



**ceutec**  
de unitec

# SISTEMA DE PUNTO DE VENTA PARA VETERINARIA

UNIVERSIDAD TECNOLÓGICA  
DE HONDURAS

T51 – DESARROLLO DE APLICACIONES WEB I

ING. JONIE MIRALDA

INTEGRANTES:  
RONY DÍAZ  
KENENN LARA  
JOSE MOLINA



# Informe Técnico – Sistema de Punto de Venta para Veterinaria

## Principios de Programación Aplicados

Durante el desarrollo del Sistema de Punto de Venta (POS) se aplicaron principios fundamentales de programación y arquitectura de software, con el objetivo de construir una aplicación modular, mantenible y segura.

### 1.1 Separación de capas

- El backend se estructuró siguiendo una arquitectura por responsabilidades:
- app.js: configuración principal de Express
- routes.js: definición de endpoints
- middleware.js: autenticación y manejo básico de errores
- db.js: conexión centralizada a MySQL
- Esta separación permitió mantener el código organizado y facilitar futuras ampliaciones.

### 1.2 Validaciones

- Se aplicaron validaciones esenciales como:
- Verificación de credenciales en el login
- Validación de existencia de productos
- Comprobación de stock antes de registrar una venta

### 1.3 Seguridad

Se implementaron prácticas como:

- Autenticación mediante JWT
- Encriptación de contraseñas con bcrypt
- Manejo de tokens mediante encabezados HTTP
- Uso de variables de entorno para credenciales sensibles



## 2. Uso de JWT

El sistema utiliza JSON Web Tokens (JWT) como mecanismo de autenticación y autorización.

### 2.1 Flujo de autenticación

#### Login

El usuario envía correo y contraseña.

El backend:

1. Verifica que el usuario exista
2. Compara la contraseña con el hash almacenado
3. Genera un token JWT con:
  - id del usuario
  - correo
  - expiración

El token se firma con JWT\_SECRET.

#### Almacenamiento del token

El frontend lo guarda en localStorage o sessionStorage.

#### Acceso a rutas protegidas

Cada petición incluye:  
Authorization: Bearer <token>

El middleware valida el token y permite el acceso.



### 3. Flujo de Git Flow Aplicado

Aunque no se siguió Git Flow de manera estricta, se trabajó con una estructura inspirada en este flujo.

#### 3.1 Ramas utilizadas

- main → versión estable
- develop → integración
- feature/\* → funcionalidades específicas

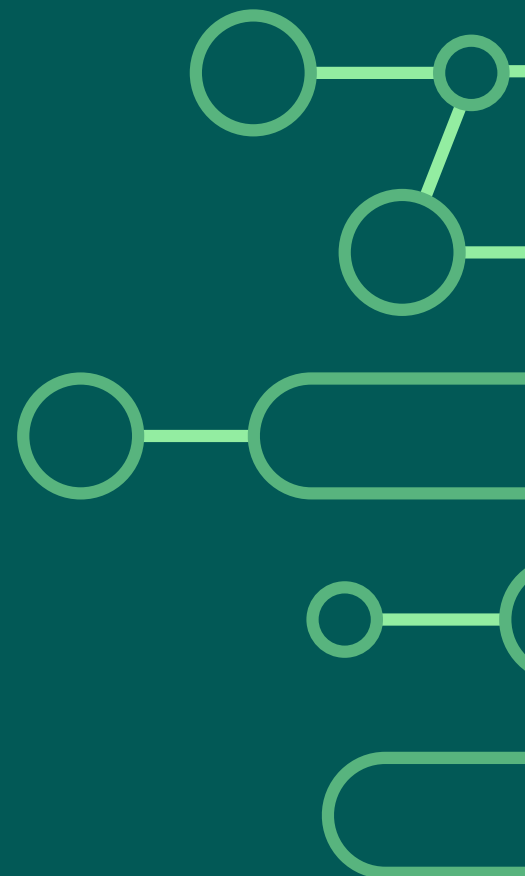
#### 3.2 Flujo ideal

- Crear rama feature/\*
- Integrar en develop
- Preparar versión en release/\*
- Publicar en main
- Corregir errores urgentes en hotfix/\*

#### 3.3 Situación real

El equipo enfrentó dificultades como:

- Cambios mezclados en develop
- Falta de sincronización entre ramas





## 4. Complicaciones Encontradas y Soluciones

### 4.1 Problemas con Angular, encontramos Dificultades con:

- Servicios para consumir la API
- Envío del token
- Actualización del estado tras el login

Solución: reorganización de servicios, uso correcto de HttpClient y configuración de headers.

### 4.2 Proceso de Login en el Frontend

El token no se enviaba correctamente, generando errores 401.

Solución:

- Corrección del header Authorization
- Ajuste del almacenamiento del token
- Implementación de un interceptor

### 4.3 Falta de un Git Flow ordenado

Los cambios se mezclaban y era difícil rastrear responsabilidades.

Solución:

- Reorganización de ramas
- Convención mínima de commits
- Merges controlados

### 4.4 Integración Backend–Frontend

Inconsistencias entre endpoints y datos esperados.

Solución:

- Revisión de rutas
- Ajuste de servicios Angular
- Pruebas con Postman antes de integrar



## Conclusión

El desarrollo del Sistema de Punto de Venta para la veterinaria permitió aplicar de manera práctica los conceptos fundamentales del desarrollo web moderno, integrando una API REST construida con Express, un sistema de autenticación seguro mediante JWT, una arquitectura modular orientada a la separación de responsabilidades y un frontend funcional desarrollado en Angular para consumir los servicios del backend.

A lo largo del proyecto se fortalecieron habilidades técnicas relacionadas con la comunicación entre capas, la gestión de datos en MySQL y la implementación de buenas prácticas de programación. Además, el trabajo colaborativo mediante Git, aunque presentó desafíos, contribuyó a la comprensión de flujos de trabajo reales en entornos de desarrollo.

A pesar de las complicaciones encontradas, especialmente en la integración entre frontend y backend y en la configuración del proceso de login, el equipo logró resolver los problemas y entregar un sistema funcional, escalable y listo para futuras mejoras, demostrando un avance significativo en el dominio de tecnologías web y metodologías de desarrollo.