

Desenvolvimento Front-End com ReactJS

Com a evolução das tecnologias de desenvolvimento web, novas técnicas e abordagens passaram a ser empregadas para melhorar os aspectos visuais das aplicações e buscando facilitar a manutenção ao longo do tempo. Como proposta para simplificar a renderização de elementos na interface do Facebook, surge, em 2011, a biblioteca ReactJS que, por meio do uso de componentes reutilizáveis, facilita o gerenciamento dos estados da aplicação. Em 2013, o código fonte da biblioteca foi aberto à comunidade, tornando-se rapidamente uma das soluções mais populares dentre os desenvolvedores Javascript.

Primeira Aplicação ReactJS

Para criar a estrutura inicial do projeto usaremos o “create-react-app” que é o starter-kit (kit de ferramentas para iniciar projetos) mais usado no mercado de trabalho. Ele não foi o primeiro a aparecer, mas de longe é o mais usado, além de que foi criado pelo Facebook que continua mantendo o projeto atualizado, dessa forma temos tranquilidade e confiança para usar o “create-react-app” em nossos projetos.

Abra o terminal do seu SO no qual você tem acesso ao NodeJS e npm que instalou anteriormente e rode o comando a seguir, para iniciar o projeto “reactjs-start” com o “create-react-app”:

```
npx create-react-app reactjs-start
```

O download e instalação dos pacotes pode demorar um pouco dependendo da sua conexão com a internet, mas ao final, caso tudo tenha corrido bem, deverá ser apresentada uma tela semelhante à imagem a seguir:

```
Success! Created reactjs-start at C:\Users\marco\Projects\reactjs-start
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 reactjs-start
  npm start

Happy hacking!
```

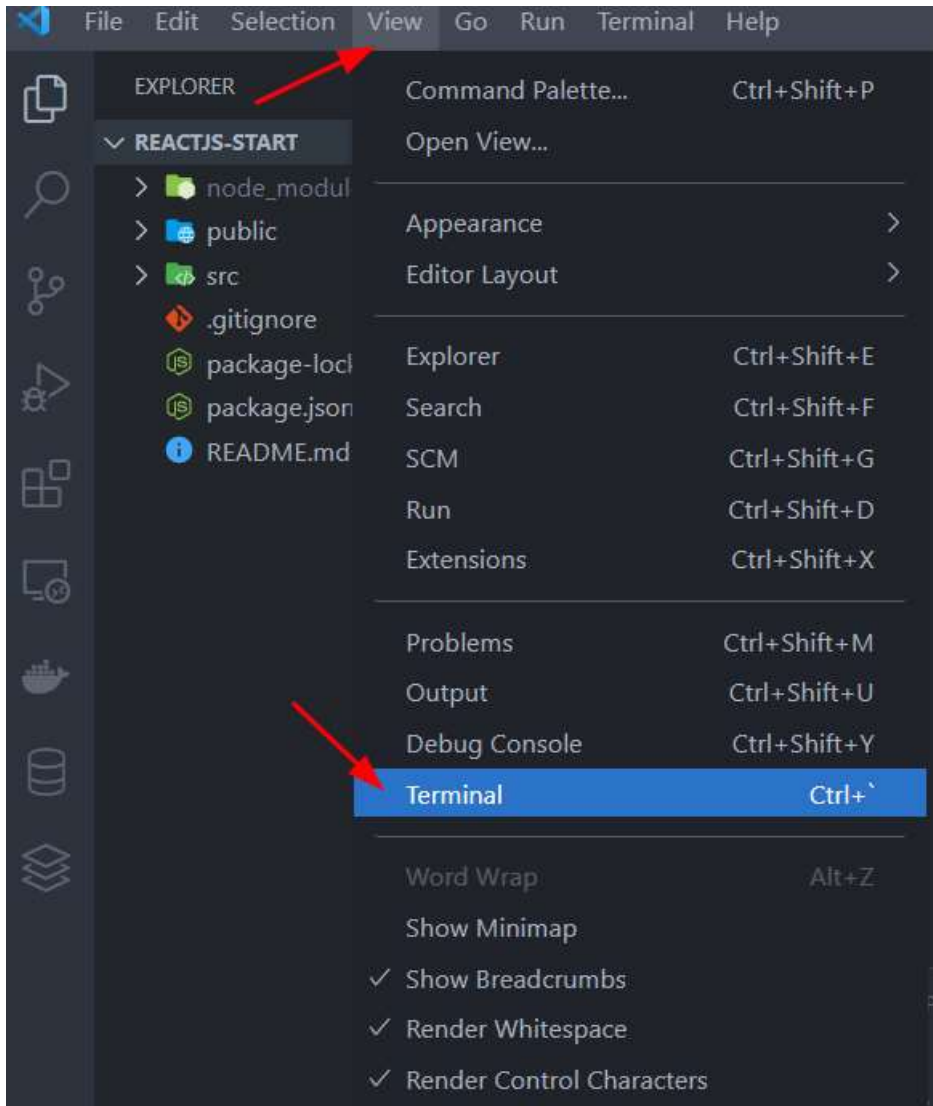
Acesse a pasta do projeto recém-criado:

```
cd reactjs-start
```

