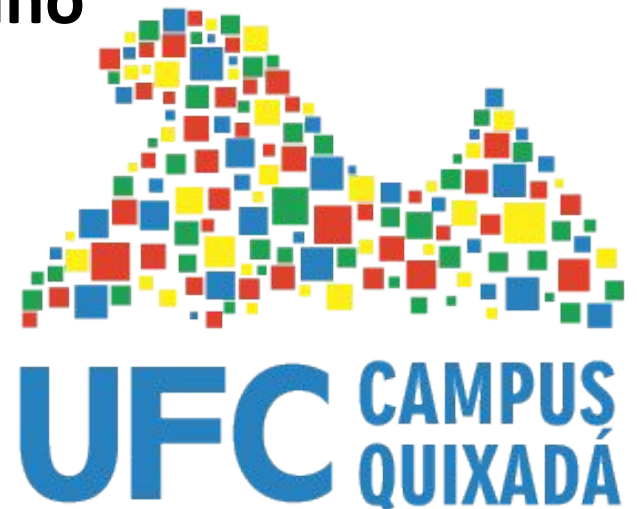


Introdução ao REACT

Prof. Sidartha Carvalho



AULA 1/6

React

- O React é uma biblioteca JavaScript de código aberto, mantida pelo Facebook, para criar interfaces de usuário (UI) interativas e dinâmicas em aplicativos web.
- Ele permite que os desenvolvedores criem componentes reutilizáveis que atualizam automaticamente quando os dados mudam, resultando em uma interface de usuário rápida e eficiente.



React

The library for web and native user interfaces

Learn React

API Reference

<https://react.dev/>

React

- O JSX (JavaScript XML) é uma extensão de sintaxe para o JavaScript que permite escrever HTML dentro do JavaScript.
- Com o JSX, os desenvolvedores podem definir a estrutura da interface de usuário de componentes React de forma semelhante ao HTML, o que melhora a legibilidade e a compreensão do código.
- Durante a compilação, o JSX é transformado em chamadas de função JavaScript, possibilitando que o React renderize os componentes na página.

```
src > FragmentReact.js > ...  
1  export default function FragmentReact() {  
2    return (  
3      <>  
4        <input type="email" placeholder="Email" />  
5        <input type="password" placeholder="Password" />  
6      </>  
7    );  
8  }  
9
```

