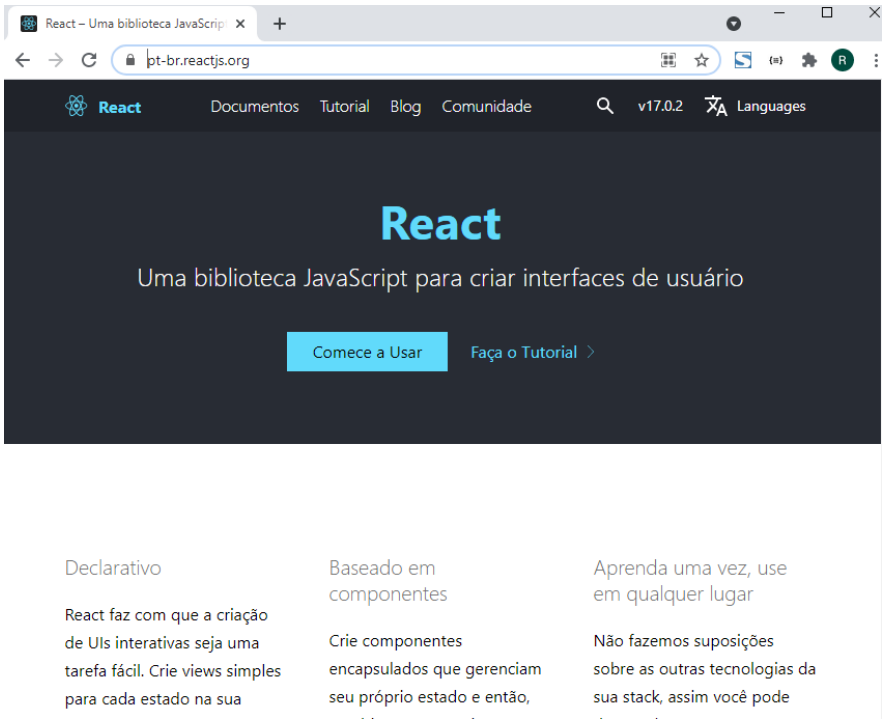
Curso de React
João Ribeiro (SYS4SOFT)

https://www.youtube.com/watch?v=C8M94QLJy0o&list=PLXik_5Br-zO9YVs9bxi7zoQlKq59VPTX1

Aula 02 - Preparação do ambiente de trabalho

https://pt-br.reactjs.org/



- Há duas formas de se utilizar o React:

1. Adicionando o React a um site (uma página HTML)
2. Criar um projeto completo de React

CDN Links

- Tanto React como ReactDOM estão disponíveis através de CDN.

```
<script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
```

- As versões acima devem ser utilizadas apenas para desenvolvimento e não são adequadas para o ambiente de produção. Versões reduzidas e otimizadas para produção estão disponíveis em:

```
<script crossorigin src="https://unpkg.com/react@17/umd/react.production.min.js"></script>
<script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.production.min.js"></script>
```

Babel

https://pt-br.reactjs.org/docs/add-react-to-a-website.html#optional-try-react-with-jsx

Experimente Rapidamente JSX

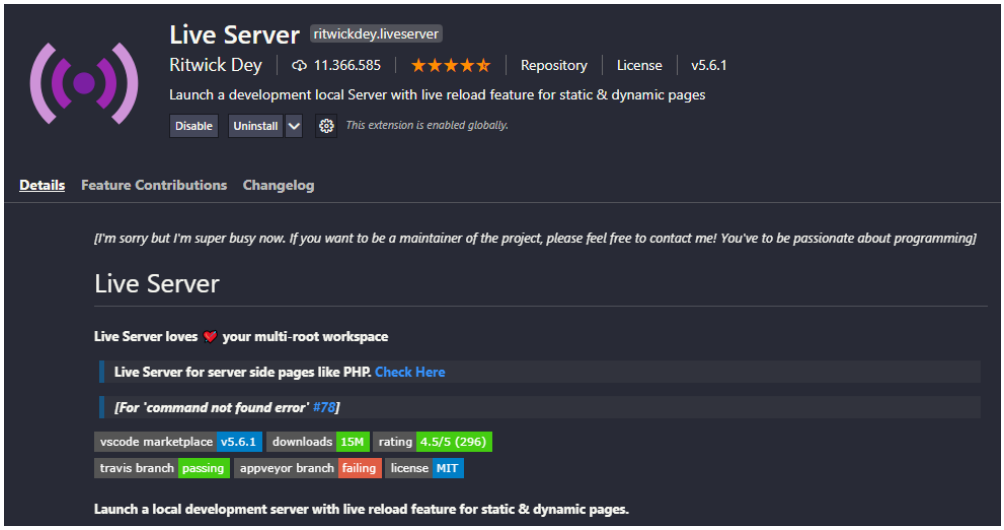
A maneira mais rápida de experimentar o JSX em seu projeto é adicionando essa tag <script> em sua página:

<script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

Agora você pode usar o JSX em qualquer tag <script> somente adicionando o atributo type="text/babel" a ele.

- Como editor use o Visual Studio Code

- Instale nele a extensão Live Server.



- O Live Serve possibilita que ao desenvolver um conjunto de código em HTML, quando o arquivo for salvo será automaticamente atualizado no browser.

- Crie uma pasta chamada React

cd React

code .

```
C:\React\app1>code .
```

- Adicione uma subpasta chamada aula_02

- E dentro dela adicione um arquivo chamado index.html

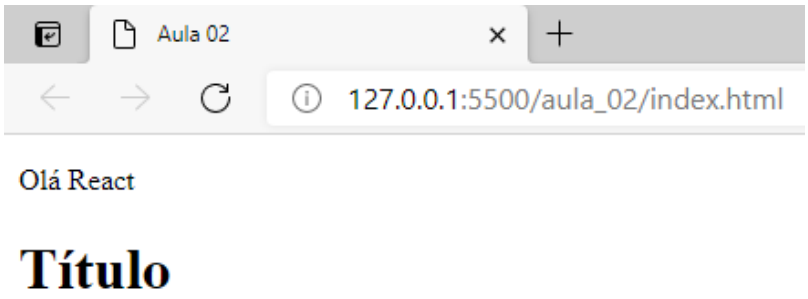
html:5

aula_02/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 02</title>
</head>
<body>
   <p>Olá React</p>
   <h1>Título</h1>
</body>
</html>
```

- Clique com o botão direito sobre o nome do arquivo e selecione a opção:

Open with Live Server

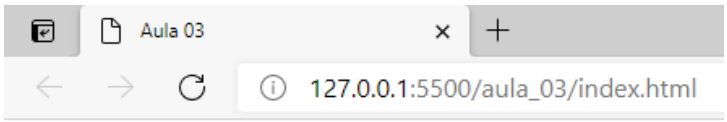


Aula 03 - Primeira experiência com ReactJS e JSX

Inserindo React dentro de uma página HTML

aula_03/index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 03</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">
      const root = document.getElementById('root');
      let nome = 'João Ribeiro';
      ReactDOM.render(<h3>O meu nome é {nome}</h3>, root);
   </script>

</body>
</html>
```

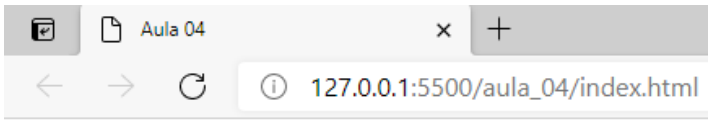


O meu nome é João Ribeiro

Aula 04 - Exemplo de um function component

aula_04\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 04</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      function MeuComponente(){
         var nome = "João";
         var sobrenome = "Ribeiro";

         return (
            <div>
               <h1>Olá React!</h1>
               <p>O meu nome é {nome} {sobrenome}</p>
            </div>
         )
      }

      const root = document.getElementById('root');
      ReactDOM.render(<MeuComponente />, root);
   </script>

</body>
</html>
```



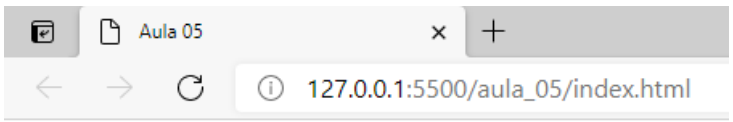
Olá React!

O meu nome é João Ribeiro

Aula 05 - Exemplo de um class component

aula_05\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 05</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class CompA extends React.Component{
         render(){
            return(
               <h3>Componente A</h3>
            )
         }
      }

      class CompB extends React.Component {
            render() {
               return(
                  <h3> Componente B</h3 >
               )
         }
      }

      class MeuComponente extends React.Component{
         render(){
            return(
               <div>
                  <CompA />
                  <hr />
                  <CompB />
               </div>
            )
         }
      }

      const root = document.getElementById('root');
      ReactDOM.render(<MeuComponente />, root);
   </script>

</body>
</html>
```



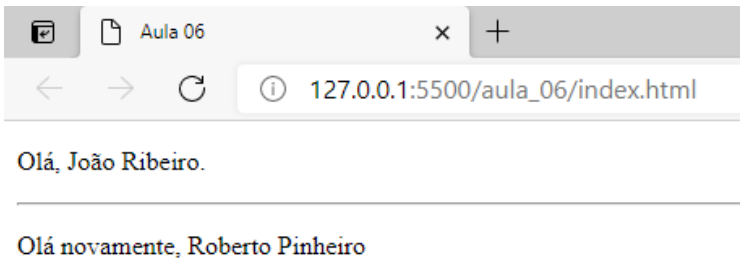
Componente A

Componente B

Aula 06 - Component props

aula_06\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 06</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      function CompA(props){
         return(
            <p>Olá, {props.nome} {props.sobrenome}.</p>
         )
      }

      class CompB extends React.Component{
         render(){
            return(
               <div>
                  <p>Olá novamente, {this.props.nome} {this.props.sobrenome}</p>
               </div>
            )
         }
      }

      class MeuComponente extends React.Component{
         render(){
            return(
               <div>
                  <CompA nome="João" sobrenome="Ribeiro" />
                  <hr />
                  <CompB nome="Roberto" sobrenome="Pinheiro" />
               </div>
            )
         }
      }

      const root = document.getElementById('root');
      ReactDOM.render(<MeuComponente />, root);
   </script>

</body>
</html>
```

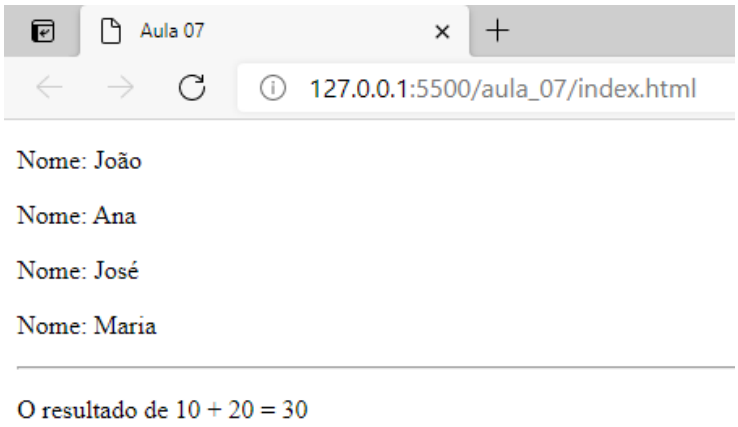

Aula 06
127.0.0.1:5500/aula_06/index.html
Olá, João Ribeiro.
Olá novamente, Roberto Pinheiro

Aula 07 - Mais aspectos sobre o uso de props

aula_07\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 07</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      function CompA(props){
         return(
            <p>Nome: {props.nome}</p>
         )
      }

      function CompB() {
         return (
            <div>
               <CompA nome="João" />
               <CompA nome="Ana" />
               <CompA nome="José" />
               <CompA nome="Maria" />
            </div>
         )
      }

      class Soma extends React.Component{
         render(){
            return(
               <div>
                  <p>O resultado de {this.props.a} + {this.props.b} = {this.props.a + this.props.b}</p>
               </div>
            )
         }
      }
```

```
    class MeuComponente extends React.Component{
      render(){
        return(
          <div>
            <CompB />
            <hr />
            <Soma a={10} b={20} />
          </div>
        )
      }
    }

    const root = document.getElementById('root');
    ReactDOM.render(<MeuComponente />, root);
  </script>

</body>
</html>
```

