



ceutec
de unitec®

PROYECTO

SISTEMA DE PUNTO DE VENTA PARA VETERINARIA

Rony Diaz

Kenenn Lara

& Jose Molina





Introducción

Este proyecto consiste en el desarrollo de un Sistema de Punto de Venta (POS) para una veterinaria, utilizando tecnologías modernas como Node.js, Express, Angular y MySQL. El sistema permite gestionar ventas, productos, clientes y facturación, integrando un backend seguro con autenticación JWT y un frontend. El objetivo principal fue aplicar buenas prácticas de programación, arquitectura por capas y trabajo colaborativo mediante Git.

Objetivos del Proyecto

- Implementar una API REST con Node.js y Express.
- Aplicar autenticación segura mediante JWT.
- Construir un frontend en Angular para consumir la API.
- Utilizar MySQL como base de datos relacional.
- Practicar buenas prácticas: validaciones, manejo de errores, modularidad.
- Trabajar en equipo utilizando Git y Git Flow.



Objetivos del Proyecto



Arquitectura General



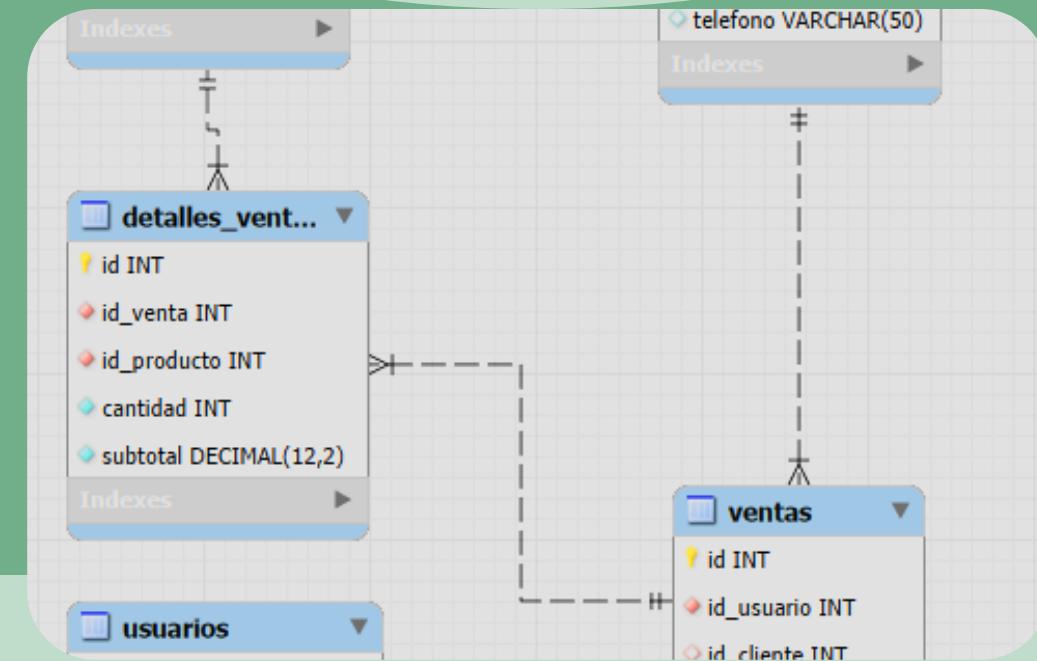
El sistema se divide en dos capas principales:

Backend (Node.js + Express): maneja la lógica, seguridad, rutas y conexión a MySQL.

Frontend (Angular): interfaz para login, registro de ventas y visualización de facturas.

