



Integrantes:

Amaro Rojas  
Santiago Aguirre  
María Verdecia  
Felipe Pérez  
Manuel Quezada

Asignatura:

Bases de Datos

Sección:

302

Abril 2025

## Índice de Contenidos

<b>Introducción .....</b>	<b>3</b>
<b>Objetivo General .....</b>	<b>3</b>
<b>Objetivos Específicos .....</b>	<b>4</b>
<b>Bases de Datos .....</b>	<b>5</b>
<b>Backend (Servidor y lógica del sistema) .....</b>	<b>6</b>
<b>Frontend (Interfaz de usuario) .....</b>	<b>6</b>
<b>Autenticación y Seguridad .....</b>	<b>7</b>
<b>Conclusión .....</b>	<b>9</b>
<b>Bibliografía .....</b>	<b>10</b>

# Introducción

En los últimos años, la industria de los videojuegos ha experimentado una transformación significativa a causa del auge del comercio electrónico. Este cambio ha obligado a muchas tiendas físicas a digitalizar sus operaciones para mantenerse competitivas. Las grandes cadenas han logrado adaptarse gracias a sus mayores recursos económicos y tecnológicos. Sin embargo, las tiendas pequeñas enfrentan barreras significativas para ingresar al mercado digital, ya que el desarrollo de plataformas propias resulta costoso y complejo. En este escenario, la empresa DEBEDE ha identificado una necesidad común entre estas tiendas: contar con una solución digital accesible que les permita ofrecer sus productos en línea de manera eficiente. El problema no es la ausencia de plataformas de venta online, sino que muchas tiendas más pequeñas de videojuegos no tienen los recursos técnicos (en algunos casos ni los económicos) para desarrollar una herramienta que les permita competir en igualdad de condiciones con las grandes cadenas. Eso reduce tanto su visibilidad en el mundo digital como su capacidad de captar y fidelizar a clientes online. El objetivo de EpsilonTek y este informe es idear e implementar una plataforma low cost que les permita a estas tiendas pequeñas de videojuegos gestionar, ordenar y vender sus juegos online. De forma que no solo sean parte del comercio electrónico, sino que puedan hacerle frente a las grandes cadenas de la industria.

Dentro de nuestros objetivos, se contempla el relevamiento de necesidades comunes de las pequeñas tiendas de videojuegos en relación con la venta online. Con estos datos en la mano, se busca diseñar una estructura de base de datos que permita mantener de forma eficiente productos, categorías de juegos, tiendas y usuarios. También, se desarrollarán funciones que resulten clave para la carga, listado y administración de los videojuegos por parte de los administradores de cada tienda. Asimismo, se plantea llevar adelante una interfaz de usuario sencilla para que los clientes puedan navegar, buscar y comprar sus juegos preferidos de forma sencilla, cómoda y segura. Por último, se someterá a una prueba de usuario para validar si el sistema propuesto es útil para los fieles jugadores de videojuegos.

## Objetivo General

Desarrollar el diseño lógico de una base de datos relacional que sustente el funcionamiento de una plataforma web destinada a la venta de videojuegos por parte de tiendas independientes, considerando los requerimientos funcionales establecidos por la empresa DEBEDE, así como la implementación de funcionalidades adicionales mediante operaciones SQL y estructuras asociadas.

## Objetivos Específicos

- Diseñar un Modelo Entidad-Relación (MER) que represente adecuadamente las entidades, atributos y relaciones necesarias para cumplir con los requerimientos del sistema.
- Traducir el MER a un Modelo Relacional (MR) compatible con PostgreSQL, definiendo claves primarias, foráneas y tipos de datos adecuados.
- Implementar un conjunto de sentencias SQL que permitan la creación y poblamiento de la base de datos.
- Desarrollar procedimientos almacenados y triggers para auditar acciones dentro del sistema.
- Proponer una arquitectura general que permita la interacción entre usuarios y la plataforma, considerando roles diferenciados (administrador, jefe de tienda, cliente).
- Diseñar y, opcionalmente, implementar un módulo web que realice operaciones CRUD sobre las entidades principales del sistema.
- Evaluar y comparar diferentes tecnologías existentes que puedan aplicarse a este tipo de plataformas.