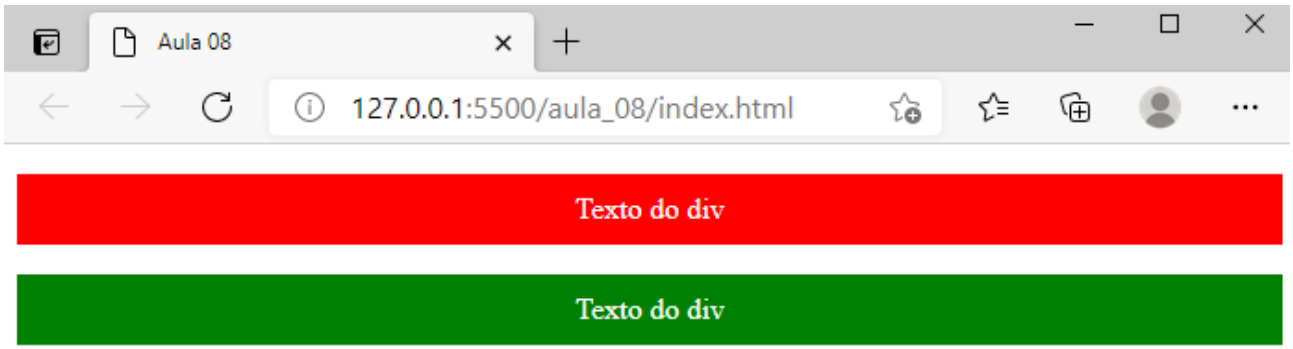


Aula 08 - Como usar CSS em REACT Components

aula_08\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 08</title>
   <style>
      .red{
         background-color: red;
         color: white;
         text-align: center;
         padding: 10px;
      }
      .green{
         background-color: green;
         color: white;
         text-align: center;
         padding: 10px;
      }
   </style>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class CompA extends React.Component{
         render(){
            return (
               <div>
                  <p className="red">Texto do div</p>
               </div >
            )
         }
      }
```

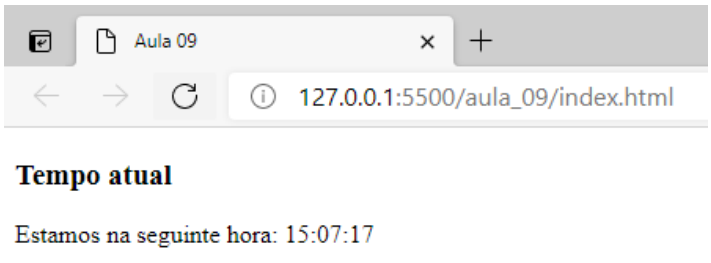
```
    class CompB extends React.Component {
      render(){
        return (
          <div>
            <p className="green">Texto do div</p>
          </div >
        )
      }
    }

    class App extends React.Component {
      render(){
        return (
          <div>
            <CompA />
            <CompB />
          </div>
        )
      }
    }

    const root = document.getElementById('root');
    ReactDOM.render(<App />, root);
  </script>

</body>
</html>
```



Aula 09 - Incluir o ReactDom Render dentro de um componente

aula_09\index.html

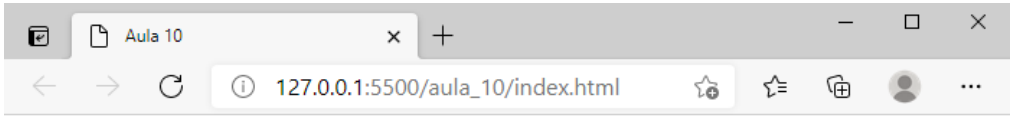
```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 09</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      function Tempo(){
         // jsx a apresentar
         const conteudo = (
            <div>
               <h3>Tempo atual</h3>
               <p>Estamos na seguinte hora: {new Date().toLocaleTimeString()}</p>
            </div>
         )

         ReactDOM.render(
            conteudo,
            document.getElementById('root')
         )
      }

      setInterval(Tempo, 1000);

   </script>

</body>
</html>
```



Aula 10 - Inline styling dentro de um componente

aula_10\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 10</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component {
         render(){

            const estilo = {
               color: "red",
               backgroundColor: "yellow",
               textAlign: "center",
               padding: "30px",
               fontSize: "50px"
            }

            return (
               <div>
                  <p style={estilo}>Texto do componente</p>
               </div>
            )
         }
      }

      const root = document.getElementById('root');
      ReactDOM.render(<App />, root);
   </script>

</body>
</html>
```

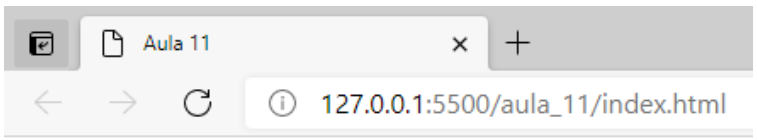


Texto do componente

Aula 11 - Introdução ao conceito de state

aula_11\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 11</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component{

         constructor(){
            super()

            this.state = {
               nome: "João",
               idade: 45
            }
         }

         render(){
            return(
               <div>
                  <p>Nome: {this.state.nome} {this.props.sobrenome}</p>
                  <p>Idade: {this.state.idade}</p>
               </div>
            )
         }
      }

      const root = document.getElementById('root');
      ReactDOM.render(<App sobrenome="Ribeiro" />, root);
   </script>

</body>
</html>
```



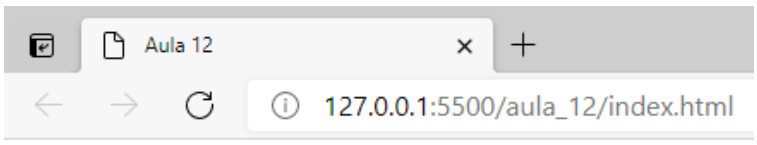
Nome: João Ribeiro

Idade: 45

Aula 12 - Diferença entre props e state

aula_12\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 12</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component{

         constructor(){
            super()

            this.state = {
               nome: "João"
            }
         }

         render(){

            this.state.nome = "Joaquim"

            return(
               <div>
                  <p>Nome: {this.state.nome}</p>
                  <p>Idade: {this.props.idade}</p>
               </div>
            )
         }
      }

      const root = document.getElementById('root');
      ReactDOM.render(<App idade="45" />, root);
   </script>

</body>
</html>
```



Nome: Joaquim

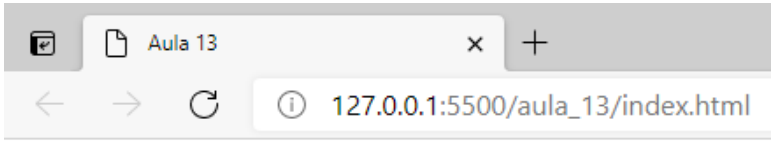
Idade: 45

Aula 13 - Utilização de state e props entre parente e child components

aula_13\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 13</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      // identificação

      class Identificacao extends React.Component{
         constructor(){
            super()
            this.state = {
               nome: "João Ribeiro",
               idade: 46
            }
         }

         render(){

            this.state.nome = "Joaquim"

            return(
               <div>
                  <Nome nome={this.state.nome} />
                  <Idade idade={this.state.idade} />
               </div>
            )
         }
      }

      class Nome extends React.Component{
         render(){
            return(
               <div>
                  <p>Nome: {this.props.nome} </p>
               </div>
            )
         }
      }
```

```
    class Idade extends React.Component{
      render(){
        return(
          <div>
            <p>Idade: {this.props.idade}</p>
          </div>
        )
      }
    }

    const root = document.getElementById('root');
    ReactDOM.render(<Identificacao />, root);
  </script>

</body>
</html>
```



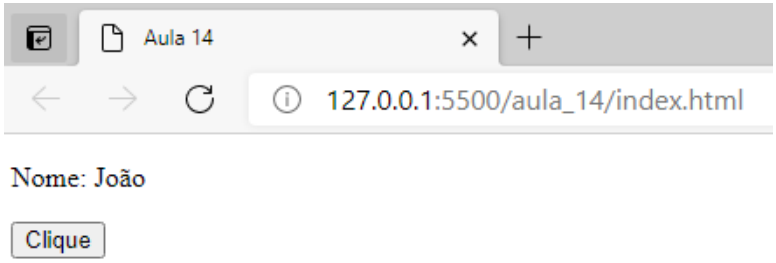
Nome: Joaquim

Idade: 46

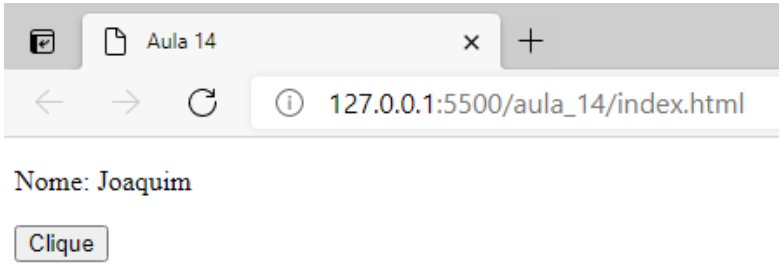
Aula 14 - Introdução aos eventos em ReactJS

aula_14\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 14</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component{

         constructor(){
            super()

            this.state = {
               nome: "João"
            }

            this.cmd_click = this.cmd_click.bind(this)
         }

         // evento click
         cmd_click(){
            this.setState({nome: "Joaquim"})
         }

         render(){
            return(
               <div>
                  <p>Nome: {this.state.nome}</p>
                  <button onClick={this.cmd_click}>Clique</button>
               </div>
            )
         }

      }

      const root = document.getElementById('root');
      ReactDOM.render(<App />, root);
   </script>

</body>
</html>
```



- Clique no botão

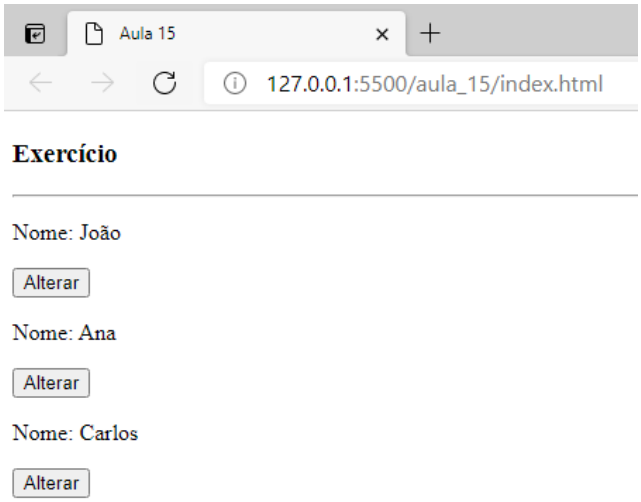


Aula 15 - Exercício prático

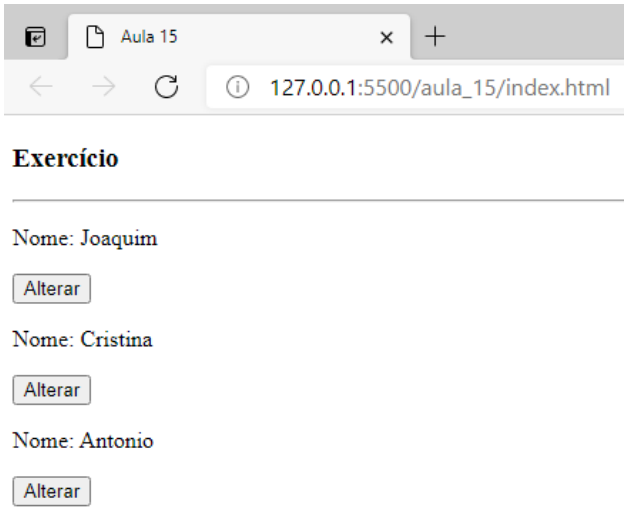
aula_15\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 15</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class CompNome extends React.Component{
         constructor(props){
            super(props)
            this.state = {
               nome: this.props.nome_inicial
            }

            //binding
            this.cmd_click = this.cmd_click.bind(this)
         }

         cmd_click(){
            this.setState({nome: this.props.nome_final})
         }

         render(){
            return(
               <div>
                  <p>Nome: {this.state.nome}</p>
                  <button onClick={this.cmd_click}>Alterar</button>
               </div>
            )
         }

      }
```

```
    class App extends React.Component{
      render(){
        return(
          <div>
            <h3>Exercício</h3>
            <hr />
            <CompNome nome_inicial="João" nome_final="Joaquim" />
            <CompNome nome_inicial="Ana" nome_final="Cristina" />
            <CompNome nome_inicial="Carlos" nome_final="Antonio" />
          </div>
        )
      }

    }

    const root = document.getElementById('root');
    ReactDOM.render(<App />, root);
  </script>

</body>
</html>
```



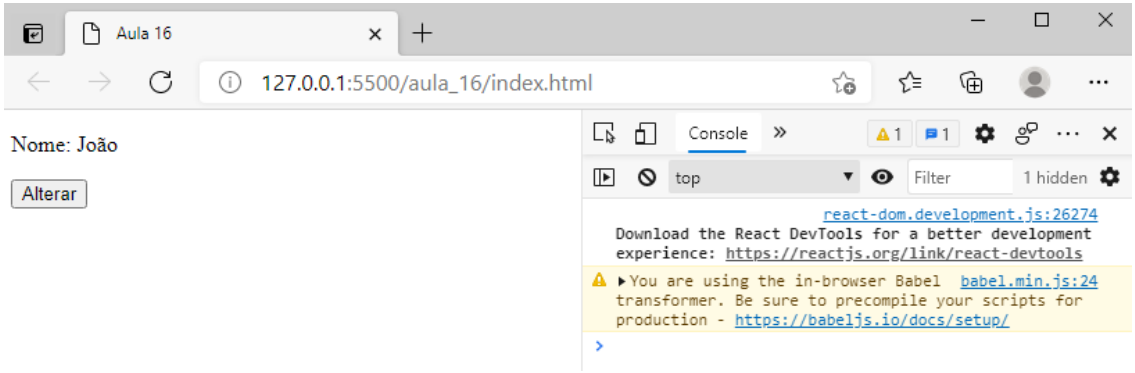
- Clicando em cada um dos três botões:



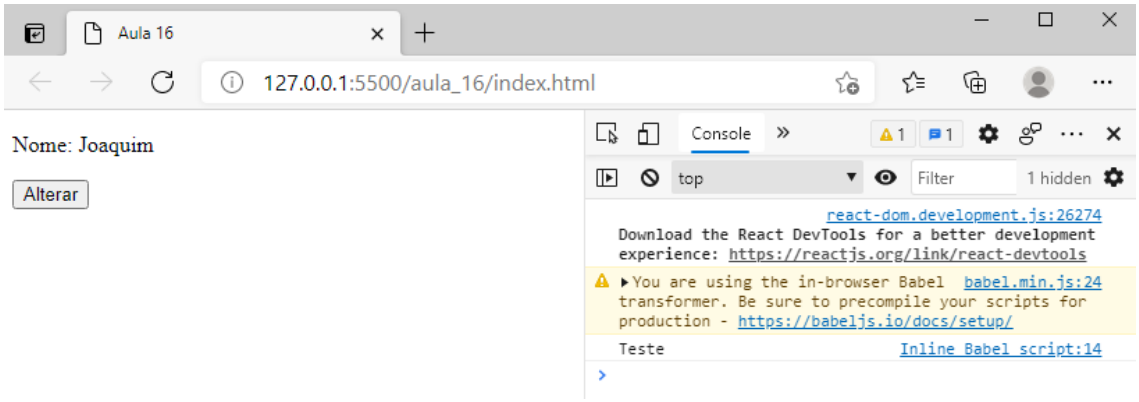
Aula 16 - Simplificar bindings com arrow functions

aula_16\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 16</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component{

         constructor(){
            super()

            this.state = {
               nome: "João"
            }
         }

         metodo = () => {
            console.log("Teste")
            this.setState({ nome: "Joaquim" })
         }

         render(){
            return(
               <div>
                  <p>Nome: {this.state.nome}</p>
                  <button onClick={this.metodo}>Alterar</button>
               </div>
            )
         }

      }

      const root = document.getElementById('root');
      ReactDOM.render(<App />, root);
   </script>

</body>
</html>
```