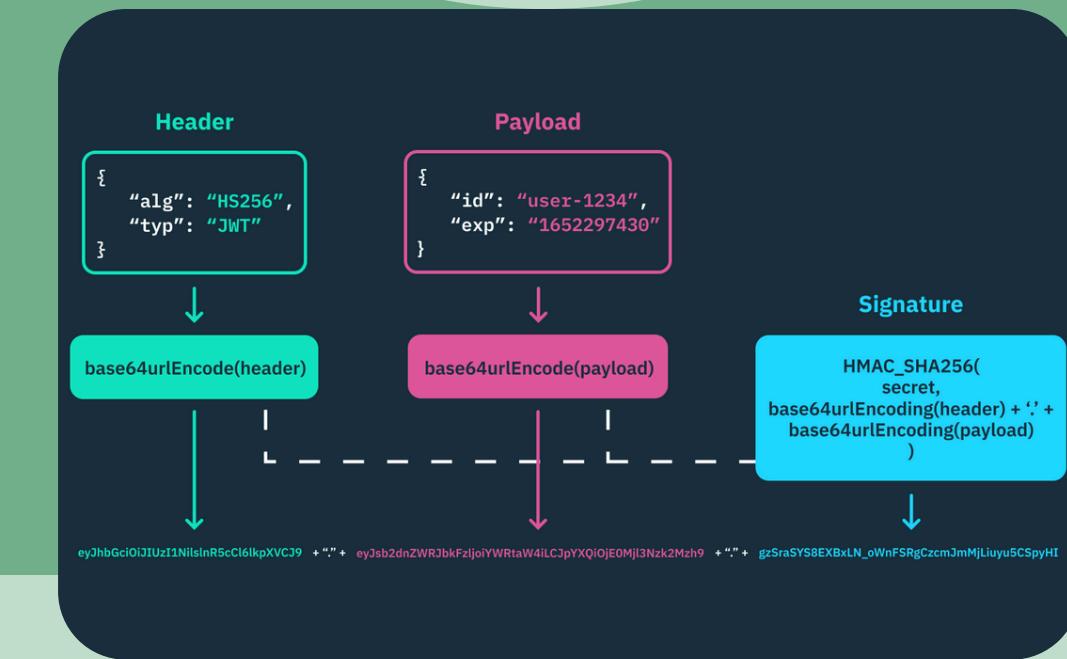
La comunicación se realiza mediante una API REST, donde Angular envía solicitudes al backend y este interactúa con la base de datos.

Modelo de Datos



El sistema utiliza un modelo relacional compuesto por las entidades: Usuario, Producto, Cliente, Venta y DetalleVenta. Las relaciones permiten registrar ventas completas, asociar productos y clientes, y mantener integridad en los datos. Este diseño facilita consultas rápidas y escalabilidad futura.

Uso de JWT



El sistema implementa JSON Web Tokens para autenticar usuarios. El flujo funciona así:

1. El usuario inicia sesión y el backend valida sus credenciales.
2. Si son correctas, se genera un token firmado con información básica del usuario.
3. El frontend almacena el token y lo envía en cada petición protegida.
4. El middleware del backend valida el token antes de permitir acceso.
5. Este mecanismo garantiza seguridad y control de acceso.

Principios de Programación Aplicados

Se aplicaron principios como separación de capas, modularidad, validaciones básicas, manejo de errores y uso de middleware. Además, se implementaron prácticas de seguridad como encriptación de contraseñas, uso de variables de entorno y protección de rutas mediante JWT.

