

Desenvolvimento Front-end com Frameworks

Fundamentos de React

Agenda

Etapas 5: Estilos em aplicações React.

- Formas Nativas de Estilizar Componentes.
- Biblioteca Styled Components.
- Bootstrap.



Formas Nativas de Estilizar Componentes

A forma mais simples de aplicar estilos em um componente é importar arquivos CSS.

O processo é semelhante a como vinculamos o arquivo CSS no HTML <head>:

```
App.jsx x
src > App.jsx > ...
1  import { BrowserRouter as Router, Route, Routes } from "react-router-dom";
2  import Layout from "../pages/Layout";
3  import Home from "../pages/Home"
4  import Fornecedores from "../pages/Fornecedores"
5  import Contatos from "../pages/Contatos"
6  import Produtos from "../pages/Produtos"
7  import Cotacoes from "../pages/Cotacoes"
8  import "../App.css"
9
10 export default function App() {
11   return (
12     <Router>
13       <Routes>
14         <Route path="/" element={<Layout />}>
15           <Route index element={<Home />} />
16           <Route path="fornecedores" element={<Fornecedores />} />
17           <Route path="contatos" element={<Contatos />}>
18             <Route path=":id" element={<Contatos />} />
19           </Route>
20           <Route path="produtos" element={<Produtos />} />
21           <Route path="cotacoes" element={<Cotacoes />} />
22         </Route>
23       </Routes>
24     </Router>
25   );
26 }
```

```
App.jsx U # App.css M X Tela1.jsx U
src > # App.css > ...
1 .container {
2   background-color: lightblue;
3   border: 1px solid black;
4   margin: 5px;
5   padding: 5px;
6   display: inline-table;
7 }
```

```
App.jsx U X # App.css M Tela1.jsx U
src > App.jsx > ...
1 import logo from './logo.svg';
2 import './App.css';
3 import Tela1 from './Tela1';
4
5 function App() {
6   return (
7     <div className="container">
8       <h3>Estudo de Estilos</h3>
9       <Tela1 />
10    </div>
11  );
12 }
13
14 export default App;
```



```
5   export default function FormFornecedor({setFornecedorId}) {  
45       <br />  
46       <input style={{ margin: "5px auto", display: "block" }} type="submit" value="Cadastrar" />  
47   </form>  
48   </div>  
49   <div className="errorsContainer">  
50       {errors.nomeFantasia?.message && (  
51           <div>{errors.nomeFantasia.message}</div>  
52       )}  
53       {errors.razaoSocial?.message && (  
54           <div>{errors.razaoSocial.message}</div>  
55       )}  
56       {errors.cnpj?.message && (  
57           <div>{errors.cnpj.message}</div>  
58       )}  
59   </div>  
60   </>  
61   );  
62 }
```

Estilos embutidos podem ser complicados de manter porque ficam literalmente espalhados pelos vários arquivos da aplicação.

Entretanto, pode ser uma solução simples em alguns casos.

```
App.jsx U # App.css M Tela1.jsx U x
src > Tela1.jsx > Tela1
1 export default function Tela1() {
2   const styles = {
3     main: {
4       backgroudColor: "#f1f1f1",
5       width: "100%",
6     },
7     inputText: {
8       padding: "10px",
9       color: "red"
10    },
11  };
12
13  return(
14    <div style={styles.main}>
15      <input type="text" style={styles.inputText} />
16    </div>
17  );
18 }
```

Uma forma interessante de elaborar melhor os estilos embutidos é criar objetos com as definições de estilos e usar esses objetos no elemento a ser estilizado.

```
Tela3.jsx # styles.module.css x
src > # styles.module.css > ...
1  .heading {
2    color: red; #f00;
3    font-size: 20px;
4  }
```

```
App.jsx U # styles.module.css U Tela2.jsx U x
src > Tela2.jsx > ...
1  import styles from './styles.module.css';
2
3  export default function Tela2() {
4
5    return (
6      <h1 className={styles.heading}>Hello World</h1>
7    )
8  }
```

Módulos CSS nos permitem usar o mesmo nome de classe em vários arquivos sem conflitos, já que cada nome de classe recebe um nome programático exclusivo.

Isto é especialmente útil em aplicações maiores.

