1. **Definição de React Router**

O React Router é uma biblioteca que gerencia rotas em aplicações React, permitindo a navegação entre páginas sem recarregamento. Isso ocorre graças ao Client-Side Routing, onde a aplicação manipula as mudanças de URL internamente. Diferente da navegação tradicional com <a href="..." />, que recarrega a página inteira, o React Router apenas troca os componentes necessários, garantindo mais velocidade. Além disso, possibilita a criação de URLs dinâmicas, como /produto/123, facilita a navegação com <Link> e navegação programática, e permite a implementação de rotas aninhadas, possibilitando layouts dinâmicos sem recarregar toda a aplicação.

1. **Implementando**

**Passo 1: Configuração Inicial**

**Criando um Novo Projeto React**

Se ainda não possui um projeto React, crie um novo executando o seguinte comando:

npx create-react-app fronts5a3 --use-npm

**Instalando o React Router**

1. Abra o terminal no Visual Studio Code e navegue até a pasta do seu projeto React:
2. cd fronts5a3
3. Instale a biblioteca React Router Dom:
4. npm install react-router-dom
5. Após a instalação, entre na pasta do projeto e inicie a aplicação:
6. npm start
7. Ao permitir a execução, uma página inicial do React será aberta no navegador.

**Passo 2: Criação das Páginas**

**Criando os Componentes de Página**

1. Dentro da pasta src do seu projeto, crie uma nova pasta chamada pages:
2. mkdir src/pages
3. cd src/pages
4. Crie quatro arquivos JavaScript para as páginas:
5. touch Dashboard.js Relatorios.js Configuracoes.js Suporte.js
6. Em cada arquivo, crie um componente funcional básico que retorna um <h1> com o nome da página. Exemplo para Dashboard.js:
7. import React from 'react';
8. function Dashboard() {
9. return <h1>Dashboard</h1>;
10. }
11. export default Dashboard;
12. Repita essa estrutura para os outros três componentes (Relatorios.js, Configuracoes.js e Suporte.js).

**Passo 3: Configuração do Roteamento**

**1. Configuração Básica do Roteamento**

1. Abra o arquivo App.js.
2. Importe os módulos necessários do React Router:
3. import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom';
4. Envolva o conteúdo do App.js com o componente <Router>, permitindo que o React Router gerencie a navegação:
5. function App() {
6. return (
7. <Router>
8. <div>
9. {/\* Adicionaremos as rotas e a navegação aqui \*/}
10. </div>
11. </Router>
12. );
13. }
14. export default App;

**2. Definição das Rotas**

1. Importe os componentes de página no App.js:
2. import Dashboard from './pages/Dashboard';
3. import Relatorios from './pages/Relatorios';
4. import Configuracoes from './pages/Configuracoes';
5. import Suporte from './pages/Suporte';
6. Adicione um menu de navegação e defina as rotas utilizando <Routes> e <Route>:
7. function App() {
8. return (
9. <Router>
10. <div>
11. <nav>
12. <ul>
13. <li><Link to="/">Dashboard</Link></li>
14. <li><Link to="/relatorios">Relatórios</Link></li>
15. <li><Link to="/configuracoes">Configurações</Link></li>
16. <li><Link to="/suporte">Suporte</Link></li>
17. </ul>
18. </nav>
19. <Routes>
20. <Route path="/" element={<Dashboard />} />
21. <Route path="/relatorios" element={<Relatorios />} />
22. <Route path="/configuracoes" element={<Configuracoes />} />
23. <Route path="/suporte" element={<Suporte />} />
24. </Routes>
25. </div>
26. </Router>
27. );
28. }
29. export default App;

**Passo 4: Implementação da Navegação**

1. O menu de navegação criado anteriormente usa <nav> para organizar os links.
2. O <Link> é utilizado no lugar de <a> para permitir a navegação sem recarregar a página.

**Passo 5: Testes e Validação**

1. Execute o seguinte comando para iniciar a aplicação:
2. npm start
3. Isso abrirá o navegador com a aplicação em execução.
4. Clique nos links de navegação e verifique se é possível alternar entre as páginas **Dashboard, Relatórios, Configurações e Suporte** sem recarregar a página.