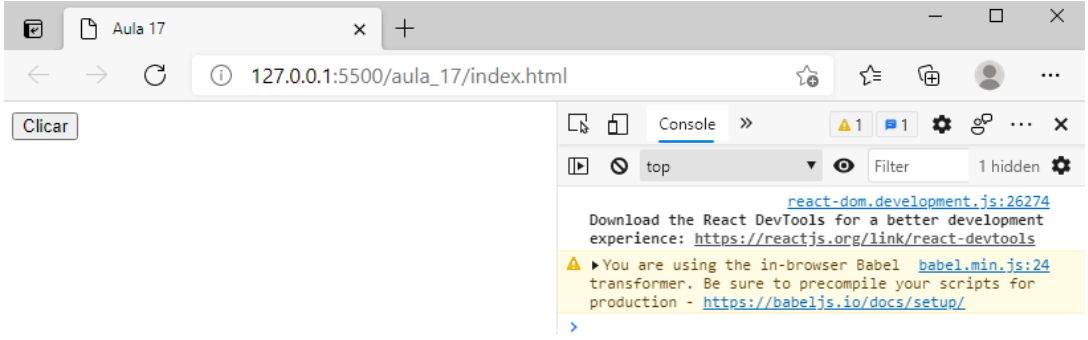
- Clicando no botão:



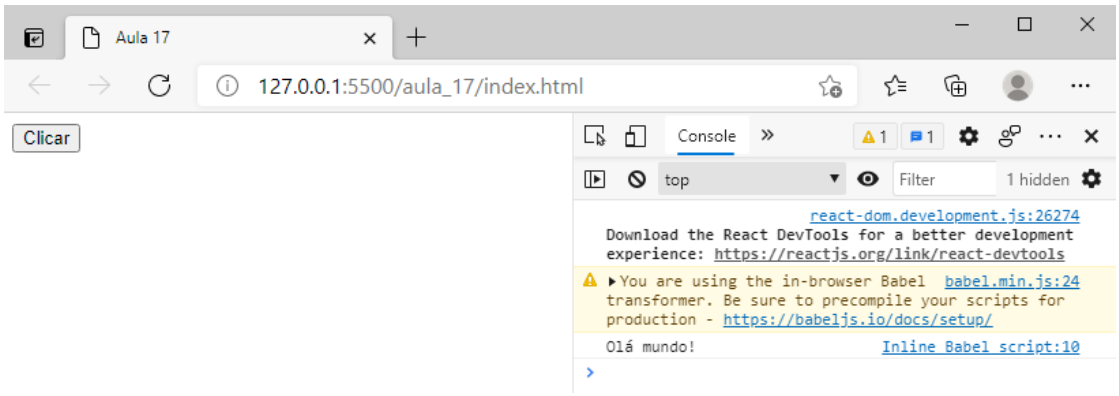
Aula 17 - Passagem de parâmetros para eventos

aula_17\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 17</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component{

         metodo = (texto) => {
            console.log(texto)
         }

         render(){
            return(
               <div>
                  <button onClick={() => this.metodo("Olá mundo!")}>Clicar</button>
               </div>
            )
         }

      }

      const root = document.getElementById('root');
      ReactDOM.render(<App />, root);
   </script>

</body>
</html>
```



- Clicando no botão:

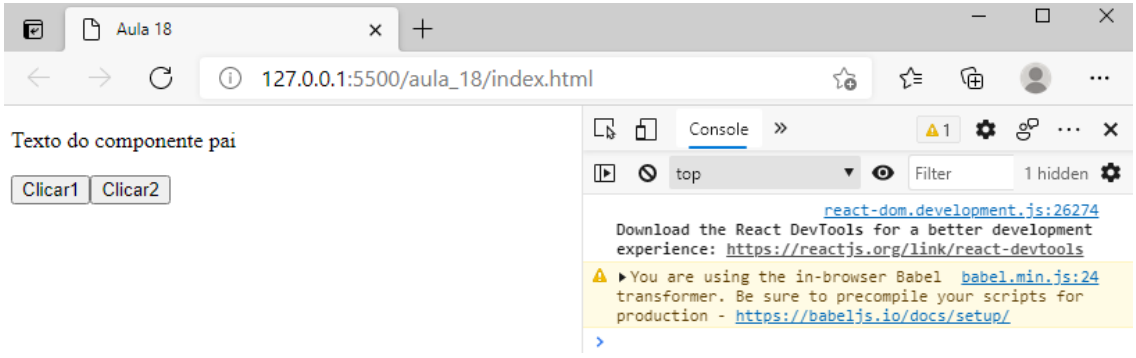


Aula 18 - Usar métodos como props

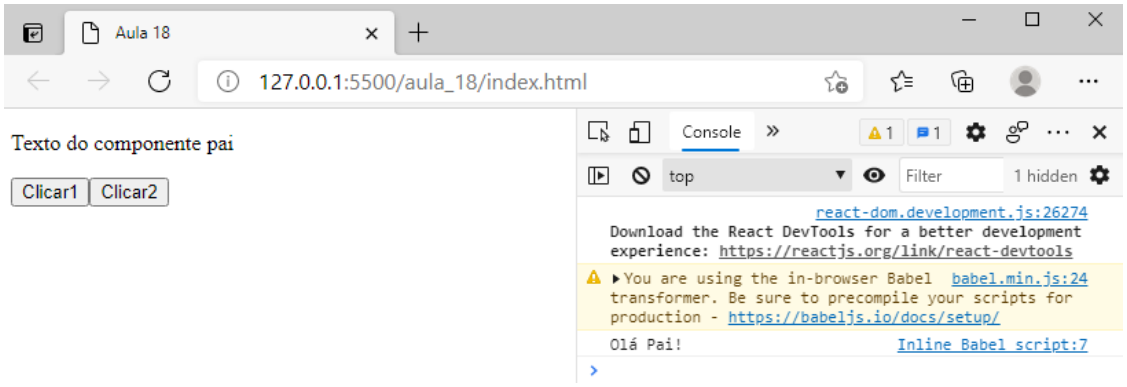
aula_18\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 18</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      // parent
      class App extends React.Component{

         met1(){
            console.log("Olá Pai!")
         }

         met2() {
            console.log("Olá Mundo!")
         }

         render(){
            return(
               <div>
                  <p>Texto do componente pai</p>
                  <CompA met1={this.met1} met2={this.met2} />
               </div>
            )
         }

      }

      // child
      class CompA extends React.Component{
         render(){
            return(
               <div>
                  <button onClick={this.props.met1}>Clicar1</button>
                  <button onClick={this.props.met2}>Clicar2</button>
               </div>
            )
         }
      }
```

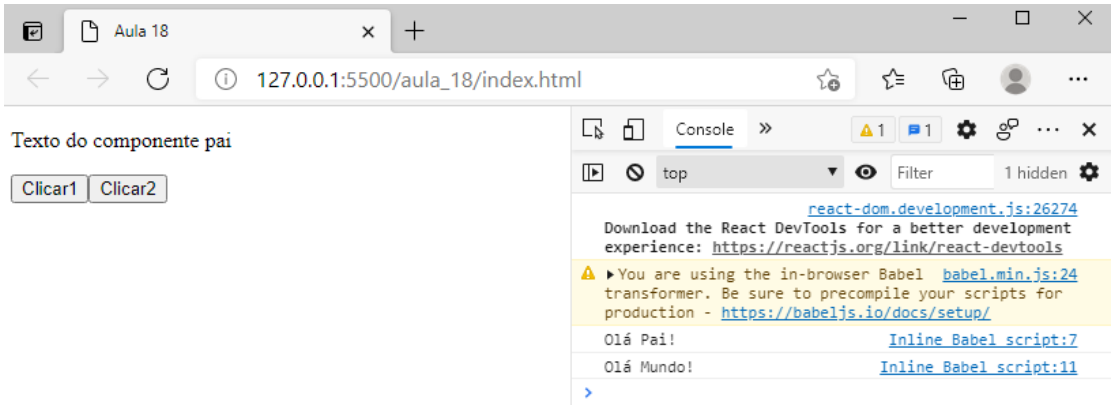
```
    const root = document.getElementById('root');
    ReactDOM.render(<App />, root);
  </script>

</body>
</html>
```



- Clicando em "Clicar1":



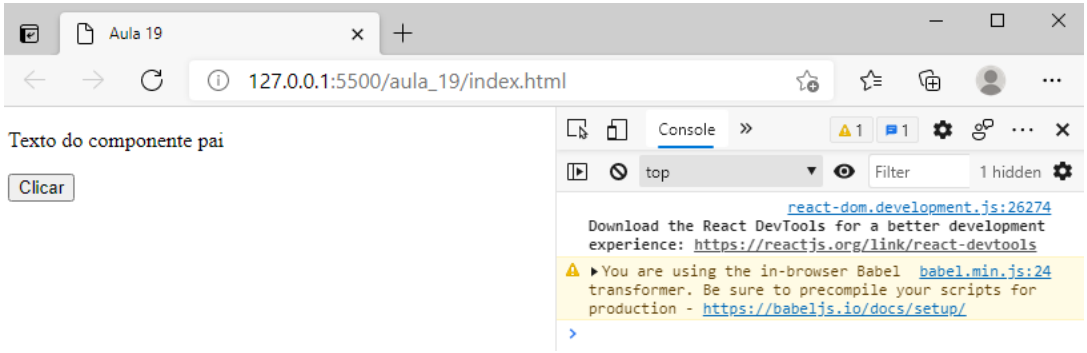
- Clicando em "Clicar2"



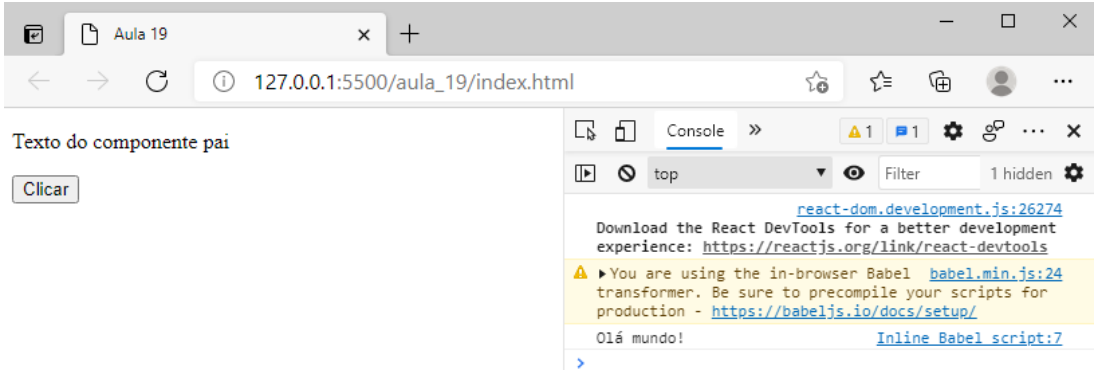
Aula 19 - Passar argumentos de child para parent component

aula_19\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 19</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      // parent
      class App extends React.Component{

         metodo(texto){
            console.log(texto)
         }

         render(){
            return(
               <div>
                  <p>Texto do componente pai</p>
                  <Child evento={this.metodo} />
               </div>
            )
         }

      }

      // child
      class Child extends React.Component{
         render(){
            return(
               <div>
                  <button onClick={()=>this.props.evento('Olá mundo!') }>Clicar</button>
               </div>
            )
         }
      }

      const root = document.getElementById('root');
      ReactDOM.render(<App />, root);
   </script>

