

React.JS Fundamentos e Intermediário 10/12/2021

Iniciaremos hoje um novo projeto com React puro.

Utilize o terminal git, terminal cmd, terminal Linux, ou outro. O prompt de comando do Windows (cmd) não funciona em algumas partes do projeto.

Para instalar o Git, digite no Google git scm.

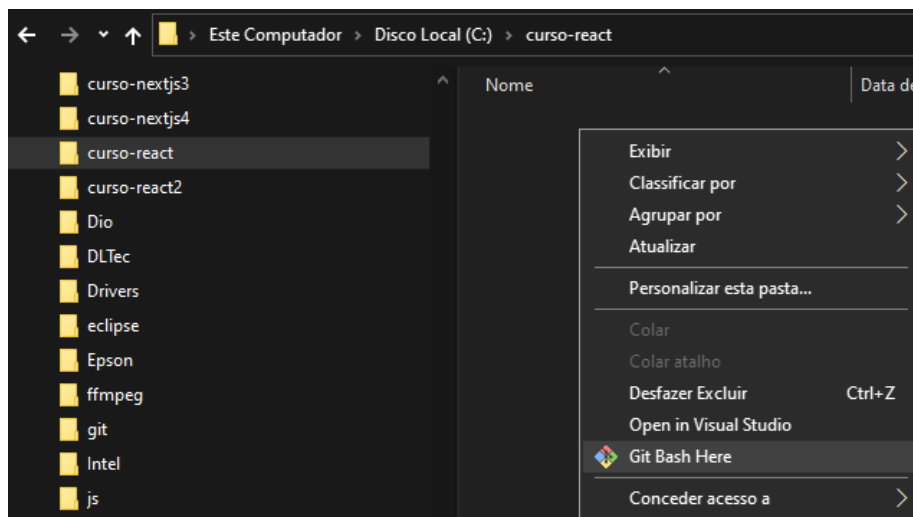
Precisamos ter o Node.js instalado na máquina, verifique com o comando `node -v`.

Se necessário faça o download do Node.js no site:

<https://nodejs.org/en/>

Para criar o projeto React utilizaremos o npx. O npx é um comando que vai se encarregar de baixar uma dependência de forma temporária para executar os comandos necessário e depois o npx exclui essas dependências sem necessariamente instalar essa dependência de forma global.

Crie a pasta curso-react e dê um Git Bash Here. Veja o modelo abaixo:



No prompt do git dê o comando:

`npx create-react-app fundamentos-react`

```
MINGW64:/c/curso-react/fundamentos-react
Alexandre@DESKTOP-78RVV7R MINGW64 /c/curso-react
$ npx create-react-app fundamentos-react

Creating a new React app in C:\curso-react\fundamentos-react.

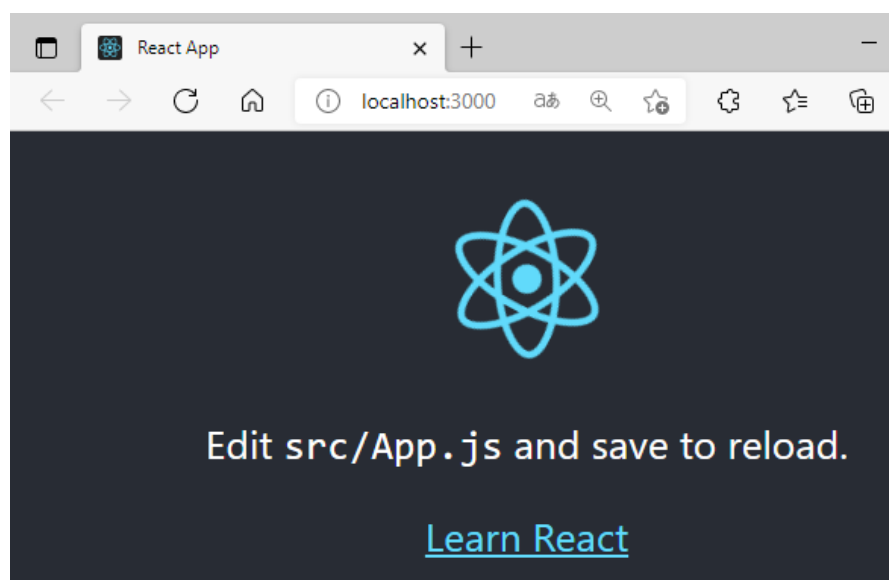
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
```

Entre na pasta do projeto com o comando `cd fundamentos-react` e inicie o servidor com `npm start`.

```
Alexandre@DESKTOP-78RVV7R MINGW64 /c/curso-react
$ cd fundamentos-react

Alexandre@DESKTOP-78RVV7R MINGW64 /c/curso-react/fundamentos-react
$ npm start

> fundamentos-react@0.1.0 start
> react-scripts start
```



