

# Tutorial React

Kid Mendes

Rodrigo Nascimento

Rodrigo Souza

# Instalação

- Nodejs - Windows/Mac

node -v



Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Download for Windows (x64)

**10.15.3 LTS**

Recommended For Most Users

**11.14.0 Current**

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.

# Instalação

- Nodejs - Linux

cd Downloads

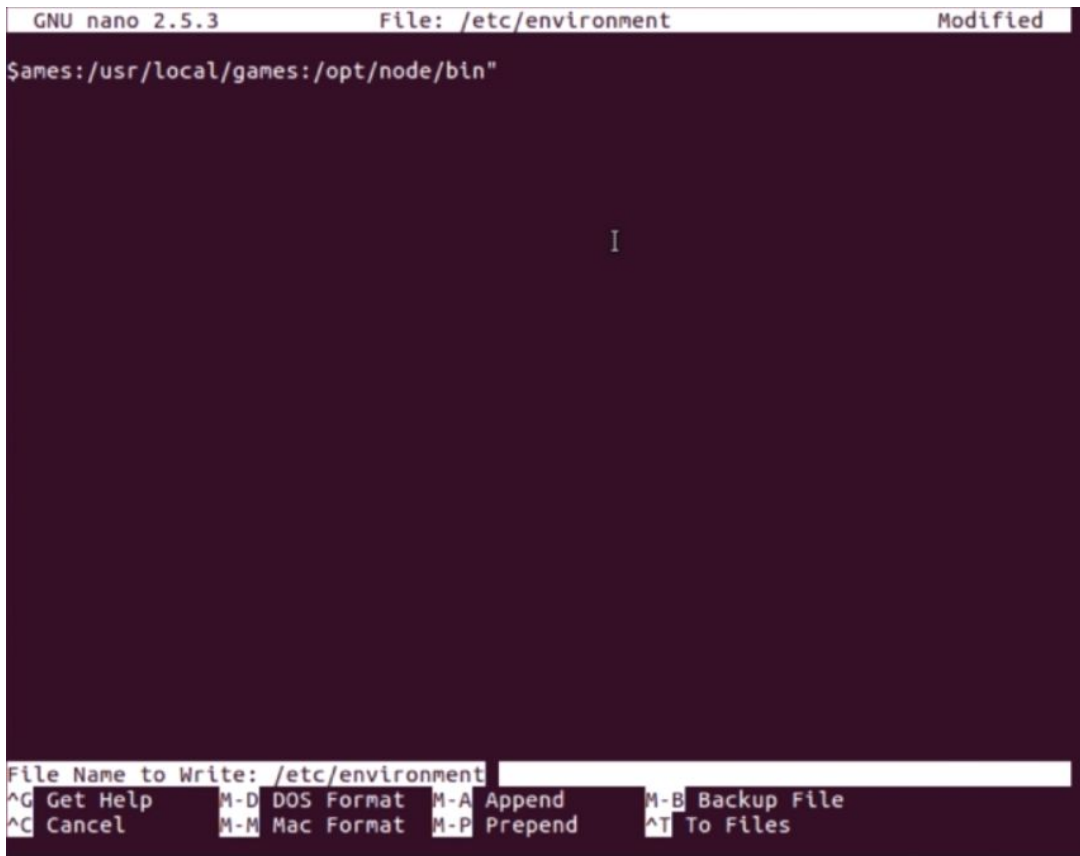
sudo mv node /opt/node

cd /opt/node

sudo nano /etc/environment

source /etc/environment

node -v



```
GNU nano 2.5.3      File: /etc/environment      Modified
$ames:/usr/local/games:/opt/node/bin"

I

File Name to Write: /etc/environment
^G Get Help      M-D DOS Format  M-A Append      M-B Backup File
^C Cancel        M-M Mac Format  M-P Prepend     ^T To Files
```

# Instalação

- Nodejs - Linux

## Installation instructions

### Node.js v11.x:

```
# Using Ubuntu
curl -sL https://deb.nodesource.com/setup_11.x | sudo -E bash -
sudo apt-get install -y nodejs

# Using Debian, as root
curl -sL https://deb.nodesource.com/setup_11.x | bash -
apt-get install -y nodejs
```

