

# React JS

★ AULA 04 ★

★ PROFESSOR  
MAROMO ★

# React JS

Rotas

Biblioteca 'react-router-dom'



# Agenda

- **Instalando a biblioteca 'react-router-dom'**
- **BrowserRouter**
- **Routes**
- **Route**
- **Link**



## Rotas

**Rotas, em desenvolvimento web, são essencialmente os diferentes URLs que uma aplicação web pode responder. Por exemplo, você pode ter uma rota para a página inicial ('/'), uma rota para a página de contato ('/contact'), e assim por diante. As rotas são uma parte fundamental de qualquer aplicação web e definem a estrutura e a organização da aplicação.**



# Rotas no Endereçamento Web

O conceito de roteamento em uma aplicação web tradicional é bastante direto: cada URL corresponde a um recurso específico no servidor, que pode ser uma página HTML, uma imagem, um arquivo CSS, entre outros.



# Rotas em Aplicação SPA

No entanto, com o advento das aplicações de página única (SPA - Single Page Applications), como as construídas com o React, o conceito de roteamento teve que ser repensado.

Em uma SPA, todo o conteúdo da aplicação é carregado na primeira solicitação, e as subsequentes mudanças de página são gerenciadas inteiramente pelo JavaScript no lado do cliente. Isso permite que as SPAs sejam muito mais responsivas do que as aplicações web tradicionais, já que não precisam recarregar a página inteira a cada solicitação.



# Contexto Roteamento em SPA

Nesse contexto, o roteamento é realizado no lado do cliente, e cada "rota" corresponde a um estado específico da aplicação, que é renderizado de acordo com a URL atual. Isso é comumente chamado de "roteamento do lado do cliente" ou "roteamento no front-end".



# Biblioteca Responsável

No React, a biblioteca **react-router-dom** é uma solução popular para gerenciar o roteamento do lado do cliente. Ela permite definir múltiplas rotas na aplicação, cada uma correspondendo a um componente React diferente, e gerencia a transição entre essas rotas de maneira eficiente. Com o **react-router-dom**, é possível criar aplicações React com vários "páginas" ou vistas, mantendo a rapidez e a responsividade de uma SPA.

**react-router-dom**

Como é que  
uso a  
biblioteca ?





Nossa aplicação funcionará da seguinte forma:

O usuário seleciona uma das opções (Front-end ou Back-end);  
A página (componente) referente à opção é renderizada na tela.

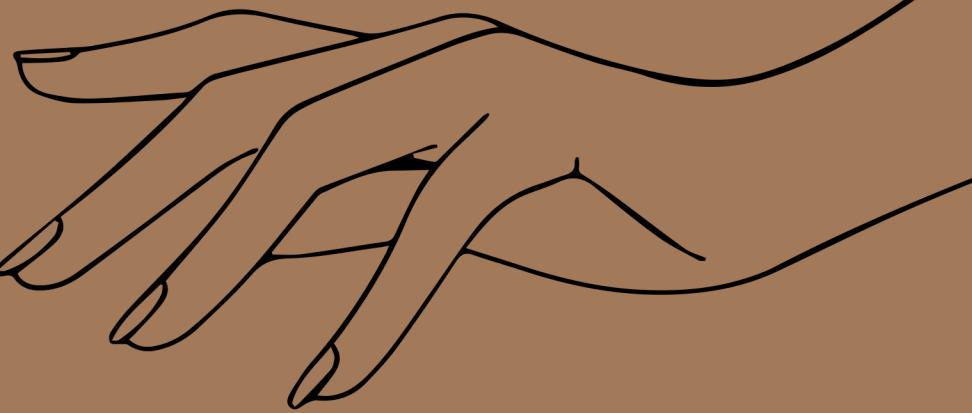
**FRONT-END**

**BACK-END**



# Renderização

- Esse resultado (sem recarregar a página) só é possível graças a biblioteca **react-router-dom**.
- Para chegarmos nesse resultado precisamos conhecer alguns componentes dessa biblioteca:
  - **BrowserRouter =>** Componente que armazena as rotas;
  - **Routes =>** Seleciona apenas um componente para exibir na tela baseado na url.
  - **Route =>** Componente que associa a rota ao componente.
  - **Link =>** Componente que permite navegar para outra página (componente).



