

## Aula 1 - Criando App

### Docupedia Export

Author:Ferro Alisson (CtP/ETS)

Date:23-Aug-2023 16:33

## Table of Contents

O que é o React?

Segundo o site oficial, Uma biblioteca javascript para criar interfaces para usuários.

Para criarmos uma aplicação React basta digitar "npx create-react-app <nome do projeto>"

```
PS U:\JS> npx create-react-app projeto_react
```

```
Creating a new React app in U:\JS\projeto_react.
```

```
Installing packages. This might take a couple of minutes.
```

```
Installing react, react-dom, and react-scripts with cra-template...
```

```
[#####.....] - idealTree:webpack-dev-server: timing idealTree:node_modules/webpack-dev-server Completed in 2043ms
```

Ao concluir a criação da pasta podemos rodar a nossa aplicação com "npm start", feito isso devemos esperar um pouco e uma pagina Web ira abrir com a tela inicial do React

```
Compiled successfully!
```

```
You can now view projeto_react in the browser.
```

```
Local: http://localhost:3000
```

```
On Your Network: http://10.234.198.168:3000
```

```
Note that the development build is not optimized.
```

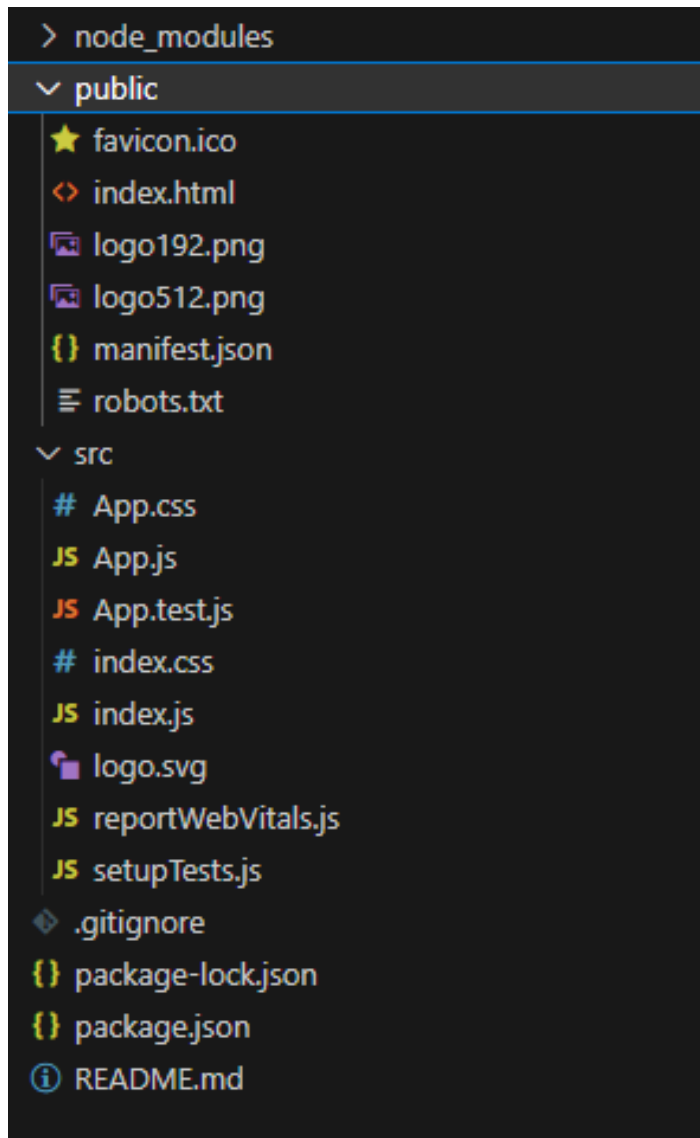
```
To create a production build, use npm run build.
```

```
webpack compiled successfully
```

```
█
```

Após concluído, uma página no navegador será exibida em localhost na 3000

A estrutura de arquivos inicialmente está da forma



Onde `src/index.js` é o ponto inicial da aplicação, e abrindo este arquivo encontramos

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

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

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

O React roda em componentes JSX, que são bem semelhantes ao HTML mas com pequenas diferenças, exemplo: podemos escrever códigos JS no nosso return, também podemos criar novas tags, className no lugar de class da tag, etc.