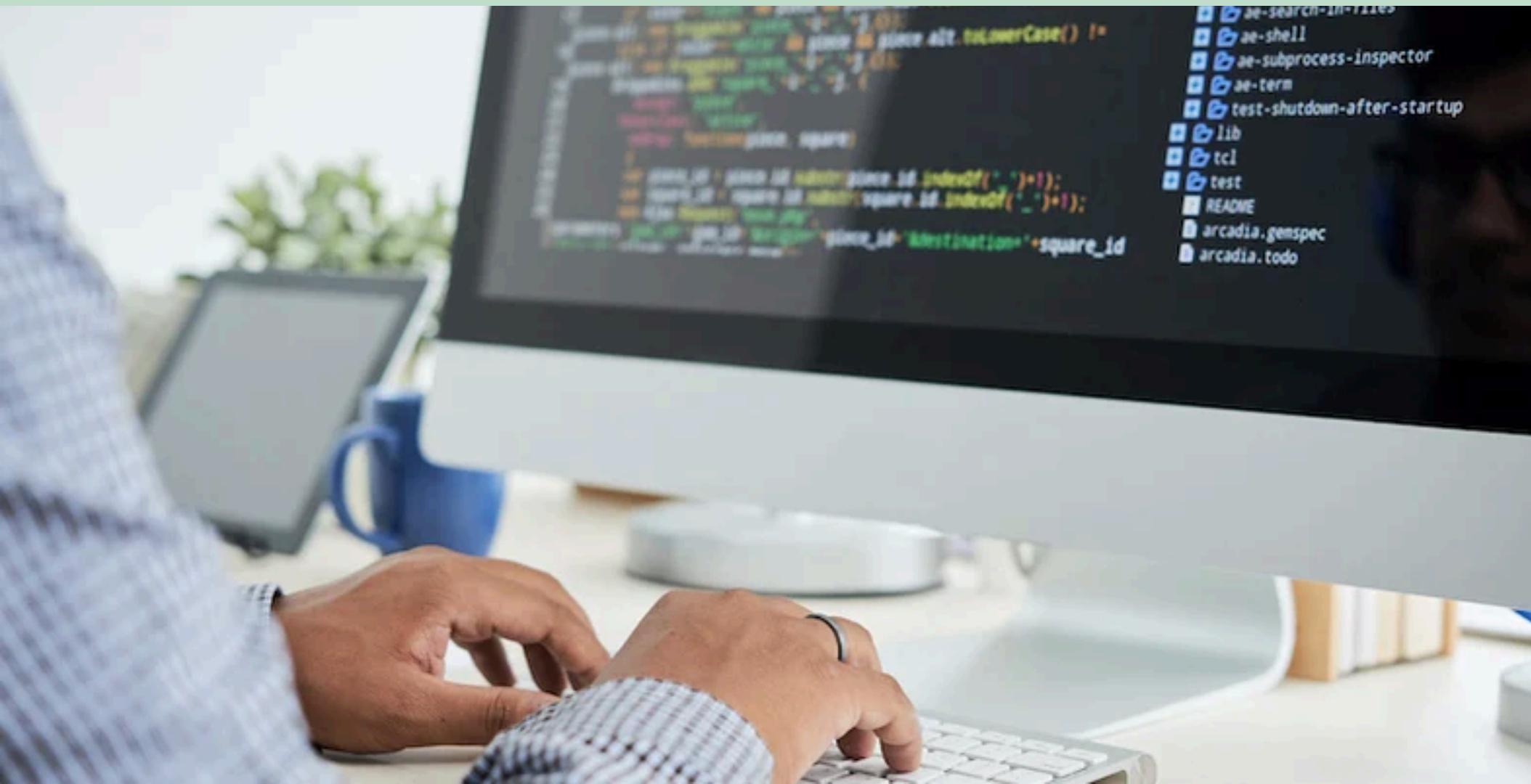
Git Flow

Aunque no se siguió de forma estricta, se trabajó con una estructura inspirada en Git Flow. Se utilizaron ramas como main, develop y feature para organizar el desarrollo. Esto permitió dividir tareas, integrar cambios y mantener un flujo de trabajo colaborativo, aunque con algunos retos en la sincronización de ramas.



Complicaciones Encontradas

Durante el desarrollo surgieron dificultades como problemas en Angular al consumir la API, errores en el envío del token durante el login y desorden en el flujo de Git.



Solución

Estas complicaciones se resolvieron mediante pruebas con Postman, reorganización del código y ajustes en los servicios de Angular.