Benefícios do React

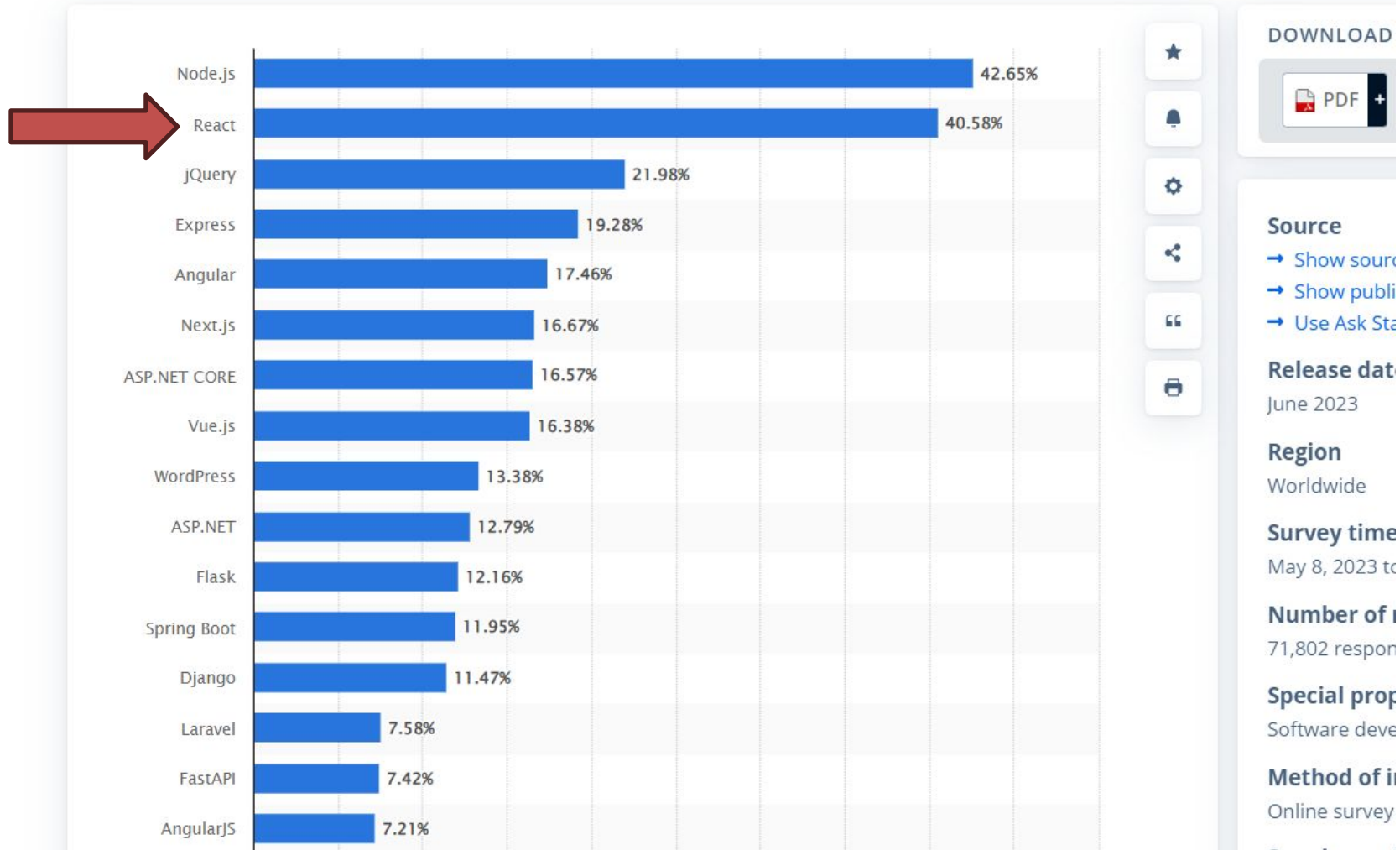
- Componentização
 - O React promove a criação de interfaces de usuário através de componentes reutilizáveis e independentes.
 - Isso simplifica o desenvolvimento, o teste e a manutenção do código, tornando-o mais modular e escalável.
- Reatividade
 - Atualização da interface de usuário quando há alteração dos dados. Isso simplifica o gerenciamento do estado da aplicação e oferece uma experiência mais fluida para o usuário.
- JSX
 - O JSX é uma extensão de sintaxe que permite escrever HTML dentro do JavaScript. Isso torna o código mais legível e fácil de entender, além de oferecer recursos familiares de HTML, como atributos e elementos aninhados.

Benefícios do React

- Performance
 - Devido ao Virtual DOM e à renderização seletiva, o React oferece um excelente desempenho, mesmo em aplicativos complexos.
 - Minimiza as operações de manipulação do DOM, resultando em uma experiência de usuário mais rápida e responsiva.
- Comunidade e suporte
 - O React é mantido pelo Facebook e possui uma comunidade grande e ativa de desenvolvedores.
 - Há muitos recursos, documentação, tutoriais e exemplos disponíveis para ajudar os desenvolvedores a aprender e resolver problemas.

Why React?

Most used web frameworks among developers worldwide, as of 2023



<https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>

Onde rodar?

- Localmente
- Compiladores online
 - codesandbox.io
 - diversos outros...

React Local

Criar projeto React Localmente

- Há diversos frameworks para a construção de aplicativos React.
- Entender cada um deles foge do nosso escopo, mas para citar:
 - Next.js
 - Remix
 - Vite
 - Gatsby
 - React Native
 - Create React App (CRA)

Criar projeto React Localmente

- Vite que vai nos ajudar a criar aplicativos React e a instalar as dependências necessárias.
- Além disso, é um dos mais rápidos.

