EveryFin - Prompt 4 Login e Cadastro Atualizado (Documento Técnico e Negocial) — Versão Detalhada

Data: 27/05/2025

---

1. Visão Geral

Objetivo:
Implementar módulos de autenticação (login e cadastro) com foco em segurança, usabilidade, acessibilidade e escalabilid

---

2. Tecnologias e Ferramentas

- **React (v18) + TypeScript**
- **React Hook Form** + **Zod** para formulários e validação de schemas
- **Axios** com interceptors para tratamento de tokens e erros
- **Tailwind CSS** com design tokens (cores, tipografia, espaçamentos)
- **Context API** para gerenciamento global de estado de autenticação
- **React Router DOM** para roteamento e proteção de rotas
- **Cypress** para testes E2E
- **Jest** + **React Testing Library** para testes unitários
- **i18next** para internacionalização (pt_BR, en_US)
- **ESLint** + **Prettier** para padronização de código

---

3. Organização de Pastas

```
/frontend
  /src
    /components
      Input.tsx
      Button.tsx
      Notification.tsx
      PasswordStrengthMeter.tsx
    /hooks
      useAuth.ts
      useAxios.ts
    /contexts
      AuthContext.tsx
    /pages
      /auth
        LoginPage.tsx
        RegisterPage.tsx
    /routes
      PrivateRoute.tsx
    /services
      authService.ts
    /utils
      cpfValidator.ts
      tokenStorage.ts
    /styles
      tailwind.config.ts
      design-tokens.ts
    /tests
      /unit
      /e2e
  vite.config.ts
  tsconfig.json
```

---

4. Login

- **Campos:** e-mail (type="email"), senha (type="password")
- **Validações (Zod):**
  - e-mail: formato válido
  - senha: mínimo 8 caracteres, ao menos 1 número, 1 letra maiúscula
- **Fluxo:**
  1. Submissão para `POST /auth/login` usando `authService.login()`
  2. Armazenamento de accessToken e refreshToken via `tokenStorage` (localStorage/secure storage)
  3. Redirecionamento para `/dashboard` em sucesso
  4. Feedback visual inline para erros (ex.: credenciais inválidas)
- **Acessibilidade:** labels associados, mensagens de erro linkáveis, foco em campos inválidos

---

5. Cadastro

- **Campos:** nome completo, e-mail, CPF (mascara e validação), senha e confirmação de senha
- **Validações (Zod):**
  - nome: mínimo 3 caracteres
  - e-mail: formato válido
  - CPF: utiliza `cpfValidator.ts` para validação de dígitos
  - senha: mínimo 8 caracteres, complexidade (número, símbolo, letra maiúscula)
  - confirmação de senha: deve combinar com senha
- **Fluxo:**
  1. Submissão para `POST /auth/register` via `authService.register()`
  2. E-mail de verificação (opcional) com token temporário
  3. Redirecionamento para página de confirmação de e-mail
- **UX:** inline validation ao digitar, indicador de força da senha, máscara CPF

---

6. Componentes Reutilizáveis

- **Input.tsx:** com suporte a ícones, máscaras (CPF), feedback de erro e acessibilidade
- **Button.tsx:** estados de loading, disabled, variants (primary, secondary)
- **Notification.tsx:** toast para sucesso, erro ou alerta
- **PasswordStrengthMeter.tsx:** barra dinâmica de força de senha

---

7. Gerenciamento de Estado e Integração

- **AuthContext:**
  - `user`, `login()`, `logout()`, `register()`
  - Revalida sessão com refreshToken em background
- **useAxios Hook:**
  - Interceptors para anexar token e tratar 401 (logout automático)
- **PrivateRoute:**
  - Redirect para login se usuário não autenticado

---

8. Testes Automatizados

- **Unitários (Jest):**
  - Validação de formulários (Zod)
  - Comportamento de Components (Input, Button)
- **E2E (Cypress):**
  - Fluxo de login válido e inválido
  - Fluxo de cadastro com validações
  - Recuperação de senha (se aplicável)

- **Cobertura:** ≥ 90% em módulos de autenticação

---

9. CI/CD e Deploy

- Workflow `frontend-auth-ci.yml` no GitHub Actions:
  1. Instalar dependências
  2. Executar lint e testes
  3. Build com Vite
  4. Deploy automático em staging (Preview) e main (produção)

---

10. Proposta Negocial

- **Entregáveis:** módulos de Login e Cadastro prontos para uso
- **Duração:** 1 semana
- **Investimento:** R$ X.XXX,XX
- **Condições:** 50% na assinatura, 50% na entrega final
- **Suporte:** bugfixes gratuitos por 15 dias

---

11. Prompt Ideal

> "Olá, ChatGPT! Carregue e analise o PDF 'EveryFin - Prompt 4 Login e Cadastro'. Gere um documento técnico e negoc
> 1. Visão geral e objetivos de autenticação;
> 2. Tecnologias, ferramentas e arquitetura de pastas;
> 3. Detalhamento de Login (validações, fluxo, acessibilidade);
> 4. Detalhamento de Cadastro (validações, fluxo, UX);
> 5. Componentes reutilizáveis e hooks;
> 6. Testes unitários e E2E;
> 7. CI/CD específico para auth;
> 8. Proposta de valor, cronograma e investimento."

Documento gerado para excelência máxima e alinhamento com padrões corporativos e de UX.