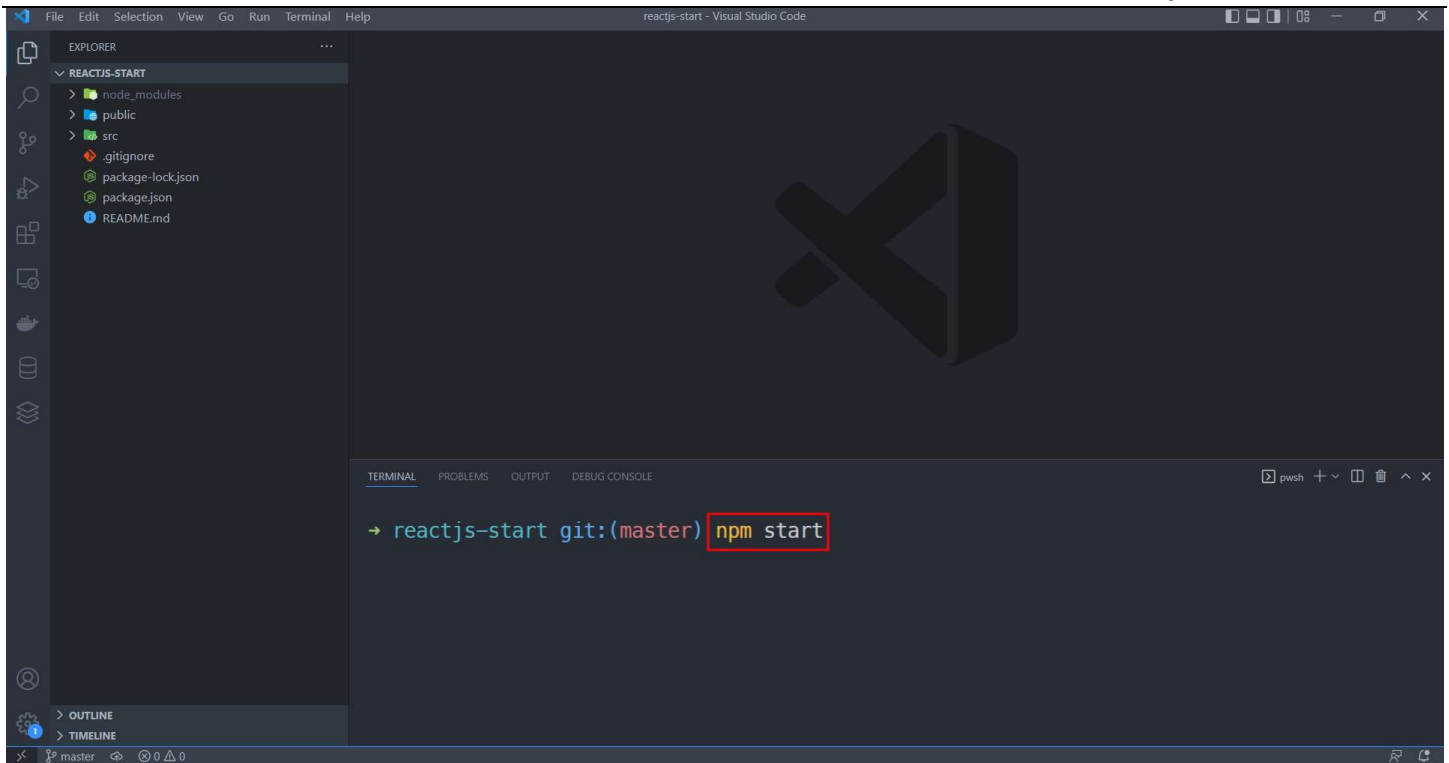
Abra a pasta de projetos no seu editor de códigos. No caso do Visual Studio Code:

```
code .
```