<https://vitejs.dev/guide/>

Pode-se testar online:

<https://stackblitz.com/edit/vitejs-vite-uftsgt?file=index.html&terminal=dev>

Criar projeto React Localmente

1. Instalar NPM (Node Package Manager)

- Para isso, vamos instalar o Node.js
- <https://nodejs.org/en/download/>

2. Criar projeto React com Vite

- Abrir o CMD do Windows
- `npm create vite@latest`

```
C:\Users\Sidarta\Downloads\vite>npm create vite@latest
Need to install the following packages:
create-vite@5.2.3
Ok to proceed? (y) y
✓ Project name: ... vite-project
? Select a framework: » - Use arrow-keys. Return to submit
  Vanilla
  Vue
> React
  Preact
  Lit
  Svelte
  Solid
  Qwik
  Others
```

- Selecionar React e JavaScript durante a criação

Criar projeto React Localmente

- Dentro da pasta criada (vite-project), executar
 - npm install

```
C:\Users\Sidarta\Downloads\vite\vite-project>dir
Volume in drive C has no label.
Volume Serial Number is 34DE-6E8F

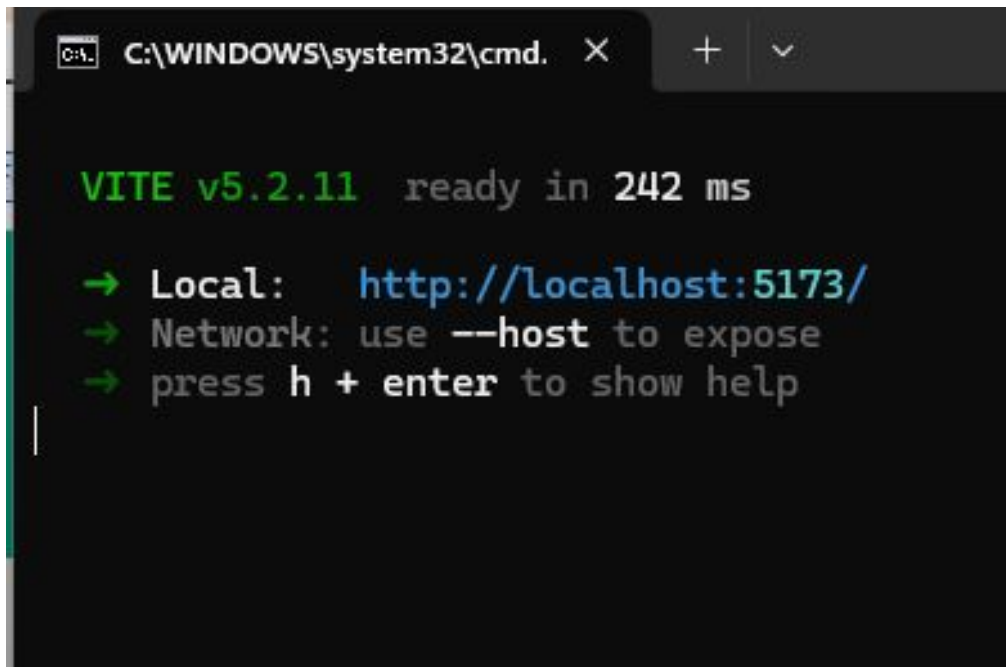
Directory of C:\Users\Sidarta\Downloads\vite\vite-project

05/07/2024  05:12 PM    <DIR>          .
05/07/2024  05:12 PM    <DIR>          ..
05/07/2024  05:11 PM             566 .eslintrc.cjs
05/07/2024  05:11 PM             253 .gitignore
05/07/2024  05:11 PM             361 index.html
05/07/2024  05:12 PM             657 package.json
05/07/2024  05:12 PM    <DIR>          public
05/07/2024  05:11 PM             451 README.md
05/07/2024  05:12 PM    <DIR>          src
05/07/2024  05:11 PM             163 vite.config.js
               6 File(s)                2,451 bytes
               4 Dir(s)  214,015,873,024 bytes free

C:\Users\Sidarta\Downloads\vite\vite-project>
```

Criar projeto React Localmente

- Para executar o seu servidor JavaScript, execute
 - `npm run dev`



```
C:\WINDOWS\system32\cmd. X + v  
  
VITE v5.2.11 ready in 242 ms  
  
→ Local: http://localhost:5173/  
→ Network: use --host to expose  
→ press h + enter to show help  
|
```



