

React JS

**GUIA COMPLETO**

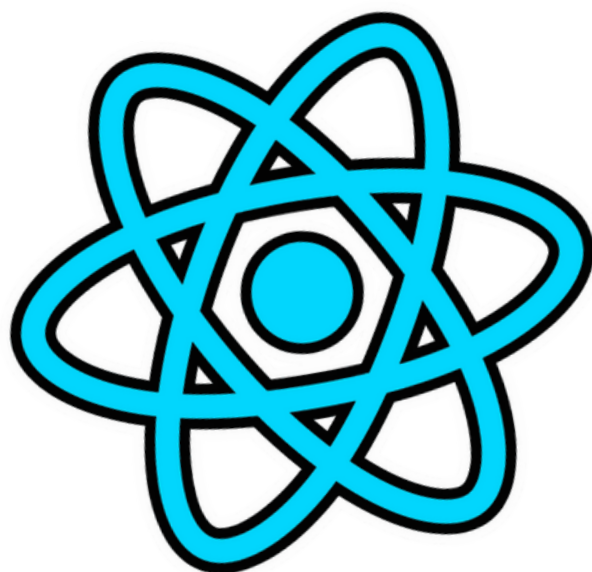
**CONSTRUINDO  
APLICAÇÕES WEB**

**SUJEITO PROGRAMADOR**

---

O Guia para dominar a tecnologia mais requisitada do mercado!

# React JS



## Índice

---

Introdução	3
O que é React?	6
Recursos e vantagens do react	8
Mão na massa	10
Primeiro projeto em react	14
React Native	21

# Introdução

A primeira coisa antes de iniciar sua jornada, antes de começar a desenvolver uma linha de código você precisa entender bem o que é Front-end.

O desenvolvedor front-end é responsável por "dar vida" à interface. Trabalha diretamente com o usuário, com o que o usuário está vendo no sistema e como ele está vendo.

E para trabalhar no front end vamos dividir em 3 camadas para ficar mais fácil você entender.

1. **HTML** Ela exibe a informação que o usuário consome, é nela que você marca um título, parágrafo, um pedaço de informação e dá significado a cada pedaço do seu projeto.

2. **CSS** É a camada responsável pela formatação e transformação do layout criado. Ou seja CSS vai dar forma à informação marcada pelo HTML, para que ela seja encaixada não só no seu devido lugar mas também controlar para que seja bem vista em qualquer dispositivo.

3. JavaScript já o javascript nós conseguimos não só trabalhar no lado front-end mas também conseguimos criar aplicações back-end que rodam no lado servidor como por exemplo usando nodejs.

[CLIQUE AQUI PARA CONHECER A FABRICA DE APLICATIVOS](#)

## Mas então onde usamos javascript?

O javascript é uma linguagem de programação que permite implementar itens complexos em projetos, por exemplo um SLIDE de algum site que você conhece quando aquele slide troca e aparece outro slide ali por baixo dos panos tem um javascript manipulado todos esses passos.

Conseguimos criar e manipular dados de forma dinamica, controlar imagens, criar animações e muito mais.

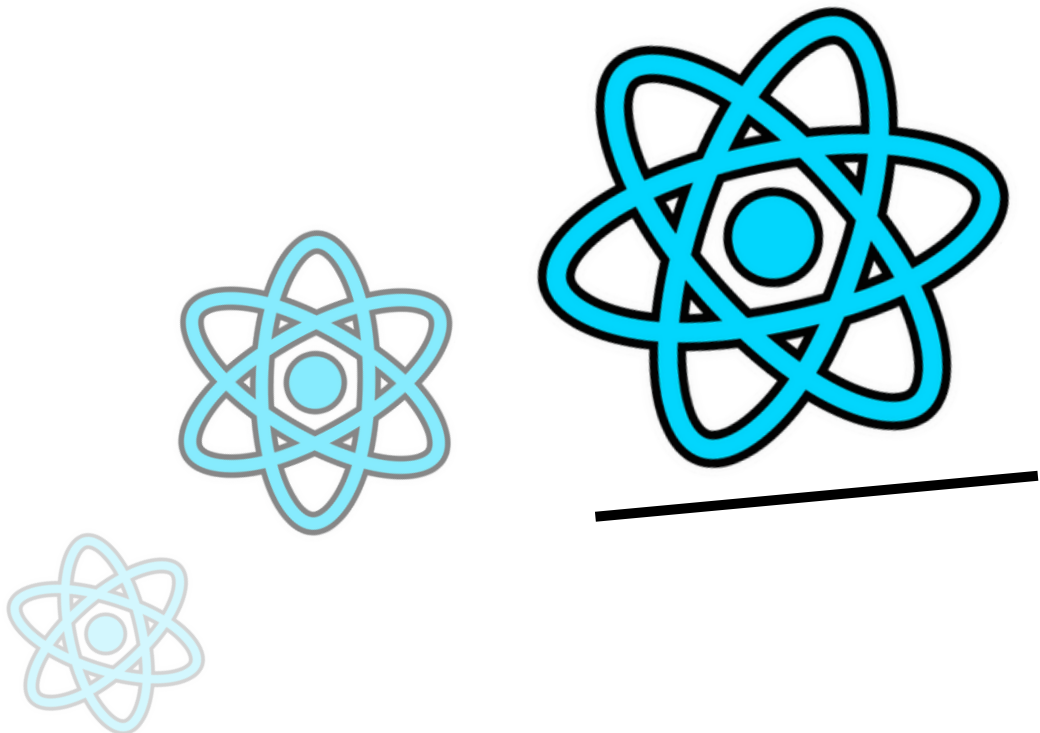


# Produtividade

Mas então se você agora está se perguntando "**Eu consigo desenvolver um sistema completo usando apenas HTML, CSS, JAVASCRIPT ?**" E a resposta é sim consegue sim sem problemas.