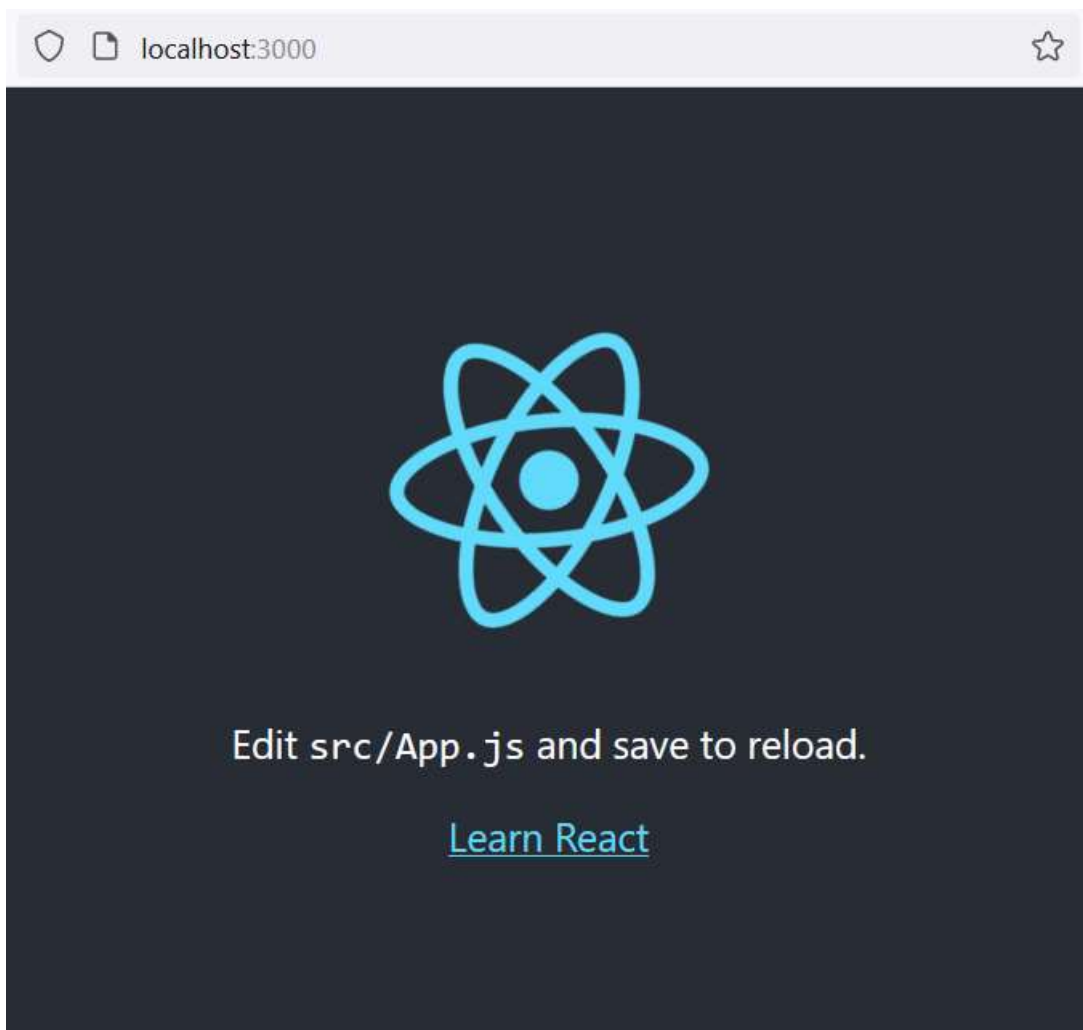
Agora que estamos com o Visual Studio Code aberto, não precisamos mais utilizar o terminal externo, podemos abrir um terminal integrado ao VSCode:



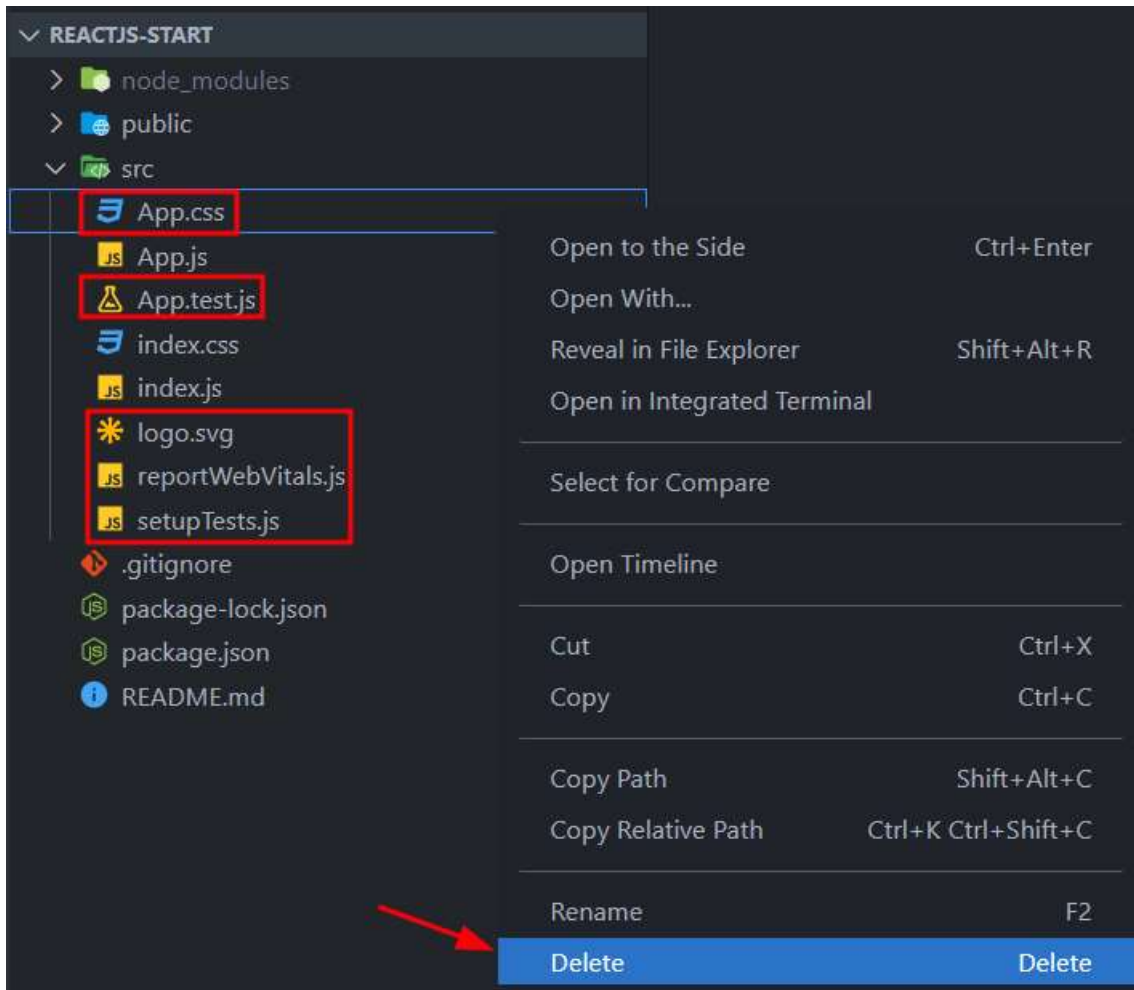
Vamos executar o comando “npm start” que subirá o nosso projeto:



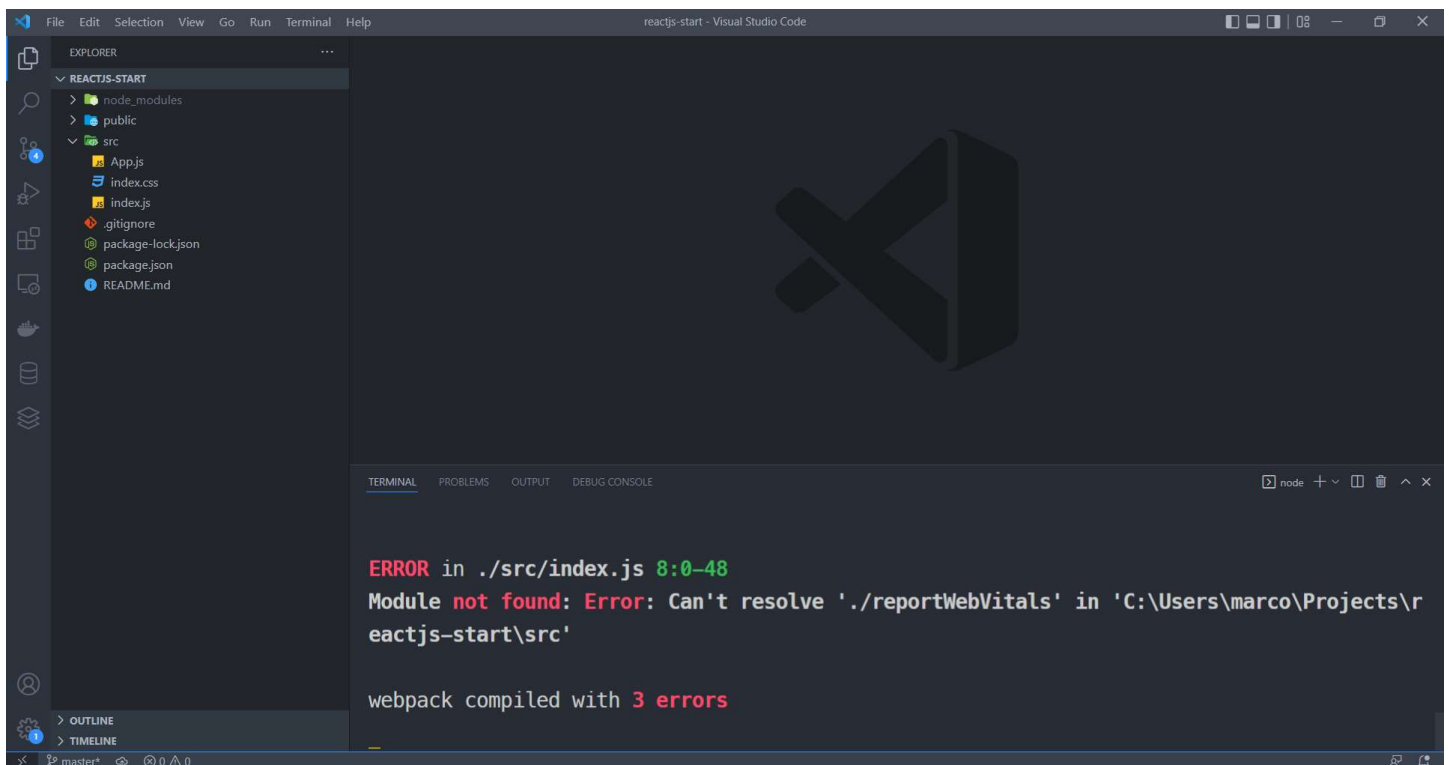
Observe que, após o carregamento da aplicação, será aberto o browser (navegador) que estiver definido como padrão no seu sistema operacional com a aplicação rodando:



Vamos remover os arquivos do modelo que não precisaremos para nossa aplicação clicando com o botão direito sobre o arquivo e selecionando “Delete” – “App.js”, “App.test.js”, “logo.svg”, “reportWebVitals.js”, “setupTests.js”:



Nosso editor ficará desta maneira:



Vamos remover do arquivo "index.js" as referências aos arquivos removidos:

```
JS index.js M X
src > JS index.js > ...
1  import React from 'react';
2  import ReactDOM from 'react-dom/client';
3  import './index.css';
4  import App from './App';
5  import reportWebVitals from './reportWebVitals';
6
7  const root = ReactDOM.createRoot(document.getElementById('root'));
8  root.render(
9    <React.StrictMode>
10     <App />
11   </React.StrictMode>
12 );
13
14 // If you want to start measuring performance in your app, pass a function
15 // to log results (for example: reportWebVitals(console.log))
16 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
17 reportWebVitals();
```

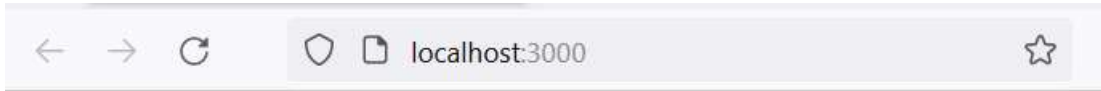
O arquivo "index.js" ficará como na listagem a seguir:

```
JS index.js M X
src > JS index.js > ...
1  import React from 'react';
2  import ReactDOM from 'react-dom/client';
3  import './index.css';
4  import App from './App';
5
6  const root = ReactDOM.createRoot(document.getElementById('root'));
7  root.render(
8    <React.StrictMode>
9     <App />
10   </React.StrictMode>
11 );
```

Vamos, agora, editar o arquivo "App.js" deixando-o como na listagem a seguir:

```
JS App.js M X
src > JS App.js > ...
1  function App() {
2    return (
3      <div>
4        <h1>Olá ReactJS!!!</h1>
5      </div>
6    );
7  }
8
9  export default App;
```

Observe que no navegador já aparece a nossa mensagem de saudação:



Olá ReactJS!!!