Vite + React

count is 0

Edit `src/App.jsx` and save to test HMR

Click on the Vite and React logos to learn more

Criar projeto React Localmente

- Para editar seu projeto React, é só abrir a pasta no VSCode 😊

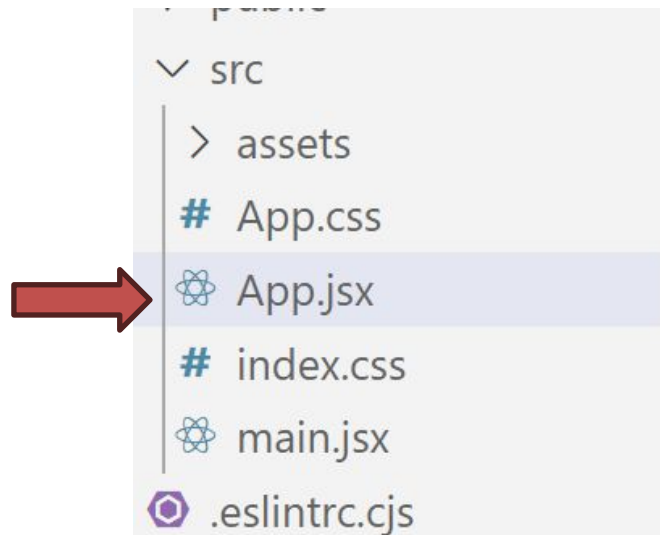


Oi PESSOAL !!!!

count is 0

Edit src/App.jsx and save to test HMR

Click on the Vite and React logos to learn more

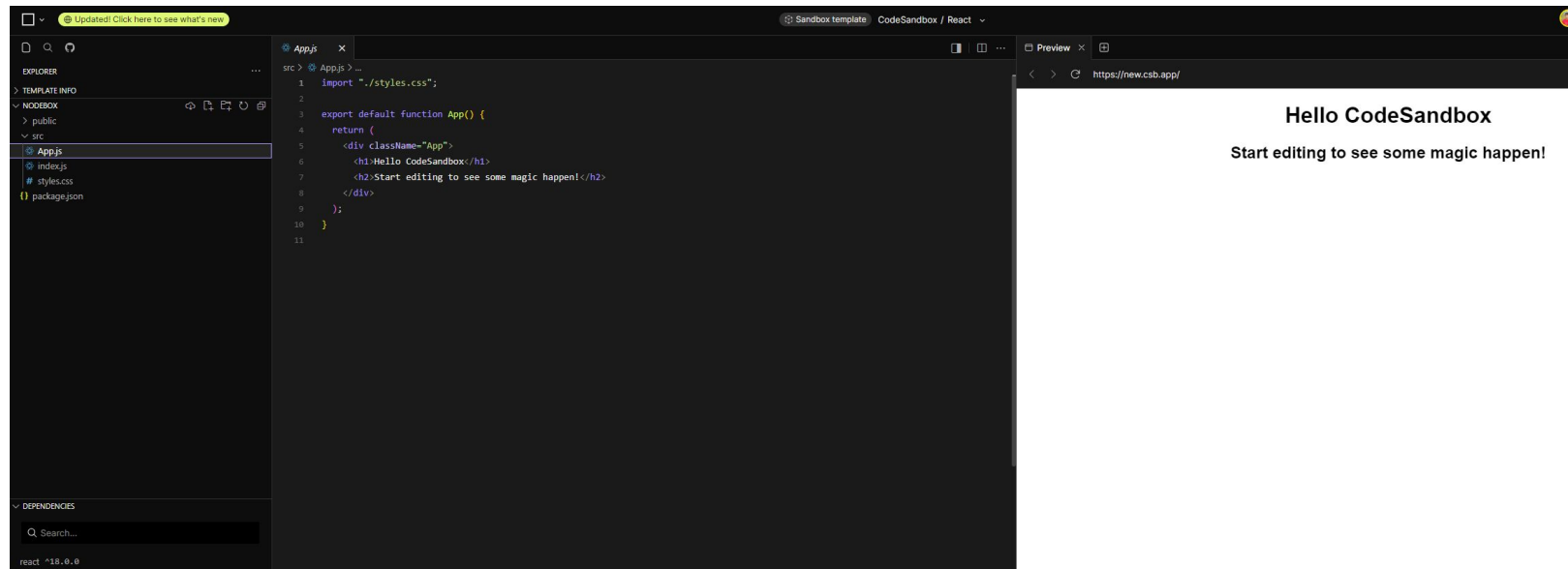


```
13 |  
14 |  
15 |   <img src={viteLogo} className=  
16 |   </a>  
17 |   <a href="https://react.dev" ta  
18 |     <img src={reactLogo} classNa  
19 |   </a>  
20 | </div>  
21 | <h1>Oi PESSOAL !!!!</h1>  
22 | <div className="card">  
    <button onClick={() => setCount  
    count is {count}
```

