



**ESTI – ESCOLA SUPERIOR DA TECNOLOGIA DA INFORMAÇÃO  
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO**

**Matéria  
Projeto de Bloco**

Rafael Soares de Souza

**Sistema Bancário - Documentação Técnica**

# Introdução

Este projeto consiste no desenvolvimento de um sistema bancário utilizando Spring Boot para o back-end e React para o front-end. O principal objetivo da aplicação é oferecer aos usuários a possibilidade de gerenciar suas contas bancárias, realizar transações, visualizar o saldo e acessar outros serviços financeiros de forma intuitiva e segura.

A arquitetura da solução foi desenhada para ser modular e escalável, com a implementação de princípios de boas práticas de engenharia de software. O projeto segue os conceitos de Domain-Driven Design (DDD), princípios SOLID e adota padrões de design como o Repository Pattern e o Service Layer Pattern.

A aplicação é composta por:

**Back-end:** Desenvolvido com Spring Boot, o back-end expõe uma API RESTful que gerencia as funcionalidades de contas e transações. A comunicação com o banco de dados é feita utilizando o Spring Data JPA com MySQL.

**Front-end:** Utilizando React para criar uma interface de usuário moderna e responsiva, o front-end se comunica com a API do back-end para consumir as informações bancárias e interagir com as funcionalidades.

**Objetivos:**

Proporcionar uma solução para gerenciamento de contas bancárias.

Permitir a realização de transações como pagamentos, transferências e consulta de saldo.

Garantir a segurança e confiabilidade dos dados financeiros dos usuários.

Criar uma interface amigável e fácil de usar, seguindo boas práticas de usabilidade.

**Tecnologias Utilizadas:**

**Spring Boot:** Framework para desenvolvimento do back-end e gerenciamento de APIs REST.

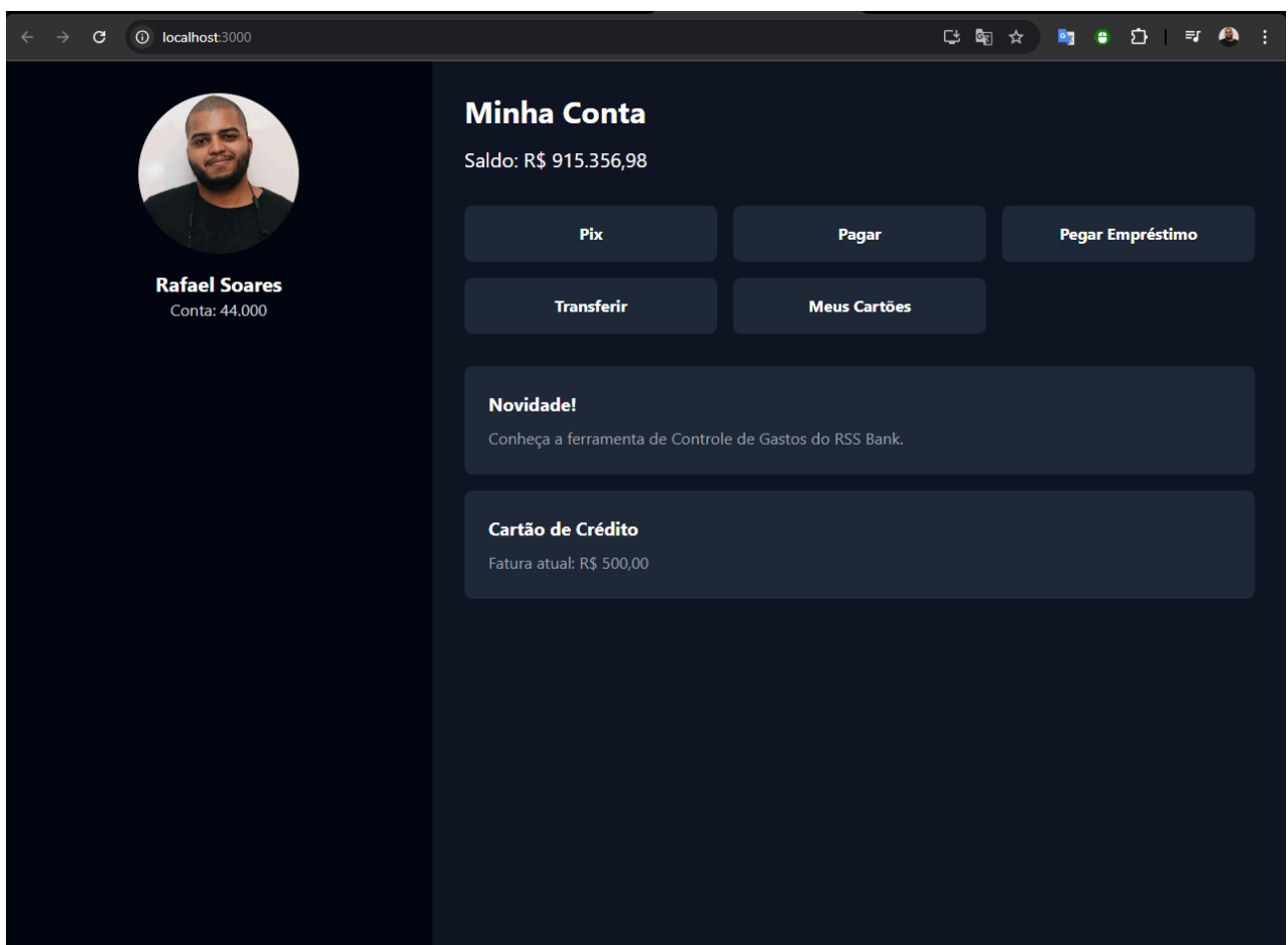
**React:** Biblioteca JavaScript para construção da interface do usuário.