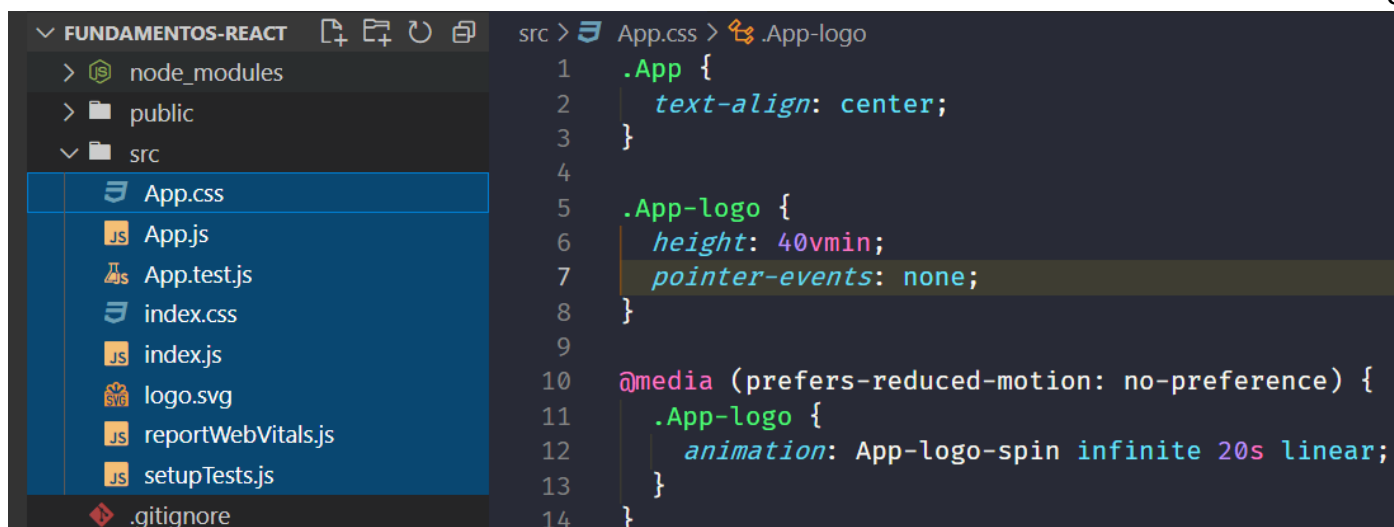
Dê um `git bash` here na pasta do projeto e abra o VSCode com o comando `code .`

```
MINGW64:/c/curso-react/fundamentos-react

Alexandre@DESKTOP-78RVV7R MINGW64 /c/curso-react/fundamentos-react
$ code .

Alexandre@DESKTOP-78RVV7R MINGW64 /c/curso-react/fundamentos-react
$ |
```

Iniciaremos nosso projeto do zero. Para isso apagaremos todos os arquivos da pasta `SRC`. Selecione o primeiro elemento, pressione `<shift> + <seta abaixo> + `



```

src > App.css > .App-logo
1  .App {
2    text-align: center;
3  }
4
5  .App-logo {
6    height: 40vmin;
7    pointer-events: none;
8  }
9
10 @media (prefers-reduced-motion: no-preference) {
11   .App-logo {
12     animation: App-logo-spin infinite 20s linear;
13   }
14 }

```

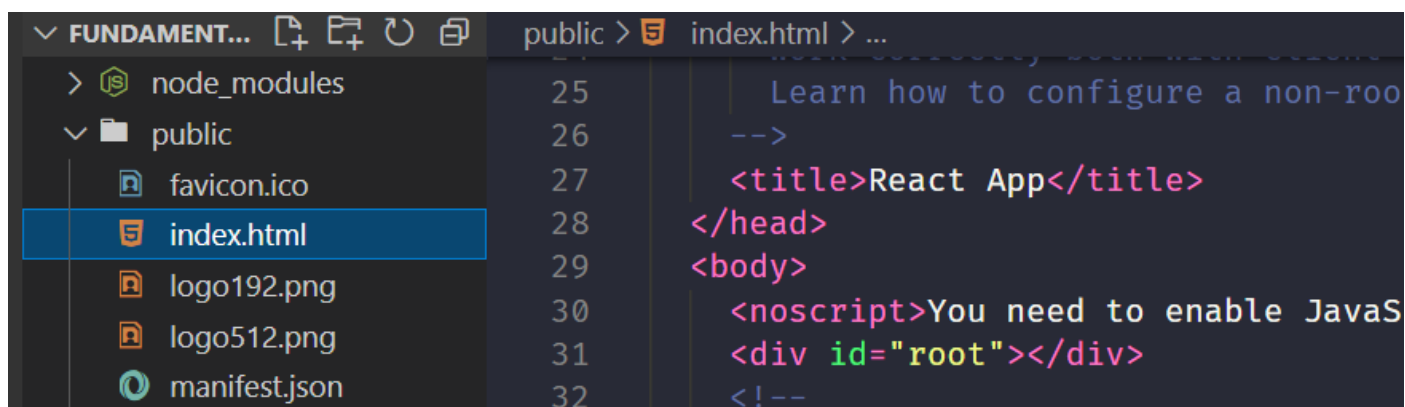
Criaremos apenas um arquivo `index.js` e será necessário que ele interaja com a biblioteca DOM do React