CodeSandBox

Criar projeto React usando codesandbox

<https://codesandbox.io/s/react-new>



React Basics

React: Conceitos Básicos

- No React tudo são componentes! Eles misturam JavaScript, HTML e CSS.
- Componente é um pedaço de interface do usuário que possui a sua própria lógica e aparência.
- O componente pode ser tão pequeno quanto um simples botão ou tão grande como uma página inteira.

React: Conceitos Básicos

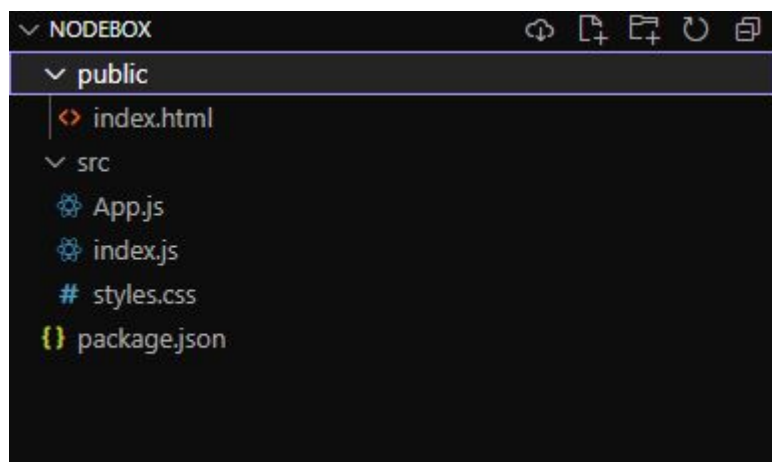
```
function MyButton() {  
  return (  
    <button>I'm a button</button>  
  );  
}
```

```
export default function MyApp() {  
  return (  
    <div>  
      <h1>Welcome to my app</h1>  
      <MyButton />  
    </div>  
  );  
}
```

- Perceba que `<MyButton />` começa com letra maiúscula. É o padrão do React!
- Todo componente no React deve começar com letra maiúscula. Por outro lado, tags HTML devem começar com letra minúscula.

React: Conceitos Básicos

- Após criar um projeto React no codesandbox, a estrutura de arquivo é similar a esta:

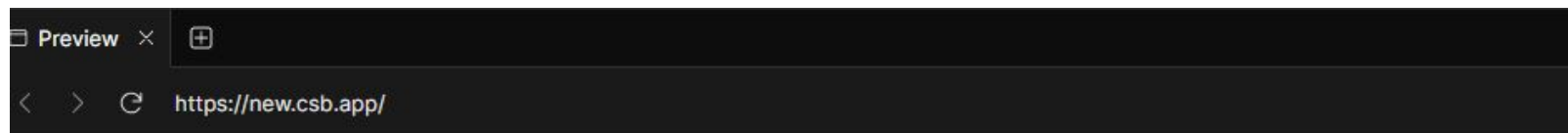
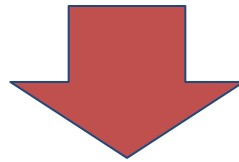


- O App.js é o arquivo principal, o primeiro a ser executado ao iniciar a aplicação.

