Sistema de Gestión de Inventario

Objetivo

Desarrollar una API REST para gestionar el inventario de una cadena de tiendas minoristas.

Requerimientos Funcionales

1. Gestión de Productos

- **GET /api/products**
  - Listar todos los productos
  - Filtros por: categoría, precio, stock
  - Paginación
- **GET /api/products/{id}**
  - Obtener detalle de un producto
- **POST /api/products**
  - Crear nuevo producto
  - Validación de datos obligatorios
- **PUT /api/products/{id}**
  - Actualizar producto existente
- **DELETE /api/products/{id}**
  - Eliminar producto

2. Gestión de Stock

- **GET /api/stores/{id}/inventory**
  - Listar inventario por tienda
- **POST /api/inventory/transfer**
  - Transferir productos entre tiendas
  - Validación de stock disponible
- **GET /api/inventory/alerts**
  - Listar productos con stock bajo

Modelos de Datos

Producto

json

```
{
  "id": "string",
```

```
  "name": "string",
  "description": "string",
  "category": "string",
  "price": "decimal",
  "sku": "string"
}
```

Inventario

json

```
{
  "id": "string",
  "productId": "string",
  "storeId": "string",
  "quantity": "integer",
  "minStock": "integer"
}
```

Movimiento

json

```
{
  "id": "string",
  "productId": "string",
  "sourceStoreId": "string",
  "targetStoreId": "string",
  "quantity": "integer",
  "timestamp": "datetime",
  "type": "enum(IN, OUT, TRANSFER)"
}
```

Requerimientos Técnicos

Base de Datos

- PostgreSQL o MongoDB
- Implementar índices para optimizar consultas frecuentes
- Manejo de transacciones para operaciones críticas

Testing

- Tests unitarios (80% cobertura mínima)
- Tests de integración para flujos críticos
- Tests de carga (500 requests/segundo)

Infraestructura

- Dockerfile y docker-compose
- Variables de entorno para configuración
- Documentación con OpenAPI/Swagger
- Logs estructurados (JSON)

Despliegue

- Instrucciones de despliegue en AWS/GCP/Azure/DigitalOcean, etc..
- Script de inicialización de base de datos
- Configuración de backups

Entregables

1. Código fuente en GitHub/GitLab invitar al proyecto a **jcantui@deacero.com**, **jfarias@deacero.com**, **etrejo@deacero.com**, **ogonzalez@deacero.com**, **jeguzman@deacero.com**
2. README con:
    - Instrucciones de instalación
    - Documentación de API
    - Decisiones técnicas
    - Diagrama de arquitectura
3. Colección de Postman/Insomnia
4. Scripts de despliegue
5. Compartir correo a **jcantui@deacero.com**, **jfarias@deacero.com**, **etrejo@deacero.com**, **ogonzalez@deacero.com**, **jeguzman@deacero.com** con los entregables

Criterios de Evaluación

1. Calidad y estructura del código
2. Manejo de errores y casos borde
3. Rendimiento y escalabilidad
4. Claridad de la documentación
5. Facilidad de despliegue

Dudas:

jcantui@deacero.com