

# Anotações do Curso ReactJS da B7Web - Módulo 01

(Por: Daniel Teixeira Quadros)

## Sumário

Aula 01 – Introdução.....	2
Aula 02 – Editor de código VS Code.....	2
Aula 03 – Instalando o Node/Npm.....	2
Aula 04 – Como o React Funciona.....	2
Aula 05 – Adicionando o React a um Site Via CDN.....	3
Aula 06 – Criando Um Projeto React.....	5

# Aula 01 – Introdução

Ferramentas e conhecimentos necessários:

- Editor de códigos
  - Base em JavaScript, HTML e CSS.
  - Familiarização com o conceito do gerenciador de pacotes do JavaScript (Node) → Que é o NPM.
  - Node v8 (NPM versão 5(ou mais) ou Yarn).
- 

## Aula 02 – Editor de código VS Code

Editor de Código VS Code.

---

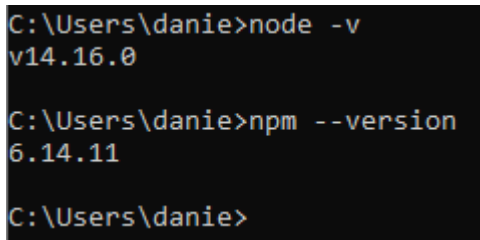
## Aula 03 – Instalando o Node/Npm

link para download: <https://nodejs.org/en/>

Para verificar se o Node e o npm já estão instalado, e também para verificarmos a versão que foram instalados, abrimos o prompt de comando e executamos os seguintes comando:

**node -v** → Para verificarmos a versão do Node

**npm --version** → Para verificarmos a versão do npm



```
C:\Users\danie>node -v
v14.16.0

C:\Users\danie>npm --version
6.14.11

C:\Users\danie>
```

Se o Prompt de comando retornar com alguma mensagem do tipo, comando não encontrado ou alguma coisa similar, é por que o(s) aplicativo(s) não está(estão) instalado(s).

---

## Aula 04 – Como o React Funciona

**Visão geral:** O processo de manipulação dos elementos na tela funciona de maneira um pouco diferente do javascript.

Com o Javascript existe uma manipulação direta do DOM (Exigindo um maior processamento).

O React, por sua vez, não faz uma manipulação direta do DOM. Ele manipula o chamado Virtual DOM, ou seja uma réplica do que está acontecendo no DOM e que está na memória do seu equipamento. Tornando o processamento muito mais rápido.

---

# Aula 05 – Adicionando o React a um Site Via CDN

CDN → URLs que dão diretamente para o arquivo Javascript de uma determinada biblioteca. Podemos adicionar esta URL diretamente no nosso site

## O PRIMEIRO MODELO É COMO ADICIONAMOS O REACT A UM SITE JÁ EXISTENTE:

Adicionamos estes dois script abaixo para incluirmos os comandos do react no nosso site.

```
<script src="https://unpkg.com/react@16/umd/react.development.js"></script>
```

**Obs. MUITO IMPORTANTE:** neste link, no último campo, utilizamos o react.development.js, porém, no momento de colocarmos o site no ar, devemos trocar este último campo por react.production.js.

```
<script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
```

Com o React, não escrevemos HTML aqui no arquivo HTML. Criamos uma DIV e rodamos o react nesta DIV específica e tudo o que montarmos será renderizado dentro desta DIV

```
<div id="site"></div>
```

Adicionamos um script no corpo(body) do site.  
E neste script criamos os seguintes elementos:

```
let botao = React.createElement('button', {}, 'Clique aqui')
```

**let botao = React.createElement('button', {}, 'Clique aqui')** → Aqui, utilizamos a forma padrão (forma raiz) para criarmos este botão.

