

# Documentação Completa do Front-End do Projeto —

## Matheus Soldesi

Autor: Matheus Soldesi

Data: 2025-09-16

Versão: 1.0

### 1. Introdução

Esta documentação detalha a arquitetura completa do front-end do projeto, desenvolvido em React + TypeScript, com estilização via Styled Components e comunicação com a API usando axios.

O objetivo é fornecer uma referência detalhada da estrutura, dos componentes, hooks, páginas, estilos, integração e fluxo de dados, garantindo clareza para manutenção e expansão futura.

### 2. Estrutura do Projeto

O front-end está organizado em diretórios principais:

- api/: funções para comunicação com o backend, incluindo chamadas paginadas e tratamento de erros.
- components/: componentes reutilizáveis da interface, como BarraBrasil, MenuBar e SimaTable.
- hooks/: hooks customizados que encapsulam lógica de estado e chamadas à API, como useSima.
- pages/: páginas da aplicação que integram componentes e hooks, como SimaPage.
- styles/: temas, estilos globais e tipagem do styled-components, garantindo consistência visual e segurança de tipos.

### 3. API — simaApi.ts

Arquivo responsável pelo acesso à API SIMA:

- Utiliza axios para requisições HTTP.
- Permite parâmetros de página e limite para chamadas paginadas.
- Retorna dados tipados como `PaginatedResponse<Sima>`.

Exemplo de uso:

```
import { getSima } from '../api/simaApi';
const response = await getSima(1, 20);
console.log(response.data);
```

### 4. Components

BarraBrasil.tsx:

- Adiciona a faixa oficial do governo brasileiro.
- Carrega dinamicamente o script e remove ao desmontar.

- Uso: <BarraBrasil />

MenuBar.tsx:

- Barra de navegação responsiva.
- Desktop: links visíveis em linha.
- Mobile: botão hambúrguer que abre menu suspenso.
- Estilização via styled-components com suporte a tema.

SimaTable.tsx:

- Tabela que recebe dados SIMA via props.
- Colunas: ID, Estação, Data/Hora, Temperatura, Umidade.
- Formata datas e exibe '-' quando dados ausentes.
- Uso: <SimaTable data={dadosSima} />

## 5. Hooks — useSima.ts

Custom hook React que centraliza a lógica de busca de dados SIMA.

- Gerencia estados: data, loading e error.
- Evita duplicação de lógica em componentes.
- Facilita reutilização.

Exemplo de uso:

```
const { data, loading, error } = useSima(1, 20);
```

## 6. Pages — SimaPage.tsx

Página principal de listagem de registros SIMA.

- Integra SimaTable e useSima.
- Controle de paginação com botões Anterior/Próxima.
- Estilização: container principal, tabela com cabeçalho colorido, linhas alternadas, botões estilizados com hover e disabled.
- Busca dados via useEffect sempre que a página muda.

## 7. Styles

GlobalStyle.ts:

- Reset de CSS: altura total, margin/padding 0.
- Define fontes e cores globais baseadas no tema.

styled.d.ts:

- Tipagem TypeScript do tema.
- Garante autocompletar e consistência.

theme.ts:

- Define cores, fontes, espaçamentos, bordas e sombras.
- Aplicado globalmente via ThemeProvider, funcionando como design system.

## 8. Fluxo de Integração

1. BarraBrasil → faixa oficial no topo.
2. MenuBar → navegação entre páginas.
3. SimaPage → integra SimaTable e useSima.
4. GlobalStyle + theme → identidade visual consistente.

Fluxo de dados:

- O hook useSima chama a API.
- Dados recebidos populam SimaTable.
- Paginação e estado de carregamento gerenciados pelo hook e página.

## **9. Conclusão**

Esta documentação detalhada fornece uma visão completa de toda a arquitetura do front-end.

- Componentes, hooks e páginas organizados.
- Estilos e tema centralizados.
- Integração clara com a API.
- Projeto totalmente documentado, tipado e preparado para manutenção e expansão futura.