Escrito por Eduardo Ferreira e Luis Eduardo:

Esses vão ser os testes do Back-end para cobrir todas as funcionalidades do gerenciamento de pedidos usando o postman:

1. Adicionar Produtos ao Estoque (Controle de Estoque)\*\*

- Método: POST

- URL: `http://localhost:8000/produtos`

- Body:

```json

{

"nome": "Produto 1",

"quantidade": 100,

"preco": 50.00

}

```

```json

{

"nome": "Produto 2",

"quantidade": 100,

"preco": 75.00

}

```

- Objetivo: Verificar se os produtos são adicionados ao estoque corretamente.

2. Criar Pedido (Criação de Pedido)

- Método: POST

- URL: `http://localhost:8000/pedidos`

- Body:

```json

{

"cliente": "João Silva",

"produtos": ["Produto 1", "Produto 2"],

"valor\_pedido": 150.75,

"valor\_frete": 20.5,

"endereco": "SQN 211, Bloco C"

}

```

- Objetivo: Criar um pedido e verificar se o pedido é armazenado corretamente e se o estoque dos produtos é atualizado.

3. Atualizar Pedido (Atualização do Pedido)\*\*

- Método: PUT

- URL: `http://localhost:8000/pedidos/{id}` (substitua `{id}` pelo ID do pedido criado anteriormente)

- Body:

```json

{

"endereco": "Rua Nova, 456",

"status": "Em Processamento"

}

```

- Objetivo: Atualizar as informações de um pedido existente.

4. Gerar Relatório de Pedidos (Relatórios de Pedidos)

- Método: GET

- URL: `http://localhost:8000/relatorio`

- Objetivo: Gerar um relatório detalhado de todos os pedidos realizados.

5. Criar Devolução (Gerenciamento de Devoluções)

- Método: POST

- URL: `http://localhost:8000/devolucoes`

- Body:

```json

{

"pedido\_id": "ID (Variável)",

"motivo": "Produto defeituoso"

}

```

- Objetivo: Registrar uma solicitação de devolução de um pedido.

6. Listar Devoluções (Gerenciamento de Devoluções)\*\*

- Método: GET

- URL: `http://localhost:8000/devolucoes`

- Objetivo: Listar todas as devoluções registradas.

7. Atualizar Devolução (Gerenciamento de Devoluções)\*\*

- Método: PUT

- URL: `http://localhost:8000/devolucoes/{id}` (substitua `{id}` pelo ID da devolução criada anteriormente)

- Body:

```json

{

"status": "Aprovado"

}

```

- \*\*Objetivo\*\*: Atualizar o status de uma devolução registrada.

### Funcionalidades Testadas

1. Criação de Pedido:

- Testamos com o método POST para criar um novo pedido.

2. Atualização de Pedido:

- Testamos com o método PUT para atualizar informações de um pedido existente.

3. Relatórios de Pedidos:

- Testamos com o método GET para listar todos os pedidos e gerar relatórios.

4. Controle de Estoque:

- Testamos com o método POST para adicionar produtos ao estoque e verificamos a atualização do estoque ao criar um pedido.

5. Notificações:

- Verificamos implicitamente, pois notificações são enviadas automaticamente ao criar e atualizar pedidos.

6. Gerenciamento de Devoluções:

- Testamos com os métodos POST para criar uma devolução, GET para listar devoluções, e PUT para atualizar o status de uma devolução.

É necessário configurar o banco de dados para testar a aplicação corretamente, o grupo utilizou postgres e segue um script para criação e configuração do banco de dados corretamente para evitar erros.

-- Database: pedido\_gerenciamento

-- DROP DATABASE IF EXISTS pedido\_gerenciamento;

CREATE DATABASE pedido\_gerenciamento;

CREATE USER username WITH ENCRYPTED PASSWORD 'password';

GRANT ALL PRIVILEGES ON DATABASE pedido\_gerenciamento TO username;

GRANT TEMPORARY, CONNECT ON DATABASE pedido\_gerenciamento TO PUBLIC;

GRANT ALL ON DATABASE pedido\_gerenciamento TO postgres;

GRANT ALL ON DATABASE pedido\_gerenciamento TO username;

DROP TABLE IF EXISTS devolucoes;

DROP TABLE IF EXISTS pedidos;

DROP TABLE IF EXISTS produtos;

-- Criação da tabela de pedidos

CREATE TABLE pedidos (

id VARCHAR PRIMARY KEY,

cliente VARCHAR,

produtos TEXT[], -- Array de textos para armazenar IDs dos produtos

valor\_pedido FLOAT,

valor\_frete FLOAT,

endereco VARCHAR,

status VARCHAR,

data\_criacao TIMESTAMP

);

-- Criação da tabela de devoluções

CREATE TABLE devolucoes (

id VARCHAR PRIMARY KEY,

pedido\_id VARCHAR REFERENCES pedidos(id),

motivo VARCHAR,

status VARCHAR,

data\_criacao TIMESTAMP

);

-- Criação da tabela de produtos

CREATE TABLE produtos (

id SERIAL PRIMARY KEY,

nome VARCHAR,

quantidade INT,

preco FLOAT

);

-- Conceder permissões

GRANT USAGE, SELECT ON SEQUENCE produtos\_id\_seq TO username;

GRANT ALL PRIVILEGES ON TABLE produtos TO username;

GRANT ALL PRIVILEGES ON TABLE pedidos TO username;

GRANT ALL PRIVILEGES ON TABLE devolucoes TO username;