Mas existem maneiras que tornam esse caminho muito mais produtivo e agil, onde além disso conseguimos pensar sempre em manutenibilidade do projeto e tornar o projeto escalavel. Exatamente eles os famosos Frameworks e bibliotecas que vão nos ajudar nessa jornada como desenvolvedor.

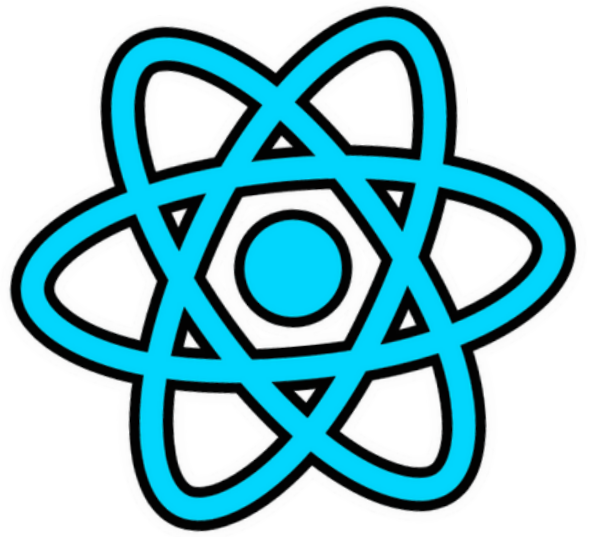
Exitem varios frameworks e biliotecas que podem ajudar mas aqui vamos trabalhar com **React JS**.



[CLIQUE AQUI PARA CONHECER À FABRICA DE APLICATIVOS](#)

# React JS

## O que é esse tal de React JS?



React é uma biblioteca JavaScript criada pelo facebook usada para criar interfaces de usuários baseada em componentes.

Podemos criar pedaços isolados do nosso projeto que damos o nome de componentes de forma pratica e facil e alem disso podemos reutilizar esses componentes em outra parte do projeto. Melhor forma de entender isso de uma vez por todas é vendo.



### Declarativo

React faz com que a criação de UIs interativas seja uma tarefa fácil. Crie views simples para cada estado na sua aplicação, e o React irá atualizar e renderizar de forma eficiente apenas os componentes necessários na medida em que os dados mudam.

Views declarativas fazem com que seu código seja mais previsível e simples de depurar.

### Baseado em componentes

Crie componentes encapsulados que gerenciam seu próprio estado e então, combine-os para criar UIs complexas.

Como a lógica do componente é escrita em JavaScript e não em templates, você pode facilmente passar diversos tipos de dados ao longo da sua aplicação e ainda manter o estado fora do DOM.

### Aprenda uma vez, use em qualquer lugar

Não fazemos suposições sobre as outras tecnologias da sua stack, assim você pode desenvolver novos recursos com React sem reescrever o código existente.

O React também pode ser renderizado no servidor, usando Node, e ser usado para criar aplicações mobile, através do [React Native](#).

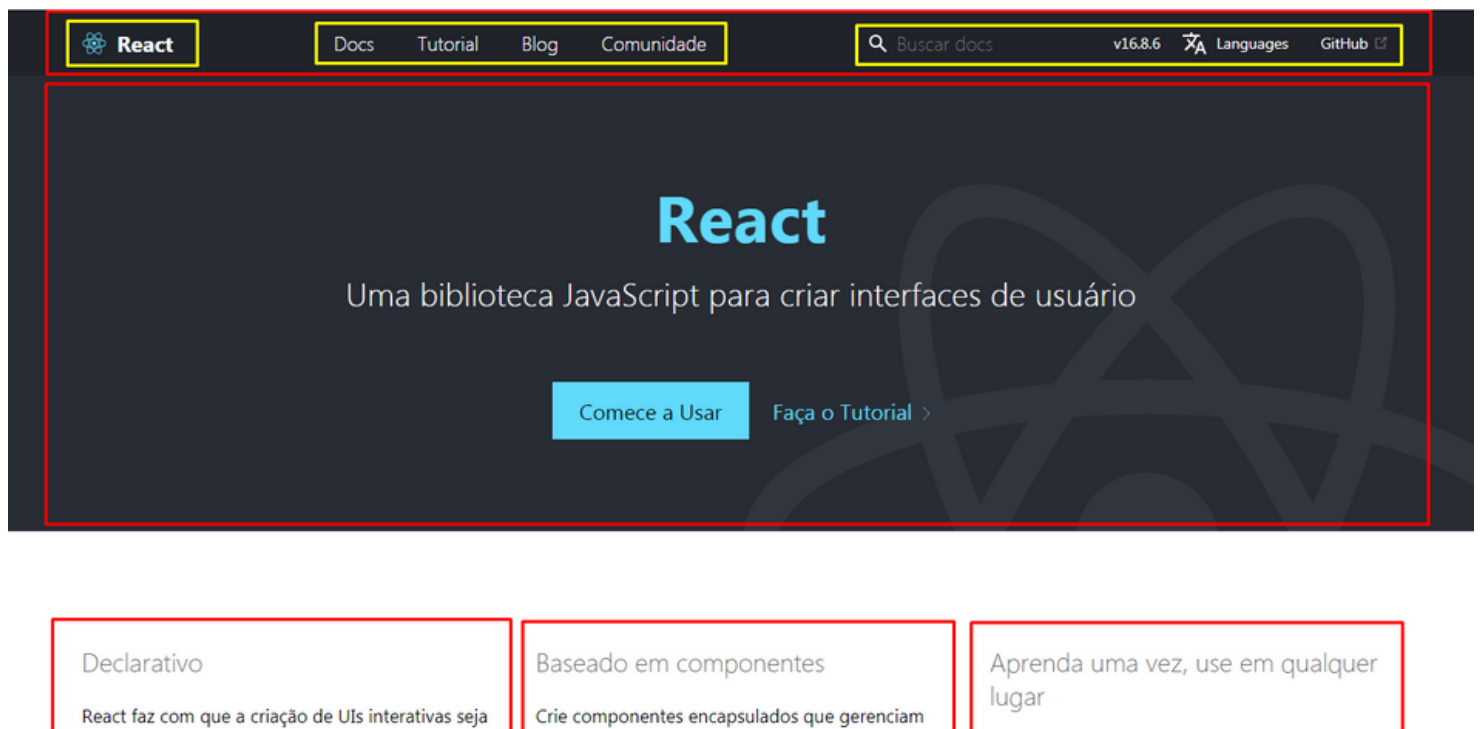
**CLIQUE AQUI VEJA IMAGEM EM TELA CHEIA**