Cada nome de classe tem escopo local para o componente específico no qual está sendo importado.

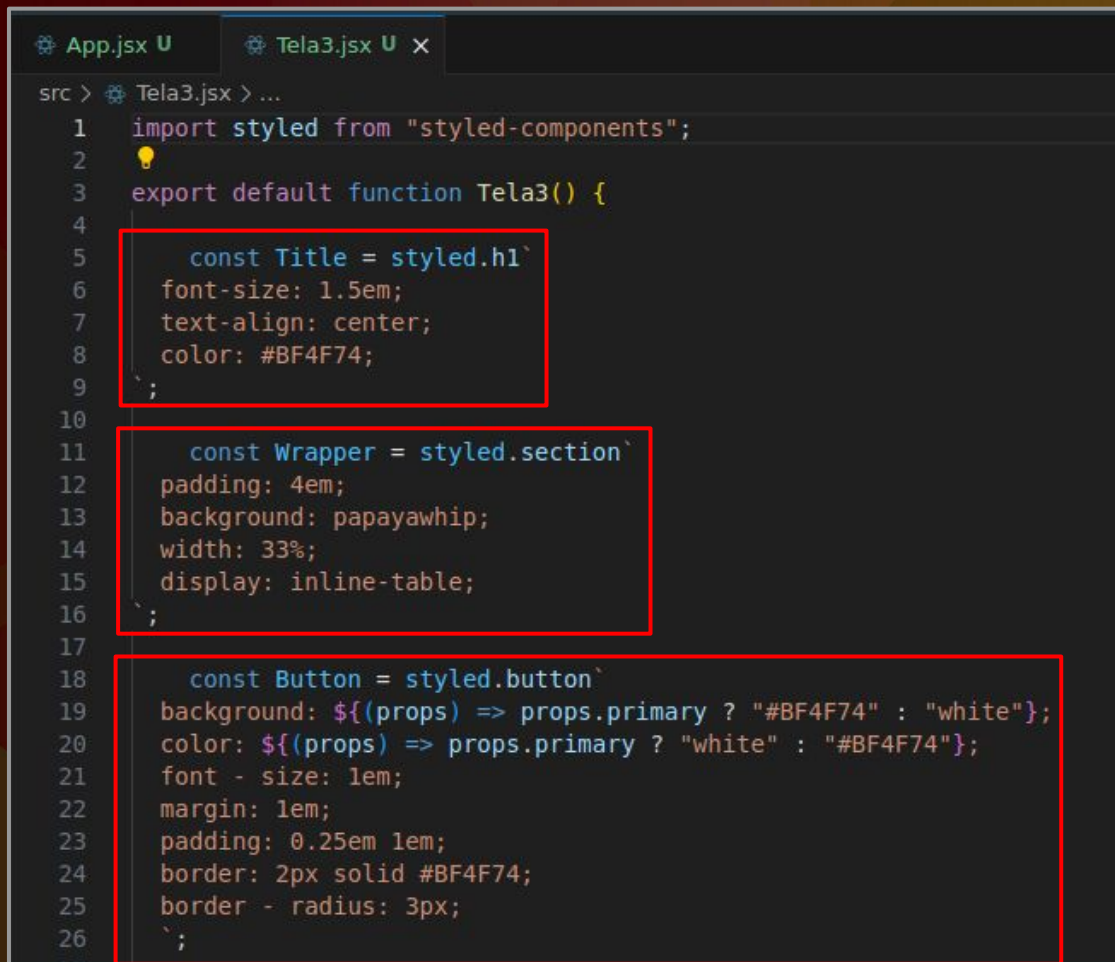
Biblioteca Styled Components

styled-components é o resultado de nos perguntarmos como poderíamos aprimorar o CSS para estilizar sistemas de componentes React.

Ao focar em um único caso de uso, conseguimos otimizar a experiência para os desenvolvedores, bem como o resultado para os usuários finais.