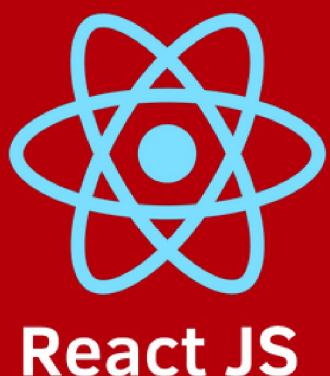
Mão na massa



# Laboratório



Desenvolvimento de uma aplicação SPA usando o React JS. Nome da aplicação **sample04**.





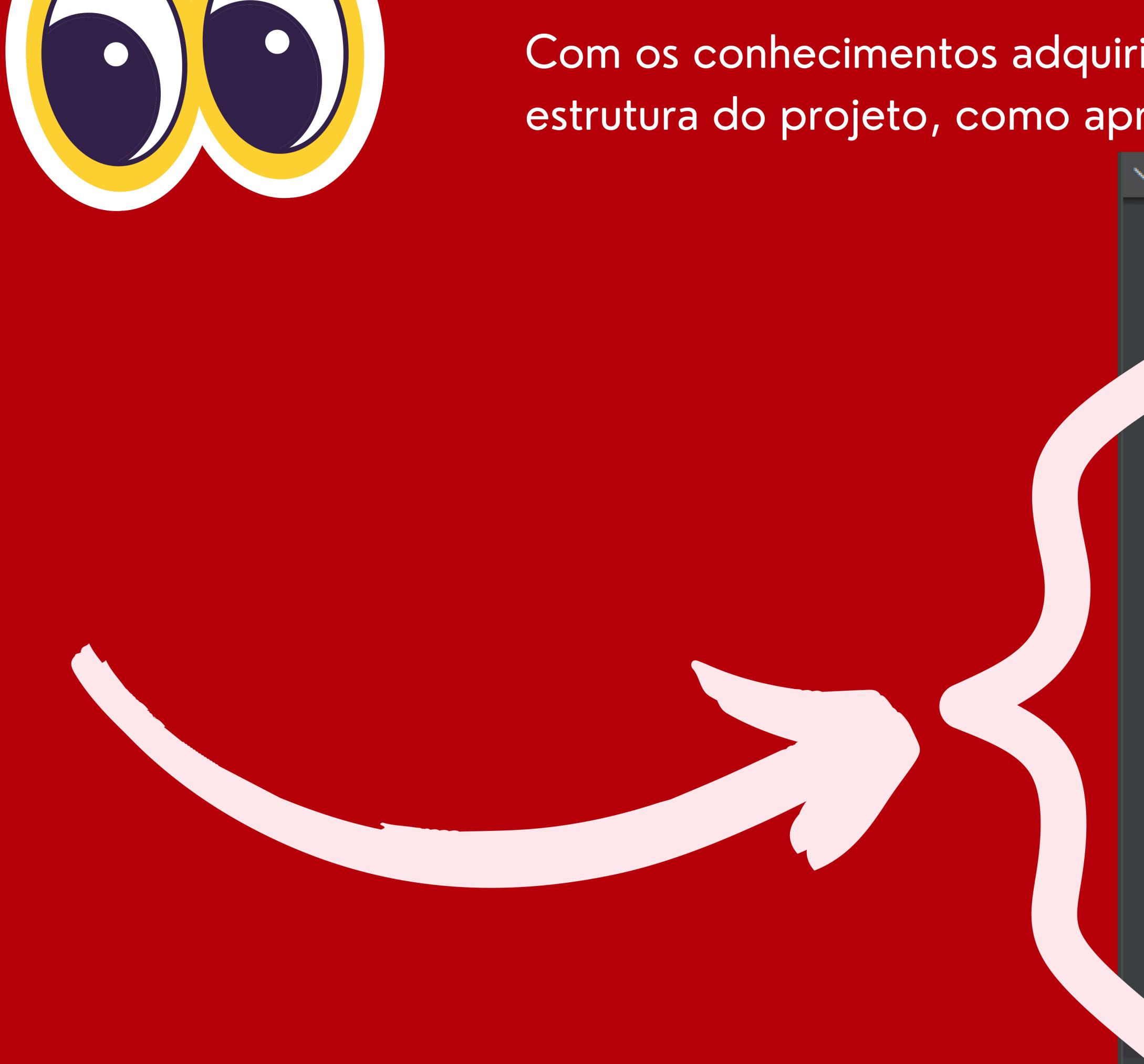
# Passo 1: Configuração do ambiente de desenvolvimento