Con el crecimiento del comercio electrónico, han surgido múltiples plataformas dedicadas a la venta y distribución digital de videojuegos. Estas plataformas permiten a los usuarios acceder a catálogos de títulos organizados por género, popularidad, ubicación y otros criterios, además de ofrecer funcionalidades como rankings, valoraciones, listas de deseos y sistemas de compra en línea. A continuación, se presentan algunas de las plataformas más relevantes en el contexto de este proyecto:

- Weplay

Weplay es una tienda chilena especializada en la venta de videojuegos, consolas y accesorios. Aunque su enfoque ha sido principalmente físico, ha desarrollado una tienda online robusta que permite a los usuarios comprar productos, revisar catálogos por categoría, y acceder a descuentos especiales. Es un ejemplo relevante en cuanto a organización del catálogo y facilidad de navegación.

- G2A

G2A es un marketplace global para la venta de claves de videojuegos digitales. La plataforma permite a vendedores autorizados publicar sus productos y a los compradores adquirirlos utilizando distintos medios de pago. G2A destaca por su sistema de filtrado por ubicación, género y popularidad, así como por su sistema de reputación y valoraciones de productos, algo que también se busca implementar en la propuesta de DEBEDE.

- Todojuegos

Plataforma chilena que, al igual que Weplay, combina tienda física con un portal web de ventas. Permite explorar videojuegos por categoría y marca, además de ofrecer una sección de “más vendidos”, lo que inspira la funcionalidad de ranking requerida en este proyecto.

- Hobbiegames

Sitio especializado en productos geek, que también ha incursionado en la venta de videojuegos. Su página permite visualizar productos con imágenes, descripciones y precios. Aunque no es tan avanzado en términos de personalización del contenido, su simplicidad puede inspirar decisiones de diseño amigables para usuarios menos técnicos.

Estas plataformas sirven como referencia para comprender las funcionalidades estándar y avanzadas que los usuarios esperan al interactuar con una tienda de videojuegos en línea. El proyecto propuesto busca integrar muchas de estas características, adaptándolas al contexto de múltiples tiendas pequeñas utilizando una sola plataforma colaborativa.

Para el desarrollo de una plataforma web que gestione videojuegos, usuarios y compras, es fundamental seleccionar tecnologías que aseguren eficiencia, escalabilidad y facilidad de mantenimiento. A continuación se comparan algunas de las tecnologías más utilizadas en cada componente clave del sistema: base de datos, backend y frontend.

## Bases de Datos

- PostgreSQL

Es una base de datos relacional de código abierto muy robusta, ideal para sistemas complejos como el de DEBEDE. Ofrece soporte para claves foráneas, integridad referencial, procedimientos almacenados, triggers, y tipos de datos avanzados. Será utilizada en este proyecto por su compatibilidad con SQL y sus capacidades de auditoría.

- MySQL

Similar a PostgreSQL en muchos aspectos, pero con menor flexibilidad en la creación de estructuras avanzadas. Es ampliamente usada, pero PostgreSQL es preferido cuando se busca cumplimiento estricto de reglas relacionales.

- MongoDB

Una base de datos NoSQL orientada a documentos. Ideal para proyectos muy flexibles y con estructuras variables, pero menos adecuada para sistemas con muchas relaciones entre entidades, como este proyecto.

## Backend (Servidor y lógica del sistema)

- Node.js (con Express.js)

Plataforma basada en JavaScript que permite construir APIs de forma rápida y escalable. Es ligera, fácil de conectar con bases de datos y ampliamente usada en aplicaciones modernas.