# React JS

Cada área **vermelha** pode ser um componente isolado onde ele cuida apenas daquela área.

Ou seja podemos ter um componente onde apenas cuidamos do **header**, outro onde cuidamos apenas do **slide** e outro componente onde cuidamos apenas das **informações**.

E esses componentes podem ainda se subdividir, vamos lá ver como isso ficaria.



**CLIQUE AQUI VEJA IMAGEM EM TELA CHEIA**

Cada área **amarela** pode ser um componente dentro do componente header onde temos um componente só pra logo, outro para apenas cuidar do menu e outro para o restante do header, podemos ter isso sem problemas e de uma forma muito pratica.

**CLIQUE AQUI PARA CONHECER À FABRICA DE APLICATIVOS**



# React JS

## Recursos do react:

**Componentes** - React é tudo sobre componentes. Você precisa pensar em tudo como um componente. Isso ajudará você a manter o código ao trabalhar em projetos de escala maior e ainda ter uma boa manutenibilidade.

**JSX** - .JSX é extensão de sintaxe JavaScript. Onde você consegue escrever e compreender de uma melhor forma, como será montado o seu component na interface.

Basicamente, usando o JSX você pode escrever estruturas concisas do tipo HTML, e no mesmo arquivo que você escreve o código JavaScript.

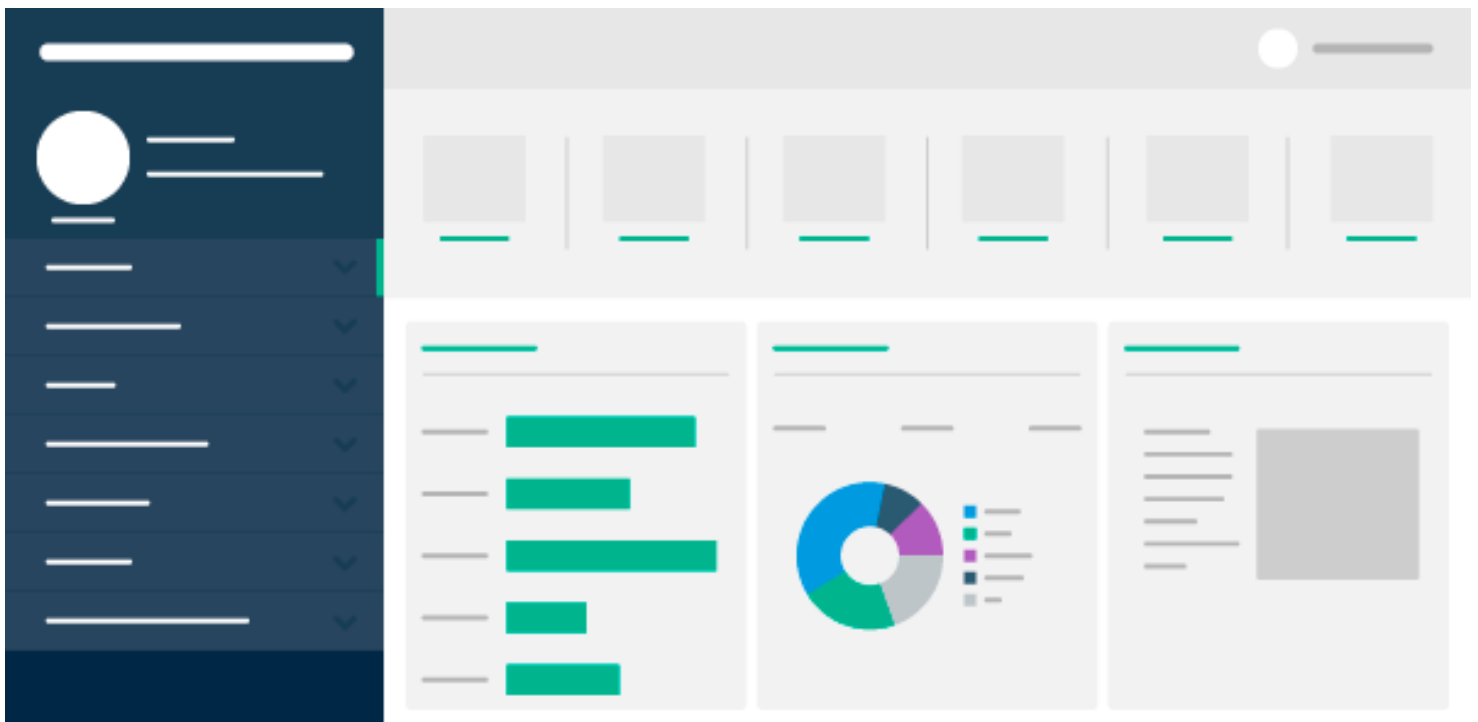
## Vantagens do React:

- Oportunidades do mercado, tanto reactjs quanto react native possuem muitas vagas no mercado de trabalho tanto como desenvolvedor front-end quanto mobile.
- Usa o virtual DOM, que é um objeto JavaScript. Isso melhorará o desempenho dos aplicativos, pois o virtual DOM JavaScript é mais rápido que o DOM comum.