1. Execute o comando `npx create-react-app sample04` para criar um novo projeto React chamado "sample04". Aguarde até que a criação do projeto seja concluída.
2. O processo anterior, de criação do projeto, pode levar alguns instantes enquanto create-react-app monta todo o **boilerplate** mínimo que uma aplicação React para a web precisa ter.
3. Entre no diretório [cd sample04] e digite [code .] para abrir o VS Code.

# Passo 2: Estrutura do projeto React



```
<SAMPLE04>
  > node_modules
  > public
  <src>
    App.css
    App.js
    App.test.js
    index.css
    index.js
    logo.svg
    reportWebVitals.js
    setupTests.js
```



Com os conhecimentos adquiridos nas aulas anteriores, defina a estrutura do projeto, como apresenta a figura:

```
▽ SAMPLE04
  > node_modules
  > public
  ▽ src
    ▽ components
      ▽ BackButton
        JS index.js
        CSS style.css
      ▽ BackEndPage
        JS index.js
        CSS style.css
      ▽ FronEndPage
        JS index.js
        CSS style.css
      ▽ HomePage
        JS index.js
        CSS style.css
    ▽ routes
      JS index.js
      CSS App.css
      JS App.js
```



# Divisão da aplicação

Nossa aplicação está dividida em:

1. App - o componente principal;
2. Routes - quem guarda e gerênciia nossas rotas;
3. HomePage- a página inicial;
4. BackEndPage - página exibida através da rota /back-end;
5. FrontEndPAge - página exibida através da rota /front-end;
6. BackButton - componente utilizado para voltar para a página inicial. Esse componente é reutilizado em BackEndPage e FronEndPage.



# Rotas do Projeto



Rota “/”

- Rota inicial, está associada ao componente HomePage

Rota “/front-end”

- Rota associada ao componente FrontEndPage

Rota “/back-end”

- Rota associada ao componente BackEndPage



# Links



Links que permitem a navegação:

- `/` => endereço url padrão.
- **Guia do React** => Link que abre uma url externa
- **Botão Voltar** => Link que volta para a página inicial (HomePage)
- **Botão Front-end** => Link que navega e exibe o componente (FrontEndPage)
- **Botão Back-end** => Link que navega e exibe o componente (BackEndPage)



## Passo 3: Preparando o ambiente para funcionamento das rotas

O primeiro passo é instalar a biblioteca react-router-dom no nosso projeto.

Instalando react-router-dom



PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL

```
PS C:\javascript\react01\sample04> npm install react-router-dom
[.....] - idealTree:sample04: sill idealTree buildDeps
```



# Passo 4: Configurando as rotas

Dentro da pasta routes, altere o arquivo index.js para definirmos as rotas de nossa aplicação. Será o componente Routes.

```
import React from "react";
import { BrowserRouter, Routes, Route } from 'react-router-dom';

import HomePage from '../components/HomePage';
import FrontEndPage from '../components/FronEndPage';
import BackEndPage from '../components/BackEndPage';

const MyRoutes = () => (
  <BrowserRouter>
    <Routes>
      <Route exact path='/' element={ <HomePage /> } />
      <Route path='/front-end' element={ <FrontEndPage /> } />
      <Route path='/back-end' element={ <BackEndPage/> } />
    </Routes>
  </BrowserRouter>
);
export default MyRoutes;
```



Explicando

A constante Routes neste código é um componente funcional em React. O papel deste componente é definir as diferentes "rotas" ou "páginas" que a aplicação terá. Ele está utilizando várias funcionalidades oferecidas pela biblioteca react-router-dom para gerenciar essas rotas de maneira eficiente e flexível.