**Tailwind CSS:** Framework CSS para estilização responsiva e moderna do front-end.

**MySQL:** Banco de dados relacional utilizado para armazenar informações das contas e transações.

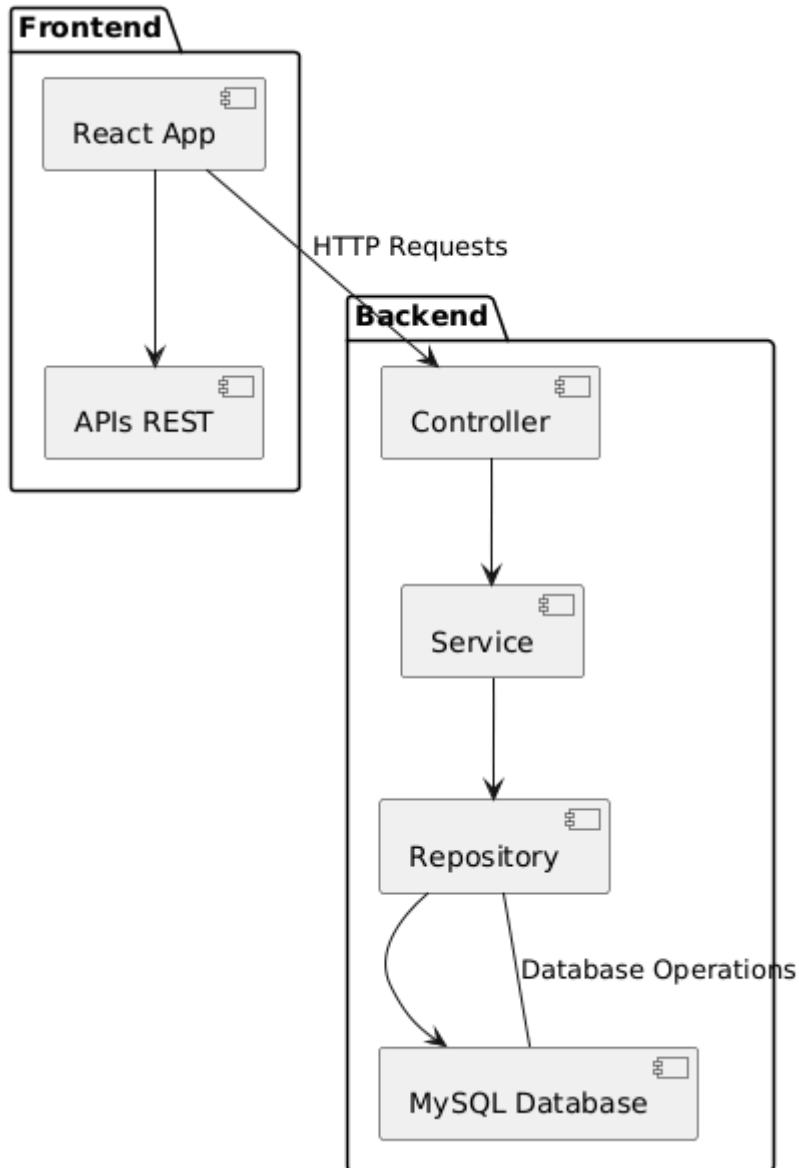
**Docker:** Para facilitar a configuração do ambiente de desenvolvimento e integração contínua.