[CLIQUE AQUI PARA CONHECER À FABRICA DE APLICATIVOS](#)



- Conseguimos desenvolver aplicações web e também aplicativos mobile para android e ios usando React Native reaproveitando todo nosso conhecimento em ReactJS.
- Usamos javascript para desenvolver nossas aplicações além da maior comunidade de desenvolvedores serem de javascript.
- React é veloz e tem componentes reutilizáveis.



[\*\*CLIQUE AQUI PARA CONHECER À FABRICA DE APLICATIVOS\*\*](#)

# React JS

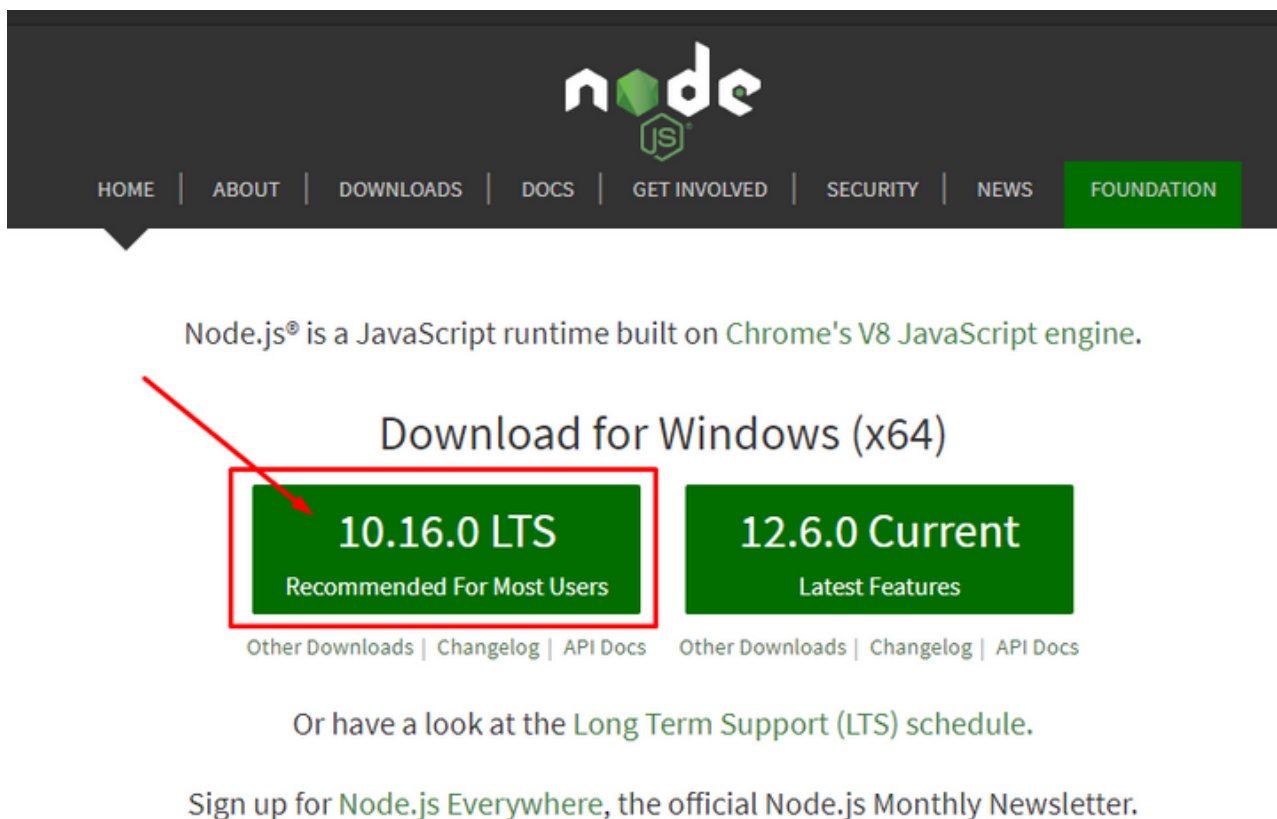
## Mão na massa:

Como começo a desenvolver usando react e aplicando tudo que aprendemos até agora?

- Primeira coisa precisamos ter o **NodeJS** instalado.

- [Clique aqui para instalar o nodeJS](#)

Recomendamos instalar a versão LTS que é a mais estável, após baixar é só seguir os passos padrão de instalação.



Node.js® is a JavaScript runtime built on [Chrome's V8 JavaScript engine](#).

Download for Windows (x64)

10.16.0 LTS	12.6.0 Current
Recommended For Most Users	Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#)   [Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [Long Term Support \(LTS\) schedule](#).

Sign up for [Node.js Everywhere](#), the official Node.js Monthly Newsletter.