</body>
</html>
```



- Clicando no botão:

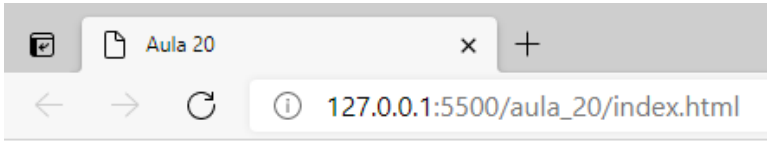


Aula 20 - Quatro formas de renderização condicional

aula_20\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 20</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component{

         constructor(){
            super()

            this.state = {
               tempoBom: true
            }
         }

         render(){

            return(
               this.state.tempoBom && <p>O tempo está bom</p>
            )

            // ================================================================

            // Renderização condicional 3 - Método ternário (condição ternária)

            //return(
            //    this.state.tempoBom ? <p>O tempo está bom</p> : <p>O tempo está ruim</p>
            //)

            // ================================================================

            // Renderização condicional 2 - Usando variável

            //let texto

            //if(this.state.tempoBom){
            //    texto = <p>O tempo está bom</p>
            //} else {
            //    texto = <p>O tempo está ruim</p>
            //}
```

```
            //return texto

            // ================================================================

            // Renderização condicional 1 - if-else

            //if(this.state.tempoBom){
            //    return <p>O tempo está bom</p>
            //} else {
            //    return <p>O tempo está ruim</p>
            //}

        }

    }

    const root = document.getElementById('root');
    ReactDOM.render(<App />, root);
  </script>

</body>
</html>
```



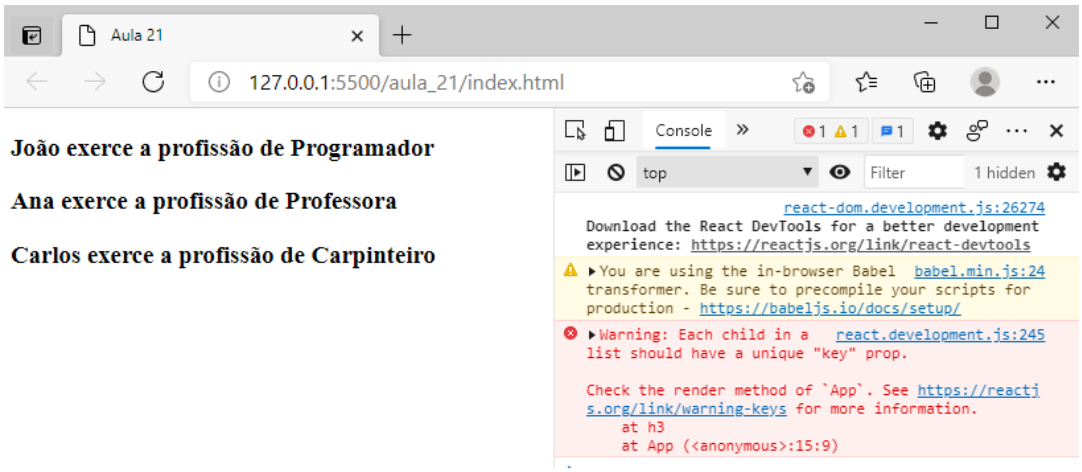
O tempo está bom

Aula 21 - Renderização de listas e função map

aula_21\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 21</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component{

         render(){

            const nomes = [
               'João',
               'Ana',
               'Carlos'
            ]

            const pessoas = [
               {
                  nome: "João",
                  profissao: "Programador"
               },
               {
                  nome: "Ana",
                  profissao: "Professora"
               },
               {
                  nome: "Carlos",
                  profissao: "Carpinteiro"
               }
            ]

            const final = pessoas.map(pessoa => <h3>{pessoa.nome} exerce a profissão de {pessoa.profissao}</h3>)

            return(
               <div>
                  {final}
               </div>
            )
         }

      }
```

```
    const root = document.getElementById('root');
    ReactDOM.render(<App />, root);

    //let nomes = ['João', 'Ana', 'Carlos']
    //let final = nomes.map(n => 'Bom dia, ' + n)

    //console.log(nomes)
    //console.log(final)

  </script>

</body>
</html>
```

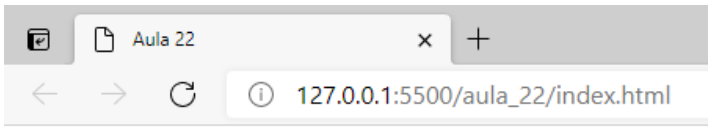


Aula 22 - Exemplo de renderização de listas de notícias

aula_22\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 22</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component{

         render(){

            const noticias = [
               {
                  titulo: "Notícia 1",
                  texto: "Texto da notícia 1",
                  autor: "Autor 1"
               },
               {
                  titulo: "Notícia 2",
                  texto: "Texto da notícia 2",
                  autor: "Autor 2"
               },
               {
                  titulo: "Notícia 3",
                  texto: "Texto da notícia 3",
                  autor: "Autor 3"
               }
            ]

            // preparação do component child Noticia

            const final = noticias.map(noticia => <Noticia noticia={noticia} />)

            return(
               <div>
                  <h2>Notícias</h2>
                  <hr />
                  {final}
               </div>
            )
         }
```

```
    }

    class Noticia extends React.Component{

      constructor(props){
        super(props)
      }

      render(){
        return(
          <div>
            <h3>{this.props.noticia.titulo}</h3>
            <p>{this.props.noticia.texto}</p>
            <small><i>{this.props.noticia.autor}</i></small>
          </div>
        )
      }
    }

    const root = document.getElementById('root');
    ReactDOM.render(<App />, root);

  </script>

</body>
</html>
```



Notícias

Notícia 1

Texto da notícia 1

Autor 1

Notícia 2

Texto da notícia 2

Autor 2

Notícia 3

Texto da notícia 3

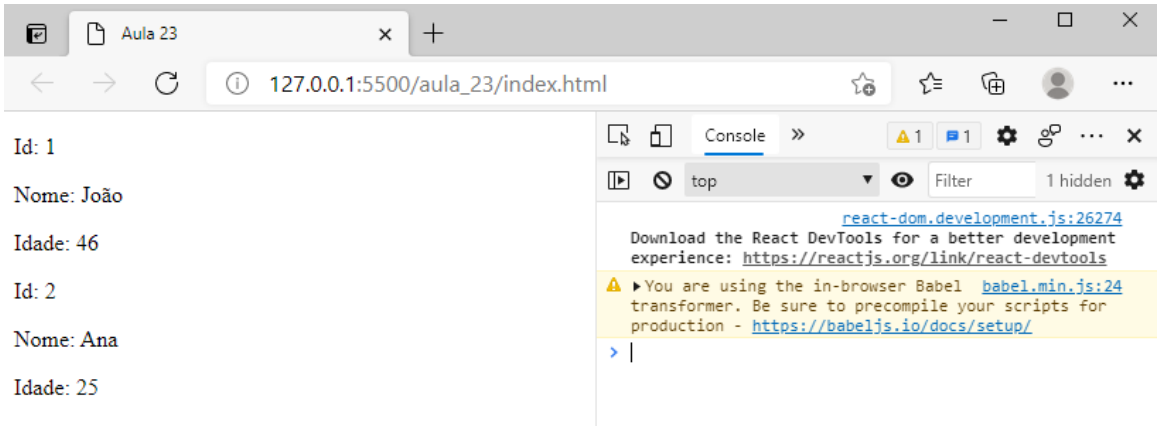
Autor 3

Aula 23 - Renderização de listas e unique keys

aula_23\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 23</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component{

         render(){

            const socios = [
               {
                  id: 1,
                  nome: "João",
                  idade: 46
               },
               {
                  id: 2,
                  nome: "Ana",
                  idade: 25
               }
            ]

            const final = socios.map(socio => <MostrarSocio key={socio.id} socio={socio} />)

            return(
               <div>
                  {final}
               </div>
            )
         }

      }

      class MostrarSocio extends React.Component{
         constructor(props){
            super(props)
         }
```

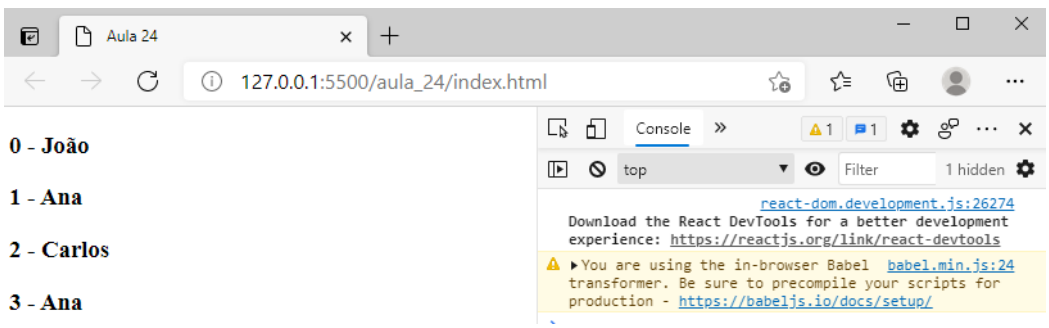
```
        render(){
            return (
                <div>
                    <p>Id: {this.props.socio.id}</p>
                    <p>Nome: {this.props.socio.nome}</p>
                    <p>Idade: {this.props.socio.idade}</p>
                </div>
            )
        }
    }

    const root = document.getElementById('root');
    ReactDOM.render(<App />, root);

  </script>

</body>
</html>
```



Aula 24 - Recorrendo ao index na renderização de coleções

aula_24\index.html

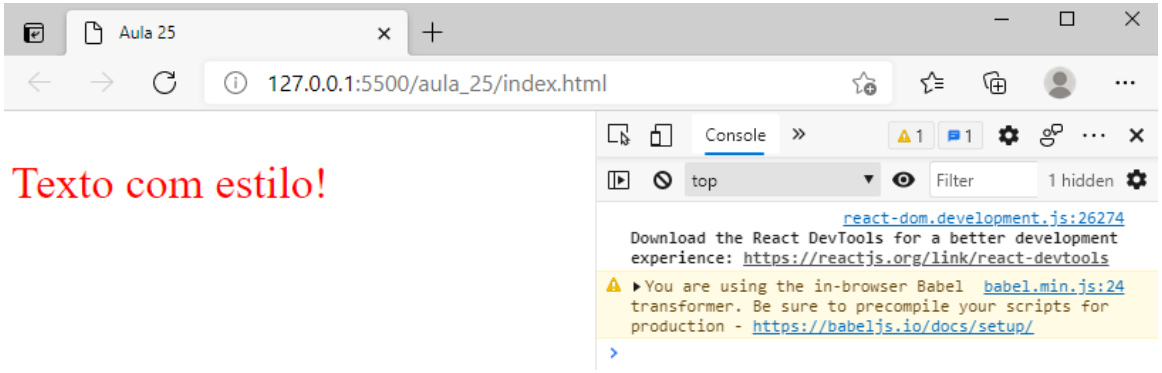
```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 24</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component{

         render(){

            const nomes = ["João", "Ana", "Carlos", "Ana"]

            const final = nomes.map((nome, index)=><h3 key={index}>{index} - {nome}</h3>)

            return(
               <div>
                  {final}
               </div>
            )
         }

      }

      const root = document.getElementById('root');
      ReactDOM.render(<App />, root);

   </script>

</body>
</html>
```



Aula 25 - Vamos voltar ao uso de CSS no React

aula_25\index.html

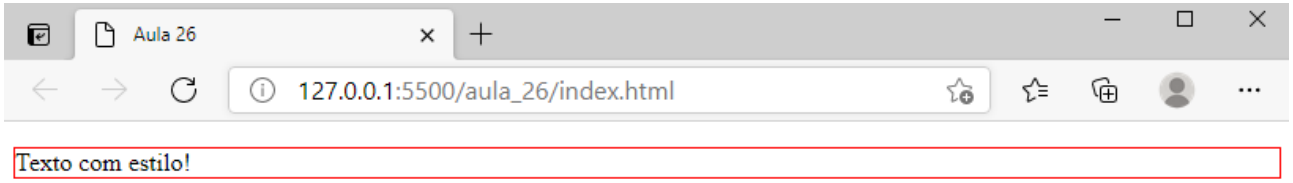
```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 25</title>
   <style>
      .cor-vermelho{
         color: red;
      }

      .tamanho{
         font-size: 2em;
      }
   </style>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component{

         constructor(props){
            super(props)
         }

         render(){

            let cor = this.props.cor ? 'cor-vermelho' : ''
            let tamanho = this.props.tamanho ? 'tamanho' : ''

            return(
               <div>
                  <p className={`${cor} ${tamanho}`}>Texto com estilo!</p>
               </div>
            )
         }
      }

      const root = document.getElementById('root');
      ReactDOM.render(<App cor={true} tamanho={true} />, root);

   </script>

</body>
</html>
```


Aula 25
127.0.0.1:5500/aula_25/index.html
Texto com estilo!
Console
top
Filter
1 hidden
react-dom.development.js:26274
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtools
You are using the in-browser Babel transformer. Be sure to precompile your scripts for production - https://babeljs.io/docs/setup/
babel.min.js:24

Aula 26 - Utilização de estilos inline

aula_26\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 26</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component{

         render(){

            let estilo = {
               color: "red",
               fontSize: "2em"
            }

            let estilo2 = {
               border: "1px solid red"
            }

            return(
               <div>
                  <p style={estilo2}>Texto com estilo!</p>
               </div>
            )
         }

      }

      const root = document.getElementById('root');
      ReactDOM.render(<App />, root);

   </script>

</body>
</html>
```

