

# Executar os processos de codificação, manutenção e documentação de aplicativos computacionais para desktop

---

SENAC PE

Thiago Nogueira

Instrutor de Educação Profissional

# React Router

# React Router

## O que é?

---

- O React Router ajuda os usuários a se moverem entre diferentes partes do seu aplicativo sem problemas, como um GPS para seu site, guiando os usuários para o lugar certo quando eles clicam em links diferentes.

# React Router

## Necessidade

---

- Você pode estar pensando: "Por que não posso simplesmente criar páginas diferentes e vinculá-las como em HTML?"
- No React, geralmente fazemos o que é chamado de Single Page Application (SPA), que é uma aplicação que carrega apenas um único documento Web e, em seguida, atualiza o conteúdo do corpo desse documento único por meio de APIs JavaScript. Isso torna seu aplicativo mais rápido e suave porque, em vez de carregar novas páginas inteiramente, apenas as partes que precisam mudar são atualizadas.


# React Router

## Necessidade

---

### Etapa 1: Instalar o React Router

- Primeiro, precisamos adicionar o React Router ao nosso projeto.



```
npm install react-router-dom
```

- Este comando diz ao npm (Node Package Manager) para buscar o React Router para nós e adicioná-lo ao nosso projeto.

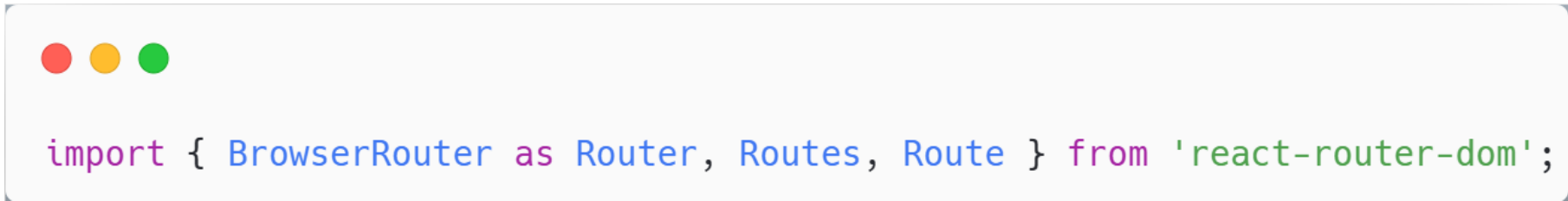
# React Router

## Necessidade

---

### Etapa 2: Importar componentes do React Router

- Primeiro, precisamos adicionar o React Router ao nosso projeto.



```
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
```

- **BrowserRouter** (Router para abreviar) é o contêiner principal que garante que todo o seu aplicativo esteja conectado ao sistema React Router.
- **Routes** é onde definiremos quais páginas (ou componentes) devem ser mostradas para diferentes URLs.
- **Route** é como as portas individuais. Cada rota diz: "Quando o usuário for para este caminho, mostre a ele este componente (ou página)".

# React Router

## Necessidade

---

### Etapa 3: Encapsular a aplicação com o BrowserRouter

- Precisamos envolver todo o nosso aplicativo com o componente BrowserRouter.
- Ele é como a espinha dorsal do nosso sistema de navegação. Ao envolver nosso aplicativo dentro do BrowserRouter, estamos dizendo ao React: "Ei, gerencie toda a navegação dentro deste aplicativo". Sem ele, o React Router não seria capaz de lidar com nenhum roteamento.

```
function App() {  
  return (  
    <Router>  
    { /* Your app content goes here */ }  
    </Router>  
  );  
}
```

# React Router

## Necessidade

---

### Etapa 4: Configurar as rotas

- É aqui que decidimos qual componente mostrar quando alguém visita uma URL específica.

```
function App() {  
  return (  
    <Router>  
      <div>  
        <Routes>  
          <Route path="/" element={<Home />} />  
          <Route path="/about" element={<About />} />  
        </Routes>  
      </div>  
    </Router>  
  );  
}
```

- Neste código, estamos dizendo ao React Router:
  - Quando o usuário visitar / (a página inicial), mostre a ele o componente Início.
  - Quando o usuário visitar /about, mostre a ele o componente About.



# React Router

## Necessidade

---

### Etapa 5: Criar os componentes Home e About

- criar os componentes Home e About que serão exibidos quando os usuários visitarem essas páginas.

```
// Home.js
function Home() {
  return <h1>Welcome to the Home Page!</h1>;
}

export default Home;
```

```
// About.js

function About() {
  return <h1>About Us</h1>;
}

export default About;
```

- Importe esses componentes para o arquivo App.js:

```
import Home from './Home';
import About from './About';
```

# React Router

## Necessidade

---

### Etapa 6: Reunindo tudo

- Arquivo App.js completo

```
// App.js

import React from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import Home from './Home';
import About from './About';

function App() {
  return (
    <Router>
    <div>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
      </Routes>
    </div>
    </Router>
  );
}
```

# Exercícios

# Exercícios

## Instruções

---


- Realizar o exercício da aula 22
- Documentar o que foi feito no README.md
- A documentação deve ter a explicação de cada bloco de código do novo componente criado bem como a explicação do que é o novo componente
- Enviar o exercício roteiro da aula 22 para um repositório no GitHub
- Anexar link do repo na atividade do Teams


*“Ensinar é impregnar  
de sentido o que  
fazemos a cada  
instante”*


*Paulo Freire*

**Obrigado!**

Thiago Nogueira

 [linkedin.com/tdn](https://www.linkedin.com/tdn)

 [thiago.nogueira@pe.senac.br](mailto:thiago.nogueira@pe.senac.br)

 (81) 9 9627-0419