O dom é document object model – modelo de objeto de documento que é exatamente a estrutura da sua página. Faremos a importação dele com o `import`.

É a partir dele que iremos renderizar algo na tela. Chamaremos o método `render` que recebe dois parâmetros: o primeiro é o que eu quero renderizar na tela, e o segundo é onde quero inserir esse elemento.

Nesse caso sabemos que temos apenas uma única página HTML no nosso projeto. E vamos inserir essa informação na nossa página React no arquivo `index.html`.

Nesse arquivo temos no `<body>` uma `<div>` chamada “root”, e é aqui onde nossa aplicação será injetada.



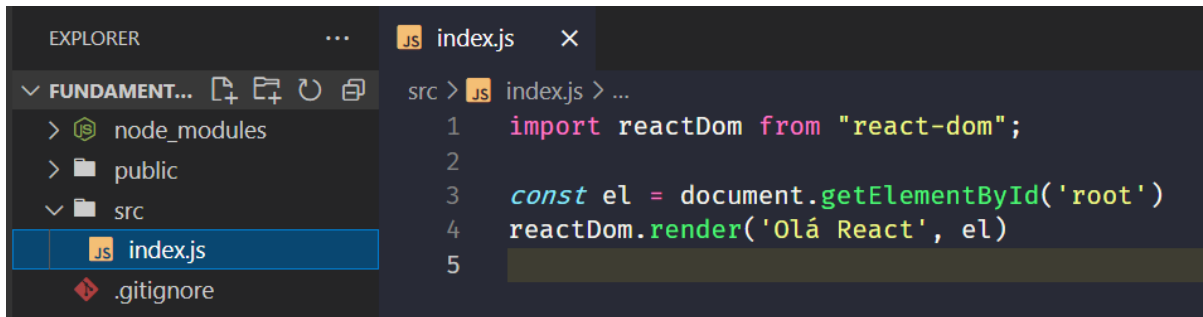
```

public > index.html > ...
25  Learn how to configure a non-root
26  -->
27  <title>React App</title>
28  </head>
29  <body>
30    <noscript>You need to enable JavaScript to run this app.</noscript>
31    <div id="root"></div>
32  <!--

```

O comando que faz isso é o:

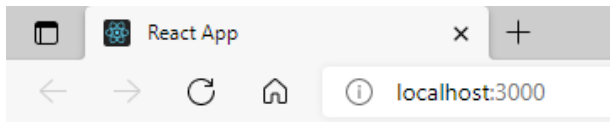
```
const el = document.getElementById('root')
```



The screenshot shows the VS Code Explorer on the left with a project structure: `FUNDAMENT...` (expanded), `node_modules`, `public`, `src` (expanded), `index.js` (selected), and `.gitignore`. The main editor shows the content of `index.js`:

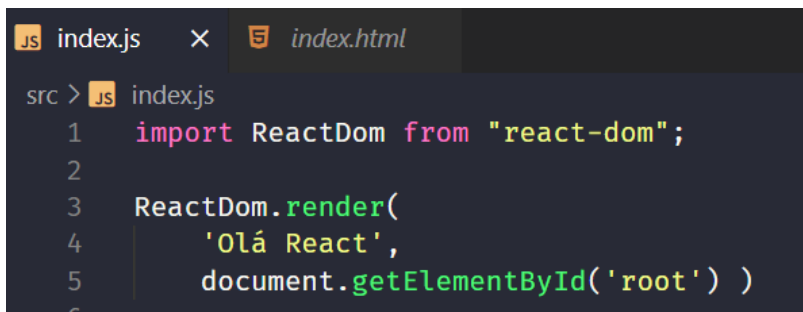
```
src > JS index.js > ...
1  import ReactDOM from "react-dom";
2
3  const el = document.getElementById('root')
4  ReactDOM.render('Olá React', el)
5
```

Recarregando a página.