- **BrowserRouter**: Este é um componente de roteamento que usa a API de histórico HTML5 para manter sua interface de usuário em sincronia com a URL.
- **Routes**: Este é um componente que renderiza apenas a primeira rota Route ou Redirect que corresponde à localização atual.
- **Route**: Este é um componente que define uma rota. Ele tem duas props principais:
  - **path**: Define o caminho da URL para a rota.
  - **element**: Define o componente que será renderizado quando essa rota é acessada.

No exemplo, há três rotas definidas:

- Uma para o caminho `/`, que renderiza o componente `HomePage`.
- Uma para o caminho `/front-end`, que renderiza o componente `FrontEndPage`.
- E outra para o caminho `/back-end`, que renderiza o componente `BackEndPage`.



# Passo 5: Importando as rotas para o App.js

Nossas rotas estão criadas e agora precisamos colocá-las em nossa aplicação. Para isso abra o arquivo App.js

```
1 import React from 'react';
2 import Routes from './routes';
3 import './App.css';
4
5 function App() {
6   return (
7     <div className="App">
8       <Routes />
9     </div>
10  );
11}
12
13 export default App;
```



Linkando para  
outras páginas





# Linkando HomePage as outras páginas

## Componentes HomePage (index.js)

```
1 import React from "react";
2 import './style.css';
3 import { Link } from 'react-router-dom';
4
5 const HomePage = () => (
6   <div>
7     <h1>Escolha o seu caminho: </h1>
8     <div className="conteudo-opcoes">
9       <Link className="opcao-front-end" to = '/front-end'>Front-end</Link>
10      <Link className="opcao-back-end" to = '/back-end'>Back-end</Link>
11    </div>
12  </div>
13 );
14 export default HomePage;
```



- Para concluirmos nossa SPA e finalizarmos a navegação da nossa SPA, precisamos permitir voltar para **HomePage** sem recarregar o navegador. Para isso vamos alterar o componente **BackButton**.
- O componente **BackButton** foi codificado apenas uma vez e reutilizado nos componentes **FrontEndPage** e **BackEndPAge**.
- Note que vamos alterar em apenas um lugar e esta alteração vai refletir em duas páginas distintas que utilizam o mesmo componente.



# Componente: BackButton

## Arquivo index.js

src > components > BackButton > **js** index.js > ...

```
1 import React from "react";
2 import './style.css';
3 import { Link } from "react-router-dom";
4
5 const BackButton = () => (
6   <div className="area-botao">
7     <Link className="botao-voltar" to="/" >Voltar</Link>
8   </div>
9 );
10 export default BackButton;
```



# Componente: BackEndPage

## Arquivo: index.js

```
import React from "react";
import './style.css';
import BackButton from '../BackButton';
const BackEndPage = () => (
  <div>
    <h1>Node.js</h1>
    <p>
      Node.js é uma plataforma de execução de código JavaScript construída sobre o motor JavaScript V8 do Google Chrome. Ele permite a execução de JavaScript no lado do servidor e não apenas no navegador, o que permite a criação de aplicações web escaláveis e de alta performance. Node.js utiliza um modelo de I/O (entrada e saída) não bloqueante orientado a eventos, tornando-o leve e eficiente, ideal para aplicações em tempo real com intensa troca de dados através de dispositivos distribuídos. Além disso, possui um rico ecossistema de bibliotecas e frameworks disponíveis através do gerenciador de pacotes npm (Node Package Manager).
    </p>
    <a href="https://nodejs.org/">Node.js - Site</a>
  </div>
);
export default BackEndPage;
```