- Componentes estilizados rastreiam quais componentes são renderizados em uma página e injetam seus estilos e nada mais, de forma totalmente automática. Combinado com a divisão de código, isso significa que seus usuários carregam a menor quantidade de código necessária.
- **styled-components** gera nomes de classe exclusivos para seus estilos. Você nunca precisa se preocupar com duplicações, sobreposições ou erros ortográficos.
- Você nunca precisa procurar arquivos diferentes para encontrar o estilo que afeta seu componente, portanto, a manutenção é muito fácil, não importa o tamanho da sua base de código

npm i styled-components



The image shows a code editor with two tabs: 'App.jsx U' and 'Tela3.jsx U x'. The 'Tela3.jsx' tab is active, showing a file path 'src > Tela3.jsx > ...'. The code in the file is as follows:

```
1 import styled from "styled-components";
2
3 export default function Tela3() {
4
5     const Title = styled.h1`
6       font-size: 1.5em;
7       text-align: center;
8       color: #BF4F74;
9     `;
10
11     const Wrapper = styled.section`
12       padding: 4em;
13       background: papayawhip;
14       width: 33%;
15       display: inline-table;
16     `;
17
18     const Button = styled.button`
19       background: ${(props) => props.primary ? "#BF4F74" : "white"};
20       color: ${(props) => props.primary ? "white" : "#BF4F74"};
21       font-size: 1em;
22       margin: 1em;
23       padding: 0.25em 1em;
24       border: 2px solid #BF4F74;
25       border-radius: 3px;
26     `;
```

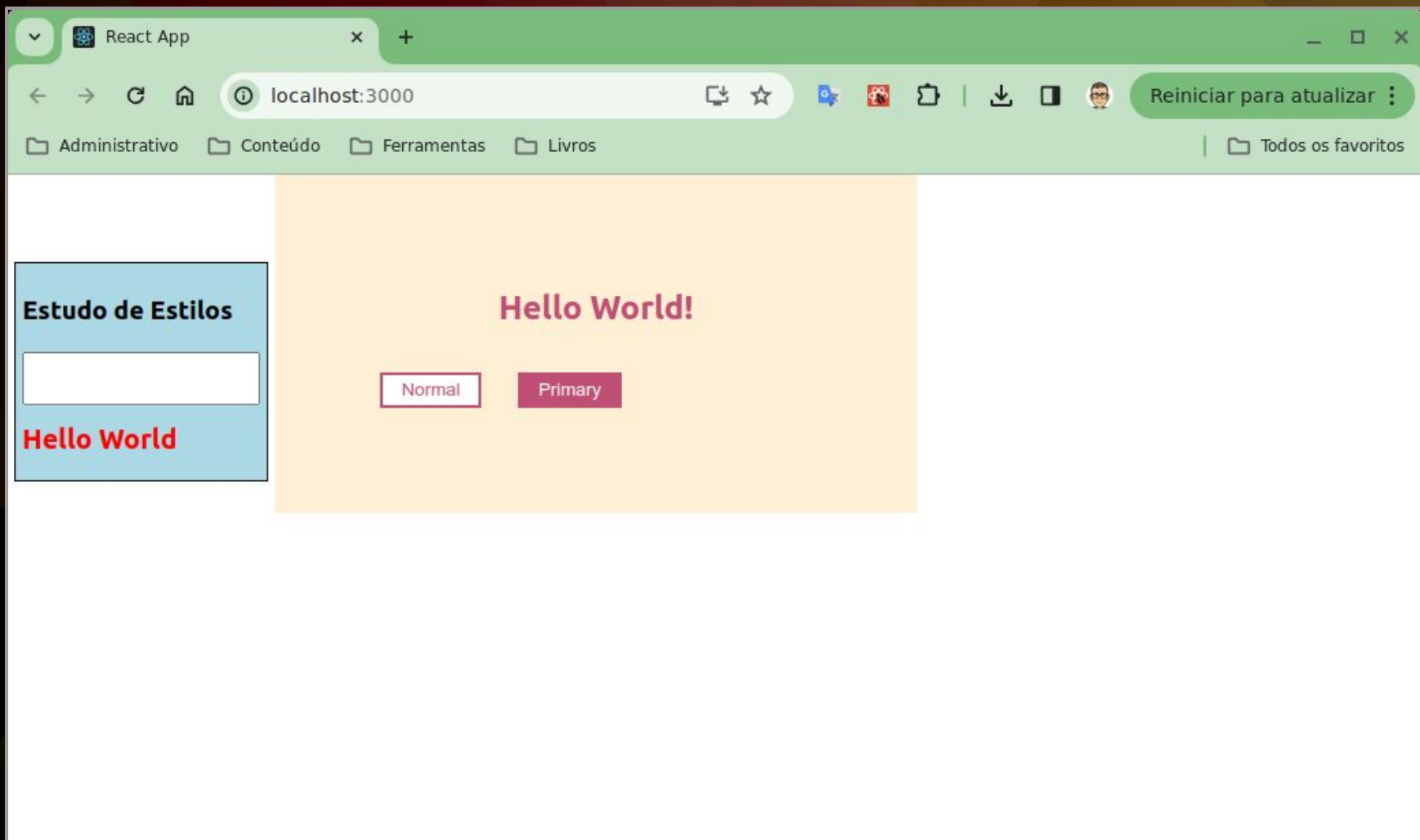
The code defines three styled components: 'Title' (a styled h1), 'Wrapper' (a styled section), and 'Button' (a styled button). The 'Button' component uses a ternary operator to set its background and color based on the 'primary' prop.

