorio

Justificación: Permite crear una aplicación de escritorio nativa que integra Spring Boot, evitando que el usuario necesite abrir un navegador.

Tecnología: WebView(JavaFX)

Descripción: Componente para renderizar contenido web dentro de la aplicación

Justificación: Me permite reutilizar las vistas HTML/CSS ya creadas con Thymeleaf dentro de una ventana de escritorio, sin tener que rehacer toda la interfaz.

4.Base de datos

Tecnología: MySQL

Descripción: Sistema gestor de BD relacional para usuarios, juegos y compras

Justificación: Es robusto, soporta múltiples usuarios simultáneos y funciona perfectamente con Azure. Además ya lo conocía de clase.

5.Hosting y servidor

Tecnología: Microsoft Azure

Descripción: Servidor en la nube para backend y base de datos

Justificación: Permite alojar la base de datos MySQL en la nube para acceder desde cualquier lugar, y ofrece servicios específicos para aplicaciones Java.

6.Gestion de proyecto

Tecnología: Maven

Descripción: Herramienta de gestión y construcción de proyectos

Justificación: Gestiona automáticamente las dependencias del proyecto y facilita la compilación y empaquetado de la aplicación.

Tecnología: GitHub

Descripción: Repositorio de código y control de versiones

Justificación: Me permite guardar versiones del código, hacer copias de seguridad y colaborar en el proyecto de forma ordenada.

7.Herramientas de desarrollo

Tecnología: Visual Studio Code

Descripción: IDE para desarrollo del código fuente Java

Justificación: Es ligero, fácil de usar y tiene extensiones para Java, Maven y Git que facilitan el desarrollo.

Tecnología: MySQL Workbench

Descripción: Herramienta gráfica para gestión de bases de datos

Justificación: Lo uso para realizar consultas SQL de forma visual y verificar que los datos se almacenan correctamente.

8.Distribucion

Tecnología: Launch4j

Descripción: Generación de ejecutable .exe nativo para Windows desde el JAR

Justificación: Convierte el JAR en un .exe que se puede ejecutar con doble clic, haciendo la aplicación más fácil de usar para cualquier usuario de Windows.

Esquema de la aplicacion: Diagrama UML

*(Por desarrollar todavía)

5.IMPLEMENTACION

Elaboracion de un prototipo

1.Implementacion:

Creación de la base datos en la nube y conectarla al proyecto

1.Entorno de desarrollo:

- Ciclo de desarrollo , prueba y paso a producción

Primero creamos un servidor online en Microsoft Azure

Inicio > Servidores de Azure Database for MySQL > mysqldemojuan

mysqldemojuan | Bases de datos ☆ ...
Servidor flexible de Azure Database for MySQL

Buscar

+ Agregar Eliminar Actualizar Comentarios

Información general
Registro de actividad
Control de acceso (IAM)
Etiquetas
Diagnosticar y solucionar problemas
Learning center
Visualizador de recursos
Configuración

Puede crear, ver y eliminar bases de datos MySQL en este servidor. Tenga en cuenta que no puede eliminar ninguna base de datos information_schema, performance_schema. Puede conectarse a las bases de datos mediante MySQL herramientas de cliente.

Nombre ↑	Juego de caract...	Colación	Tipo de esque...
mysql	utf8mb4	utf8mb4_0900_...	System
information_schema	utf8mb3	utf8mb3_gener...	System
performance_schema	utf8mb4	utf8mb4_0900_...	System
sys	utf8mb4	utf8mb4_0900_...	System
tiendajuegos	utf8mb3	utf8mb3_gener...	User

Abrir en Power BI

Nos vamos a Mysql Workbench y seleccionamos el servidor de Microsoft Azure

MySQL Connections +

Local instance MySQL80
root
localhost:3306

TiendaJuegos
adminuser
mysqldemojuan.mysql.database.azure...

tiendajuegos

- Tables
 - articulo
 - biblioteca_usuario
 - compra
 - detalle_compra
 - usuario
- Views
- Stored Procedures
- Functions

Result Grid

	id	desarrollador	descripcion	genero	nombre
▶	1	CD Projekt RED	Juego de rol futurista	RPG	Cyberpunk 2077
	2	CD Projekt RED	Aventura épica de fantasía	RPG	The Witcher 3
	3	Rockstar Games	Acción en mundo abierto	Acción	Grand Theft Auto V
	4	Mojang Studios	Construcción y supervivencia	Sandbox	Minecraft
	5	Valve	Shooter táctico competitivo	FPS	Counter-Strike 2
	6	Numantian Games	Juego de zombies	RPG	They are billions
	7	EA SPORT	Juego de deporte	Simulación	FIFA 19
*	NULL	NULL	NULL	NULL	NULL

