

## Notas ReactJS

Site: <https://reactjs.org/>

Componentes: <https://reactstrap.github.io/>

### No VS Code

Em qualquer diretório

yarn global add create-react-app

Do diretório que você quer criar a pasta

yarn create react-app <nome do projeto>

Todo componente obrigatoriamente deve ter um método render.

O método render retorna um JSX, ou seja, um html com js

O React tem 2 formas de criar componente, são elas:

#### 1. Componente Stateless:

```
1  import React from 'react';
2
3  // Component stateless
4  const Header = () => (
5    <h1>Header</h1>
6  )
7
8  // Component stateless
9  const Header = () => {
10    return (<h1>Header</h1>)
11  }
12
13  // Component stateless
14  const Header = function () {
15    <h1>Header</h1>
16  }
17
18  // Component stateless
19  const Header = function () {
20    return (<h1>Header</h1>)
21  }
```

Acima temos 4 formas de desenvolver um componente stateless. Componente Stateless significa que o componente não precisa trabalhar com o conceito de estado (state). Assim temos um código mais simples e menos verboso. Além disso, é possível desenvolver esse componente utilizando arrow functions => ou functions padrão, veja acima.

O mais comum é trabalharmos assim:

```
1  import React from 'react';
2
3  // Component stateless
4  const Header = () => (
5    <h1>Header</h1>
6  )
```

Perceba que esse método não tem o return. Por que isso? Quando o método tem apenas uma linha de retorno, podemos criá-lo sem o return, apenas colocando o que ele vai retornar.

## 2. Componente com State

Quando o componente necessitar trabalhar com State, fazer dessa forma:

```
1  import React from 'react';
2
3  // Componente que trabalhará com State
4  class Header extends Component {
5    render() {
6      return <h1>Header Completo</h1>
7    }
8  }
```

## Métodos de ciclo de vida do React

O React tem métodos de ciclo de vida. São métodos que são chamados na montagem do componente na tela. São eles:

### 1. componentDidMount()

Esse método é executado assim que o componente for renderizado (montado) em tela. Sempre que você precisar executar uma ação assim que o componente for mostrado em tela, utilize esse método.

## **Método Render**

Sempre que tivermos uma variável no state e essa variável for alterada o método render será executado novamente. Pois, o método render sempre vai ficar escutando o state.

## **Resumo dos princípios do React JS**

- Componentes (Component)
- Roteamento com React-Router
- Métodos de ciclos de vida
- Utilização de APIs
- Estados (State)
- Propriedades (Props)
- Estilização em CSS

CRUD completo em React JS:

<https://www.tutsmake.com/first-crud-node-express-js-mysql-example/>