React: Conceitos Básicos

```
import "./styles.css";

export default function App() {
  return (
    <div className="App">
      <h1>Hello CodeSandbox</h1>
      <h2>Start editing to see some magic happen!</h2>
    </div>
  );
}
```



Hello CodeSandbox

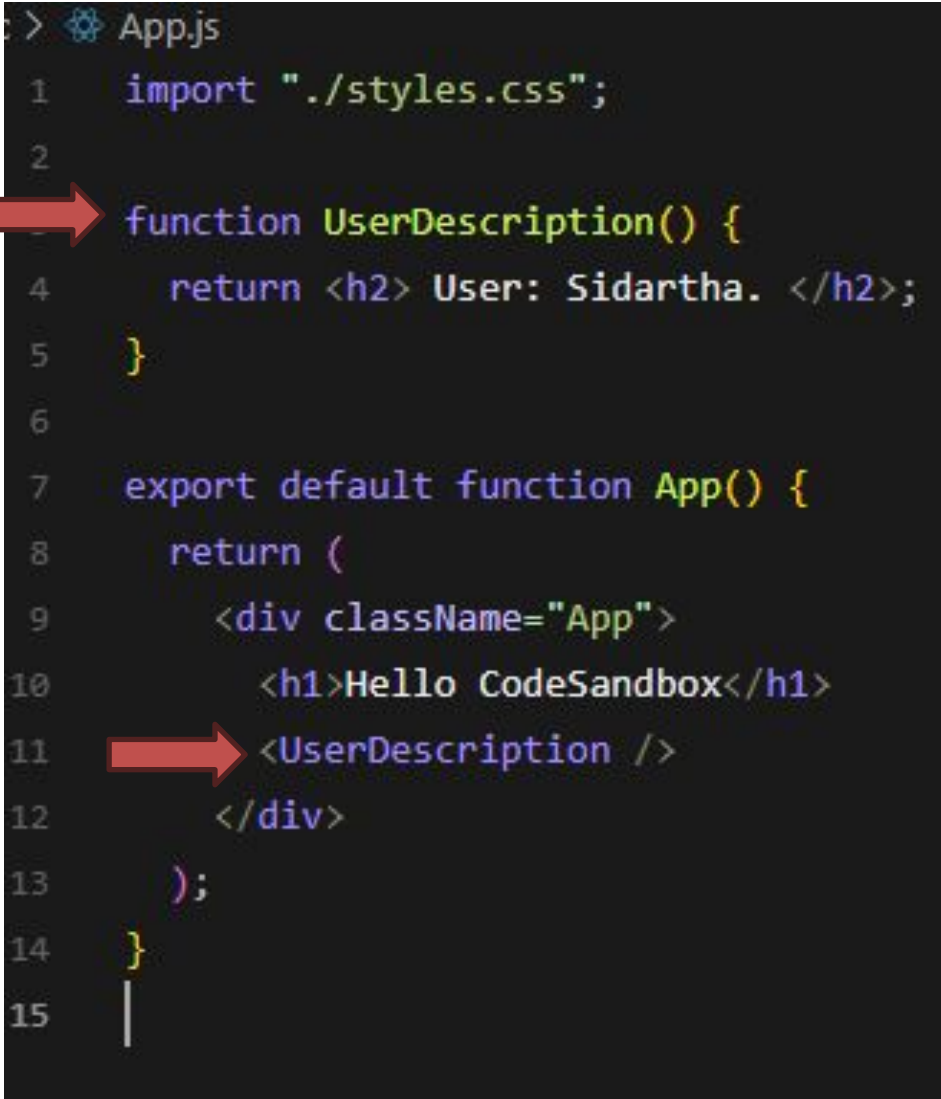
Start editing to see some magic happen!

Criar componentes

- Pode-se criar componentes dentro do mesmo arquivo ou em arquivos separados.
- Ao criar em arquivos separados, é necessário importar os componentes.

Criar componentes: mesmo arquivo

Componente
“UserDescription”
criado no mesmo
arquivo



```
App.js
1  import './styles.css';
2
3  function UserDescription() {
4      return <h2> User: Sidartha. </h2>;
5  }
6
7  export default function App() {
8      return (
9          <div className="App">
10             <h1>Hello CodeSandbox</h1>
11             <UserDescription />
12          </div>
13      );
14  }
15  |
```

The image shows a code editor with a dark background. The file name 'App.js' is at the top. The code defines a 'UserDescription' function that returns an h2 element with the text 'User: Sidartha.'. Below this, the 'App' function is defined, which returns a div containing an h1 element and a 'UserDescription' component. Two red arrows point to the 'UserDescription' function definition and its usage in the 'App' component.