App.jsx U

Tela3.jsx U x

src > Tela3.jsx > ...

```
3  export default function Tela3() {  
28    return (  
29      <>  
30        <Wrapper>  
31          <Title>  
32            Hello World!  
33          </Title>  
34          <div>  
35            <Button>Normal</Button>  
36            <Button primary>Primary</Button>  
37          </div>  
38        </Wrapper>  
39      </>  
40    )  
41  }
```

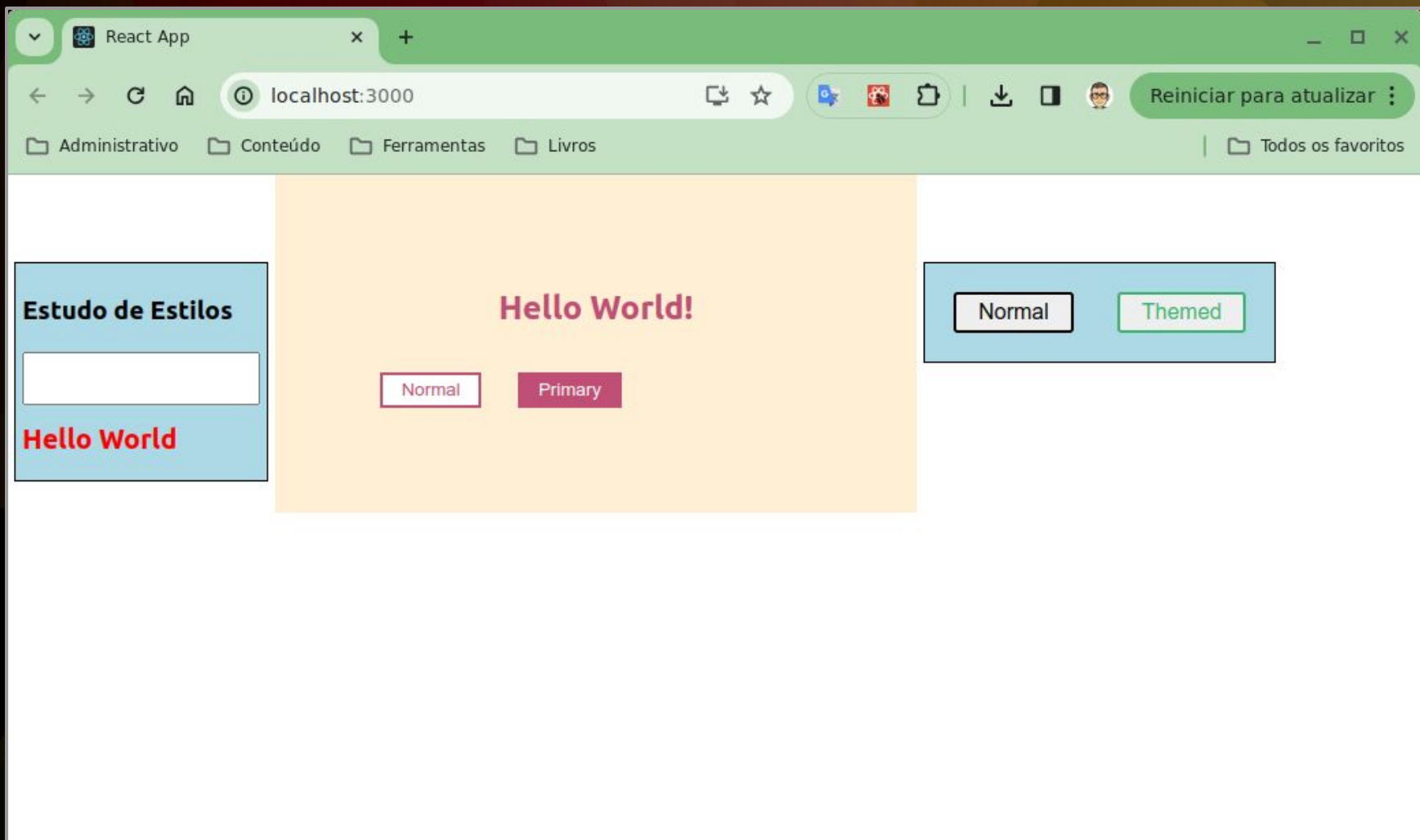


App.jsx U

Tela4.jsx U X

src > Tela4.jsx > ...

```
1  import styled, { ThemeProvider } from "styled-components";
2
3  export default function Tela4() {
4
5      const Button = styled.button`
6          font-size: 1em;
7          margin: 1em;
8          padding: 0.25em 1em;
9          border-radius: 3px;
10         color: ${props => props.theme.main};
11         border: 2px solid ${props => props.theme.main};
12     `;
13
14     const theme = {
15         main: "mediumseagreen"
16     };
17
18     return (
19         <div className="container">
20             <Button>Normal</Button>
21
22             <ThemeProvider theme={theme}>
23                 <Button>Themed</Button>
24             </ThemeProvider>
25         </div>
26     )
27 }
28
```





Bootstrap

React App

React Bootstrap | React Boo

← → ↻ 🏠

react-bootstrap.netlify.app


🔍 ☆

🗨️ ⚙️ 📁

📄 📱 👤

Reiniciar para atualizar ⋮

Administrativo | Conteúdo | Ferramentas | Livros | Todos os favoritos

 **React Bootstr...**

Getting Started

Components

v2.10.2 (Bootstrap 5) ▼

GitHub ↗

🌙

🔍 Search CTRL K


React Bootstrap

The most popular front-end framework, rebuilt for React.

Get started

Components

Current version: 2.10.2



Design and Development tips