### Node.js v10.x:

```
# Using Ubuntu
curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
sudo apt-get install -y nodejs

# Using Debian, as root
curl -sL https://deb.nodesource.com/setup_10.x | bash -
apt-get install -y nodejs
```

# Instalação

- React

```
npx create-react-app myApp
```

```
cd myApp
```

```
npm start
```



# Instalação

- Gerenciamento de dependências

npm i --save express mongoose

npm i --save-dev mocha chai

npm i

```
package.json x
1  {
2    "name": "myapp",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "react": "^16.8.6",
7      "react-dom": "^16.8.6",
8      "react-scripts": "2.1.8"
9    },
10   "scripts": {
11     "start": "react-scripts start",
12     "build": "react-scripts build",
13     "test": "react-scripts test",
14     "eject": "react-scripts eject"
15   },
16   "eslintConfig": {
17     "extends": "react-app"
18   },
19   "browserslist": [
20     ">0.2%",
21     "not dead",
22     "not ie <= 11",
23     "not op_mini all"
24   ]
25 }
26
```

# Como funciona?

- O que é React?

**React é uma biblioteca JavaScript** para a criação de interface de usuários criada pelo Facebook. Atualmente o **React** é amplamente utilizado para a **web** e para **aplicações mobiles** com o **React Native**.



# Como funciona?

- Framework React

**React é uma biblioteca** que permite a componentização de interfaces de usuário e unidades menores, porém **React** não trabalha sozinho. No contexto da **web** **React** trabalha com uma biblioteca chamada **ReactDOM** e a junção dessas duas bibliotecas caracteriza um **framework** chamado **React**.






# Como funciona?

- Como React funciona?

**React** provê uma sintaxe semelhante ao html chamada **JSX** que permite programar de forma declarativa como cada **view** deve se apresentar ao usuário de acordo com os dados passados à **view**, sem se preocupar em como o **DOM** irá mudar para se adaptar à mudança desses dados.

Em **React**, sempre que os dados em uma **view** são alterados ele sabe por meio de um **DOM** virtual quais elementos do **DOM** devem ser alterados sem precisar atualizar a **view** inteira, garantindo maior performance à aplicação.



# Como funciona?

- Exemplo

LIVE JSX EDITOR	JSX?	RESULT
<pre>class HelloMessage extends React.Component {   render() {     return (       &lt;div&gt;         Hello {this.props.name}       &lt;/div&gt;     );   } }</pre> <pre>ReactDOM.render(   &lt;HelloMessage name="Rodrigo" /&gt;,   document.getElementById('hello-example') );</pre>		Hello Rodrigo

# Projeto React

1. Comando para criar um projeto React

```
npx create-react-app react-tutorial
```

2. Comando para executar projeto React

```
cd react-tutorial  
npm start
```

# Projeto React

- Divisão de arquivos



# Projeto React

- index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import * as serviceWorker from './serviceWorker';

ReactDOM.render(<App />, document.getElementById('root'));

// If you want your app to work offline and load faster, you can change
// unregister() to register() below. Note this comes with some pitfalls.
// Learn more about service workers: https://bit.ly/CRA-PWA
serviceWorker.unregister();
```

# Components

- Simple Components
  - Apresentar dados estáticos;
  - Não é possível manipular a busca de dados;
- Class Components
  - Utilizado para fontes de dados dinâmicos;
  - Lida com qualquer dado que possa mudar;



# Class Components

- Table.js

```
import React, { Component } from 'react'

class Table extends Component{
  render(){
    return(
      <table>
        <thead>
          <tr>
            <th>Name</th>
            <th>Job</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>Charlie</td>
            <td>Janitor</td>
          </tr>
          <tr>
            <td>Mac</td>
            <td>Bouncer</td>
          </tr>
          <tr>
            <td>Deek</td>
            <td>Aspiring actress</td>
          </tr>
          <tr>
            <td>Dennis</td>
            <td>Bartender</td>
          </tr>
        </tbody>
      </table>
    )
  }
}

export default Table;
```

# Simple Components

- Table.js

```
class Table extends Component{  
  render(){  
    return(  
      <table>  
        <TableHeader />  
        <TableBody />  
      </table>  
    )  
  }  
}  
  
export default Table;
```

```
import React, { Component } from 'react'  
  
const TableHeader = () =>{  
  return(  
    <thead>  
      <tr>  
        <th>Name</th>  
        <th>Job</th>  
      </tr>  
    </thead>  
  )  
}  
  
const TableBody = () =>{  
  return(  
    <tbody>  
      <tr>  
        <td>Charlie</td>  
        <td>Janitor</td>  
      </tr>  
      <tr>  
        <td>Mac</td>  
        <td>Bouncer</td>  
      </tr>  
      <tr>  
        <td>Dee</td>  
        <td>Aspiring actress</td>  
      </tr>  
      <tr>  
        <td>Dennis</td>  
        <td>Bartender</td>  
      </tr>  
    </tbody>  
  )  
}
```



# Simple Components

- Table.js

