



Taller de Desarrollo de API: Sistema de Gestión de Productos y Pedidos

Objetivo del Taller:

El objetivo de este taller es que los coders desarrollen una API RESTful que permita gestionar productos, categorías y pedidos. El propósito principal es que cada entidad tenga sus respectivos endpoints CRUD (Create, Read, Update, Delete) y que se sigan buenas prácticas de separación de responsabilidades. Además, se incluirán endpoints especiales que permitirán búsquedas avanzadas y filtros específicos.

Requisitos:

1. **Entidades:**
 - a. Product (Producto)
 - b. Category (Categoría)
 - c. Order (Pedido)
2. **Base de datos:**
 - a. Debe ser gestionada con MySQL.
 - b. Usa migraciones y seeders para poblar datos iniciales.
3. **Autenticación:**
 - a. Implementar JWT para proteger endpoints sensibles.
4. **Estructura del Proyecto:**
 - a. Utiliza el principio de **Responsabilidad Única** (S de SOLID). Cada controlador debe tener un propósito específico.
 - b. Aplica el patrón de **repositorios y servicios** para manejar la lógica de negocio y acceso a la base de datos.

Instrucciones Detalladas:

Paso 1: Crear la Entidad Product y sus Endpoints

1. **Lógica:**



- a. Un producto pertenece a una categoría. Debes diseñar el modelo de Product con las propiedades necesarias como nombre, descripción, precio, stock y categoría.
 - b. Asegúrate de manejar correctamente la relación con Category utilizando claves foráneas.
2. **Endpoints:**
- a. **GET** /api/v1/products – Obtener todos los productos.
 - i. Respuesta esperada: una lista de productos con información detallada.
 - b. **GET** /api/v1/products/{id} – Obtener un producto específico por su ID.
 - i. Respuesta esperada: los detalles de un solo producto.
 - c. **POST** /api/v1/products – Crear un nuevo producto.
 - i. Datos esperados: nombre, descripción, precio, stock, categoría.
 - d. **PUT** /api/v1/products/{id} – Actualizar un producto.
 - i. Datos actualizables: cualquier campo de un producto.
 - e. **DELETE** /api/v1/products/{id} – Eliminar un producto.

¡Importante! Antes de pasar a la siguiente entidad, asegúrate de que todos los endpoints de Product estén funcionando correctamente, incluyendo la relación con Category. Cada producto debe estar vinculado a una categoría existente.

Paso 2: Crear la Entidad Category y sus Endpoints

1. **Lógica:**
- a. La categoría es una entidad simple que sirve para organizar productos. Cada producto pertenece a una única categoría, pero una categoría puede tener múltiples productos.
 - b. Crea el modelo de Category con las propiedades: nombre y descripción.
2. **Endpoints:**
- a. **GET** /api/v1/categories – Obtener todas las categorías.
 - b. **GET** /api/v1/categories/{id} – Obtener una categoría específica por ID.
 - c. **POST** /api/v1/categories – Crear una nueva categoría.
 - i. Datos esperados: nombre y descripción.
 - d. **PUT** /api/v1/categories/{id} – Actualizar una categoría.
 - e. **DELETE** /api/v1/categories/{id} – Eliminar una categoría.



¡Tip! Usa seeders para poblar la base de datos con algunas categorías iniciales (ej. "Electrónica", "Hogar", "Ropa").

Paso 3: Crear la Entidad Order y sus Endpoints

1. **Lógica:**
 - a. Un pedido (Order) contiene uno o varios productos y está vinculado a un cliente.
 - b. Debes diseñar la entidad Order con propiedades como fecha, cliente (nombre, dirección, contacto), y un listado de productos.
 - c. Crea una relación entre Order y Product para poder agregar varios productos a un solo pedido.
2. **Endpoints:**
 - a. **GET** /api/v1/orders – Obtener todos los pedidos.
 - b. **GET** /api/v1/orders/{id} – Obtener los detalles de un pedido por ID.
 - c. **POST** /api/v1/orders – Crear un nuevo pedido.
 - i. Datos esperados: productos incluidos, información del cliente, fecha.
 - d. **PUT** /api/v1/orders/{id} – Actualizar un pedido.
 - e. **DELETE** /api/v1/orders/{id} – Eliminar un pedido.

¡Ojo! Cada vez que se crea un pedido, asegúrate de que los productos asociados se restan del inventario (stock).

Paso 4: Endpoints Especiales

1. **GET** /api/v1/products/search/{keyword} – Buscar productos por palabra clave.
 - a. **Descripción:** Este endpoint permitirá buscar productos por su nombre o descripción.
 - b. **Lógica:** Implementa una búsqueda que consulte el nombre y la descripción del producto para devolver coincidencias.
2. **GET** /api/v1/orders/customer/{customerName} – Obtener todos los pedidos de un cliente.
 - a. **Descripción:** Este endpoint devuelve todos los pedidos realizados por un cliente específico.



- b. **Lógica:** El cliente debe ser buscado por su nombre (o algún identificador) y se debe devolver una lista de pedidos.
- 3. **GET /api/v1/products/low-stock** – Listar productos con bajo stock.
 - a. **Descripción:** Mostrar todos los productos que tienen menos de una cantidad mínima de stock (ej. 5 unidades).
 - b. **Lógica:** Puedes establecer un umbral de stock bajo (por ejemplo, 5 unidades) y mostrar productos que necesiten ser reabastecidos.
- 4. **GET /api/v1/orders/date/{date}** – Listar pedidos realizados en una fecha específica.
 - a. **Descripción:** Este endpoint muestra todos los pedidos creados en una fecha específica.
 - b. **Lógica:** Filtra los pedidos por la fecha en que fueron creados.
- 5. **GET /api/v1/categories/{id}/products** – Listar productos de una categoría específica.
 - a. **Descripción:** Devuelve todos los productos que pertenecen a una categoría específica.
 - b. **Lógica:** Utiliza el ID de la categoría para listar todos los productos asociados a ella.

Consideraciones adicionales:

- **JWT y Seguridad:**
 - Asegúrate de que algunos endpoints sensibles, como la creación de pedidos o la eliminación de productos, estén protegidos con autenticación JWT.
- **Validación:**
 - Implementa validaciones para los campos (ej. precios no negativos, stock mayor que cero, nombre de producto no vacío).
- **Data Transfer Objects (DTOs):**
 - Usa DTOs para evitar que las entidades del dominio sean expuestas directamente en las respuestas de la API.
- **Swagger:**
 - Documenta tu API con Swagger para que los endpoints queden bien definidos y fáciles de explorar.

Este taller te ayudará a comprender cómo crear una API bien estructurada, dividiendo responsabilidades y asegurando que los endpoints estén bien diseñados y protegidos. Recuerda



trabajar en una entidad y sus endpoints completamente antes de pasar a la siguiente. ¡Buena suerte!