

DOCUMENTACIÓN COMPLETA - API PET-SHOP INVENTORY

PROBLEMÁTICA RESUELTA

Contexto del Negocio:

Un Pet-Shop necesita un sistema de gestión de inventario que permita controlar productos, categorías, proveedores, movimientos de stock y registro de mascotas de clientes.

Problemas Específicos Solucionados:

1. Control manual de inventario → Sistema automatizado
2. Falta de trazabilidad → Historial completo de movimientos
3. Sin roles definidos → 3 niveles de acceso (admin, empleado, cliente)
4. Stock desactualizado → Alertas de stock bajo y actualización automática
5. Datos desorganizados → Estructura relacional con categorías y proveedores

ARQUITECTURA DE ENTIDADES

1. USUARIOS (User)

Propósito: Control de acceso y autenticación

2. PRODUCTOS (Product)

Propósito: Gestión completa del inventario

3. CATEGORÍAS (Category)

Propósito: Organización de productos

4. MOVIMIENTOS DE INVENTARIO (InventoryMovement)

Propósito: Auditoría y trazabilidad

5. PROVEEDORES (Supplier)

Propósito: Gestión de cadena de suministro

6. MASCOTAS (Pet)

Propósito: Registro de clientes y sus mascotas

SISTEMA DE SEGURIDAD

- Autenticación JWT
- Sistema de Roles (RBAC)
- Guards Implementados

SISTEMA DE VALIDACIÓN (DTOs)

- DTOs de Usuarios
- DTOs de Productos
- DTOs de Auth
- DTOs de Movimientos

FLUJOS PRINCIPALES

1. Gestión de Inventario

El empleado registra la entrada = Stock se actualiza automáticamente

El sistema de alerta de Stock bajo = El empleado realiza el pedido

La venta registra la salida = El Stock disminuye automáticamente

2. Flujo de Autenticación

El usuario se registra = Hash de la contraseña = Y la creación de la BD

El usuario hace login = Se validan las credenciales = Y así se genera el JWT

El acceso a rutas = Middleware verifica el token = Y los Guards validan el rol

3. Auditoría de Movimientos

Cada movimiento se registra = Incluye el usuario, la fecha y el motivo

El historial consultable por producto = Crea una trazabilidad completa

Movimientos inmutables = Crea una buena integridad en los datos

CARACTERÍSTICAS TÉCNICAS

Arquitectura

- Framework: NestJS (TypeScript)
- Base de Datos: PostgreSQL
- ORM: TypeORM
- Autenticación: JWT + Passport

Características de Seguridad

- Contraseñas hasheadas con bcrypt
- Validación de datos con class-validator
- Protección contra inyección SQL
- Tokens JWT con expiración
- Control de acceso por roles

Escalabilidad

- Estructura modular
- Fácil agregar nuevas entidades
- Preparado para módulo de facturación futuro
- API RESTful estándar

VALOR AGREGADO

Para el Administrador:

- Control completo del sistema
- Reportes y auditorías
- Gestión de usuarios

Para el Empleado:

- Operaciones diarias de inventario
- Alertas preventivas de stock
- Registro de mascotas

Para el Cliente:

- Consulta de productos
- Registro de sus mascotas
- Experiencia personalizada