Order Management System

Ângelo Victor de Lima

Funcionalidades Principais

Order Processing (Processamento de Pedidos):

- **Descrição:**
  Centraliza o processamento de pedidos, coordenando a validação de usuários e produtos, descontando o estoque e publicando mensagens para o microsserviço de logística.
- **Responsabilidades:**

  Receber requisições para criar pedidos.

  Validar a existência do cliente no sistema (via Customer Management).

  Verificar a disponibilidade do produto e quantidade em estoque (via Product Catalog).

  Enviar o pedido para o microsserviço de logística (via RabbitMQ).
- **Tecnologias:**

  Spring Boot para desenvolvimento do microsserviço.

  Spring Cloud Stream para integração com RabbitMQ.

  WebClient para chamadas REST assíncronas aos microsserviços de clientes e produtos.

Customer Management (Gestão de Clientes):

- **Descrição:**
  Gerencia todas as operações relacionadas aos clientes, como criação, leitura, atualização e exclusão de dados, e validação da existência de clientes.
- **Responsabilidades:**

  Realizar operações CRUD nos registros de clientes.

  Validar a existência de um cliente a partir de um ID fornecido pelo microsserviço de pedidos.

  Armazenar e recuperar informações dos clientes.
- **Tecnologias:**

  Spring Boot para implementação do serviço.

  Spring Data JPA para persistência de dados no banco PostgreSQL.

Product Catalog (Catálogo de Produtos):

- **Descrição:**
  Gerencia o catálogo de produtos, incluindo a importação em massa de dados, consulta de informações e controle de estoque.
- **Responsabilidades:**

  Realizar operações CRUD nos registros de produtos.

  Validar disponibilidade de produtos e quantidade em estoque.

  Descontar o estoque após validação do pedido.

  Importar dados de produtos de fontes externas, como arquivos CSV, utilizando Spring Batch.
- **Tecnologias:**

  Spring Boot para construção do microsserviço.

  Spring Data JPA para operações no banco de dados PostgreSQL.

  Spring Batch para processamento em massa de dados.

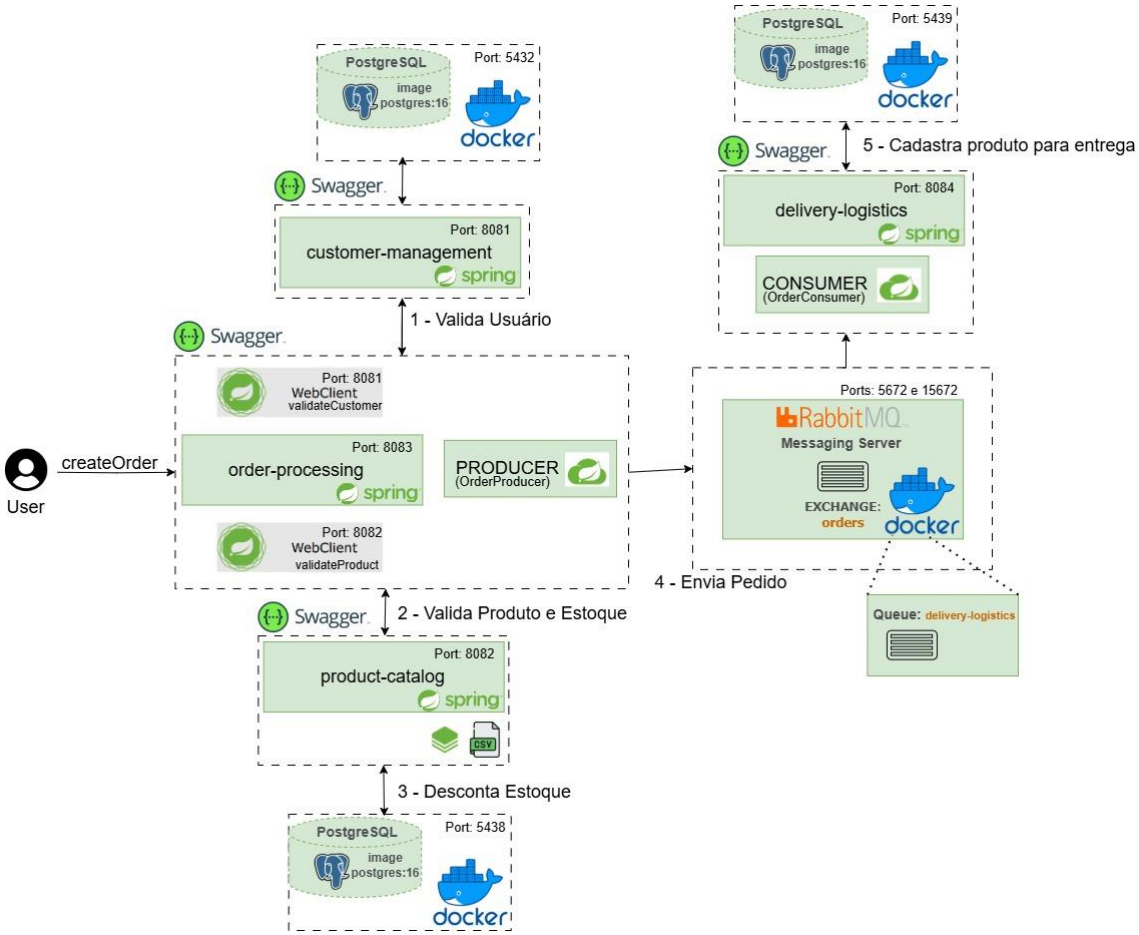
Delivery Logistics (Logística de Entrega):