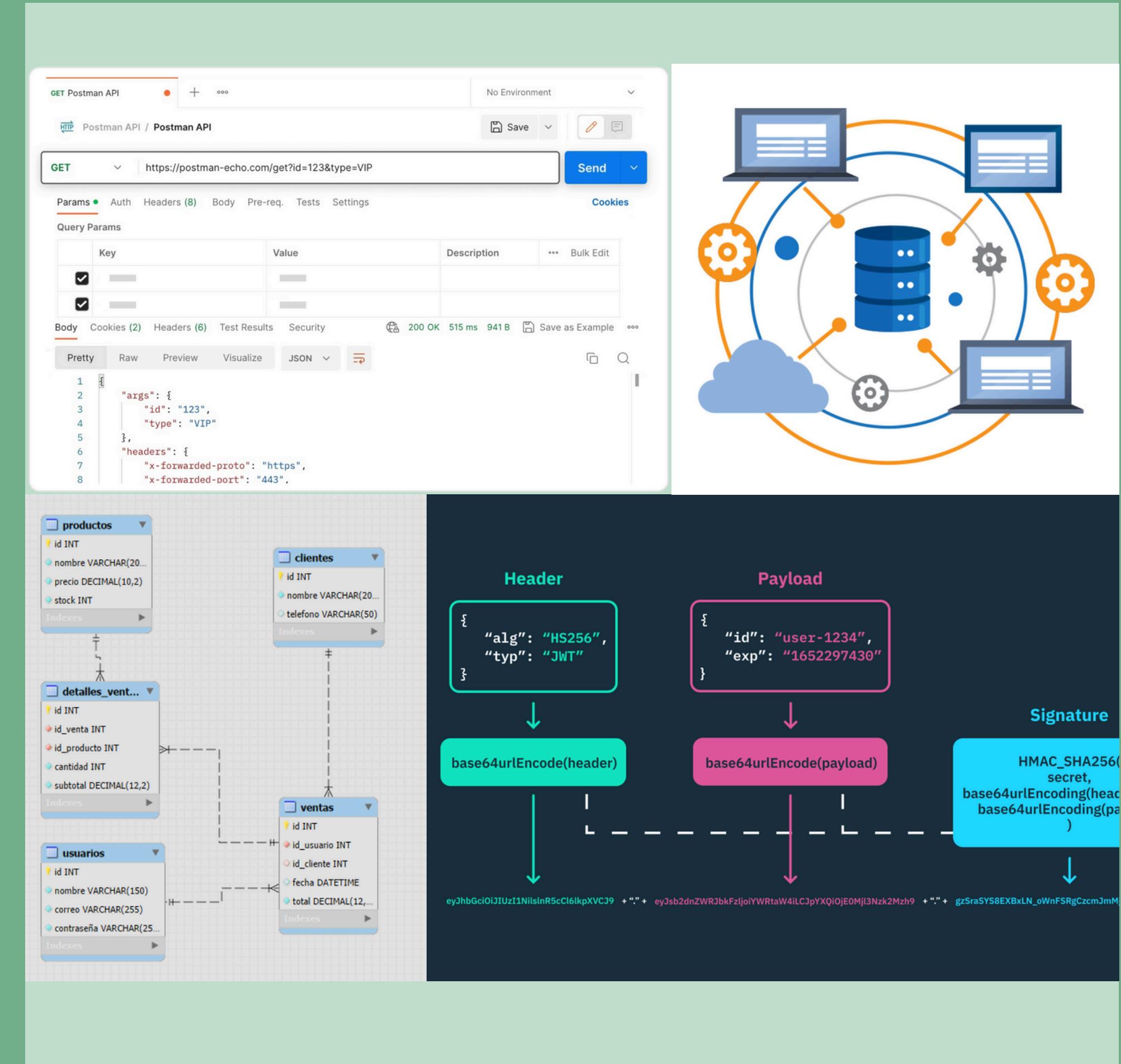
Desarrollo

También hubo inconsistencias entre los endpoints del backend y los servicios del frontend.



Conclusión

El proyecto permitió integrar conocimientos clave del desarrollo web moderno, combinando backend, frontend, seguridad y bases de datos. A pesar de los desafíos técnicos, el equipo logró construir un sistema funcional, modular y escalable. La experiencia fortaleció habilidades en programación, arquitectura, autenticación y trabajo colaborativo, dejando una base sólida para futuras mejoras.



Muchas gracias por su atención!

