

# Análise de qualidade do Sistema de Moedas G1

## Arquitetura e tecnologias utilizadas:

O projeto em análise consiste em um projeto de backend e frontend contidos no mesmo repositório. Para o projeto de backend foi utilizado o framework Spring, baseando a implementação em uma arquitetura MVC. O projeto de backend está bem organizado, com as camadas com responsabilidades bem definidas, e com cada uma respeitando os conceitos de responsabilidade dentro da arquitetura MVC. Um ponto positivo para as Services do projeto. Essa camada manteve as lógicas de aplicação retirando da camada de Controller a responsabilidade de cálculos que não estão relacionados com o roteamento das informações dentro do sistema. Isso simplificou a manutenção e deixou o sistema menos acoplado. O frontend do projeto, foi implementado utilizando o framework de desenvolvimento React. Esse framework divide a implementação das views em componentes que podem ser reutilizados de maneira simples dentro do projeto. A escolha desse framework foi muito boa, pois além de ser uma tecnologia de código recente com uma grande comunidade ativa, o reuso de código melhora a manutenção e o tempo de implementação do projeto.

## Organização no GitHub:

A organização do projeto no GitHub foi boa e consistente, tendo a maioria das features integradas ao código principal através de Pull Requests. Apesar disso, o repositório não possui uma branch de desenvolvimento entre o código principal e as features. Seria bom criar uma branch intermediária (develop) para realizar a gestão do código, mantendo o histórico e evitando a perda de código caso seja necessário remover determinadas features do código principal. Um outro ponto de melhoria opcional, seria realizar o merge das Pull Requests usando da estratégia Squash. Com essa estratégia o histórico do código seria preservado de maneira linear, facilitando a gestão das features no código principal. Na estratégia de gestão atual, caso mais de um desenvolvedor estivesse trabalhando em features diferentes ao mesmo tempo, o histórico ficaria com vários nodes referentes ao merge das Pull Requests realizadas em paralelo.

## Dificuldade para configuração do ambiente:

A configuração do ambiente foi feita de maneira rápida, pois ambos os frameworks escolhidos para o front e back da aplicação, possuem ferramentas bem desenvolvidas para a execução do projeto. No backend, com poucos comandos o Spring builda a aplicação com todas as dependências necessárias. De forma similar, o frontend builda e executa a aplicação através de um gerenciador de pacotes npm a critério do utilizador. Um ponto de melhoria seria adicionar na documentação do projeto os comandos e dependências necessárias para a execução do projeto.

## Sugestões de melhorias:

- Criar uma branch intermediária entre as branches de feature e o código principal do projeto.
- Mergear as PRs usando da estratégia squash.
- Adicionar na documentação do projeto os comandos e dependencias necessárias para executar o projeto.