in your inbox. Every weekday.

ADS VIA CARBON

npm i react-bootstrap bootstrap

```
App.jsx U  Tela5.jsx U x
src > Tela5.jsx > ...
1  import Button from 'react-bootstrap/Button';
2  import 'bootstrap/dist/css/bootstrap.min.css';
3
4  export default function Tela5() {
5
6      return (
7          <>
8              <div className="container">
9                  <Button variant="primary">Primary</Button>{' '}
10                 <Button variant="secondary">Secondary</Button>{' '}
11                 <Button variant="success">Success</Button>{' '}
12                 <Button variant="warning">Warning</Button>{' '}
13                 <Button variant="danger">Danger</Button>{' '}
14                 <Button variant="info">Info</Button>{' '}
15                 <Button variant="light">Light</Button>{' '}
16                 <Button variant="dark">Dark</Button>
17                 <Button variant="link">Link</Button>
18             </div>
19         </>
20     )
21 }
```


App.jsx U

Tela5.jsx U

Tela6.jsx U X

src > Tela6.jsx > Tela6

```
1  import 'bootstrap/dist/css/bootstrap.min.css';
2  import Button from 'react-bootstrap/Button';
3  import Card from 'react-bootstrap/Card';
4
5  export default function Tela6() {
6
7      return (
8          <Card style={{ width: '18rem', margin: "10px" }}>
9              <Card.Body>
10                 <Card.Title>Lorem Ipsum</Card.Title>
11                 <Card.Text>
12                     Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam quis viverra felis.
13                 </Card.Text>
14                 <Button variant="primary">Clique Aqui</Button>
15             </Card.Body>
16         </Card>
17     )
18 }
```

