Especificação do Sistema de Internet Banking

Objetivo Geral

Desenvolver uma aplicação fullstack de Internet Banking com backend em Spring Boot, frontend em React, banco de dados PostgreSQL e microserviço de envio de e-mails. O sistema permitirá o cadastro de usuários, contas correntes, e a realização de operações bancárias (depósito, saque, pagamento), além da geração de extrato. Cada transação e cadastro gerará um e-mail de notificação.

Tecnologias Utilizadas

- **Frontend:** React + Axios
- **Backend Principal:** Java + Spring Boot (Spring Web, Spring Data JPA, Validation)
- **Banco de Dados:** PostgreSQL
- **Microserviço de E-mail:** Spring Boot com JavaMailSender
- **Documentação:** Swagger

Funcionalidades

1. Cadastro de Usuário

- Nome, CPF, e-mail, senha (com hash)
- Conta corrente criada automaticamente
- E-mail de boas-vindas enviado após cadastro

2. Conta Corrente

- Número da conta, agência (0001), saldo
- Associada a um usuário

3. Operações Bancárias

a) Depósito

- Entrada: conta, valor (> 0)
- Atualiza saldo e envia e-mail

b) Saque

- Entrada: conta, valor (> 0, saldo suficiente)
- Atualiza saldo e envia e-mail

c) Pagamento

- Entrada: conta, valor, descrição
- Atualiza saldo e envia e-mail

d) Extrato

- Lista de operações com filtros (tipo, data)

Microserviço de Envio de E-mail

- Recebe requisição via REST (POST /emails)
- Envia e-mails para:
    - Cadastro de usuário
    - Depósito
    - Saque
    - Pagamento
- Estrutura do payload:

```
{
  "to": "usuario@email.com",
  "subject": "Depósito realizado",
  "body": "Você recebeu um depósito de R$500,00. Saldo atual: R$800,00"
```

}

Modelo ER (Entidade-Relacionamento)

Usuario

- id (PK), nome, cpf (unique), email (unique), senhaHash, dataCadastro

Conta

- id (PK), numero, agencia, saldo, usuario_id (FK)

Operacao

- id (PK), tipo (ENUM: DEPOSITO, SAQUE, PAGAMENTO), valor, dataHora, descricao, conta_id (FK)

Relacionamentos:

- Um usuário tem uma conta
- Uma conta possui várias operações

Critérios de Avaliação

Critério	Pontuaçao
Funcionalidades corretas	30 pts
Microserviço de e-mail funcional	20 pts
Modelo de dados relacional correto	10 pts
API REST bem estruturada e documentada	10 pts
Frontend responsivo e funcional	20 pts
Boas práticas de código	10 pts