



Taller: Backend - Node JS

Objetivo: Definir el anteproyecto de una aplicación web que implemente autenticación de usuarios, gestión de datos, generación de reportes y consumo de una API externa.

Proyecto: Velvet Room

Velvet Room será una plataforma de eCommerce diseñada para satisfacer las necesidades de los consumidores de videojuegos y de merchandising de la cultura geek en general.

Debido a la falta de un espacio centralizado donde los usuarios puedan comprar videojuegos, accesorios gaming, ropa, figuras coleccionables y demás artículos que coincidan con su fandom favorito, se decidió por crear una plataforma en la que se pueda hacer compra (y también venta) de artículos mencionados ya sean oficiales, nuevos, de segunda mano o fan made.

Además, contaría con la inclusión de vendedores independientes para que puedan también ofrecer sus productos a una audiencia global, no solo teniendo material oficial.

Objetivos Terminales:

1. Crear una plataforma intuitiva y segura para la compra y venta de productos relacionados con videojuegos y cultura geek.
2. Implementar autenticación robusta y gestión de roles para garantizar la seguridad del sistema.
3. Ofrecer herramientas de generación de reportes para administradores y vendedores.

4. Integrar una API externa para mejorar la experiencia del usuario, en este caso para el procesamiento de los pagos.

Autenticación de Usuarios

Para la seguridad al momento de registrar y loguear a los usuarios, se implementará JWT (JSON Web Tokens) como método de autenticación debido a su escalabilidad y portabilidad.

Los tokens se generarán al iniciar sesión y se almacenarán en el frontend (localStorage) para su uso en solicitudes autenticadas.

Roles y Permisos de Usuario

- **Cliente:**
 - Puede navegar por el catálogo de productos.
 - Puede agregar productos al carrito y realizar compras.
 - Puede ver su historial de compras.
- **Vendedor:**
 - Puede registrar nuevos productos en el catálogo.
 - Puede gestionar sus propios productos (editar, eliminar).
 - Puede ver reportes de ventas de sus productos.
- **Administrador:**
 - Puede gestionar todos los productos, usuarios y reportes (agregar, eliminar o modificar).

Módulos:

1. Gestión de Productos:

Entidades Principales

Producto: Representa un artículo a la venta (videojuego, figura coleccionable, ropa geek, etc.).

Atributos: ID, nombre, descripción, precio, stock, categoría, imágenes, vendedor.

Operaciones CRUD

Crear Producto (POST /api/products): Solo accesible para vendedores y administradores.

Leer Productos (GET /api/products): Accesible para todos los usuarios.

Actualizar Producto (PUT /api/products/:id): Solo accesible para el vendedor que lo creó o administradores.

Eliminar Producto (DELETE /api/products/:id): Solo accesible para el vendedor que lo creó o administradores.

Endpoints Necesarios

/api/products: Listar todos los productos.

/api/products/:id: Obtener detalles de un producto específico.

/api/products/categories: Listar categorías disponibles.

2. Gestión de Compras

Entidades Principales

Pedido: Representa una compra realizada por un cliente.

Atributos: ID, fecha, estado (pendiente, completado), total, productos, cliente.

Carrito: Representa los productos seleccionados por un cliente antes de realizar la compra.

Atributos: ID, productos, cantidad, cliente.

Operaciones CRUD

Crear Pedido (POST /api/orders): Accesible para clientes.

Ver Historial de Pedidos (GET /api/orders): Accesible para clientes y administradores.

Actualizar Estado de Pedido (PUT /api/orders/:id): Accesible para administradores.

Endpoints Necesarios

/api/orders: Listar pedidos (filtrados por cliente o administrador).

/api/cart: Gestionar el carrito de compras.

3. Gestión de Usuarios

Entidades Principales

Usuario: Representa a un cliente, vendedor o administrador.

Atributos: ID, nombre, email, contraseña, rol, dirección, historial de compras.

Operaciones CRUD

Registrar Usuario (POST /api/auth/register): Accesible para todos.

Iniciar sesión (POST /api/auth/login): Accesible para todos.

Obtener Datos del Usuario (GET /api/users/me): Accesible para usuarios autenticados.

Endpoints Necesarios

/api/auth/register: Registrar un nuevo usuario.

/api/auth/login: Iniciar sesión y obtener token JWT.

/api/users/me: Obtener datos del usuario autenticado.

Módulo de Reportes:

Tipo de Reportes Generados

- Reporte de Ventas Globales:
Total de ventas por mes/año.

Productos más vendidos.

- Reporte de Actividad de Usuarios:

Número de usuarios registrados por mes.

Usuarios más activos (compradores frecuentes).

- Reporte de Inventarios:

Productos con bajo stock.

Categorías más populares.

Formatos de Salida

Gráficos (barras, líneas, pasteles) para visualización interactiva.

Tablas exportables a PDF o Excel.

Endpoints Necesarios

/api/reports/sales: Generar reporte de ventas.

/api/reports/users: Generar reporte de actividad de usuarios.

/api/reports/inventory: Generar reporte de inventarios.

Integración con una API Externa

API Seleccionada

Pensamos a partir de los objetivos del anteproyecto, que Stripe API sería una opción ideal para el procesamiento de pagos.

[Authentication | Stripe API Reference](#)

Descripción de los Datos

Datos enviados:

Monto total de la compra.

Información del cliente (nombre, email, dirección).

Lista de productos comprados.

Datos recibidos:

Confirmación del pago.

ID de transacción.

Estado del pago (éxito, fallido, pendiente).

Endpoints Necesarios

/api/payments: Procesar pagos utilizando Stripe.

Diagrama de Casos de Uso de Contexto

Visual Paradigm Professional (David Henao Salazar (Universidad Icesi))