No arquivo src/index.js se observarmos o código, está criando uma constante root e no buscando um elemento no HTML com id "root", que abre uma tag React.StrictMode, StrictMode, segundo a documentação, é uma ferramenta para destacar problemas potenciais em um aplicativo, não renderiza nenhuma interface visível ao usuário. e dentro dessa tag temos o componente App, que está no nosso arquivo src/App.js

```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
      <a
```

```
        className="App-link"
        href="https://reactjs.org"
        target="_blank"
        rel="noopener noreferrer"
      >
        Learn React
      </a>
    </header>
  </div>
);
}

export default App;
```

E nossa renderização no navegador será semelhante a imagem abaixo

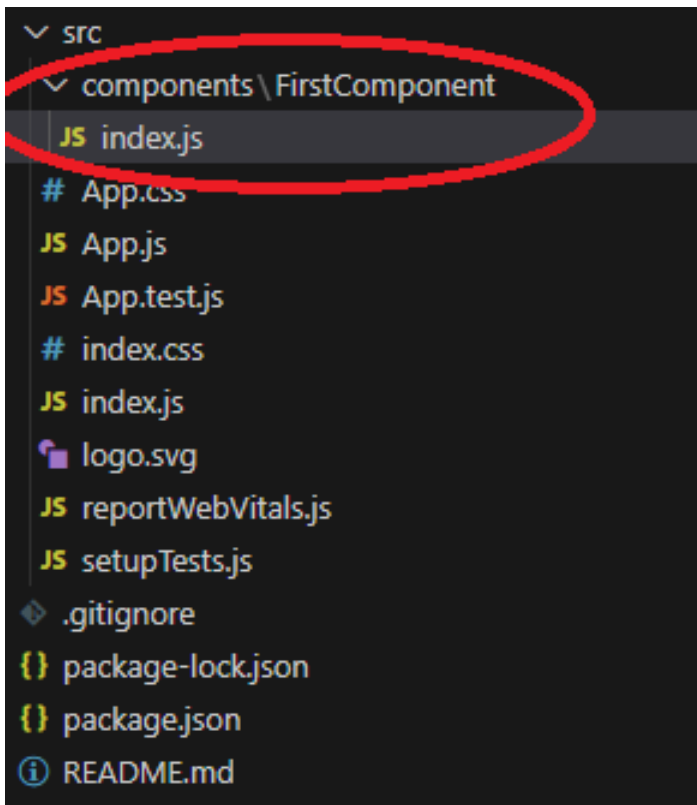


Edit `src/App.js` and save to reload.

[Learn React](#)

Dentro do return podemos observar a semelhança do JSX com HTML.

Vamos criar nosso primeiro componente, para isso é só criar um novo arquivo. Vamos antes criar o seguinte diretório `src/components/FirstComponent` e lá criamos nosso arquivo `index.js`



e dentro desse arquivo vamos colocar o seguinte código

```
export default function FirstComponent(){  
  return(  
    <p>  
      Meu primeiro componente  
    </p>  
  )  
}
```

mas ao salvar nada ainda foi alterado, precisamos chamar nosso componente no App, em src/App.js colocamos o código



```
import FirstComponent from './components/FirstComponent'

function App() {
  return (
    <FirstComponent />
  );
}

export default App;
```

agora sim nossa App exibe nosso texto,

O React foi criado pensando no reaproveitamento de componentes, como por exemplo, no facebook temos várias publicações, cada publicação tem seu card, e elas são reaproveitadas. Nós podemos fazer isso também, basta chamar várias vezes o nosso componente, veja o seguinte código

```
import React from 'react'
import FirstComponent from './components/FirstComponent'

function App() {
  return (
    <React.Fragment>
      <FirstComponent />
      <FirstComponent />
      <FirstComponent />
    </React.Fragment>
  );
}

export default App;
```

Observe que precisamos importar o React e utilizar a tag React.Fragment, isso ocorre pois cada componente só pode ter uma tag pai no return, quando tínhamos apenas um FirstComponent não era necessário pois só tinha um elemento pai no return, quando passamos a ter mais de um é necessário adiciona essa tag, ou também podemos somente utilizar o atalho <></> que abrimos uma tag vazia e fechamos no final do código, ficando da seguinte maneira

```
import FirstComponent from './components/FirstComponent'
```

```
function App() {  
  return (  
    <>  
      <FirstComponent />  
      <FirstComponent />  
      <FirstComponent />  
    </>  
  );  
}  
  
export default App;
```

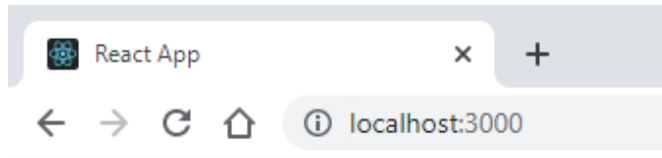
Observe que como a tag `React.Fragment` foi substituída por uma tag vazia, não é mais necessário importar o `React` no `App` para adicionar estilo no nosso componente, é semelhante ao HTML puro, basta colocar um nome na classe ou a tag e importar o `css`, vamos criar então um arquivo em `src/components/FirstComponent/styles.css`.

```
@import url('https://fonts.googleapis.com/css2?family=Wix+Madefor+Display&display=swap');  
  
.paragrafo {  
  color: red;  
  font-family: 'Wix Madefor Display';  
}
```

Com isso, adicionamos a importação e o `className` na tag

```
import './styles.css';  
  
export default function FirstComponent(){  
  return(  
    <p className='paragrafo'>  
      Meu primeiro componente  
    </p>  
  )  
}
```

Observe que foi adicionado estilo agora ao nosso parágrafo



Meu primeiro componente

Meu primeiro componente

Meu primeiro componente