- Código

En la aplicación de spring boot en **“application.properties”** lo modificamos para añadir la conexión al servidor

```
src > main > resources > application.properties
1
2 {
3   # CONFIGURACIÓN Microsoft Azure
4   spring.datasource.url=jdbc:mysql://mysqldemojuan.mysql.database.azure.com:3306/tiendajuegos?sslMode=REQUIRED&serverTimezone=UTC
5   spring.datasource.username=adminuser
6   spring.datasource.password=Poppy@azure
7   spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
8 }
9
10 # Configuración JPA/Hibernate para MySQL
11 spring.jpa.hibernate.ddl-auto=update
12 spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
13 spring.jpa.show-sql=false
```

2.Entorno de ejecución:

- Ejemplos y posibilidades


Las posibilidades son infinitas obviamente con una limitante de dinero que se tenga.

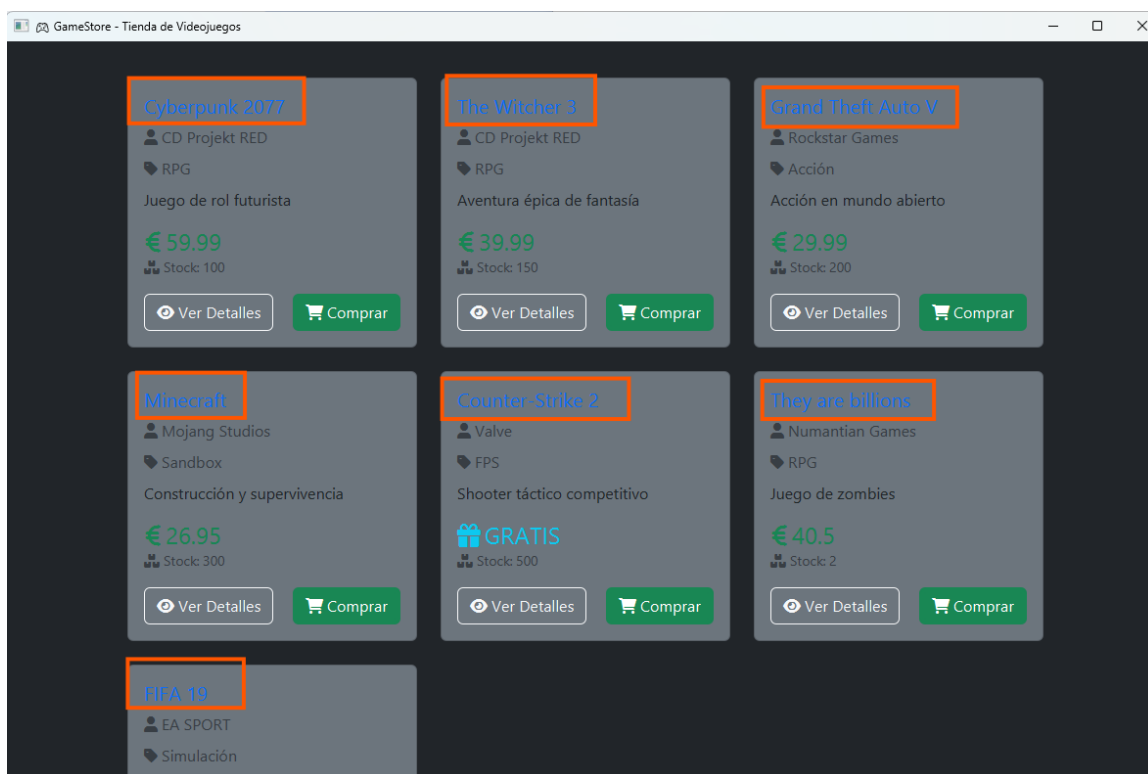
Los servidores online de Microsoft Azure son buenos, pero son caros de contratar.

Esto permite que cualquier usuario que use la aplicación sea de donde sea pueda usar la aplicación y ver actualizado a tiempo real la aplicación.

- Test

Como podemos ver los datos son los correctos y no falla nada, los datos que están en Mysql Workbench son los mismos que en la aplicación de pc.