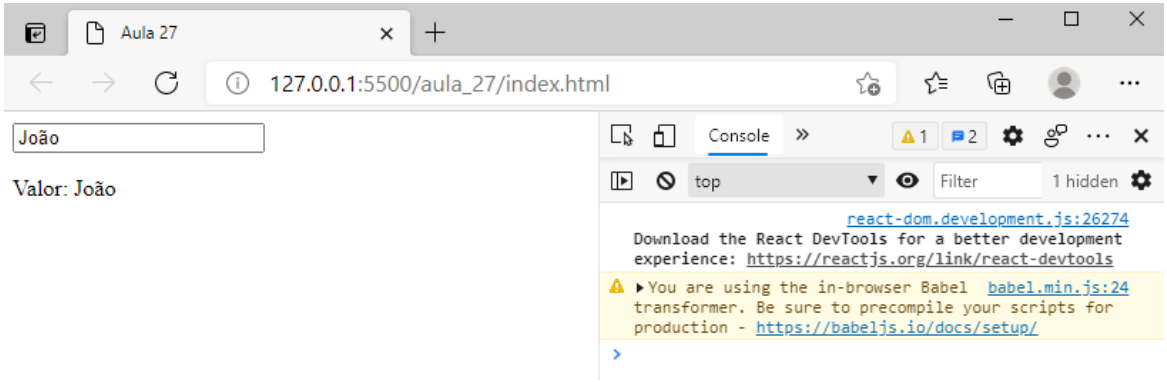


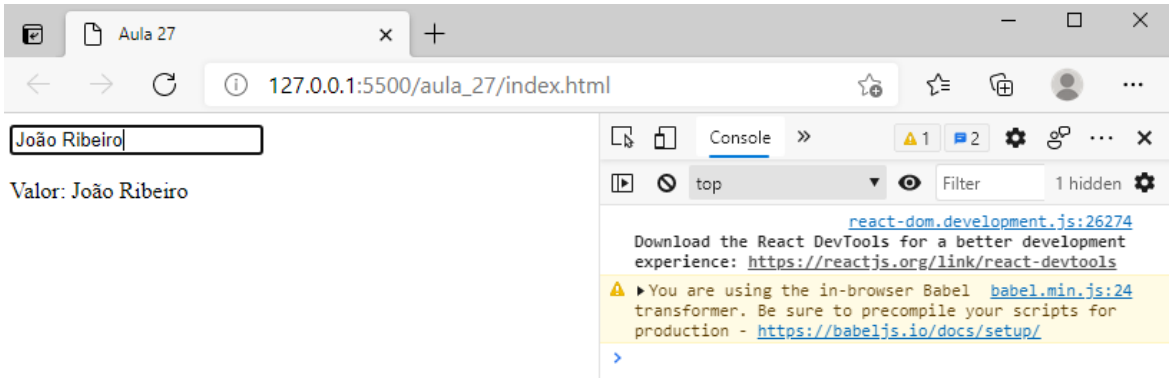
Texto com estilo!

Aula 27 - React e inputs de formulários

aula_27\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 27</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component{

         constructor(){
            super()
            this.state = {
               nome: "João"
            }
         }

         alterarNome = (event) => {
            this.setState({nome: event.target.value})
         }

         render(){

            return(
               <div>
                  <input type="text" value={this.state.nome} onChange={this.alterarNome}
                  />

                  <p>Valor: {this.state.nome}</p>
               </div>
            )
         }
      }

      const root = document.getElementById('root');
      ReactDOM.render(<App />, root);

   </script>

</body>
</html>
```

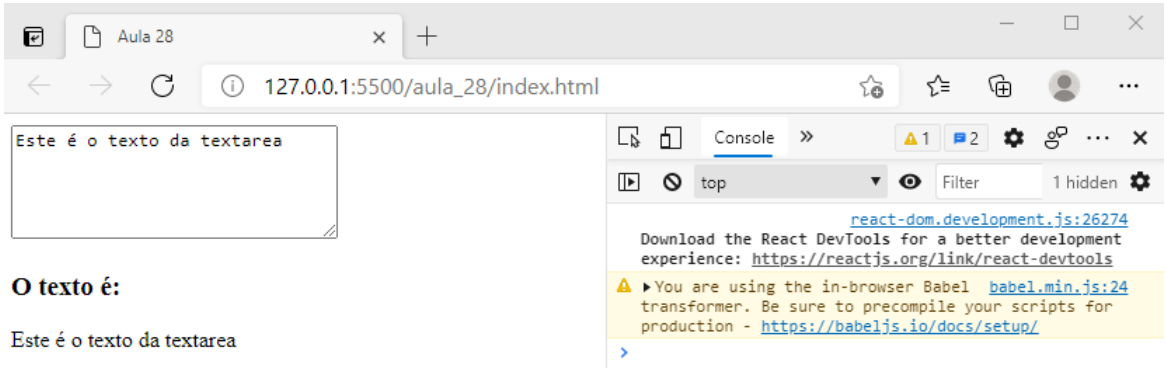

Aula 27
127.0.0.1:5500/aula_27/index.html
João
Valor: João
Console
top
Filter
1 hidden
react-dom.development.js:26274
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtools
You are using the in-browser Babel transformer. Be sure to precompile your scripts for production - https://babeljs.io/docs/setup/
babel.min.js:24

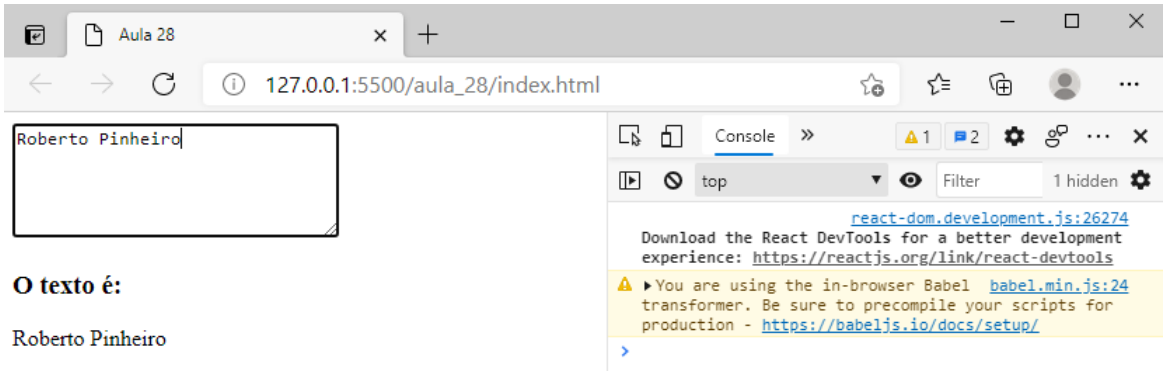

Aula 27
127.0.0.1:5500/aula_27/index.html
João Ribeiro
Valor: João Ribeiro
Console
top
Filter
1 hidden
react-dom.development.js:26274
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtools
You are using the in-browser Babel transformer. Be sure to precompile your scripts for production - https://babeljs.io/docs/setup/
babel.min.js:24

Aula 28 - Exemplo com textarea

aula_28\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 28</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component{

         constructor(){
            super()

            this.state = {
               texto: "Este é o texto da textarea"
            }
         }

         alterarTexto = (event) => {
            this.setState({texto: event.target.value})
         }

         render(){

            return(
               <div>
                  <textarea cols="30" rows="5" value={this.state.texto} onChange={this.alterarTexto}></textarea>
                  <h3>O texto é:</h3>
                  <p>{this.state.texto}</p>
               </div>
            )
         }

      }

      const root = document.getElementById('root');
      ReactDOM.render(<App />, root);

   </script>

</body>
</html>
```


Aula 28
127.0.0.1:5500/aula_28/index.html
Este é o texto da textarea
O texto é:
Este é o texto da textarea
Console
top
Filter
1 hidden
react-dom.development.js:26274
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtools
You are using the in-browser Babel transformer. Be sure to precompile your scripts for production - https://babeljs.io/docs/setup/
babel.min.js:24

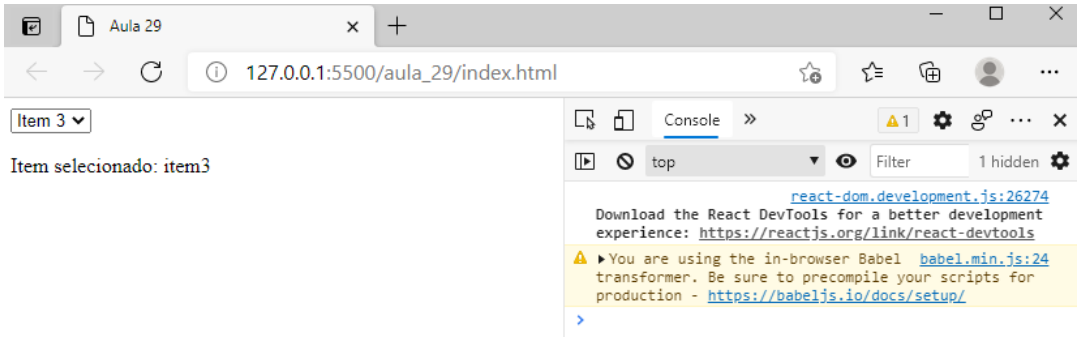

Aula 28
127.0.0.1:5500/aula_28/index.html
Roberto Pinheiro
O texto é:
Roberto Pinheiro
Console
top
Filter
1 hidden
react-dom.development.js:26274
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtools
You are using the in-browser Babel transformer. Be sure to precompile your scripts for production - https://babeljs.io/docs/setup/
babel.min.js:24

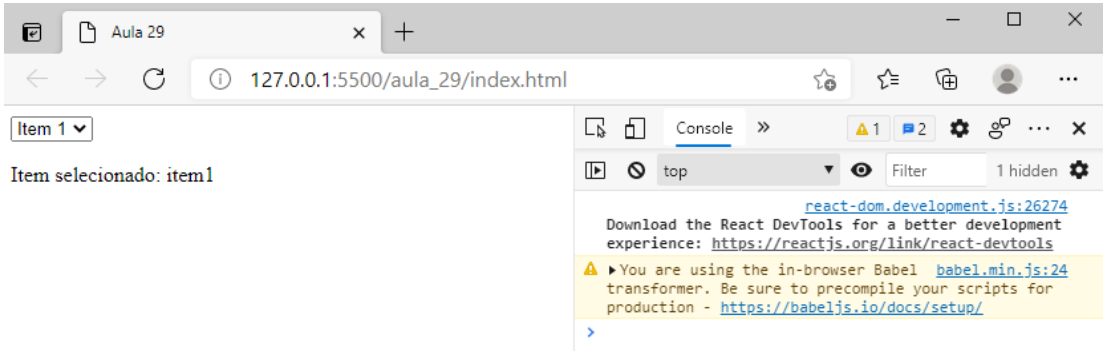
Aula 29 - Exemplo com select

aula_29\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 29</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component{

         constructor(){
            super()

            this.state = {
               item: "item3"
            }
         }

         alterarItem = e => {
            this.setState({item: e.target.value})
         }

         render(){

            return(
               <div>
                  <select value={this.state.item} onChange={this.alterarItem}>
                     <option value="item1">Item 1</option>
                     <option value="item2">Item 2</option>
                     <option value="item3">Item 3</option>
                  </select>

                  <p>Item selecionado: {this.state.item}</p>

               </div>
            )
         }
      }

      const root = document.getElementById('root');
      ReactDOM.render(<App />, root);

   </script>
```

```
</body>
</html>
```


Aula 29
127.0.0.1:5500/aula_29/index.html
Item 3
Item selecionado: item3
Console
top
Filter
1 hidden
react-dom.development.js:26274
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtools
You are using the in-browser Babel transformer. Be sure to precompile your scripts for production - https://babeljs.io/docs/setup/
babel.min.js:24

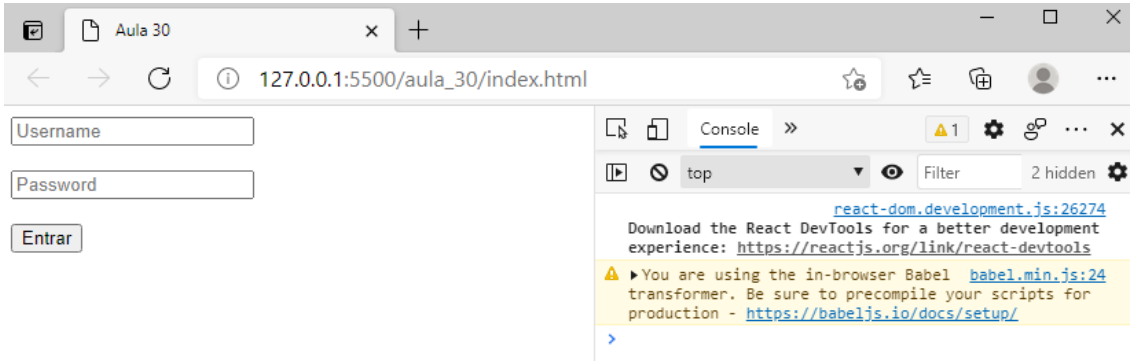

Aula 29
127.0.0.1:5500/aula_29/index.html
Item 1
Item selecionado: item1
Console
top
Filter
1 hidden
react-dom.development.js:26274
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtools
You are using the in-browser Babel transformer. Be sure to precompile your scripts for production - https://babeljs.io/docs/setup/
babel.min.js:24

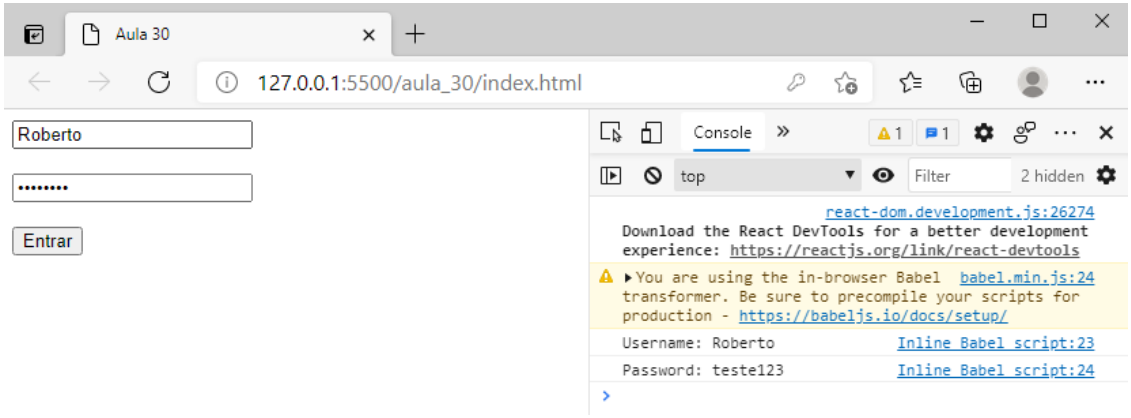
Aula 30 - Formulário de Login

aula_30\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 30</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component{

         constructor(){
            super()

            this.state = {
               username: '',
               password: ''
            }
         }

         alterarUsername = e => {
            this.setState({username: e.target.value})
         }

         alterarPassword = e => {
            this.setState({password: e.target.value})
         }

         submeterForm = e => {
            console.log("Username: " + this.state.username)
            console.log("Password: " + this.state.password)
            e.preventDefault()
         }

         render(){

            return(
               <div>
                  <form onSubmit={this.submeterForm}>
                     <input type="text" value={this.state.username} onChange={this.alterarUsername}
placeholder="Username" />
                     <br /><br />
                     <input type="password" value={this.state.password} onChange={this.alterarPassword}
placeholder="Password" />
```

```
                <br /><br />
                <input type="submit" value="Entrar" />
              </form>
            </div>
          )
        }

      }

      const root = document.getElementById('root');
      ReactDOM.render(<App />, root);

    </script>

</body>
</html>
```



