

## # Documentação Técnica - Projeto API de Pedidos

### ## Viso Geral

Este projeto tem como objetivo construir uma API RESTful que permite:

- Consultar os itens de um pedido e os CDs associados
- Processar pedidos e determinar os Centros de Distribuição com estoque para cada item

Baseado no desafio proposto para liderança técnica no Mercado Livre.

---

### ## Requisitos do Desafio

#### ### Funcionais:

- Um pedido pode conter até \*\*100 itens\*\*.
- Deve existir uma API para \*\*consultar pedidos\*\* com itens e CDs.
- Deve existir uma API para \*\*processar pedidos\*\* e retornar os CDs com estoque.
- Deve-se usar a API de consulta de CDs (mockada).

#### ### Não Funcionais:

- Pode-se usar qualquer linguagem/framework.
- Não requer frontend.
- API externa pode ser mockada internamente.

---

### ## Tecnologias Utilizadas

- Java 17
- Spring Boot 3.x
- Maven
- Lombok
- JPA (Hibernate)
- Flyway (migrations)
- MySQL 8+
- Insomnia / curl para testes de API

---

### ## Estrutura de Pacotes (principais)

```
```\nmeli.pedidos.api\n  cliente\n  produto\n  centrodistribuicao\n  pedido\n  itempedido\n  estoque\n  ```\n
```

---

### ## Modelo de Dados (Entidades)

#### ### Cliente

- id (PK)
- nome
- email

### ### Produto

- id (PK)
- nome
- sku

### ### CentroDistribuicao

- id (PK)
- nome
- estoque

### ### Pedido

- id (PK)
- cliente\_id (FK)
- status

### ### ItemPedido

- id (PK)
- pedido\_id (FK)
- produto\_id (FK)
- cd\_id (FK)
- quantidade

---

## ## Migrations - Flyway (SQL)

### Scripts:

- V1\_\_create\_table\_clientes.sql
- V1\_\_create\_table\_produtos.sql
- V1\_\_create\_table\_pedido.sql
- V1\_\_create\_table\_itens\_pedido.sql
- V1\_\_create\_table\_centros\_distribuicao.sql
- V1\_\_create\_table\_estoque\_cd.sql

---

## ## application.properties

``properties

spring.datasource.url=jdbc:mysql://localhost:3306/meli\_api  
spring.datasource.username=root  
spring.datasource.password=root  
spring.jpa.hibernate.ddl-auto=none  
spring.jpa.show-sql=true  
spring.flyway.enabled=true  
spring.flyway.locations=classpath:db/migration  
server.port=8080  
...

---

## ## Endpoints REST

### 1. Consultar itens e CDs de um pedido

```
**GET** `/pedidos/{id}/itens-cds`
```

### 2. Processar pedido com lista de itens

```
**POST** `/pedidos/processar`
```

```
**Request:**
```

```
```json
```

```
{
  "itens": [
    { "produtoId": 1 },
    { "produtoId": 2 }
  ]
}
```

### 3. Mock da API externa (CDs por item)

```
**GET** `/centros-distribuicao?produtoId=1`
```

---

## Testes via curl

```
```bash
```

```
curl -X GET http://localhost:8080/pedidos/1/itens-cds
```

```
curl -X POST http://localhost:8080/pedidos/processar \
  -H "Content-Type: application/json" \
  -d '{ "itens": [ {"produtoId": 1}, {"produtoId": 2} ] }'
```

---

## Execuo do Projeto

```
```bash
```

```
./mvnw clean spring-boot:run
```

```
```
```

---

## Observaes Finais

- Arquitetura limpa com controllers, services, DTOs e repositrios
- Uso de Lombok reduz boilerplate
- Flyway permite versionamento seguro

---

## Inserts para popular o banco de dados (MySQL)

```
```sql
```

```
-- CLIENTES
```

```
INSERT INTO clientes (nome, email) VALUES
```

```

('Joo Silva', 'joao.silva@email.com'),
('Maria Oliveira', 'maria.oliveira@email.com');

-- PRODUTOS
INSERT INTO produtos (nome, sku) VALUES
('Notebook Dell Inspiron', 'SKU-NTB-001'),
('Mouse Logitech', 'SKU-MSE-002'),
('Teclado Mecnico Redragon', 'SKU-KBD-003');

-- CENTROS DE DISTRIBUIO (CDs)
INSERT INTO centros_distribuicao (nome, endereco, status) VALUES
('CD So Paulo', 'Rua A, SP', 'ATIVO'),
('CD Rio de Janeiro', 'Rua B, RJ', 'ATIVO'),
('CD Recife', 'Rua C, PE', 'INATIVO');

-- ESTOQUE POR CD
INSERT INTO estoque_cd (cd_id, produto_id, quantidade) VALUES
(1, 1, 15), -- SP - Notebook
(1, 2, 30), -- SP - Mouse
(2, 2, 10), -- RJ - Mouse
(2, 3, 50); -- RJ - Teclado

-- PEDIDOS
INSERT INTO pedidos (cliente_id, status) VALUES
(1, 'NOVO'),
(2, 'NOVO');

-- ITENS DO PEDIDO
-- Pedido 1: Joo comprou 1 notebook (vai do CD SP) e 1 mouse (vai do CD RJ)
INSERT INTO itens_pedido (pedido_id, produto_id, quantidade, cd_id) VALUES
(1, 1, 1, 1),
(1, 2, 1, 2);

-- Pedido 2: Maria comprou 2 teclados (vai do CD RJ)
INSERT INTO itens_pedido (pedido_id, produto_id, quantidade, cd_id) VALUES
(2, 3, 2, 2);

```