Olá React

Poderia pegar diretamente pelo id



The screenshot shows two files open in VS Code: `index.js` and `index.html`. The `index.js` file content is:

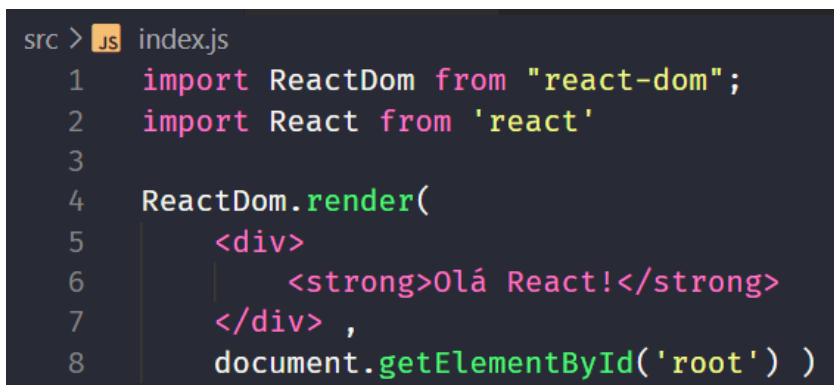
```
src > JS index.js
1  import ReactDOM from "react-dom";
2
3  ReactDOM.render(
4    'Olá React',
5    document.getElementById('root') )
6
```

Nesse momento estamos usando o ReactDOM, ainda não é o React.
Para usarmos o React utilizaremos o JSX através do import React.



The screenshot shows the `index.js` file in VS Code with the following code:

```
src > JS index.js
1  import ReactDOM from "react-dom";
2  import React from 'react'
3
4  ReactDOM.render(
5    <div>Olá React!</div> ,
6    document.getElementById('root') )
7
```



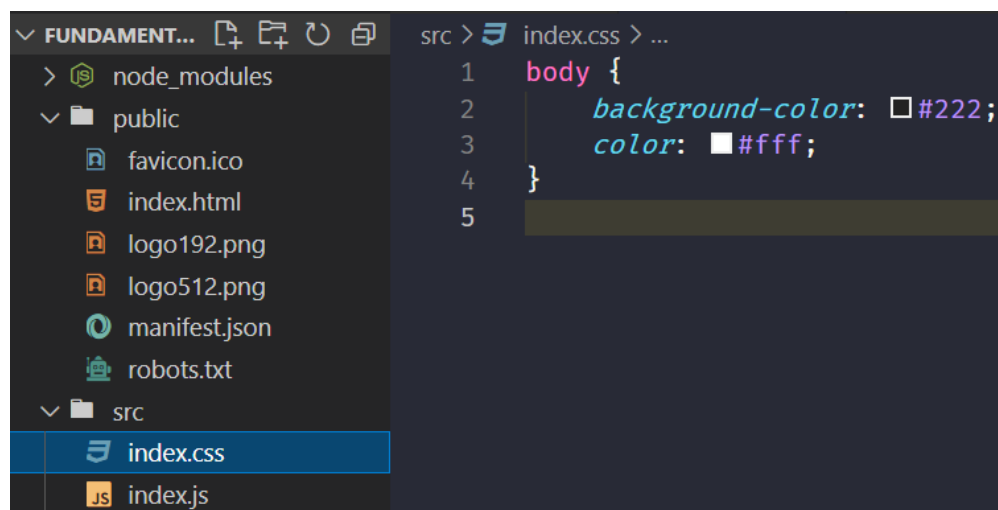
The screenshot shows the `index.js` file in VS Code with the following code:

```
src > JS index.js
1  import ReactDOM from "react-dom";
2  import React from 'react'
3
4  ReactDOM.render(
5    <div>
6      <strong>Olá React!</strong>
7    </div> ,
8    document.getElementById('root') )
9
```

Podemos ainda colocar dentro de uma função essa string.

