

SmartTransfer

O **SmartTransfer** consiste em uma aplicação simples de agendamento de transferências bancárias.

Funcionalidades da aplicação

- Tela de login;
- Tela de listagem de usuários;
- Tela de criação de usuários;
- Tela de agendamento de transferência;
- Tela de histórico de transferências (extrato).

Nota: A tela de histórico de transferências exibe apenas as transferências realizadas pelo usuário logado. Ou seja, o usuário só poderá visualizar suas próprias transações.

Tecnologias utilizadas

- Node.js 22
- Vue 3
- Vue Router
- PNotify
- Axios
- TypeScript
- Vite

Como executar o projeto

```
# Clone o repositório
git clone https://github.com/NaskIII/SmartTransfer-front.git

# Acesse a pasta do projeto
cd SmartTransfer-front

# Instale as dependências
npm install

# Inicie o projeto
npm run dev
```

Decisões técnicas

- Implementei um layout padrão em `./src/layouts/appLayout.vue`, com as rotas sendo renderizadas dentro desse layout.

Isso proporciona maior flexibilidade para modificar o layout geral da aplicação sem impactar diretamente os componentes filhos.

- O design é simples, porém funcional, e se adapta bem entre telas desktop e mobile.
- Algumas funcionalidades estão implementadas apenas na API (como atualização e inativação de usuários, busca por ID, filtros completos).
Decidi não incluí-las no front-end para manter o escopo do projeto focado e direto, evitando sobrecarga desnecessária.
- Por se tratar de uma aplicação com páginas simples, optei por **não utilizar componentes reutilizáveis**. A introdução de uma arquitetura de componentes genéricos seria **overengineering** neste contexto.
Em um projeto corporativo real, todos os componentes seriam reutilizáveis e customizáveis em tempo de execução.
- Como implementei a funcionalidade completa de usuários, optei por incluir o número da conta no cadastro do usuário. Dessa forma, ao realizar uma requisição de agendamento, o back-end recupera o usuário logado e retorna automaticamente o número da conta de origem, eliminando a necessidade de fornecer essa informação no momento da transferência.

Sobre o cálculo de taxas

A lógica de cálculo das taxas foi implementada **somente no back-end**, pelas seguintes razões:

- Por se tratar de uma operação monetária, é inseguro confiar no front-end para executar cálculos. O código do front-end roda no ambiente do usuário, sobre o qual não tenho controle ou garantias.
 - O back-end Java possui suporte robusto a operações matemáticas e financeiras, oferecendo maior confiabilidade e precisão para este tipo de lógica.
 - Sempre que um cálculo precisa ser realizado, os dados são enviados para o back-end, onde o cálculo é executado. Em seguida, a API retorna o resultado para o front-end, que exibe o valor ao usuário.
-