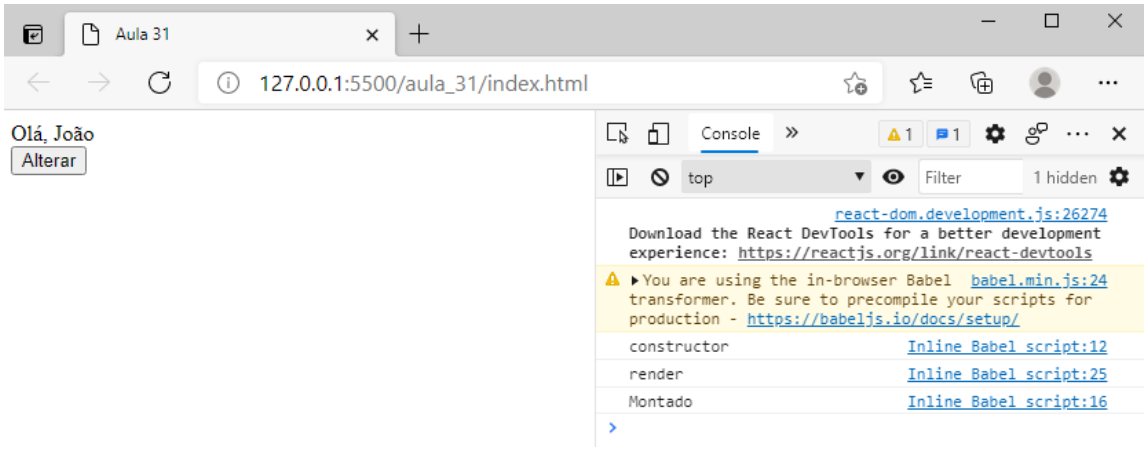


Aula 31 - Breve introdução aos lifecycle methods

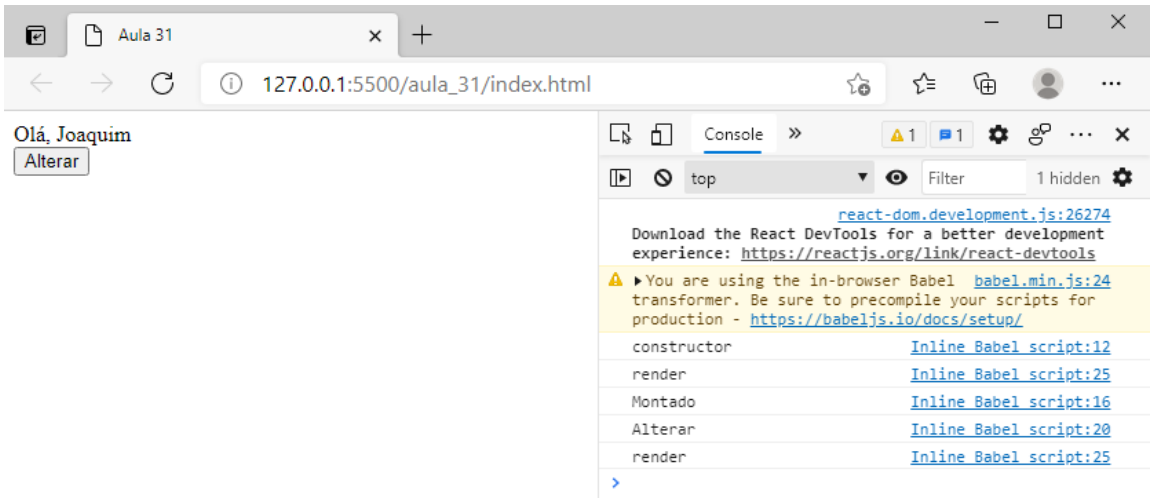
aula_31\index.html

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Aula 31</title>
</head>
<body>

   <div id="root"></div>

   <script crossorigin src="https://unpkg.com/react@17/umd/react.development.js"></script>
   <script crossorigin src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
   <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

   <script type="text/babel">

      class App extends React.Component{

         constructor(){
            super()

            this.state = {
               nome: "João"
            }

            console.log("constructor")
         }

         componentDidMount(){
            console.log("Montado")
         }

         alterar = () => {
            console.log("Alterar")
            this.setState({nome: "Joaquim"})
         }

         render(){
            console.log("render")
            return(
               <div>
                  Olá, {this.state.nome}
                  <div>
                     <button onClick={this.alterar}>Alterar</button>
                  </div>
               </div>
            )
         }

      }
```

```
    const root = document.getElementById('root');
    ReactDOM.render(<App />, root);

  </script>

</body>
</html>
```



- Clique no botão "Alterar":



Aula 32 - Criar uma React APP

Crie um novo React App

Use uma toolchain integrada para uma melhor experiência de usuário e desenvolvedor.

Esta página descreve algumas toolchains populares com React que ajudam em tarefas como:

- Escalar para muitos arquivos e componentes.
- Usar bibliotecas de terceiros através do npm.
- Detectar erros comuns cedo.
- Edição em tempo real de CSS e JS em desenvolvimento.
- Otimizar a saída para produção

Create React App

Create React App é um ambiente confortável para aprender React, e é a melhor maneira de começar um single-page application em React.

Além de configurar seu ambiente de desenvolvimento para utilizar as funcionalidades mais recentes do JavaScript, ele fornece uma experiência de desenvolvimento agradável, e otimiza o seu app para produção. Será necessário ter Node >= 10.16 e npm >= 5.6 na sua máquina. Para criar um novo projeto, rode:

npx create-react-app my-app
cd my-app
npm start

```
C:\React>node -v
v14.15.3

C:\React>npm -v
6.14.9

C:\React>npx -v
6.14.9
```

npx create-react-app app2

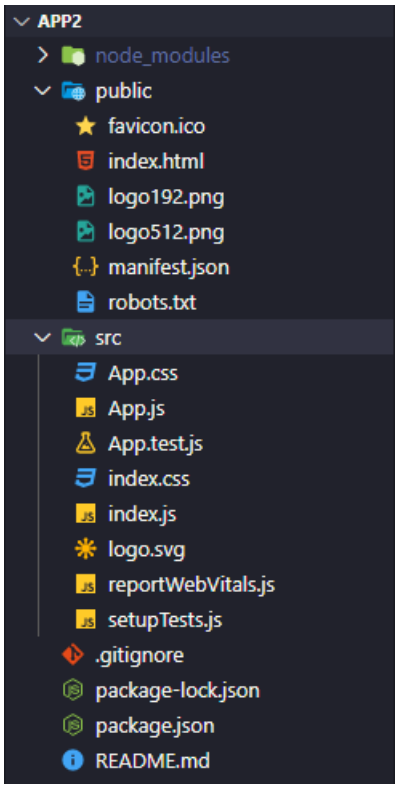
```
C:\React>npx create-react-app app2
npx: instalou 67 em 40.329s

Creating a new React app in C:\React\app2.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...


> core-js@2.6.12 postinstall C:\React\app2\node_modules\babel-runtime\node_modules\core-js
> node -e "try{require('./postinstall')}catch(e){}"


> core-js@3.11.0 postinstall C:\React\app2\node_modules\core-js
> node -e "try{require('./postinstall')}catch(e){}"


> core-js-pure@3.11.0 postinstall C:\React\app2\node_modules\core-js-pure
> node -e "try{require('./postinstall')}catch(e){}"


> ejs@2.7.4 postinstall C:\React\app2\node_modules\ejs
> node ./postinstall.js
```

.
.
.

```
Created git commit.

Success! Created app2 at C:\React\app2
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd app2
  npm start

Happy hacking!

C:\React>
```

cd app2
code .

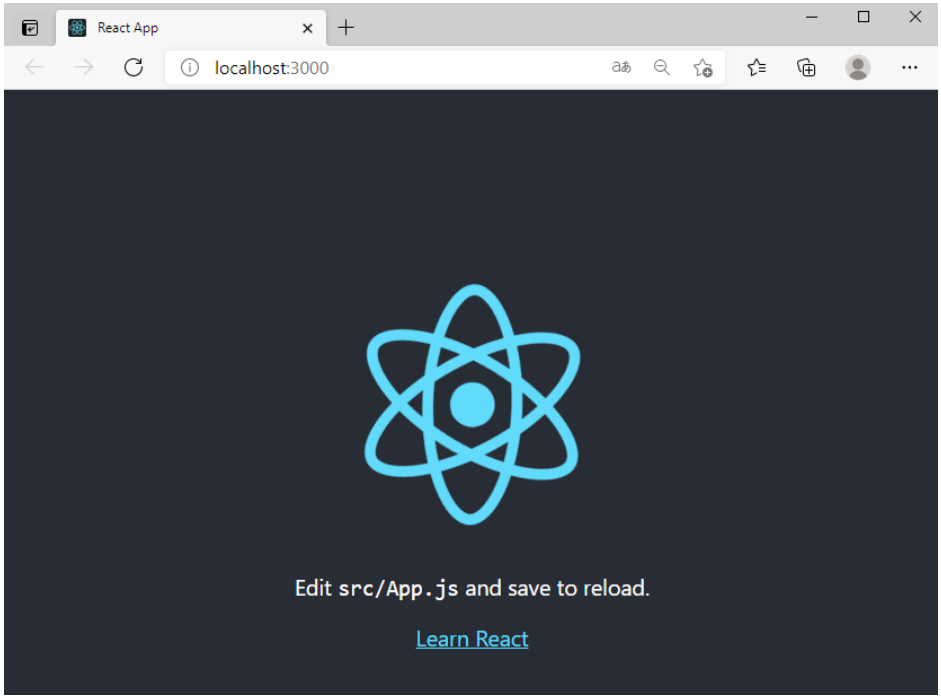
```
C:\React\app2>code .
```



- Para levantar o servidor, no terminal, entre com o comando:

npm start

```
Compiled successfully!