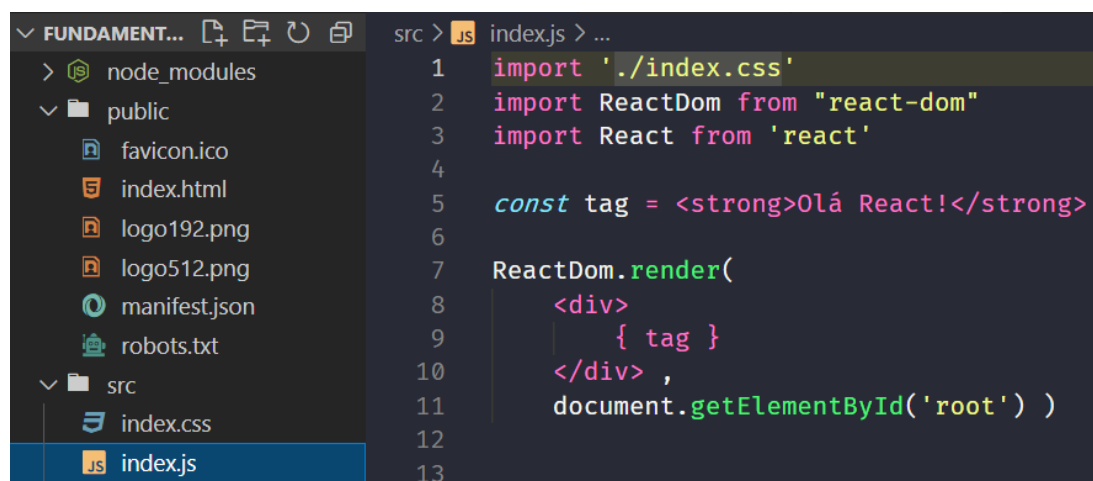
```
src > JS index.js > ...
1  import ReactDOM from "react-dom";
2  import React from 'react'
3
4  const tag = <strong>Olá React!</strong>
5
6  ReactDOM.render(
7    <div>
8      { tag }
9    </div> ,
10   document.getElementById('root') )
```

Vamos criar um css



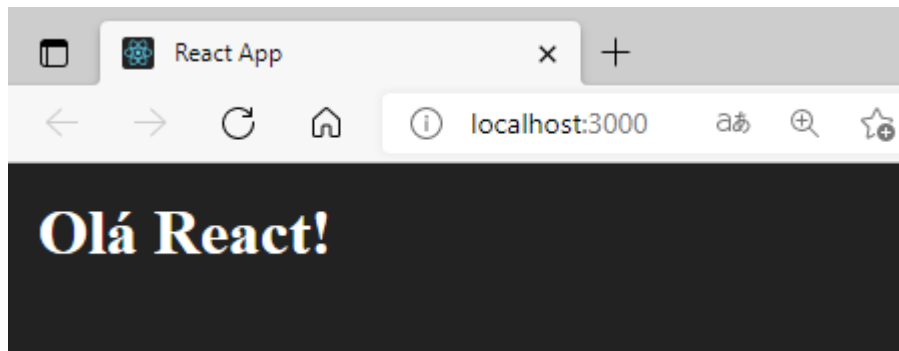
```
src > index.css > ...
1  body {
2    background-color: #222;
3    color: #fff;
4  }
5
```

E fazer o import no arquivo index.js



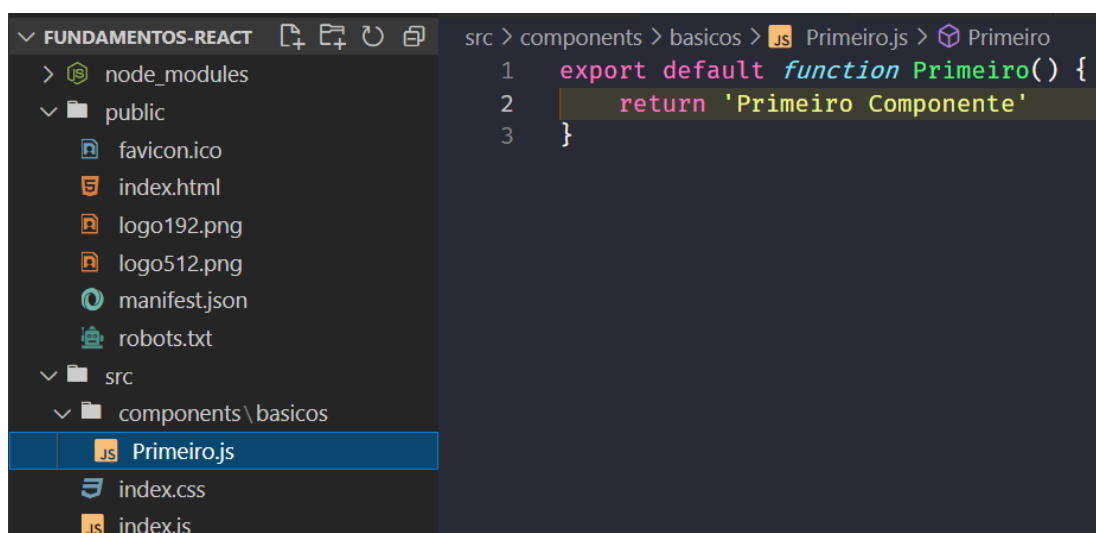
```
src > JS index.js > ...
1  import './index.css'
2  import ReactDOM from "react-dom"
3  import React from 'react'
4
5  const tag = <strong>Olá React!</strong>
6
7  ReactDOM.render(
8    <div>
9      { tag }
10   </div> ,
11   document.getElementById('root') )
12
13
```

A página ficará com essa aparência.