# Componente: FrontEndPage

## Arquivo: index.js

```
import React from "react";
import './style.css';
import BackButton from '../BackButton';
const FrontEndPage = () => (
  <div>
    <h1>React.js</h1>
    <p>
      React.js, frequentemente chamado apenas de React, é uma biblioteca JavaScript de código aberto criada pelo Facebook para construir interfaces de usuário, especialmente para aplicações de página única. Seu principal objetivo é ser rápido, escalável e simples, trabalhando com componentes que podem ser reutilizados, permitindo aos desenvolvedores criar interfaces de usuário complexas de maneira eficiente e fácil de manter. Ele também suporta a renderização no lado do servidor e pode ser usado para desenvolver aplicações móveis por meio do React Native.
    </p>
    <a href="https://reactjs.org/">React.js - Site</a>
  </div>
);
export default FrontEndPage;
```

# Arquivos de Apoio. Estilos

**Arquivos que não serão comentados, apenas para melhorar a estética. As folhas de estilos.**

**São eles:**

- **BackButton -> style.css**
- **HomePage -> style.css**
- **App.css**

# BackButton - style.css

```
1  .area-botao {  
2      width: 100%;  
3      margin-top: 100px;  
4  }  
5  
6  .botao-voltar {  
7      padding: 20px;  
8      text-decoration: none;  
9      background-color: #f7df1e;  
10     color: #000;  
11     font-weight: 700;  
12 }
```

```
.conteudo-opcoes {  
    display: flex;  
    justify-content: space-between;  
    margin-top: 50px;  
}  
  
.conteudo-opcoes a {  
    padding: 20px;  
    text-decoration: none;  
    font-weight: 500;  
}  
  
.opcao-front-end {  
    background-color: #61dafb;  
    color: blue;  
}  
  
.opcao-back-end {  
    background-color: #43853d;  
    color: white  
}
```

# HomePage - style.css

```
1 .App {  
2   text-align: center;  
3   display: flex;  
4   justify-content: center;  
5   align-items: center;  
6   width: 100vw;  
7   height: 100vh;  
8   padding: 20px;  
9   box-sizing: border-box;  
10 }
```

## App.css



## Escolha o seu caminho:

Front-end

Back-end

### React.js

React.js, ou simplesmente React, é uma biblioteca JavaScript de código aberto para construção de interfaces de usuário, desenvolvida pelo Facebook. A sua principal característica é a criação de componentes reutilizáveis, que permitem aos desenvolvedores construir aplicações web complexas com interfaces dinâmicas de maneira mais eficiente e organizada. React também é conhecido pelo uso do Virtual DOM, uma abstração do DOM real, que ajuda a otimizar a renderização e o desempenho das aplicações. Além disso, a biblioteca pode ser utilizada tanto no desenvolvimento de aplicações de página única (Single Page Applications - SPA) quanto em partes de websites mais tradicionais, e é uma das tecnologias principais no ecossistema de desenvolvimento JavaScript moderno.

[React.js - Site](#)