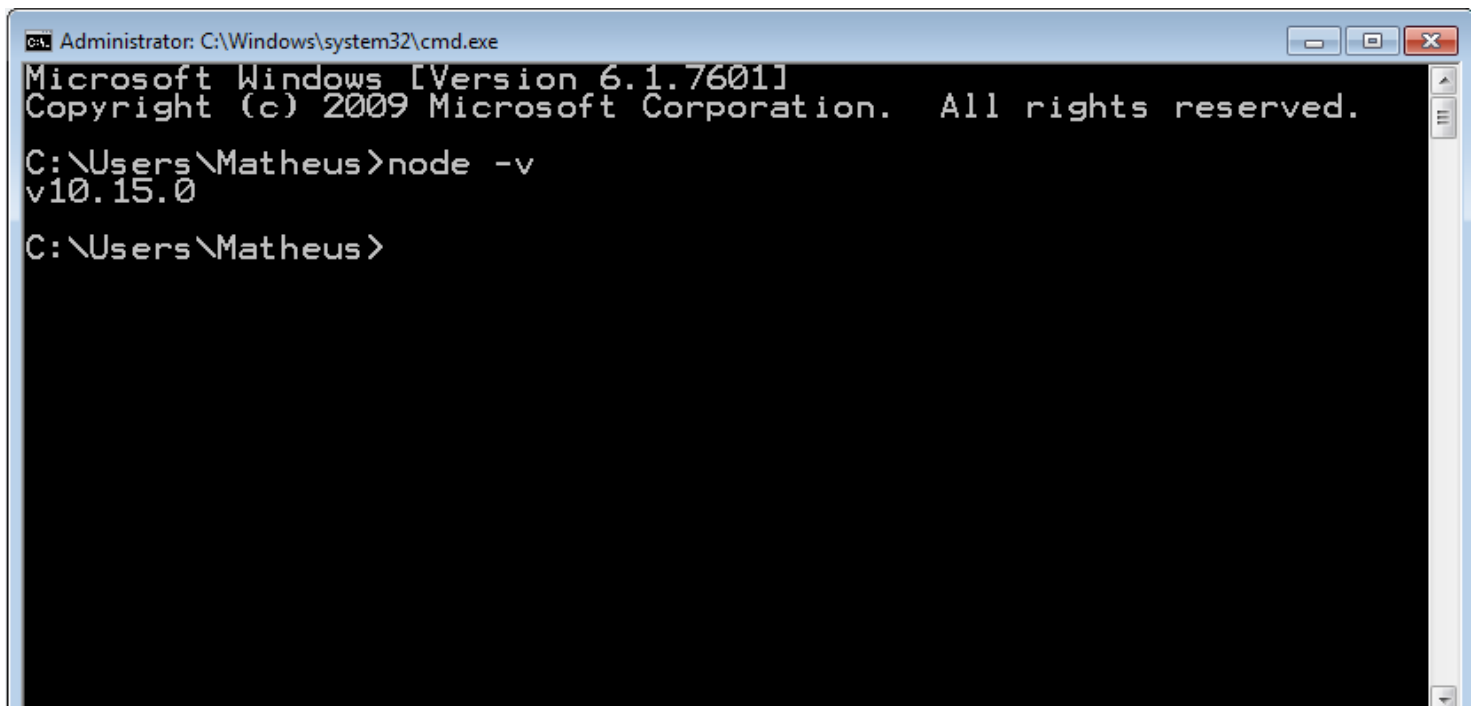
[CLIQUE AQUI PARA CONHECER À FABRICA DE APLICATIVOS](#)

# Mão na massa:

Agora par ter certeza que o nodeJS foi instalado corretamente abra seu cmd e digite:

**node -v**

Ele deve retornar a versão atual do node. **Exemplo abaixo:**



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Matheus>node -v
v10.15.0

C:\Users\Matheus>
```

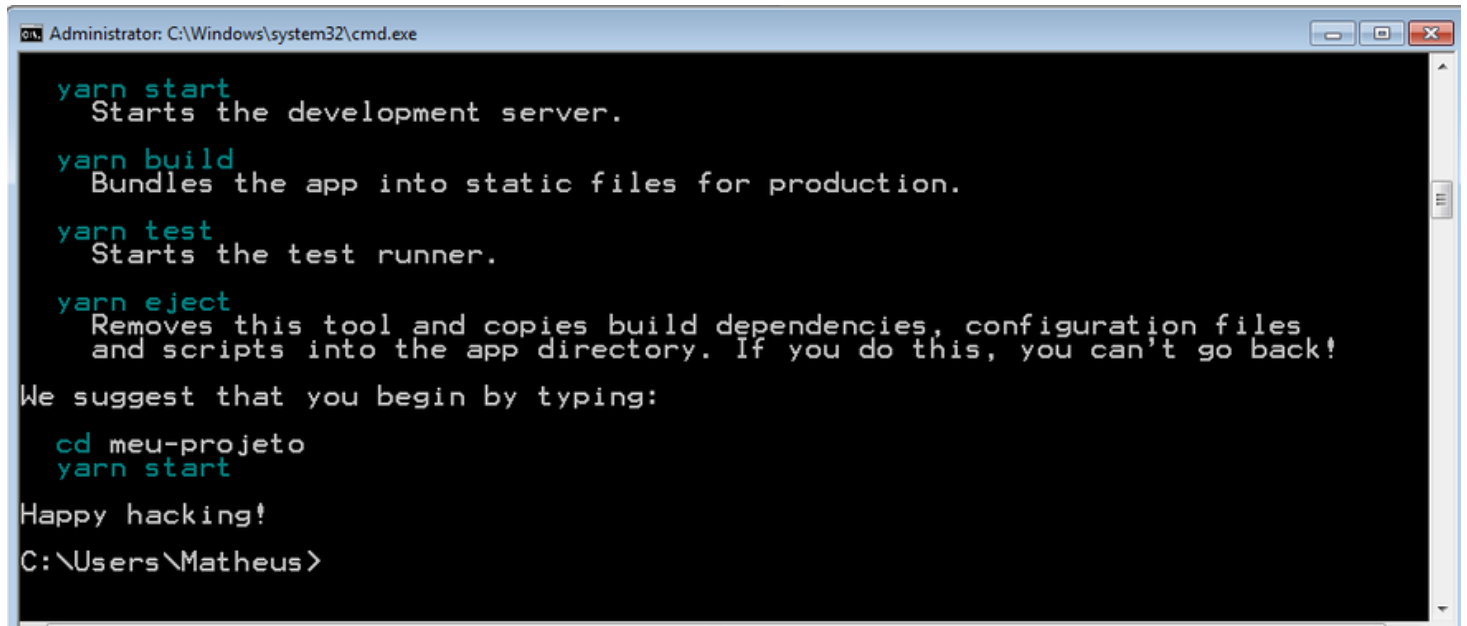
Para criarmos um novo projeto com react podemos rodar o comando no cmd:

**npx create-react-app meu-projeto**

Após rodar este comando é só aguardar, ele irá baixar tudo que é necessario para começar a desenvolver nosso projeto.

# Mão na massa:

Quando acabar ele vai ter um resultado parecido com esse:

A screenshot of a Windows command prompt window titled "Administrator: C:\Windows\system32\cmd.exe". The window has a black background with green and white text. It displays the following content:

```
yarn start
  Starts the development server.

yarn build
  Bundles the app into static files for production.

yarn test
  Starts the test runner.

yarn 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 meu-projeto
  yarn start

Happy hacking!
C:\Users\Matheus>
```

Ele vai ter criado uma pasta com nome do projeto que criou que no nosso caso criamos **meu-projeto**.

No meu caso ele criou dentro da pasta **Users\Matheus** e lá dentro **meu-projeto**.

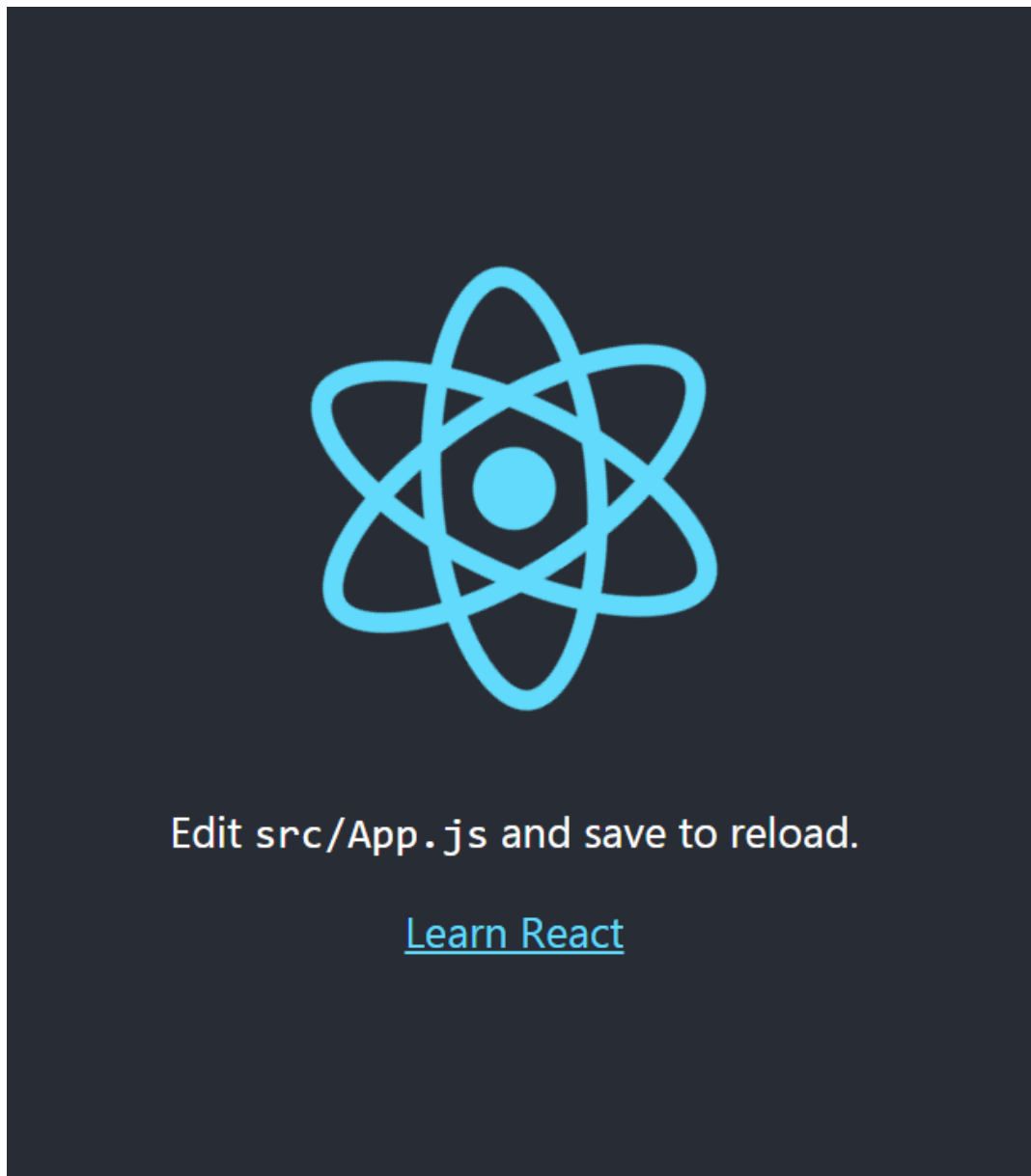
E sim tudo pronto já podemos rodar nosso projeto. No seu cmd você pode dar **cd meu-projeto** para entrar dentro da pasta do projeto pelo cmd e agora rode o comando:

**npm start**

Ele vai rodar seu projeto é só aguardar...