You can now view app2 in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.0.157:3000

Note that the development build is not optimized.
To create a production build, use npm run build.
```



Aula 33 - Primeiro contato com a estrutura da APP

- Para iniciar o servidor, no terminal, entre com o comando:

npm start

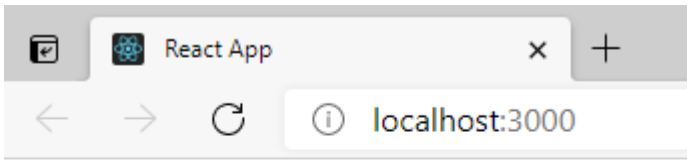
- Para interromper o servidor, pressione:

<ctrl><c>

- E confirme

src\App.js

```
import React from 'react';

function App() {
  return (
    <div>
      <p>teste</p>
      <h2>Olá mundo!</h2>
    </div>
  );
}

export default App;
```



teste

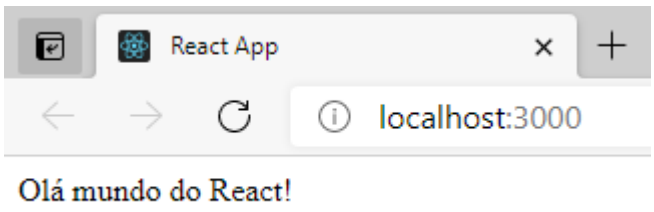
Olá mundo!

Aula 35 - Alteração de código e mais algumas dicas

- Inicialmente, no diretório raiz, compacte os arquivos originais das pastas public e src dentro de um arquivo chamado base_original.rar

src\App.js

```
import React from 'react'

class App extends React.Component{
  render(){
    return(
      <div>
        Olá Mundo do React!
      </div>
    )
  }
}

export default App;
```



Aula 36 - Criação de versão simplificada do projeto

Removendo arquivos desnecessários para o projeto

- Na pasta src, exclua os seguintes arquivos:

- App.css
- index.css
- App.test.js
- logo.svg
- reportWebVitals.js
- setupTests.js

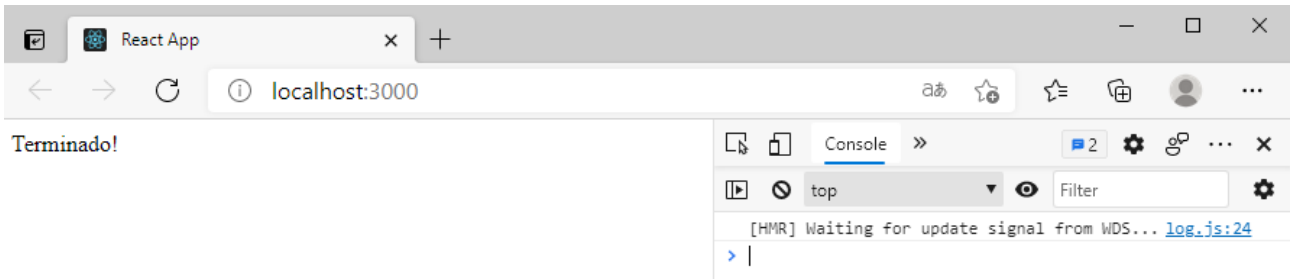
- Altere os arquivos seguintes:

src\index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

ReactDOM.render(<App />, document.getElementById('root'));
```

src\App.js

```
import React from 'react'

class App extends React.Component{
 render(){
  return(
   <div>
    Terminado!
   </div>
  )
 }
}

export default App;
```



- Na pasta public, exclua o seguinte arquivo:

- logo512.png

- No arquivo manifest.json, exclua a referência a esse arquivo:

public\manifest.json

```
{
  "short_name": "React App",
  "name": "Create React App Sample",
  "icons": [
    {
      "src": "favicon.ico",
      "sizes": "64x64 32x32 24x24 16x16",
      "type": "image/x-icon"
    },
    {
      "src": "logo192.png",
      "type": "image/png",
      "sizes": "192x192"
    }
  ],
  "start_url": ".",
  "display": "standalone",
  "theme_color": "#000000",
  "background_color": "#ffffff"
}
```

- No diretório raiz, compacte os arquivos originais das pastas public e src dentro de um arquivo chamado base_simples.rar

Aula 37 - Import e Export explicado de forma simples

- Dentro da pasta src crie uma subpasta chamada components e dentro adicione um arquivo chamado Identicacao.js

src\components\Identificacao.js

```
import React from 'react';
import Nome from './Nome';
import Sobrenome from './Sobrenome'

class Identificacao extends React.Component{
   render(){
      return(
         <div>
            <Nome />
            <Sobrenome />
         </div>
      )
   }
}

export default Identificacao;
```

src\components\Nome.js

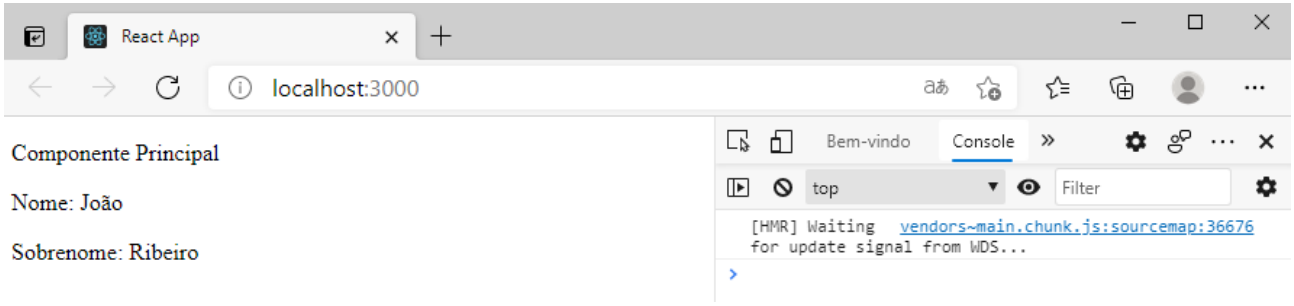
```
import React from 'react';

class Nome extends React.Component{
   render(){
      return(
         <div>
            <p>Nome: João</p>
         </div>
      )
   }
}

export default Nome;
```

src\components\Sobrenome.js

```
import React from 'react';

class Sobrenome extends React.Component{
   render(){
      return(
         <div>
            <p>Sobrenome: Ribeiro</p>
         </div>
      )
   }
}

export default Sobrenome;
```

src\App.js

```
import React from 'react';
import Identificacao from './components/Identificacao';

class App extends React.Component{
  render(){
    return(
      <div>
        <p>Componente Principal</p>
        <Identificacao />
      </div>
    )
  }
}

export default App;
```



Aula 38 - React snippets e várias dicas

Visual Studio Code - Extensões para o React

- Instale a extensão Reactjs code snippets
- Instale a extensão JS JSX Snippets
- Instale a extensão HTML to JSX

- Na pasta src/components, adicione um componente chamado Identificacao.js

- Neste arquivo digite rcc e no menu de contexto que irá se abrir, selecione rcc

Automaticamente será apresentado:

```
import React, { Component } from 'react';

class Identificacao extends Component {
   render() {
      return (
         <div>

         </div>
      );
   }
}

export default Identificacao;
```

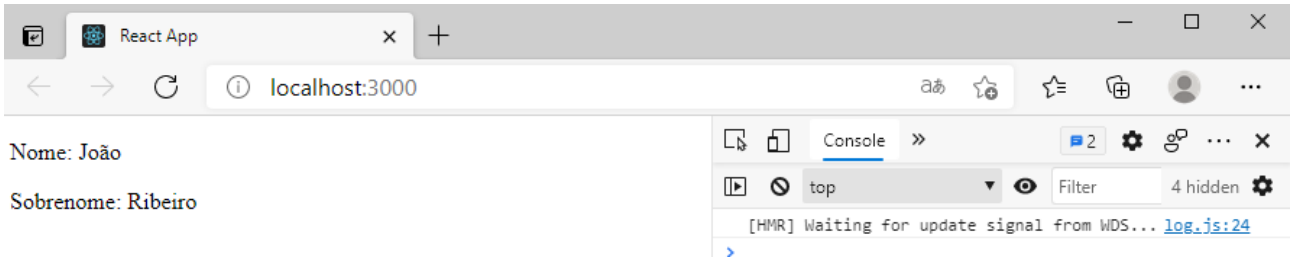
- Dentro de div acrescente:

src\components\Identificacao.js

```
import React, { Component } from 'react';

class Identificacao extends Component {
   render() {
      return (
         <div>
            <p>Nome: {this.props.nome}</p>
            <p>Sobrenome: {this.props.sobrenome} </p>
         </div>
      );
   }
}

export default Identificacao;
```

src\App.js

```
import React from 'react';
import Identificacao from './components/Identificacao';

class App extends React.Component{

 state = {
  nome: "João",
  sobrenome: "Ribeiro"
 }

 render(){
    return(
     <div>
      <Identificacao nome={this.state.nome} sobrenome={this.state.sobrenome} />
     </div>
    )
  }
}

export default App;
```



Aula 39 - Fluxo de dados numa React APP

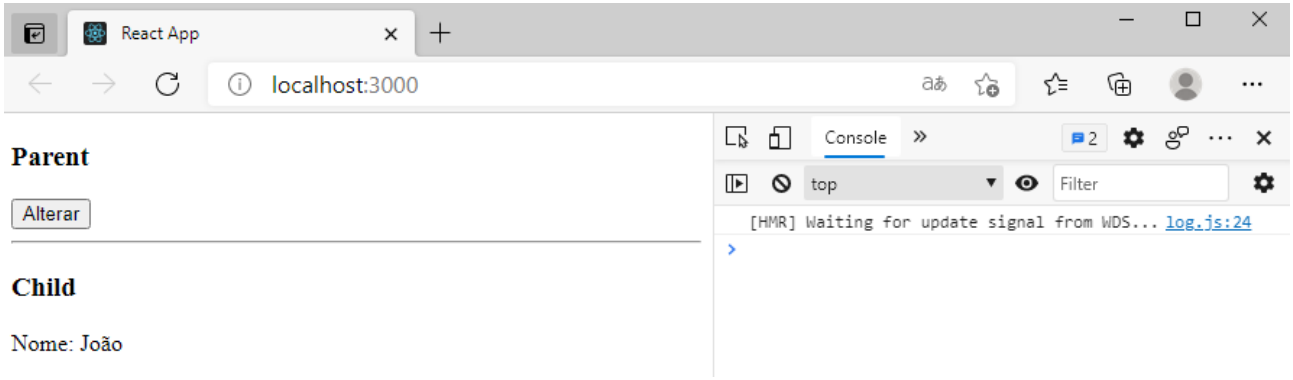
Passando dados de um Parent para um Child

src\App.js

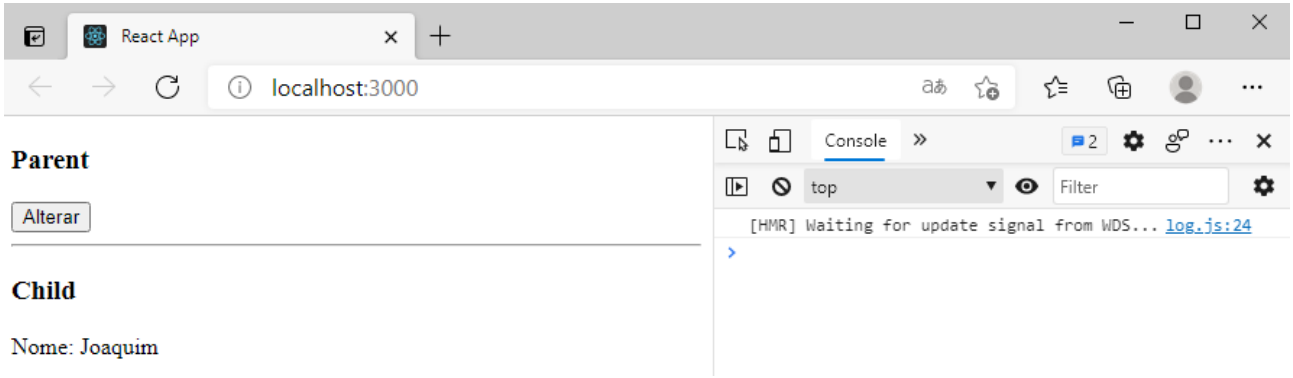
```
import React from 'react';
import Child from './components/Child';

class App extends React.Component{

 state = {
   nome: "João"
 }

 alterar = () => {
   this.setState({
     nome: "Joaquim"
   });
 }

 render(){
    return(
      <div>
        <h3>Parent</h3>
        <button onClick={this.alterar}>Alterar</button>
        <hr />
        <Child nome={this.state.nome} />
      </div>
    )
  }
}

export default App;
```

src\components\Child.js

```
import React, { Component } from 'react';

class Child extends Component {
   render() {
      return (
         <div>
            <h3>Child</h3>
            <p>Nome: {this.props.nome}</p>
         </div>
      );
   }
}

export default Child;
```



- Clicando no botão:



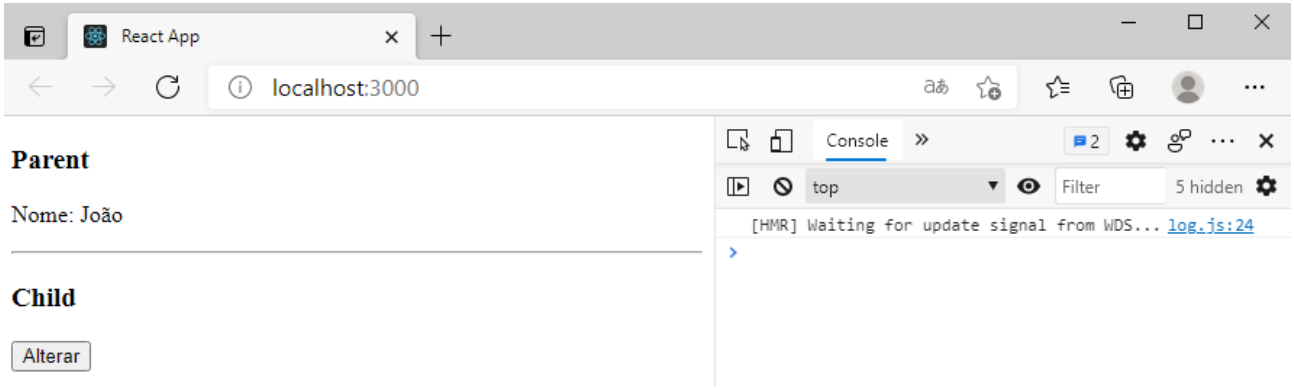
Passando dados de um Child para um Parent

src\App.js

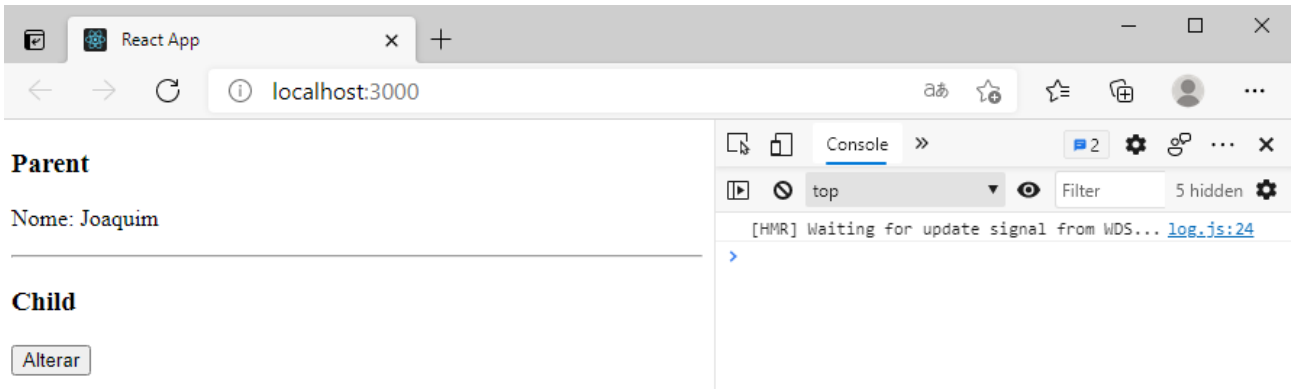
```
import React from 'react';
import Child from './components/Child';

class App extends React.Component{

  state = {
    nome: "João"
  }

  Alterar = () =>{
    this.setState({
      nome: "Joaquim"
    })
  }

  render(){
     return(
       <div>
         <h3>Parent</h3>
         <p>Nome: {this.state.nome}</p>
         <hr />
         <Child funcaoAlterar = {this.Alterar} />
       </div>
     )
   }
}

export default App;
```

src\components\Child.js

```
import React, { Component } from 'react';

class Child extends Component {

   Alterar = () => {
      // alterar o nome no parent component
      this.props.funcaoAlterar()
   }

   render() {
      return (
         <div>
            <h3>Child</h3>
            <button onClick={this.Alterar}>Alterar</button>
         </div>
      );
   }
}

export default Child;
```