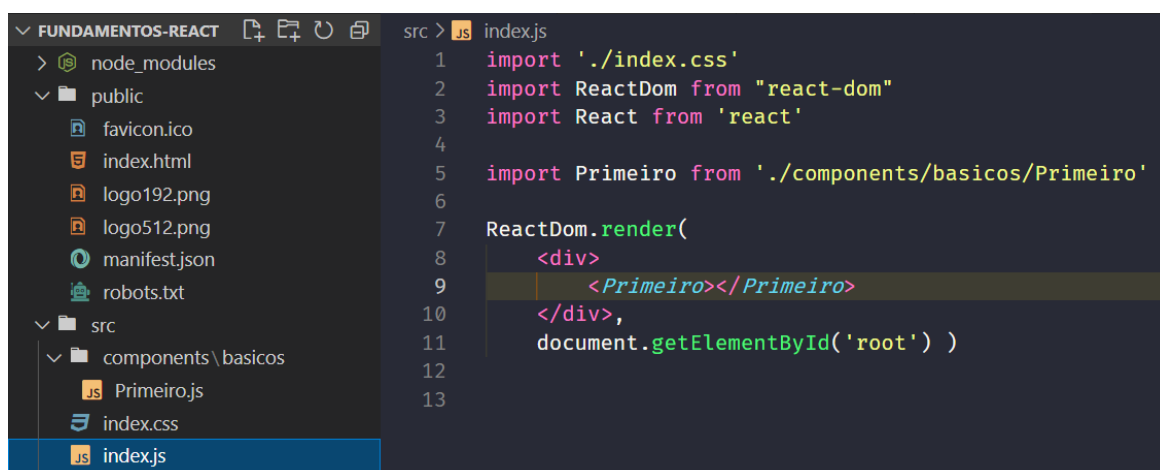


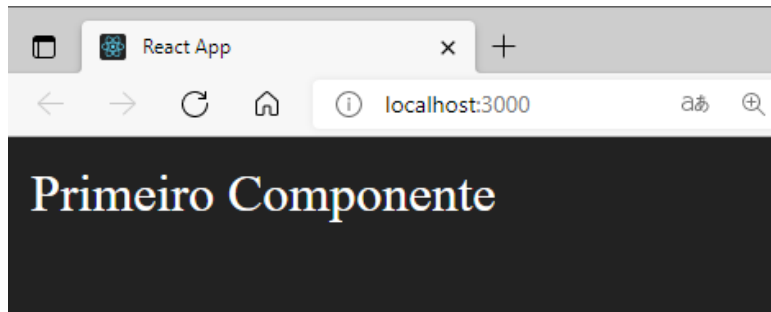
Criaremos nosso primeiro componente na pasta componentes e na subpasta básicos no arquivo Primeiro.js

Essa função deve ficar disponível para fora da pasta, para isso usamos o export default



Preciso referenciar esse arquivo no index.js com o import Primeiro e já posso chamar esse meu componente <Primeiro></Primeiro>





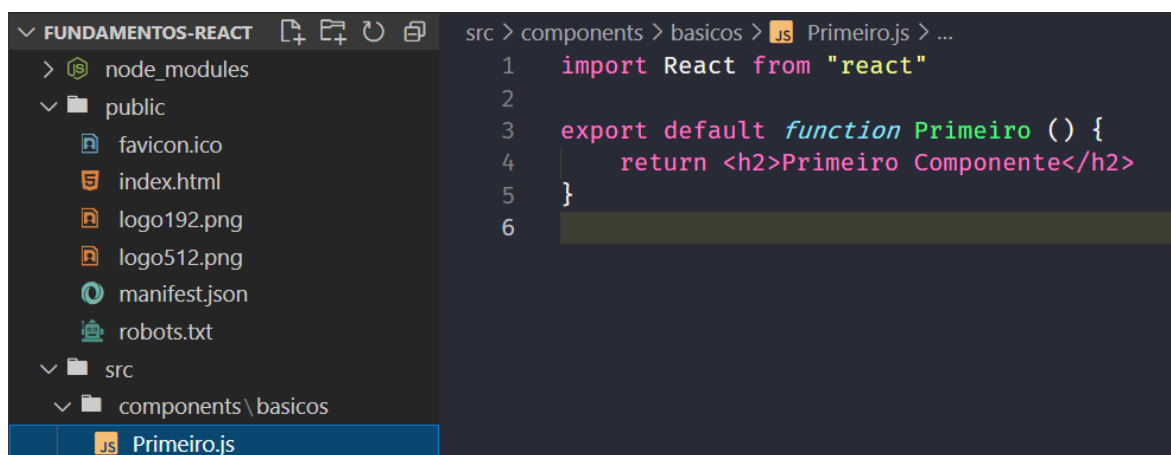
Posso referenciar esse meu componente com outros nomes.

```
src > JS index.js
1  import './index.css'
2  import ReactDOM from 'react-dom'
3  import React from 'react'
4
5  import Comp from './components/basicos/Primeiro'
6
7  ReactDOM.render(
8    <div>
9      <Comp></Comp>
10   </div>,
11   document.getElementById('root') )
```