Criar componentes: arquivo separado

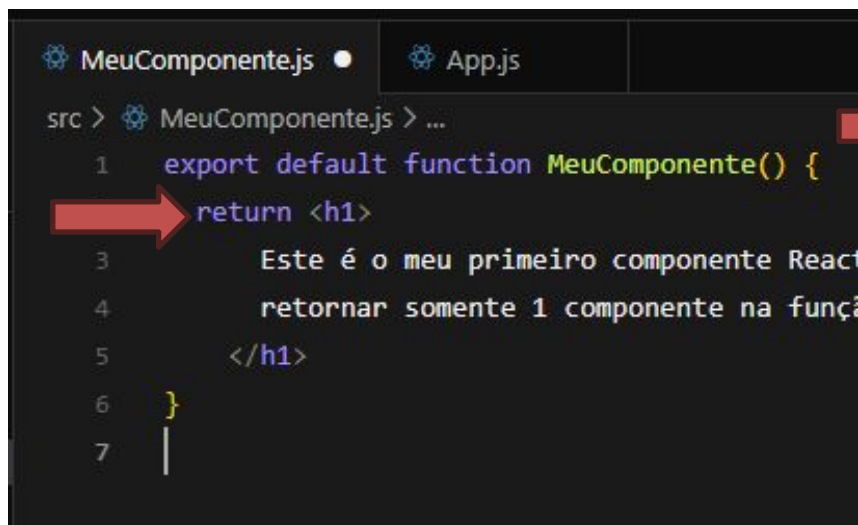
```
MeuComponente.js • App.js
src > MeuComponente.js > ...
1 export default function MeuComponente() {
2   return <h1>
3     Este é o meu primeiro componente React! Não esqueça de fechar as TAGs e
4     retornar somente 1 componente na função.
5   </h1>;
6 }
7 |
```



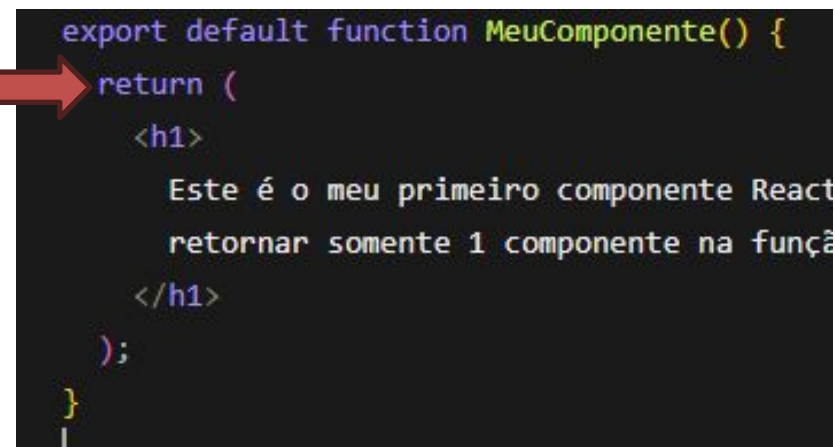
```
MeuComponente.js App.js X
src > App.js > ...
1 import './styles.css';
2 import MeuComponente from './MeuComponente.js';
3
4 function UserDescription() {}
5
6 export default function App() {
7   return (
8     <div className="App">
9       <h1>Hello CodeSandbox</h1>
10      <MeuComponente />
11    </div>
12  );
13 }
14
```

Criar componentes: arquivo separado

- Ao criar o *return* do componente, o elemento HTML deve estar na mesma linha do return!
 - caso contrário, será acionado um erro.
- Outra forma de escrever é usando ().



