

Desafio: Sistema de Cadastro de Cartões

O objetivo deste desafio é criar um sistema backend para gerenciar o cadastro de portadores de cartão e seus respectivos cartões. O sistema será composto por dois microsserviços Spring Boot: Produto Service e Cartão Service.

1. Produto Service:

Responsável por gerenciar os tipos de cartão (produtos) oferecidos.

- **Funcionalidades:**

- Cadastro de novos produtos.
- Busca de produto por ID.
- Cadastro completo de clientes (portador e cartão).
- Busca de cliente por ID (incluindo dados do portador, cartão e produto associado).
- Envio de dados do cartão para o Cartão Service via SQS após o cadastro.

- **Dados do Produto:**

- ID
- Descrição (ex: "Cartão Gold")
- Status (ATIVO, CANCELADO)
- Data de Criação
- Data da última atualização

- **Dados do Portador (na versão completa do serviço):**

- ID
- Nome
- CPF
- Status (ATIVO, BLOQUEADO, CANCELADO)
- Data de Nascimento
- Data de Criação
- Data da última atualização

2. Cartão Service:

Responsável por gerenciar os cartões emitidos para os clientes.

- **Funcionalidades:**

- Recebimento de dados de cartão via SQS (consumidor).
- Busca de cartão por ID (incluindo dados do cartão e do produto associado). Os dados do produto devem ser armazenados em cache (Redis).

- **Dados do Cartão:**

- ID
- Número do Cartão
- Senha do Cartão
- Status (ATIVO, BLOQUEADO, CANCELADO)
- ID do Portador
- Nome Personalizado do Cartão (apelido dado pelo portador)
- ID do Produto (referência ao tipo de cartão)
- Data de Criação
- Data da última atualização

Requisitos Técnicos:

- Java 17
- Maven 3+
- Spring Boot 2.6.7
- PostgreSQL
- Redis
- AWS SQS (simular com LocalStack)
- Docker
- Docker Compose
- Swagger (Springdoc-openapi)
- Testes Unitários para ambos os serviços.

Deploy e Documentação:

O projeto deve ser entregue com Docker Compose para facilitar o deploy. A documentação deve incluir informações sobre os dados, como executar o projeto, decisões técnicas, conceitos utilizados e o fluxo de cadastro.