Companion - Frontend

Sobre o Projeto

Este é o frontend do projeto Companion, desenvolvido com React e Vite para proporcionar uma experiência moderna e performática.

Tecnologias Utilizadas

- **React**: Biblioteca para construção de interfaces de usuário
- **Vite**: Ferramenta de build rápida e moderna
- **ESLint**: Linting para manter qualidade do código
- **Jest**: Framework de testes unitários
- **Cypress**: Framework para testes end-to-end

Pré-requisitos

Antes de começar, certifique-se de ter instalado em sua máquina:

- Node.js (versão 16 ou superior)
- npm ou yarn

Instalação

1. Clone o repositório:

```
git clone <url-do-repositorio>
cd frontend
```

2. Instale as dependências:

```
npm install
```

Scripts Disponíveis

Desenvolvimento

```
npm run dev
```

Inicia o servidor de desenvolvimento. A aplicação estará disponível em `http://localhost:5173`

Build de Produção

```
npm run build
```

Cria uma versão otimizada para produção na pasta `dist`

Preview da Build

```
npm run preview
```

Visualiza a build de produção localmente

Testes

```
npm test
```

Executa os testes unitários com Jest

```
npm run test:watch
```

Executa os testes em modo watch (observando mudanças)

Testes E2E

```
npm run cypress:open
```

Abre a interface gráfica do Cypress para testes interativos

```
npm run cypress:run
```

Executa todos os testes E2E em modo headless

Linting

```
npm run lint
```

Verifica problemas de código com ESLint

```
npm run lint:fix
```

Corrige automaticamente problemas que podem ser resolvidos pelo ESLint

Estrutura do Projeto

```
src/
├── components/          # Componentes reutilizáveis
├── assets/             # Imagens, ícones e outros recursos
├── __tests__/          # Testes unitários
├── __mocks__/          # Mocks para testes
├── App.jsx             # Componente principal da aplicação
├── main.jsx            # Ponto de entrada da aplicação
└── index.css           # Estilos globais

cypress/
├── e2e/                # Testes end-to-end
└── support/            # Arquivos de configuração do Cypress

public/                 # Arquivos estáticos
```

Configuração de Desenvolvimento

ESLint

O projeto utiliza ESLint para manter a qualidade do código. As regras estão configuradas no arquivo `eslint.config.js`.

Plugins do Vite

Atualmente, o projeto utiliza:

- @vitejs/plugin-react: Usa Babel para Fast Refresh
- @vitejs/plugin-react-swc: Usa SWC para Fast Refresh (alternativa mais rápida)

Expandindo a Configuração do ESLint

Para aplicações em produção, recomendamos usar TypeScript com regras de lint type-aware habilitadas. Consulte o template TS para informações sobre como integrar TypeScript e `typescript-eslint` no seu projeto.

Contribuição

1. Faça um fork do projeto
2. Crie uma branch para sua feature (`git checkout -b feature/AmazingFeature`)
3. Commit suas mudanças (`git commit -m 'Add some AmazingFeature'`)
4. Push para a branch (`git push origin feature/AmazingFeature`)

5. Abra um Pull Request