

****Tema:** Criação de uma API Rest para um E-commerce de Produtos para Pets**

**Objetivo Geral**

Desenvolver uma API Rest utilizando ****Spring Boot****, que permite gerenciar um e-commerce de produtos para pets, desde o cadastro de produtos até o processamento de pedidos. O aluno deverá implementar a API com boas práticas de desenvolvimento, documentar o código, e realizar testes utilizando o ****Insomnia****.

**Especificações Técnicas**

**1. Funcionalidades obrigatórias da API**

A API deve oferecer os seguintes endpoints:

**1.1. Gestão de Produtos**

- ****Cadastrar um Produto:**** Receber nome, descrição, preço, quantidade em estoque e categoria.
- ****Consultar Produtos:**** Buscar todos os produtos ou filtrar por categoria.
- ****Atualizar Produto:**** Permitir atualizar qualquer atributo de um produto.
- ****Remover Produto:**** Excluir um produto pelo ID.

**1.2. Gestão de Clientes**

- ****Cadastrar Cliente:**** Receber nome, e-mail, telefone e endereço.
- ****Consultar Clientes:**** Listar todos os clientes ou buscar por ID.
- ****Atualizar Cliente:**** Permitir atualizar os dados de um cliente.
- ****Remover Cliente:**** Excluir cliente pelo ID (se não houver pedidos vinculados).

**1.3. Gestão de Pedidos**

- ****Criar Pedido:**** Associar cliente e lista de produtos (com quantidade). Validar disponibilidade em estoque e calcular valor total.
- ****Consultar Pedidos:**** Buscar todos os pedidos ou filtrar por cliente ou status.
- ****Atualizar Pedido:**** Alterar status do pedido (ex.: "em andamento", "enviado", "concluído").
- ****Cancelar Pedido:**** Marcar como cancelado e devolver produtos ao estoque.

**2. Requisitos Técnicos**

- ****Tecnologias:****
 - Linguagem: ****Java****
 - Framework: ****Spring Boot****
 - Banco de Dados: ****PostgreSQL**** (ou outro banco relacional)
 - Ferramenta de Teste: ****Insomnia****
- ****Arquitetura:****
 - Aplicar o padrão ****MVC**** e utilizar ****DAO/Repository**** para acesso ao banco de dados.
 - Implementar ****serviços**** para a lógica de negócio.

- ****Validações:****

- Implementar validações básicas, como formato de e-mail válido e valores positivos para preços e quantidades.
- Utilizar ****Bean Validation**** para validar as requisições.

- ****Documentação:****

- Documentar todos os endpoints utilizando ****Swagger**** ou comentários claros no código.

**3. Testes no Insomnia**

Os alunos devem:

1. Criar um workspace no Insomnia para a API.
2. Testar todas as operações CRUD implementadas (GET, POST, PUT, DELETE).
3. Salvar os exemplos de requisições e respostas (incluindo casos de sucesso e erro).
4. Organizar as requisições em coleções no Insomnia, separadas por categorias (Produtos, Clientes, Pedidos).

**4. Dicas e Observações**

- Utilize o [Spring Initializr](<https://start.spring.io/>) para configurar o projeto.
- Consulte a documentação oficial do Spring Boot para dúvidas sobre implementações específicas.