- Python (con Django o Flask)

Django es un framework de alto nivel que incluye muchas funcionalidades predefinidas para autenticación, administración y gestión de datos. Flask es más ligero y flexible, ideal para proyectos a medida. Python es una excelente opción por su claridad y comunidad activa.

- PHP (con Laravel)

Laravel es un framework potente para PHP que facilita la creación de sistemas complejos con arquitectura MVC. Aunque menos usado actualmente en startups modernas, sigue siendo una alternativa viable.

## Frontend (Interfaz de usuario)

- React

Biblioteca JavaScript desarrollada por Meta. Ideal para interfaces reactivas y dinámicas. Funciona muy bien con APIs RESTful y es ampliamente adoptada por la comunidad.

- Vue.js

Framework progresivo que combina simplicidad con potencia. Es muy intuitivo para desarrolladores principiantes y excelente para proyectos medianos.

- Angular

Framework completo y robusto mantenido por Google. Muy útil en aplicaciones grandes, pero tiene una curva de aprendizaje más alta.

## Autenticación y Seguridad

- JWT (JSON Web Tokens)

Método moderno y seguro para autenticar usuarios mediante tokens en el frontend. Ideal para sistemas distribuidos como el que se propone.

- OAuth 2.0

Protocolo más complejo, pero útil si se planea integrar con servicios externos como Google o Facebook.

Tecnologías elegidas para el proyecto

Para este proyecto, se ha decidido utilizar las siguientes tecnologías:

- Base de datos: PostgreSQL
- Backend: Python con Flask (por su simplicidad y adaptabilidad)
- Frontend: React (si se implementa el CRUD visual)
- Autenticación: JWT para login/registro de usuarios

Estas herramientas permitirán cumplir con todos los requerimientos funcionales propuestos, manteniendo una estructura clara, modular y escalable.

El sistema propuesto consiste en una plataforma web que permita a múltiples tiendas de videojuegos publicar sus productos, categorizarlos y venderlos a través de un portal común. Los usuarios podrán registrarse en la plataforma, navegar el catálogo, agregar productos a su carrito o lista de deseos, realizar compras y dejar valoraciones. Todo esto será gestionado por una base de datos relacional que asegura integridad, seguridad y trazabilidad de las operaciones.

Roles del sistema

- Administrador:

Tiene acceso completo al sistema. Puede realizar operaciones CRUD (crear, leer, actualizar, eliminar) sobre todas las entidades: usuarios, tiendas, productos, compras, valoraciones, etc. También puede revisar reportes y rankings.

- Jefe de tienda:

Representa a una tienda. Puede subir videojuegos con su información (nombre, descripción, stock, precio, URL de imagen), editarlos, eliminarlos y ver sus estadísticas de ventas. Solo tiene acceso a los productos de su propia tienda.

- Cliente:

Usuario final que puede navegar el catálogo, usar filtros (por ubicación, categoría, tipo de videojuego), añadir productos a su carrito de compras o a la lista de deseos, comprar videojuegos (si hay stock disponible) y dejar una valoración única por videojuego.

Funcionalidades clave (requerimientos cubiertos)

- Registro/Login con autenticación segura (JWT).
- Gestión de productos por tienda (CRUD limitado para jefes de tienda).
- Carrito de compras y lista de deseos personal por cliente.
- Proceso de compra con generación de boleta y selección de medio de pago.
- Filtrado de productos por ubicación geográfica y tipo.
- Valoraciones de videojuegos (una por usuario).
- Rankings: más vendidos, más deseos.
- Control de stock para validar la compra.
- Administración global del sistema (CRUD completo para administrador).

Distribución lógica entre frontend, backend y base de datos

- Frontend:

Interfaz amigable para clientes y jefes de tienda. Permite navegar, comprar y administrar productos. Utiliza React para consumo de la API.

- Backend:

Aplicación desarrollada en Flask que actúa como intermediario entre el frontend y la base de datos. Maneja autenticación, lógica de negocio, validaciones y exposición de endpoints.