```
npm
yarn 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 meu-projeto
  yarn start
Happy hacking!
C:\Users\Matheus>cd meu-projeto
C:\Users\Matheus\meu-projeto>npm start
_
```

Após rodar tudo que é necessário irá abrir um navegador com seu projeto já rodando...



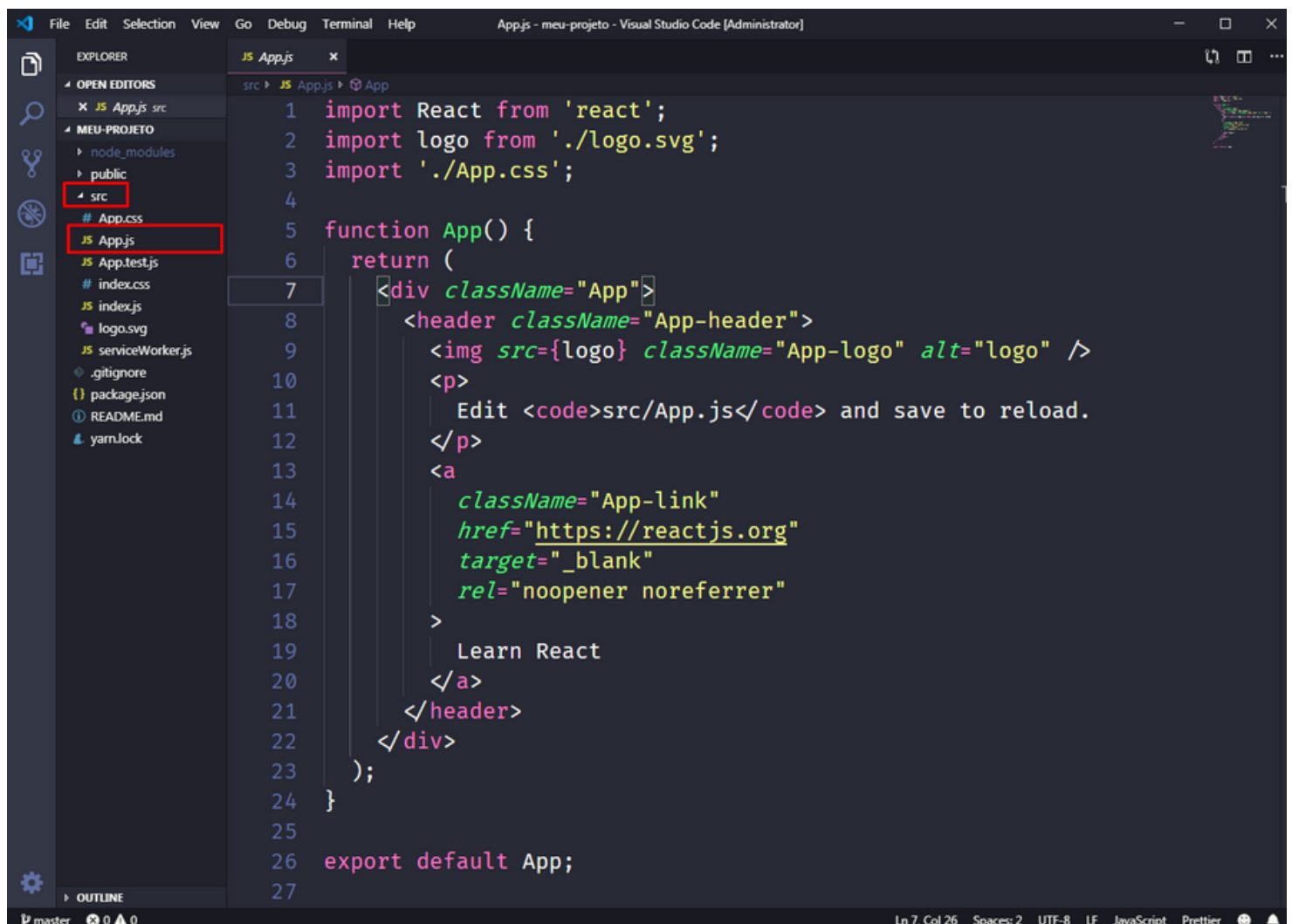
[CLIQUE AQUI PARA CONHECER A FABRICA DE APLICATIVOS](#)

# Primeiro projeto

Agora você pode minimizar o cmd deixar seu projeto rodando e abrir a pasta do seu projeto no seu editor preferido.

Aqui no ebook vou usar o editor [Visual Studio Code](#)

Podemos abrir a pasta **src/** e dentro dela o arquivo **App.js**



```
1 import React from 'react';
2 import logo from './logo.svg';
3 import './App.css';
4
5 function App() {
6   return (
7     <div className="App">
8       <header className="App-header">
9         <img src={logo} className="App-logo" alt="logo" />
10        <p>
11          Edit <code>src/App.js</code> and save to reload.
12        </p>
13        <a
14          className="App-link"
15          href="https://reactjs.org"
16          target="_blank"
17          rel="noopener noreferrer"
18        >
19          Learn React
20        </a>
21      </header>
22    </div>
23  );
24 }
25
26 export default App;
```

[CLIQUE AQUI PARA CONHECER À FABRICA DE APLICATIVOS](#)

Vamos alterar nosso código:

```
JS App.js x
src JS App.js App
1 import React from 'react';
2 import logo from './logo.svg';
3 import './App.css';
4
5 function App() {
6   return (
7     <div className="App">
8       <header className="App-header">
9         <img src={logo} className="App-logo" alt="logo" />
10        <p>
11          Sujeito Programador!
12        </p>
13        <a
14          className="App-link"
15          href="https://reactjs.org"
16          target="_blank"
17          rel="noopener noreferrer"
18        >
19          Learn React
20        </a>
21      </header>
22    </div>
23  );
24 }
25
26 export default App;
```

O React possui o sistema **live reload** ou seja após salvar o arquivo que editou você pode ir no navegador e já ver todas alterações feitas em tempo real.

Vamos ver então como ficou nosso projeto.





Sujeito Programador!

[Learn React](#)

O arquivo `app.js` é o arquivo que é chamado quando nossa aplicação é iniciada ou seja, o primeiro componente que é montado em tela é o **App.js**

Vamos entender a estrutura de um componente e como montamos um componente.

Veja um exemplo na proxima pagina:

[CLIQUE AQUI PARA CONHECER À FABRICA DE APLICATIVOS](#)

```

c ▸ JS App.js ▸ App
1  import React from 'react';
2
3  function App() {
4      return (
5          <div>
6              <h1>Hello World</h1>
7          </div>
8      );
9  }
10
11  export default App;
12

```

Na primeira linha temos o **import do React**, precisamos importar o react para conseguir escrever nosso código **JSX** que é este código que fica dentro do **return ( );** que se parece muito com HTML5.

Então temos nosso componente App.

```

function App(){
  return(
    <div>
      <h1>Aprendendo React JS</h1>
    </div>
  );
}

```

E por ultimo temos `export default App;` exportamos nosso componente para que o arquivo `index.js` possa chamar ele quando iniciar nosso projeto.

Podemos criar também componentes totalmente nossos, vamos lá voltar ao código.

Embaixo do `import React from 'react';` podemos criar um novo componente:

```
function Nome(){  
  return(  
    <h2>Sujeito Programador</h2>  
  );  
}
```

- Então todo nosso código ficou assim:

```
1  import React from 'react';  
2  
3  function Nome(){  
4    return(  
5      <h2>Sujeito Programador</h2>  
6    );  
7  }  
8  
9  function App(){  
10    return(  
11      <div>  
12        <h1>Aprendendo React JS</h1>  
13      </div>  
14    );  
15  }  
16  
17  export default App;
```

Se você ainda estiver com a sua aplicação quando salvar o arquivo vai ver que ainda não mudou nada na aplicação, isso por que apenas criamos um **componente Nome** que tem mostra um texto mas não chamamos ele dentro do nosso componente principal que é o **App**, vamos fazer isso então:

Então se o componente App é o principal então é dentro dele que precisamos colocar o componente Nome:

```
import React from 'react';

function Nome(){
  return(
    <h2>Sujeito Programador</h2>
  );
}

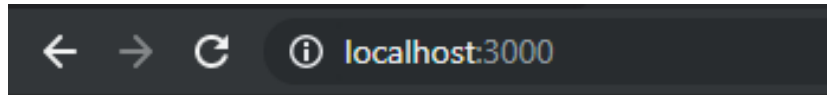
function App(){
  return(
    <div>
      <h1>Aprendendo React JS</h1>
      <Nome />
    </div>
  );
}

export default App;
```

Podemos chamar ele como se fosse chamar uma **Tag HTML** como no exemplo acima, chamamos ele abaixo do **h1**.

Agora podemos salvar o arquivo e vamos ao navegador ver as alterações.

A saída deve ser esta:



# Aprendendo React JS

## Sujeito Programador

Aqui você já deve ter começado a sentir o poder e as infinitas possibilidades que conseguimos criar.

**Show de bola!** Já temos nosso primeiro projeto criado em React e podemos usar todo poder dessa biblioteca incrível.

**Conheça a documentação do reactjs**

[Clique aqui para acessar.](#)

**Documentação em português**

[Clique aqui para acessar.](#)



[CLIQUE AQUI PARA CONHECER À FABRICA DE APLICATIVOS](#)

# React Native

Mas o que é esse **React Native**?

React Native é uma biblioteca criada também pelo Facebook para desenvolvimento de **aplicativos** nativos para **Android** e **iOS** utilizando JavaScript.

E com React Native conseguimos utilizar todo poder do JavaScript e também todo poder do React, por que dentro do React Native nós também usamos a biblioteca React, sim essa mesma que você aprendeu nas páginas acima. Ou seja conseguimos reutilizar muito do nosso código e também do aprendizado para desenvolver sistemas e aplicações web quanto desenvolver aplicativos para Android e iOS.

Aqui vai alguns aplicativos que você deve conhecer que são feitos em React Native:



Conseguimos reaproveitar todo conhecimento em ReactJS e aplicar para desenvolver aplicativos isso é **incrível!**

**Conheça site do React Native:**

[Clique aqui para acessar.](#)

[CLIQUE AQUI PARA CONHECER A FABRICA DE APLICATIVOS](#)

# Conclusão

O Mercado para **React JS** e **React Native** está super aquecido no brasil e muito bem pago alem dos gigantes da tecnologia que aplicam em seus produtos/serviços.

Então se você quer ser um desenvolvedor web reconhecido, onde tem vagas no mercado que pagam bem a **Stack React** é o caminho, você vai encontrar facilmente vagas que pagam desde **R\$ 3mil** até **R\$ 12mil** pra mais.

Mas fica tranquilo que **não acabou** preparamos algo **exclusivo pra você** que chegou até aqui.

Para você se tornar um desenvolvedor profissional criar aplicações completas do zero, desenvolver aplicativos, sistemas, tudo isso usando React e mudar sua carreira de uma vez por todas.

**SIM! QUERO SER UM(A) DESENVOLVEDOR(A) PROFISSIONAL!**





Chegamos ao fim dessa jornada incrível de muito aprendizado.

## Acompanhe nossas redes sociais:



**SujeitoProgramador**



**SujeitoProgramador**



**SujeitoProgramador**