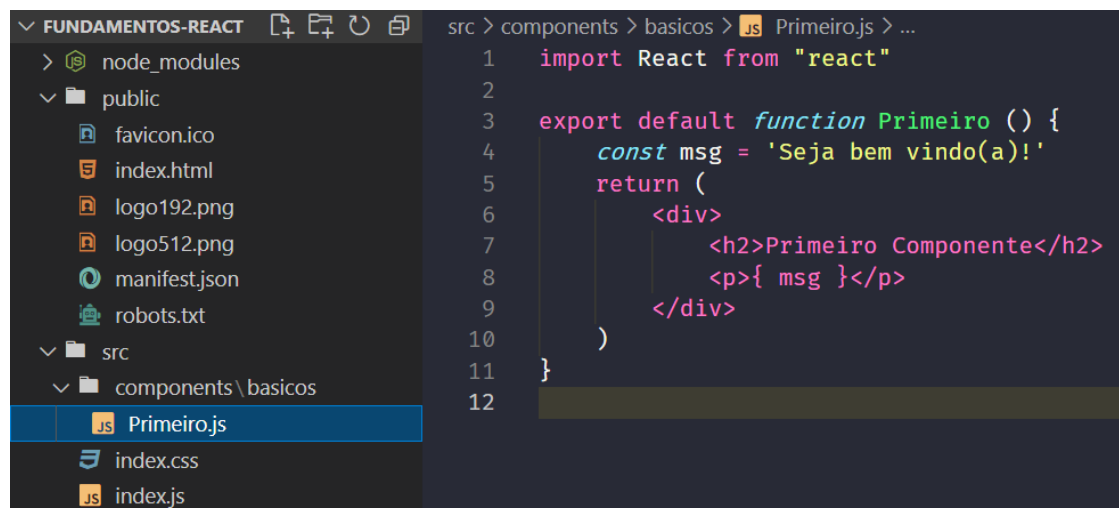
Posso deixar também como uma função anônima.

```
src > components > basicos > JS Primeiro.js > ...
1  export default function () {
2    return 'Primeiro Componente'
3  }
```

Também colocarei a tag `<h2>` na nossa função, para isso, importaremos o React com o import, colocando o nosso JSX dentro de um componente



Trabalhando com constantes dentro da função usando o JSX



The screenshot shows the VS Code interface. On the left, the file explorer displays the project structure: 'FUNDAMENTOS-REACT' (expanded), 'node_modules', 'public' (with files like favicon.ico, index.html, logo192.png, logo512.png, manifest.json, robots.txt), 'src' (expanded), and 'components\basicos' (expanded). The file 'Primeiro.js' is selected. The main editor shows the code for 'Primeiro.js' with line numbers 1 to 12. The code is as follows:

```
1 import React from "react"
2
3 export default function Primeiro () {
4   const msg = 'Seja bem vindo(a)!'
5   return (
6     <div>
7       <h2>Primeiro Componente</h2>
8       <p>{ msg }</p>
9     </div>
10  )
11 }
12
```

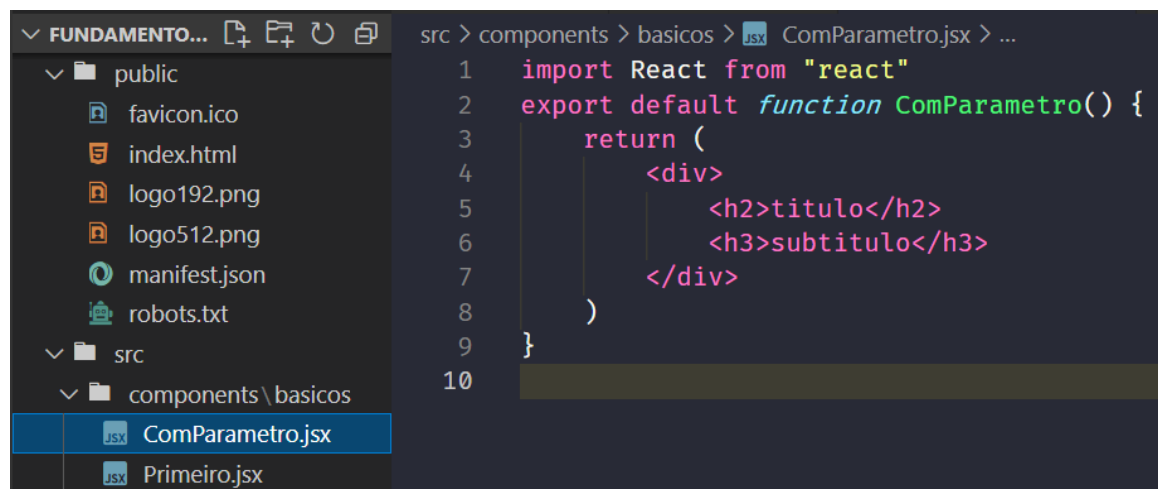
Agora passaremos parâmetros para nosso componente.