Voltar



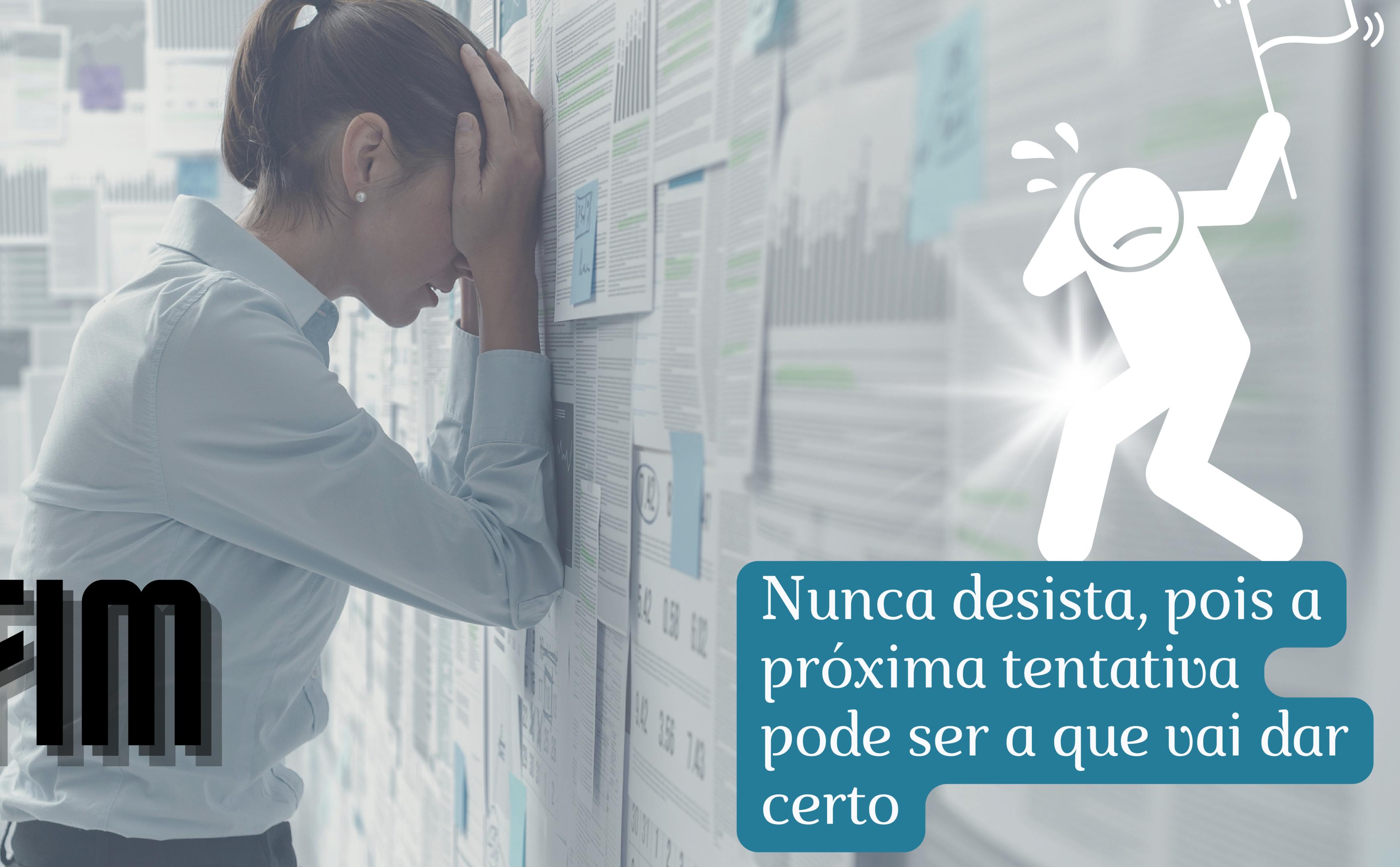
# Conclusão

Rotas desempenham um papel crítico em qualquer aplicação web moderna, pois proporcionam uma experiência de navegação intuitiva e eficiente ao usuário, além de tornar a aplicação mais organizada e de fácil manutenção para os desenvolvedores.

Ao trabalhar com React, a biblioteca react-router-dom é um excelente recurso para gerenciar rotas, permitindo a construção de aplicações de página única robustas e dinâmicas.

Ela oferece ferramentas poderosas como BrowserRouter, Routes e Route que possibilitam a criação de várias rotas e sub-rotas, melhorando assim a escalabilidade e a interatividade da aplicação. Aprender a manipular rotas é, portanto, uma habilidade fundamental para desenvolvedores React.

**FIM**



Nunca desista, pois a  
próxima tentativa  
pode ser a que vai dar  
certo