- Base de datos (PostgreSQL):

Gestiona todas las entidades del sistema (usuarios, roles, videojuegos, tiendas, categorías, compras, valoraciones, etc.), incluyendo triggers y procedimientos almacenados para auditar acciones relevantes.



### Ejemplo de flujo: Compra de un videojuego

1. El cliente inicia sesión y navega por la plataforma.
2. Agrega un videojuego al carrito.
3. El sistema valida que haya stock.
4. El cliente elige un medio de pago y confirma la compra.
5. Se genera una boleta.
6. El stock del producto se actualiza.
7. El videojuego pasa a estar disponible en su cuenta para valoración.

## Conclusión

Este diseño modular y escalable asegura que el sistema pueda evolucionar en el tiempo y adaptarse a nuevos requerimientos o integraciones futuras.

El presente informe ha permitido establecer una base sólida para el desarrollo de una plataforma web orientada a la venta de videojuegos por parte de tiendas independientes. A partir del análisis del contexto actual y de plataformas similares, se identificaron las necesidades y funcionalidades clave que debe cubrir el sistema, tanto desde la perspectiva del usuario final como desde la administración del negocio.

La propuesta presentada aborda de forma integral los desafíos técnicos y funcionales, incorporando un diseño lógico de base de datos basado en PostgreSQL, la diferenciación clara de roles de usuario y la planificación de funcionalidades como el carrito de compras, las valoraciones, el control de stock y los rankings. Asimismo, se seleccionaron tecnologías modernas y adecuadas para cada componente del sistema, considerando su robustez, escalabilidad y comunidad de soporte.

Este proyecto no solo representa una oportunidad de aprendizaje en diseño y modelado de bases de datos, sino también un ejercicio práctico en la planificación de soluciones reales y aplicables al mercado. Las próximas etapas del trabajo contemplan la implementación del Modelo Entidad-Relación (MER), la escritura del código SQL necesario para levantar la base de datos, y la eventual conexión con una aplicación web que permita la interacción efectiva de los distintos tipos de usuarios.

# Bibliografía

## Bases de Datos

- Denzer, P. (2002, 23 octubre). *PostgreSQL*. Recuperado de <http://profesores.elo.utfsm.cl/~agv/elo330/2s02/projects/denzer/informe.pdf>
- Iskrend. (2024, 28 febrero). *What is PostgreSQL and How Is It Used in Web Hosting?* AwardSpace.com. <https://www.awardspace.com/kb/what-is-postgresql/>
- PostgreSQL Global Development Group. (2025). PostgreSQL Documentation. <https://www.postgresql.org/docs>
- Erickson, J. (2024, 29 agosto). *MySQL: Understanding What It Is and How It's Used*. <https://www.oracle.com/mysql/what-is-mysql/>
- Pirtle, Mitch (2011, 3 marzo), *MongoDB for Web Development* (1st ed.), Addison-Wesley Professional, p. 360

## Estado del Arte

- G2A. (2025). G2A Marketplace. <https://www.g2a.com>
- Weplay. (2025). Tienda Weplay. <https://www.weplay.cl>
- Todojuegos. (2025). Tienda Todojuegos Chile. <https://www.todojuegos.cl>
- Hobbiegames. (2025). Hobbiegames Chile. <https://www.hobbiegames.cl>

## Tecnologías y herramientas

- Flask. (2025). Flask Documentation. <https://flask.palletsprojects.com/>
- Singh, R. (2024, 2 enero). *What is Flask and use cases of Flask?* DevOpsSchool.com. <https://www.devopsschool.com/blog/what-is-flask-and-use-cases-of-flask/>
- React. (2025). React – A JavaScript library for building user interfaces. <https://reactjs.org/>
- JWT.io. (2025). JSON Web Tokens. <https://jwt.io/introduction/>
- Mozilla Developer Network. (2025). OAuth 2.0 Overview. <https://developer.mozilla.org/en-US/docs/Web/Security/OAuth>