

Data: 27/05/2025

---

## ## 1. Visão Geral do Frontend

### **\*\*Objetivo:\*\***

Entregar uma interface de usuário moderna, responsiva e intuitiva para o EveryFin, garantindo experiência consistente e

---

## ## 2. Tecnologias e Ferramentas

- **\*\*React (v18) + TypeScript\*\***
- **\*\*Tailwind CSS\*\*** com design tokens
- **\*\*React Router\*\*** para navegação SPA
- **\*\*React Hook Form\*\*** e **\*\*Zod\*\*** para formulários e validação
- **\*\*Axios\*\*** com interceptors customizados
- **\*\*Context API\*\*** e **\*\*Custom Hooks\*\*** para autenticação e estado global
- **\*\*i18next\*\*** para internacionalização (pt\_BR, en\_US)
- **\*\*Vite\*\*** como bundler otimizado
- **\*\*ESLint\*\*** e **\*\*Prettier\*\*** para padronização de código
- **\*\*Jest\*\*** + **\*\*React Testing Library\*\*** para testes de componentes

---

## ## 3. Organização de Pastas

...

```
/frontend
/src
  /assets      # Imagens, SVGs, fontes
  /components  # Botões, Cards, Modals, Inputs
  /hooks       # useAuth, useFetch, useForm
  /contexts    # AuthContext, ThemeContext
  /pages       # Login, Dashboard, Entradas, Saidas, Relatorios
  /routes      # Proteção de rotas e configuração de rotas
  /services    # apiClient, authService, transactionService
  /styles      # tailwind.config.ts, design tokens
  /utils       # formatação, data, helpers
  /tests       # testes de unidade e integração
tailwind.config.ts
vite.config.ts
tsconfig.json
...
```

---

## ## 4. Componentes e Layout

- **\*\*Sistema de Grid\*\*** com Tailwind para responsividade
- **\*\*Cards\*\*** com sombra suave e bordas arredondadas
- **\*\*Botões\*\*** customizáveis via design tokens
- **\*\*Modals\*\*** acessíveis com gestão de foco
- **\*\*Formulários\*\*** com feedback em tempo real

---

## ## 5. State Management e Autenticação

- **\*\*AuthContext\*\*** gerenciando tokens e usuário
- **\*\*Proteção de rotas\*\*** com HOC ``withAuth``

- **Persistência** em localStorage com refresh automático

---

## ## 6. Estilo e Design Tokens

- Definição central em `tailwind.config.ts`
- Tokens de cor: primary, secondary, semantic
- Tipografia: fontSizes e lineHeights
- Espaçamentos escaláveis

---

## ## 7. Testes Automatizados

- **Cobertura de 90%** em componentes críticos
- Testes de interação e snapshots
- Ferramentas de coverage e relatórios

---

## ## 8. CI/CD e Deploy

- Workflow `frontend-ci.yml`: rodar lint, testes e build
- Deploy automático para Vercel (branch main)
- Preview Deploy para cada PR

---

## ## 9. Proposta Negocial - Frontend

### **Valor Agregado:**

- Interface moderna e consistente
- Acessibilidade e performance
- Flexibilidade para temas e i18n
- Manutenibilidade com estrutura modular

### **Cronograma Frontend:**

- Setup e configuração inicial: 1 semana
- Design tokens e componentes base: 1 semana
- Páginas de autenticação: 1 semana
- Dashboard e relatórios: 2 semanas
- Testes e QA: 1 semana

### **Investimento:**

- Valor único: R\$ X.XXX,XX
- Condições: 50% no início, 50% na entrega

---

## ## 10. Prompt Ideal para Frontend Atualizado

- > "Olá, ChatGPT! Carregue e analise o PDF 'EveryFin - Prompt 3 Frontend Completo'. Gere um documento técnico e neg
- > 1. Tecnologias, ferramentas e bundler;
- > 2. Organização de pastas e design tokens;
- > 3. Componentes e layout responsivo;
- > 4. State management e autenticação;
- > 5. Testes automatizados e cobertura;
- > 6. CI/CD e deploy;
- > 7. Proposta de valor, cronograma e investimento.
- >
- > Corrija eventuais erros, complemente com boas práticas e entregue em PDF detalhado."

---

\*Documento expandido para excelência máxima e alinhamento com padrões UX e de negócios.\*