- **Descrição:**
  Gerencia a logística de entrega de pedidos, desde a atribuição de entregadores até o rastreamento das entregas e atualizações de status.
- **Responsabilidades:**

  Receber os pedidos enviados pelo microsserviço de processamento de pedidos via RabbitMQ.

  Registrar os pedidos para entrega no banco de dados.
- **Tecnologias:**
  - **Spring Boot** para construção do serviço.
  - **Spring Data JPA** para persistência dos dados no banco PostgreSQL.
  - **Spring Cloud Stream** para consumo de mensagens relacionadas aos pedidos.

Fluxo do Sistema - Criar Pedido



A imagem ilustra a arquitetura e o fluxo operacional do sistema de gerenciamento de pedidos, demonstrando como os microsserviços se comunicam e interagem para realizar um pedido de forma integrada. Abaixo, detalhamos o fluxo em cinco etapas principais:

Validação de Usuário:

- O serviço **customer-management** (porta 8081) valida se o usuário existe no sistema.
- Ele utiliza o banco de dados PostgreSQL (porta 5432) para armazenar informações de clientes.

Validação de Produto e Estoque:

- O microsserviço **product-catalog** (porta 8082) verifica a disponibilidade do produto e a quantidade em estoque.
- Ele interage com um banco de dados PostgreSQL (porta 5438) e possui suporte à carga em massa de produtos via arquivos CSV.
- A funcionalidade de carga pode ser acionada rodando o projeto, por agendamento (toda 1 hora da manhã) e de forma manual, entrando no

swagger ou outra ferramenta e executando uma requisição get para o endpoint “/batch/run”

- A validação é acionada por chamadas feitas pelo microsserviço de processamento de pedidos usando WebClient.

Desconto no Estoque:

- Após a validação, o **product-catalog** atualiza o estoque do produto, garantindo a consistência dos dados.

Envio do Pedido:

- O microsserviço **order-processing** (porta 8083) é responsável por criar o pedido e publicá-lo no servidor de mensageria RabbitMQ.
- Um **OrderProducer** envia a mensagem para o Exchange orders, que direciona a mensagem para a fila do microsserviço de logística de entrega.

Cadastro de Produto para Entrega:

- O serviço **delivery-logistics** (porta 8084) consome as mensagens da fila delivery-logistics no RabbitMQ (portas 5672 e 15672).
- Ele registra os pedidos para entrega no banco de dados PostgreSQL (porta 5439).

Endpoints Detalhados

Os endpoints relacionados ao fluxo de trabalho das funcionalidades principais foram detalhados no Swagger.

customer-management: http://localhost:8081/swagger-ui.html

product-catalog: http://localhost:8082/swagger-ui.html

order-processing: http://localhost:8083/swagger-ui.html

delivery-logistics: http://localhost:8084/swagger-ui.html

Repositório do projeto

Passo a passo de como executar o projeto no README.

Link do repositório do projeto no GitHub:

https://github.com/angelovlima/order-management

Ferramentas/Framework

Ferramentas e Tecnologias Utilizadas

- **Linguagem: Java 17**
- **Frameworks: Spring Boot 3.3.0, Spring Data JPA, Spring Cloud Stream, Spring Batch, Spring WebClient**
- **Testes: Junit5, Mockito, MockMvc e Assertj**
- **Banco de Dados: PostgreSQL, H2Database**
- **Message broker: RabbitMQ**
- **Ferramentas de Build: Maven**
- **Documentação: Swagger**
- **Conteinerização: Docker**

Arquitetura do Projeto

A arquitetura do sistema adota os princípios da **Clean Architecture**, estruturando as responsabilidades em camadas bem definidas para promover alta coesão e baixo acoplamento:

- **Controller:** Atua como ponto de entrada para as requisições HTTP, direcionando as interações para os casos de uso e retornando respostas ao cliente.
- **Domain:** Encapsula as regras de negócio e abstrações centrais do domínio, mantendo independência de frameworks e tecnologias externas.
- **Use Case:** Implementa a lógica de aplicação, orquestrando as operações entre as entidades de domínio e as interfaces de infraestrutura, garantindo a execução dos fluxos de negócio.
- **Gateway:** Fornece a implementação concreta para acesso a dados persistentes e integrações externas, como repositórios de banco de dados e sistemas de mensageria, por meio de contratos bem definidos.