Utilizamos o:

**React.createElement** → Para criar um elemento

**button** → É o elemento que está sendo criado.

**{}** → O segundo parâmetro são atributos eventuais que ele precisa ter. Neste caso não adicionaremos nenhum.

**'Clique aqui'** → Texto do botão

**ReactDOM.render()** → Esta é a função que renderiza o React, a função inicial.

Esta função possui dois parâmetros:

**botao** → Primeiro parâmetro é o que eu quero renderizar neste site.

**document.getElementById('site')** → Segundo parâmetro é qual a div será renderizada

Nosso HTML Inicialmente ficou assim:

```

m01a05 > <> index.html > html > body
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <script src="https://unpkg.com/react@16/umd/react.development.js"></script>
8      <script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
9      <title>m01a05</title>
10 </head>
11 <body>
12     <div id="site"></div>
13     <script>
14         let botao = React.createElement('button', {}, 'Clique aqui')
15         ReactDOM.render(botao, document.getElementById('site'))
16     </script>
17 </body>
18 </html>

```

## BABEL:

Neste momento, aprendemos a utilizar o Babel, ele nos permite a trabalharmos com uma tecnologia chamada de JSX.

Para utilizá-lo, precisamos adicionar mais um scrit no <head> do nosso site

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

**type="text/babel"** → Precisamos adicionar este type no nosso script para que ele utilize o Babel para a interpretação do script.

```
<script type="text/babel">
```

**let botao = <button>Clique aqui</button>** → Embora, o conteúdo da variável botão sejam comandos html, neste caso, dentro do script, isto não é html. Ao termos incluído o babel no nosso site ele converte este html no comando que havíamos criado anteriormente (React.createElement('button', {}, 'Clique aqui')) tornando o nosso código muito mais simples.

```
let botao = <button>Clique aqui</button>
```

Ficamos com o nosso html da seguinte forma:

```

m01a05 > <> index.html > html > body > script
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <script src="https://unpkg.com/react@16/umd/react.development.js"></script>
8      <script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
9      <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
10     <title>m01a05</title>
11 </head>
12 <body>
13     <div id="site"></div>
14     <script type="text/babel">
15         let botao = <button>Clique aqui</button>
16         ReactDOM.render(botao, document.getElementById('site'))
17     </script>
18 </body>
19 </html>

```

É muito importante lembrar que o método aprendido nesta aula é utilizado apenas para incluir o React em um trecho de um site já existente. O Babel é uma tecnologia que exige muito processamento e não é ideal para utilizarmos em um site construído totalmente utilizando o React

A PARTIR DA PRÓXIMA AULA APRENDEREMOS OS PROCEDIMENTOS PARA A CRIAÇÃO DE PROJETOS EM REACT.

## Aula 06 – Criando Um Projeto React

Primeiramente, para criarmos um projeto em React, precisamos acessar o Terminal do VS Code ou o Prompt de comando e executar o seguinte comando:

**npx create-react-app nomedoprojeto** → Este comando deve ser executado dentro da pasta que criaremos o nosso projeto e ele instalará o pacote React no nosso projeto

```

D:\Informática\Programação\2021\Daniel\B7Web\ReactJS\modulo01>npx create-react-app m01a06
npx: installed 67 in 8.861s

Creating a new React app in D:\Informática\Programação\2021\Daniel\B7Web\ReactJS\modulo01\m01a06.

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

```

We suggest that you begin by typing:

```

cd m01a06
npm start

```

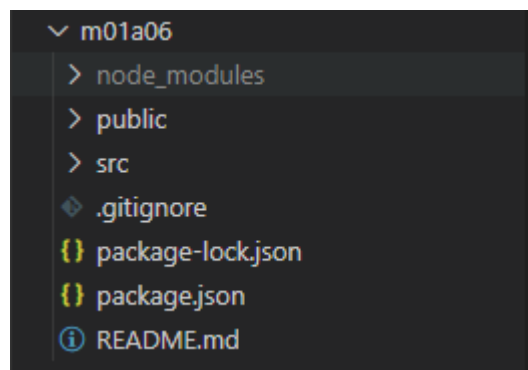
Happy hacking!

```

D:\Informática\Programação\2021\Daniel\B7Web\ReactJS\modulo01>

```

No VS Code Ficou assim:



Se por acaso, não vier o projeto completo desta forma, no caso de já haver sido instalado uma versão mais antiga do react, será necessário reinstalar. Para isto, desinstalamos e instalamos novamente. Se for este o caso, utilizamos o seguinte comando para a desinstalação:

**npm uninstall create-react-app -g**

No terminal do VS Code ou no prompt de comando, dentro da pasta do nosso projeto, precisamos executar o React. Para isto executamos o seguinte comando **npm start**

```
PS D:\Informática\Programação\2021\Daniel\B7Web\ReactJS\modulo01\m01a06> npm start

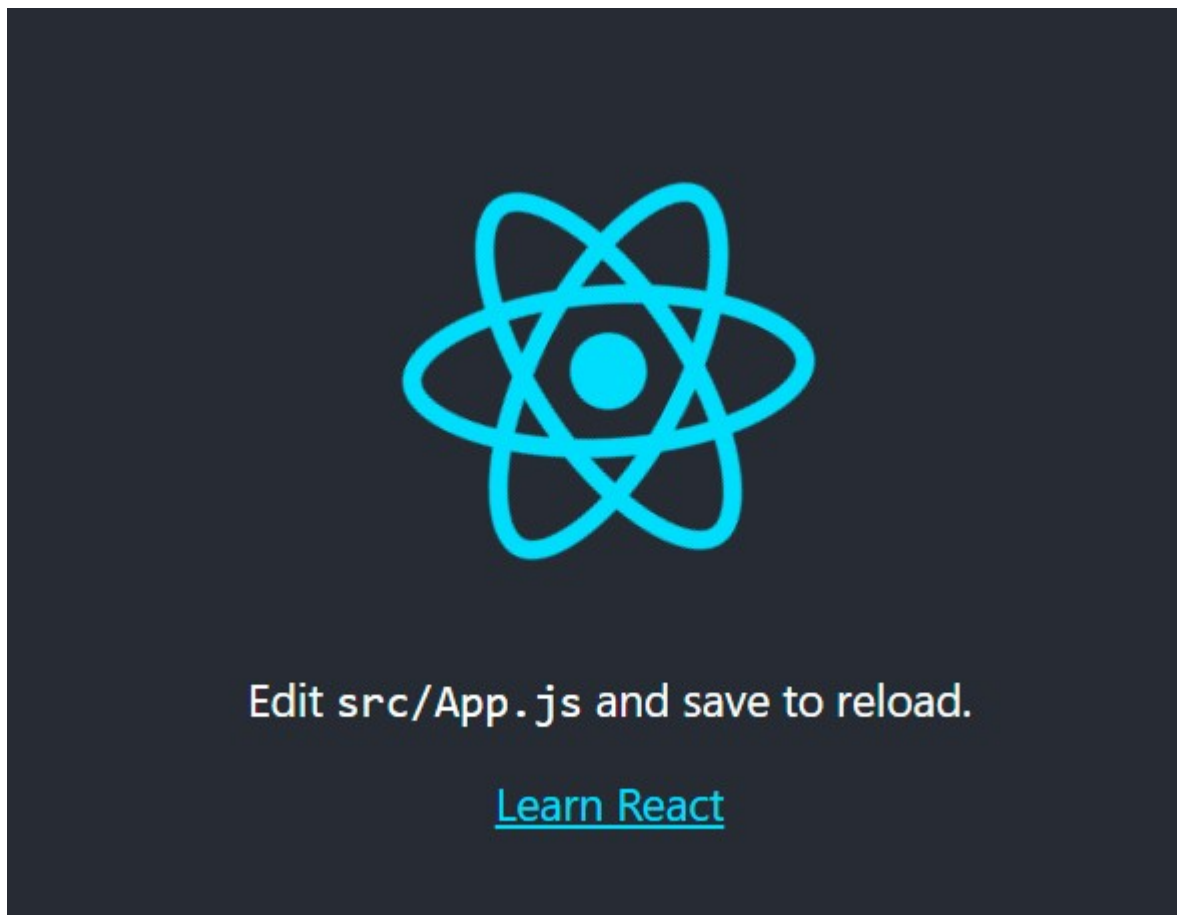
> m01a06@0.1.0 start D:\Informática\Programação\2021\Daniel\B7Web\ReactJS\modulo01\m01a06
Compiled successfully!

You can now view m01a06 in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.3.3:3000

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

E o VS Code já abrirá o nosso projeto no navegador:



Se por algum motivo precisarmos parar de rodar o nosso projeto, no terminal, utilizamos as teclas de atalho **Ctrl+C**

---

Todo o nosso projeto está na pasta **src** e ele inicia rodando o arquivo **index.js**