- Clicando no botão:



Aula 40 - Três formas de fluxo de dados

https://towardsdatascience.com/passing-data-between-react-components-parent-children-siblings-a64f89e24ecf

Passando dados entre componentes do React - parent, child, siblings

React é uma biblioteca JavaScript criada pelo Facebook. O tratamento de dados no React pode ser um pouco complicado, mas não tão complicado quanto pode parecer. Atualmente, compilei três métodos de tratamento de dados no React: -

1. De pai para filho usando acessórios
2. De filho para pai usando callbacks
3. Entre irmãos:
   (i) Combine os dois métodos acima
   (ii) Usando Redux
   (iii) Usando Context API

Entre irmãos

Quando eu era um iniciante, demorou muito para decidir qual método escolher para compartilhar dados entre irmãos. Existem três métodos conhecidos para compartilhar dados entre irmãos e todos eles têm suas próprias vantagens e desvantagens.

Método 1: Combine os dois métodos acima de compartilhamento de dados.

Esse método, entretanto, não funcionará para estruturas de diretório complicadas, pois será necessário escrever grandes trechos de código para enviar dados entre componentes em níveis distantes uns dos outros. Os dados, então, terão que ser empurrados e puxados através de cada nível intermediário.

Método 2: Redux

Use um armazenamento global mantendo os estados de todos os componentes filhos que são necessários para interagir e consumir os dados necessários do armazenamento - Redux

Método 3: Usar context API

Já existem muitos artigos e blogs sobre por que o React atualizou para context API e qual é o melhor em quais termos.

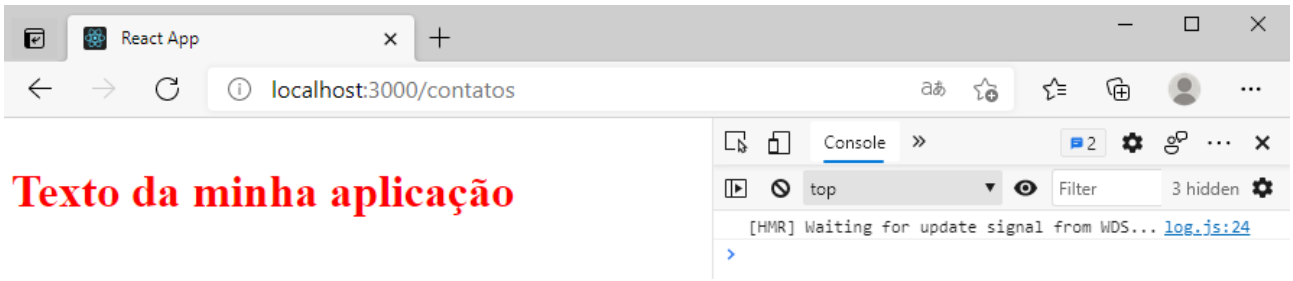
Aula 41 - Importar CSS para React APP

src\App.css

```
.cor-texto {
  color: red;
  font-size: 2em;
}
```

src\App.js

```
import React from 'react';
import './App.css';

class App extends React.Component{

  render(){
     return(
       <div>
         <h3 className="cor-texto">Texto da minha aplicação</h3>
       </div>
     )
   }
}

export default App;
```



Aula 42 - Introdução ao React Router

- Para poder usar um sistema de roteamento no React devemos instalar nas nossas dependências o React Router Dom:

- Para isso, no terminal, entre com o seguinte comando:

npm install react-router-dom

src\App.js

```
import React from 'react';
import Navegacao from './components/Navegacao';
import Home from './components/Home';
import Servicos from './components/Servicos';
import Contatos from './components/Contatos';

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

class App extends React.Component{

 render(){
    return(
     <div>
      <Router>

       <Navegacao />

       <Switch>
        <Route exact path="/">
         <Home />
        </Route>

        <Route path="/servicos">
         <Servicos />
        </Route>

        <Route path="/contatos">
         <Contatos />
        </Route>
       </Switch>

      </Router>
     </div>
    )
   }
}

export default App;
```

src\components\Navegacao.js

```
import React from 'react';
import {Link} from 'react-router-dom';

const Navegacao = () => {
   return(
      <div>
         <Link to="/">Home</Link> | <Link to="/servicos">Serviços</Link> | <Link to="/contatos">Contatos</Link>
      </div>
   )
}

export default Navegacao
```

src\components\Home.js

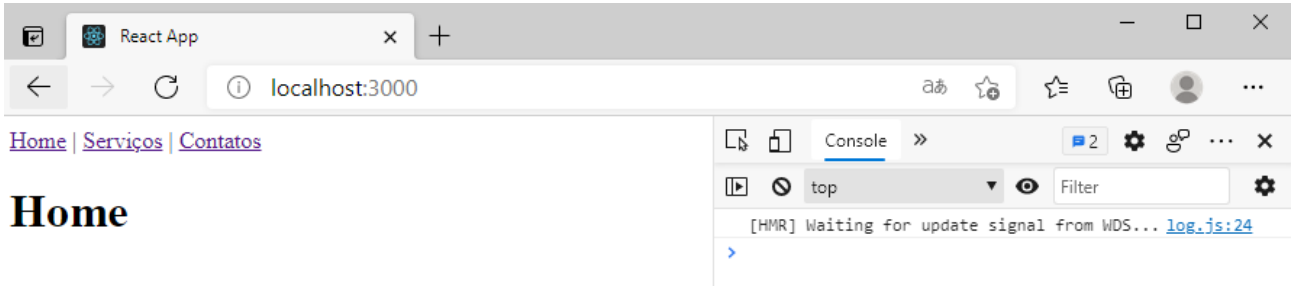
```
import React from 'react';

const Home = () => {
   return(
      <div>
         <h1>Home</h1>
      </div>
   )
}

export default Home
```

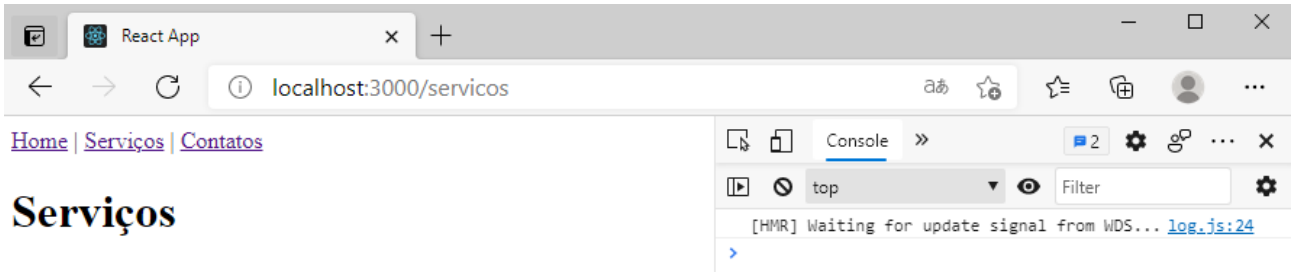
src\components\Servicos.js

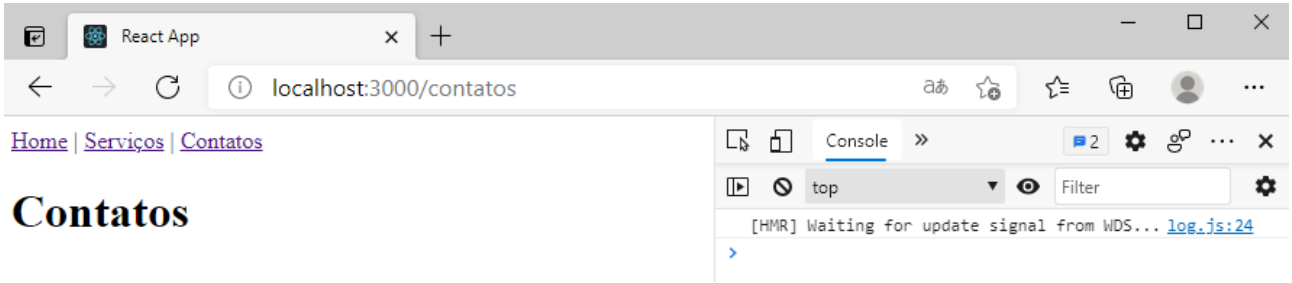
```
import React from 'react';

const Servicos = () => {
   return(
      <div>
         <h1>Serviços</h1>
      </div>
   )
}

export default Servicos
```

src\components\Contatos.js

```
import React from 'react';

const Contatos = () => {
   return(
      <div>
         <h1>Contatos</h1>
      </div>
   )
}

export default Contatos
```


React App
localhost:3000
Home | Serviços | Contatos
Home
Console
top
Filter
[HMR] Waiting for update signal from WDS... log.js:24


React App
localhost:3000/servicos
Home | Serviços | Contatos
Serviços
Console
top
Filter
[HMR] Waiting for update signal from WDS... log.js:24


React App
localhost:3000/contatos
Home | Serviços | Contatos
Contatos
Console
top
Filter
[HMR] Waiting for update signal from WDS... log.js:24

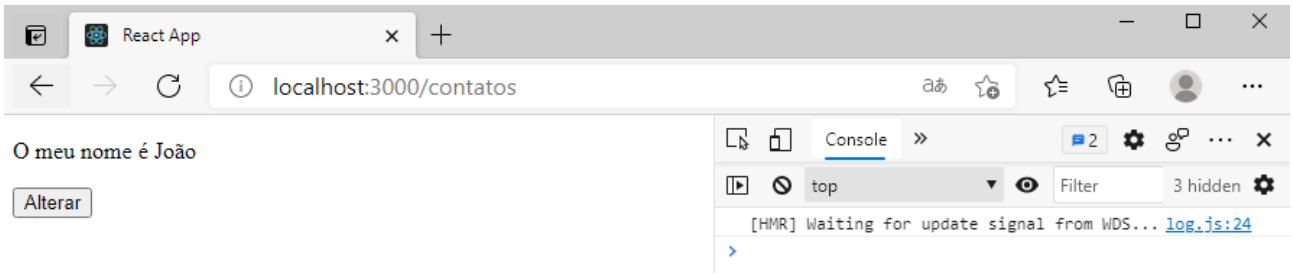
Aula 43 - Introdução a React Hooks

https://pt-br.reactjs.org/docs/hooks-intro.html

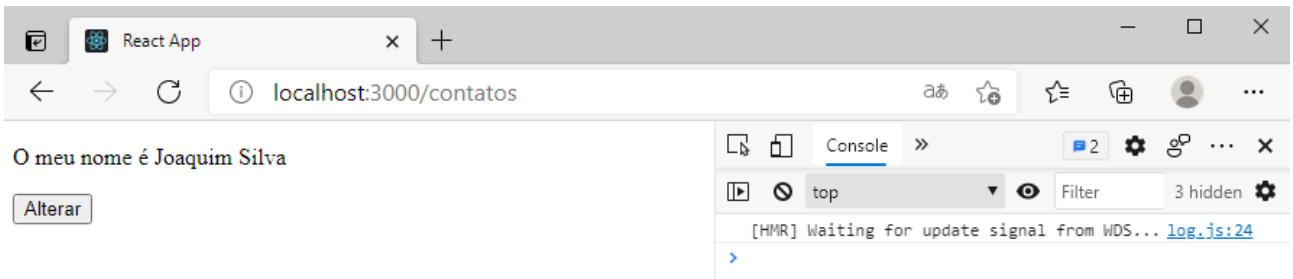
Hooks são uma nova adição ao React 16.8. Eles permitem que você use o state e outros recursos do React sem escrever uma classe.

src\App.js

```
import React, {useState} from 'react';

const App = () => {

 const[Nome, setNome] = useState('João');

 const AlterarNome = () => setNome('Joaquim Silva');

 return(
  <div>
   <p>O meu nome é {Nome}</p>
   <button onClick={() => AlterarNome()}>Alterar</button>
  </div>
 )
}

export default App;
```



- Clique no botão:



Aula 44 - Final da primeira parte

- Em um próximo curso veremos como utilizar recursos mais avançados do React.