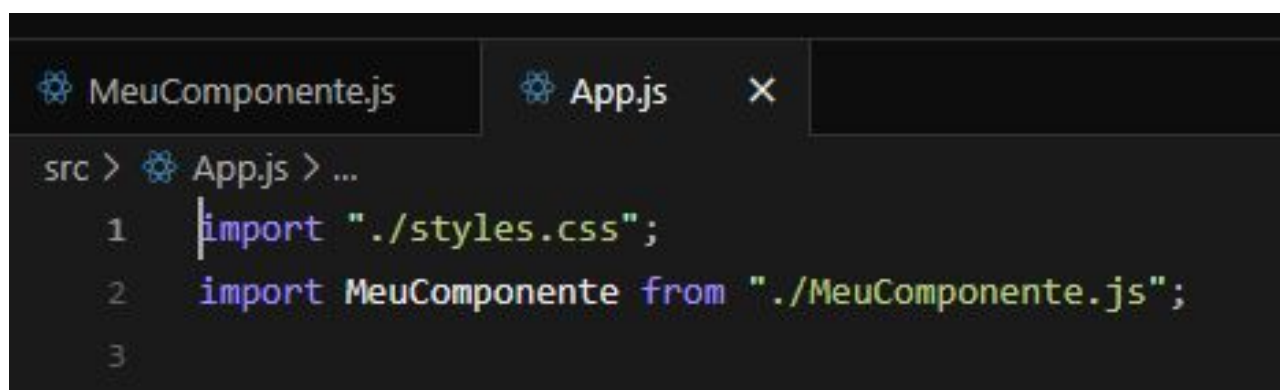
```
src > MeuComponente.js > ...
1  export default function MeuComponente() {
2    return <h1>
3      Este é o meu primeiro componente React
4      retornar somente 1 componente na função
5    </h1>
6  }
7  |
```



```
export default function MeuComponente() {
2    return (
3      <h1>
4        Este é o meu primeiro componente React
5        retornar somente 1 componente na função
6      </h1>
7    );
8  }
9  |
```

Criar componentes: arquivo separado

- O nome usado na importação é o que deve ser chamado na hora de utilizar o componente (*MeuComponente*).
- O nome do componente importado não exige ser o mesmo nome do arquivo, mas é uma boa prática manter os mesmos nomes.



The screenshot shows a code editor with two tabs: 'MeuComponente.js' and 'App.js'. The 'App.js' tab is active, showing the following code:

```
src > App.js > ...  
1  import './styles.css';  
2  import MeuComponente from './MeuComponente.js';  
3
```

Criar componentes: arquivo separado

- Se precisar retornar mais de uma TAG HTML no retorno do seu componente, além de usar o `return ()` também é preciso adicionar um novo elemento, como um `<form>` ou uma `<div>`.
- O React só permite o retorno de um elemento no componente!

Componentes React

```
MeuComponente.js  MeuDuploComponente.js X  App.js
src > MeuDuploComponente.js > ...
1  export default function MeuDuploComponente() {
2    return (
3      <div>
4        <input type="email" placeholder="Email" />
5        <input type="password" placeholder="Password" />
6      </div>
7    );
8  }
9
```

Hello CodeSandbox

Primeiro componente React! Não esqueça de fazer retornar somente 1 componente na função.

Email	Password
-------	----------

Componentes React

- Você também pode chamar o componente quantas vezes quiser!

```
App.js > App
1 import './styles.css';
2 import MeuComponente from './MeuComponente.js';
3 import MeuDuploComponente from './MeuDuploComponente.js';
4
5 function UserDescription() {}
6
7 export default function App() {
8   return (
9     <div className="App">
10       <h1>Hello CodeSandbox</h1>
11
12       <MeuComponente />
13
14       <MeuDuploComponente />
15
16       <MeuDuploComponente />
17
18       <MeuDuploComponente />
19     </div>
20   );
21 }
22
```

Hello CodeSandbox

primeiro componente React! Não esqueça de fechar a: retornar somente 1 componente na função.



Email	Password
Email	Password
Email	Password

Componentes React

- Você também pode exportar mais de uma função no mesmo arquivo.

```
src > Calculadora.js > sub
1  function add(x, y) {
2    return x + y;
3  }
4
5  function sub(x, y) {
6    return x - y;
7  }
8
9  export { add, sub };
10
```

```
App.js  x  Calculadora.js
src > App.js > ...
1  import "../styles.css";
2  import MeuComponente from "../MeuComponente.js";
3  import MeuDuploComponente from "../MeuDuploComponente.js";
4  import { add, sub } from "../Calculadora";
5
6  let soma = add(5, 10);
7
8  export default function App() {
9    return (
10      <div className="App">
11        <h1>Hello CodeSandbox</h1>
12        <MeuComponente />
13        <MeuDuploComponente />
14        <h3>{soma}</h3>
15      </div>
16    );
17  }
```