 GameStore-Desktop.exe



2.Implementacion:

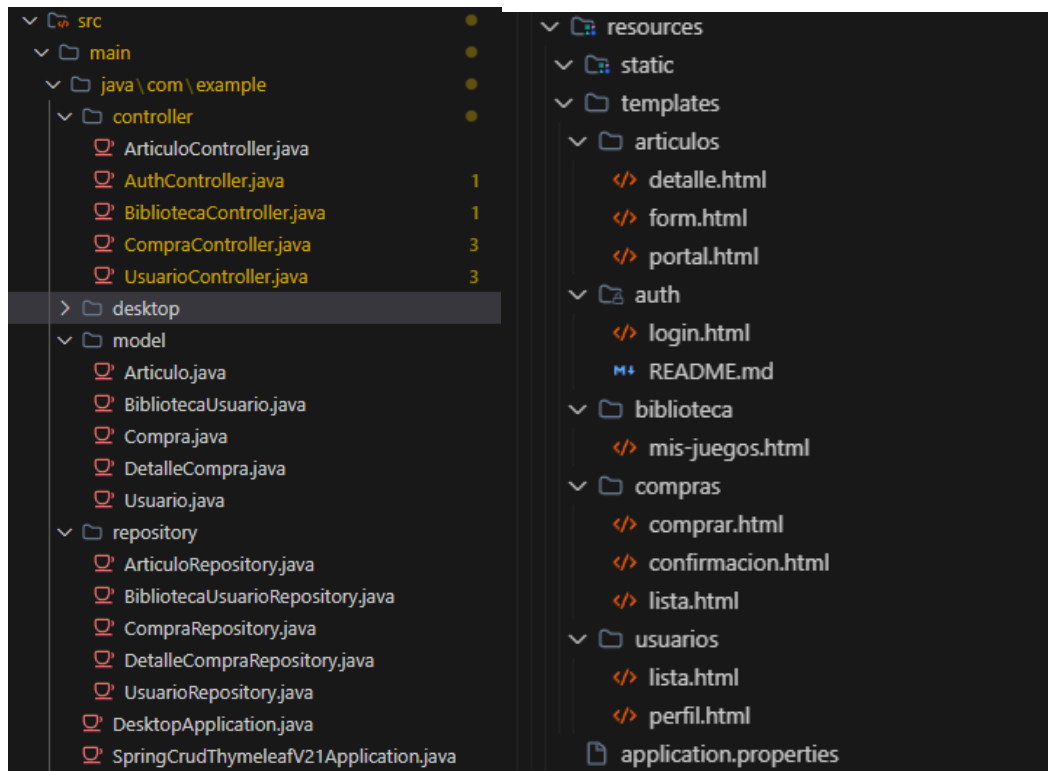
Creación del programa en general

1.Entorno de desarrollo:

- Ciclo de desarrollo , prueba y paso a produccion

Planteamos la idea en un diagrama entidad relacion conceptual

Vamos a visual



- Código

2.Entorno de ejecución:

- Ejemplos y posibilidades

- Test

6.PRUEBA

Elaboración de pruebas

1.Prueba carga de video en el servidor

Utilidad, valoración y posibles errores

Participación en la evolución de los usuarios o clientes finales

Resultados obtenidos

2.Prueba ejecutable .exe en otros ordenadores

7.MANTENIMIENTO

Juan Antonio
Peñaranda Madrid

8.CONCLUSIONES

Pixel Shop tiene potencial de éxito en un mercado dominado por gigantes, aprovechando el nicho de desarrolladores indie que buscan alternativas. El éxito dependerá de:

- **Ejecución:** Deberá haber calidad técnica desde el día 1
- **Construcción de comunidad:** Crear comunidad y hacerse notar entre los desarrolladores y consumidores
- **Adaptabilidad:** Tendrá que adaptarse a los competidores actuales y ofrecer servicios que los otros no ofrezcan
- **Sostenibilidad financiera:** Crecer orgánicamente sin quemar capital

9.BLIBIOGRAFÍA

1. Información del proyecto

Estudio de mercado:

<https://www.xataka.com/videojuegos/microsoft-va-a-steam-recorta-al-12-comision-que-se-queda-cada-videojuego-comprado-microsoft-store>

Steam:

<https://partner.steamgames.com/doc/gettingstarted/appfee>

<https://store.steampowered.com/sub/163632/>

Epic games:

<https://store.epicgames.com/en-US/distribution>

GOG:

<https://www.gog.com/en/>

<https://www.1d3.com/blog/video-game-distribution-platforms-gog-com>

<https://www.pcgamer.com/steam-and-gog-take-30-revenue-cut-suggests-fez-creator-phil-fish/>

Itchio.io:

<https://itch.io/>

<https://www.itchio-spanish.com/account.html>

<https://es.wikipedia.org/wiki/Itch.io>

Microsoft Store:

<https://learn.microsoft.com/en-us/windows/apps/publish/publish-your-app/why-distribute-through-store>

<https://learn.microsoft.com/en-us/windows/apps/publish/faq/get-started-with-the-microsoft-store>

2. **Planificación del proyecto**
3. **Información y desarrollo**
4. **Diseño**
5. **Implementación**
6. **Prueba**

7. **Mantenimiento**
8. **Conclusiones**