O componente criado anteriormente era um componente funcional, pois é baseado em função.

Existe componente baseado em classe.

Mudaremos a extensão dos arquivos JS para JSX apenas para facilitar nosso trabalho na IDE. Por padrão o React aceita as duas extensões.

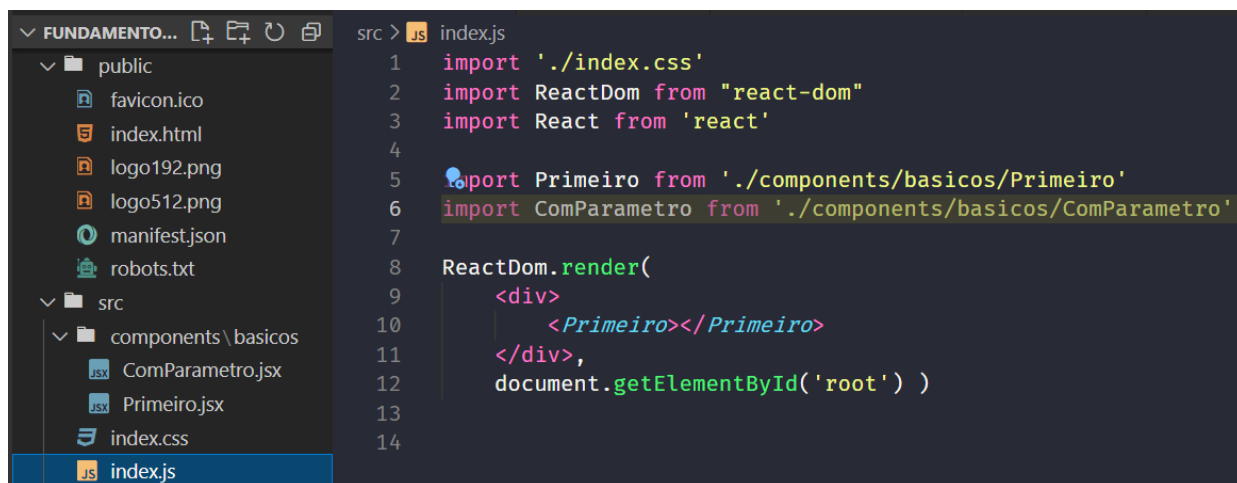
Vamos criar o arquivo ComParametro.jsx



The screenshot shows the VS Code interface. On the left, the file explorer displays the project structure: 'FUNDAMENTO...' (expanded), 'public' (with files like favicon.ico, index.html, logo192.png, logo512.png, manifest.json, robots.txt), 'src' (expanded), and 'components\basicos' (expanded). The file 'ComParametro.jsx' is selected. The main editor shows the code for 'ComParametro.jsx' with line numbers 1 to 10. The code is as follows:

```
1 import React from "react"
2 export default function ComParametro() {
3   return (
4     <div>
5       <h2>titulo</h2>
6       <h3>subtitulo</h3>
7     </div>
8   )
9 }
10
```


E fazer a importação dele no arquivo index.js



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a 'public' folder containing 'favicon.ico', 'index.html', 'logo192.png', 'logo512.png', 'manifest.json', and 'robots.txt'. There is also a 'src' folder containing a 'components\basicos' subfolder with 'ComParametro.jsx' and 'Primeiro.jsx', and 'index.css' and 'index.js' files. The 'index.js' file is selected. The code editor shows the following code:

```
src > JS index.js
1  import './index.css'
2  import ReactDOM from "react-dom"
3  import React from 'react'
4
5  import Primeiro from './components/basicos/Primeiro'
6  import ComParametro from './components/basicos/ComParametro'
7
8  ReactDOM.render(
9    <div>
10      <Primeiro></Primeiro>
11    </div>,
12    document.getElementById('root') )
13
14
```