Este projeto visa, além de ser uma aplicação funcional, proporcionar uma estrutura sólida que sirva como base para a evolução futura do sistema, com a possibilidade de escalar para uma arquitetura de microsserviços.



## Diagrama

O diagrama acima ilustra o fluxo de uma transação bancária no sistema, detalhando a interação entre o usuário, o front-end, o back-end (controlador, serviço e repositório), e o banco de dados. Esse diagrama descreve o processo completo de como uma solicitação de transação é realizada no sistema e os diferentes componentes envolvidos.



Passos Detalhados:

Usuário Solicita a Transação:

O usuário inicia o processo ao solicitar uma transação, como um pagamento ou transferência, por meio da interface do frontend (React).

Envio da Solicitação ao Back-end:

O frontend envia uma solicitação HTTP para o controlador (Controller) no back-end. Esse controlador é responsável por receber e interpretar a requisição do usuário.

## Processamento da Lógica de Negócio (Service):

O controlador delega a solicitação ao serviço (Service), onde a lógica de negócios é processada. Neste ponto, são aplicadas as regras necessárias, como verificar se o usuário tem saldo suficiente para completar a transação.

## Salvando a Transação no Banco de Dados (Repository):

Após a validação, o serviço utiliza o repositório (Repository) para interagir com o banco de dados e salvar a transação. O repositório encapsula a lógica de persistência e gerencia as operações no banco de dados.

## Execução de Query no Banco de Dados:

O repositório executa uma query no banco de dados (MySQL, no caso) para registrar a transação financeira. Isso pode incluir a criação de uma nova transação e a atualização do saldo da conta.

## Confirmação de Sucesso:

O banco de dados confirma que a query foi executada com sucesso, retornando uma resposta positiva ao repositório.

## Envio da Resposta ao Serviço e Controlador:

O repositório retorna uma mensagem de sucesso para o serviço, que, por sua vez, repassa essa confirmação para o controlador. O controlador então gera uma resposta HTTP que é enviada de volta ao frontend.

## Exibição da Confirmação ao Usuário:

O frontend exibe ao usuário a confirmação de que a transação foi concluída com sucesso, completando o ciclo da operação.

## Componentes Envolvidos:

**Usuário:** A pessoa que está interagindo com a aplicação e iniciando o processo de transação.

**Frontend (React):** Responsável por enviar a solicitação de transação e exibir as respostas ao usuário.

**Controller (Spring Boot):** A camada que gerencia as requisições HTTP e direciona as operações para o serviço correto.

**Service** (Lógica de Negócio): Onde a lógica de negócio, como validações e regras, são processadas antes de acessar o banco de dados.

**Repository**: Gerencia as operações de leitura e gravação no banco de dados, utilizando Spring Data JPA.

**Banco de Dados** (MySQL): O armazenamento persistente das informações de contas e transações.

