

FitScore - Documentação e Diagrama

1 Diagrama do Fluxo do Usuário

flowchart TD

```
A[Início - Página Principal] --> B[Login ou Registrar]
B -->|Usuário cadastrado| C[Formulário FitScore]
B -->|Novo usuário| D[Registrar Usuário]
D --> C
C --> E[Calcular FitScore]
E --> F[Salvar dados no localStorage]
F --> G[Dashboard]
G --> H[Visualizar FitScore e Classificação]
B -->|Não autenticado| I[Sidebar mostra Login e Registrar]
C -->|Autenticado| J[Sidebar mostra Dashboard, Form, Relatório]
```

Explicação do diagrama: - A aplicação começa mostrando a página principal. - Se o usuário não está logado, ele pode **Registrar** ou **Login**. - Após login ou registro, ele acessa o **Formulário**. - O formulário calcula o **FitScore** e a **Classificação**, salva no `localStorage`. - Depois, o usuário vai para o **Dashboard**, que lê os dados do `localStorage` e mostra na tabela. - A **Sidebar** muda os links dependendo se o usuário está logado ou não.

2 Explicação Detalhada do Código

Components / Sidebar.tsx

- Sidebar fixa com links.
- Usa `useContext(AuthContext)` para verificar login.
- Usa `useLocation` para saber qual página está aberta.
- `useNavigate` permite navegar programaticamente.
- `useState(mostrarBotaoLogin)` controla os botões centrais de Login e Registrar.
- Funções `handleLoginClick` e `handleRegisterClick` mudam estado e navegam.

Pages / Dashboard.tsx

- Mostra tabela com dados do `localStorage` (`fitscore`).
- `useEffect` carrega os dados ao montar o componente.
- Se não houver dados, mostra mensagem.

Pages / Form.tsx

- Formulário com perguntas divididas em blocos.
- `respostas` guarda respostas do usuário.

- `handleChange` atualiza respostas.
- `calcularFitScore` e `getClassificacao` calculam score e classificação.
- `handleSubmit` salva tudo no `localStorage`.

Pages / Usuario / Login.tsx

- Formulário de login.
- Compara email e senha com `localStorage`.
- Se correto, chama `auth.login()` e navega para `/form`.
- Spinner é mostrado durante carregamento.

Pages / Usuario / Registrar.tsx

- Formulário de cadastro.
- Salva usuário no `localStorage`.
- Chama `auth.login()` e navega para `/form`.
- Spinner é mostrado durante carregamento.

App.tsx

- Contém `Sidebar` fixo.
- `<Routes>` define as páginas com `Route`.
- Layout usa `flex` para colocar sidebar e conteúdo lado a lado.

main.tsx

- Inicia React.
- Envolve App em `BrowserRouter` e `AuthProvider`.
- Garante que todas as rotas e componentes tenham acesso ao contexto de autenticação.

Conclusão

- A aplicação gerencia autenticação simples usando `localStorage`.
- Sidebar muda de acordo com login.
- Form calcula FitScore e Dashboard mostra resultados.
- Tudo organizado com React, hooks, context API e TailwindCSS.