```
class Table extends Component{  
  render(){  
    return(  
      <table>  
        <TableHeader />  
        <TableBody />  
      </table>  
    )  
  }  
}  
  
export default Table;
```

```
import React, { Component } from 'react'  
  
const TableHeader = () =>{  
  return(  
    <thead>  
      <tr>  
        <th>Name</th>  
        <th>Job</th>  
      </tr>  
    </thead>  
  )  
}  
  
const TableBody = () =>{  
  return(  
    <tbody>  
      <tr>  
        <td>Charlie</td>  
        <td>Janitor</td>  
      </tr>  
      <tr>  
        <td>Mac</td>  
        <td>Bouncer</td>  
      </tr>  
      <tr>  
        <td>Dee</td>  
        <td>Aspiring actress</td>  
      </tr>  
      <tr>  
        <td>Dennis</td>  
        <td>Bartender</td>  
      </tr>  
    </tbody>  
  )  
}
```

# Props

- São uma forma eficaz de passar dados existentes para um componente React;
- Somente para leitura, os dados não podem ser alterados.

```
import React, { Component } from 'react'

const TableHeader = () =>{
  return(
    <thead>
      <tr>
        <th>Name</th>
        <th>Job</th>
      </tr>
    </thead>
  )
}

const TableBody = props =>{
  const rows = props.characterData.map((row, index) =>{
    return(
      <tr key={index}>
        <td>{row.name}</td>
        <td>{row.job}</td>
      </tr>
    )
  })
  return <tbody>{rows}</tbody>
}

class Table extends Component{
  render(){
    const { characterData } = this.props;

    return(
      <table>
        <TableHeader />
        <TableBody characterData={characterData} />
      </table>
    )
  }
}

export default Table;
```

# State

- Qualquer dado que deve ser salvo e modificado sem necessariamente ser adicionado a um banco de dados;
- Ex: Itens de um carrinho de compra.
- Form.js

```
import React, { Component } from 'react'

class Form extends Component{
  constructor(props) {
    super(props)

    this.initialState = {
      name: '',
      job: '',
    }

    this.state = this.initialState
  }

  handleChange = event =>{
    const { name, value } = event.target

    this.setState({
      [name]: value,
    })
  }

  submitForm = () => {
    this.props.handleSubmit(this.state)
    this.setState(this.initialState)
  }
}
```

# State

- Qualquer dado que deve ser salvo e modificado sem necessariamente ser adicionado a um banco de dados;
- Ex: Itens de um carrinho de compra.
- Form.js

```
submitForm = () => {  
  this.props.handleSubmit(this.state)  
  this.setState(this.initialState)  
}  
  
render() {  
  const { name, job } = this.state;  
  
  return (  
    <form>  
      <label>Name</label>  
      <input type="text" name="name" value={name} onChange={this.handleChange} />  
      <label>Job</label>  
      <input type="text" name="job" value={job} onChange={this.handleChange} />  
      <input type="button" value="Submit" onClick={this.submitForm} />  
    </form>  
  );  
}  
}  
  
export default Form;
```

# Projeto React

<

>

↺

⌵

🌐

localhost:3000

Name	Job	Remove
João	Advogado	Delete
Maria	Médica	Delete

Name

Job

Submit

# Fonte

- Rascia, Tania. “Getting Started with React - An Overview and Walkthrough”. Disponível em: <https://www.taniarascia.com/getting-started-with-react/>
- <https://reactjs.org>
- “Guia Completo de React: Primeiros Passos com React”. <https://www.devmedia.com.br/react/>



The background is a solid pink color. In the top right corner, there is a decorative pattern of overlapping geometric shapes, including triangles and squares, in various shades of pink and magenta.

Obrigado!