

# Introdução ao React + Redux

## Kura Web

<https://www.youtube.com/watch?v=CAURWhm--7Y&t=3s>

Resumo feito por Roberto Pinheiro

- Instale o Node

Para verificar se tudo foi instalado corretamente e confirmar qual foi a versão instalada, dentro da pasta git entre com o comando:

**node -v**

```
λ node -v
v10.15.3
```

- Acesse a URL: <https://github.com/facebook/create-react-app>

Entre com o comando: **npm install -g create-react-app**

```
λ npm install -g create-react-app
C:\Users\Beto\AppData\Roaming\npm\create-react-app -> C:\Users\Beto\AppData\Roaming\npm\node_modules\create-react-app\index.js
+ create-react-app@3.0.1
added 91 packages from 45 contributors in 32.164s

λ npx create-react-app react-project

Creating a new React app in c:\arquivos2\git\react-project.

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

> core-js@2.6.9 postinstall c:\arquivos2\git\react-project\node_modules\babel-runtime\node_modules\core-js
> node scripts/postinstall || echo "ignore"

> core-js-pure@3.1.4 postinstall c:\arquivos2\git\react-project\node_modules\core-js-pure
> node scripts/postinstall || echo "ignore"

+ react-scripts@3.0.1
+ react@16.8.6
+ react-dom@16.8.6
added 1386 packages from 675 contributors and audited 902127 packages in 354.007s
found 0 vulnerabilities

Success! Created react-project at c:\arquivos2\git\react-project
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 react-project

  npm start

Happy hacking!

c:\arquivos2\git (master -> origin)
λ |
```

Para carregar o servidor, entre com:

**npm start**

```
c:\arquivos2\git\react-project (master -> origin)
λ npm start

> react-project@0.1.0 start c:\arquivos2\git\react-project
> react-scripts start

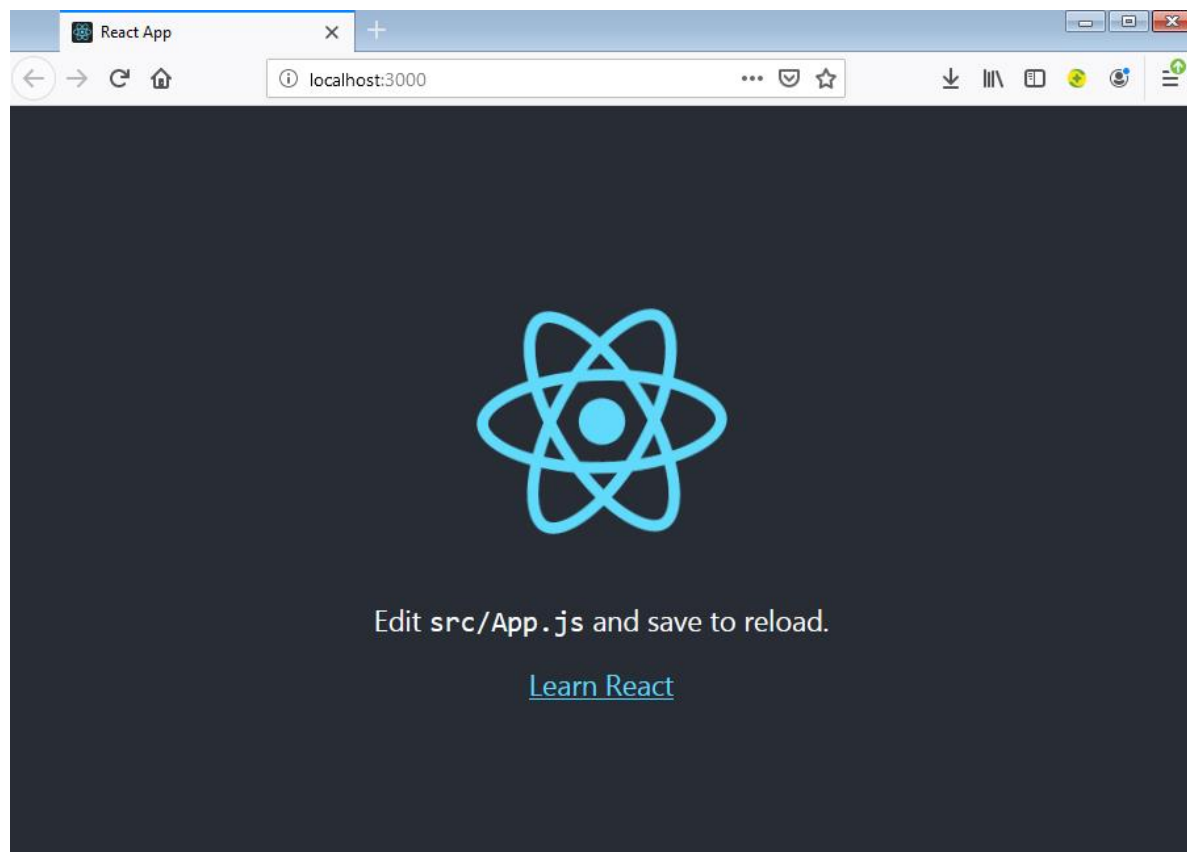
Compiled successfully!

You can now view react-project in the browser.

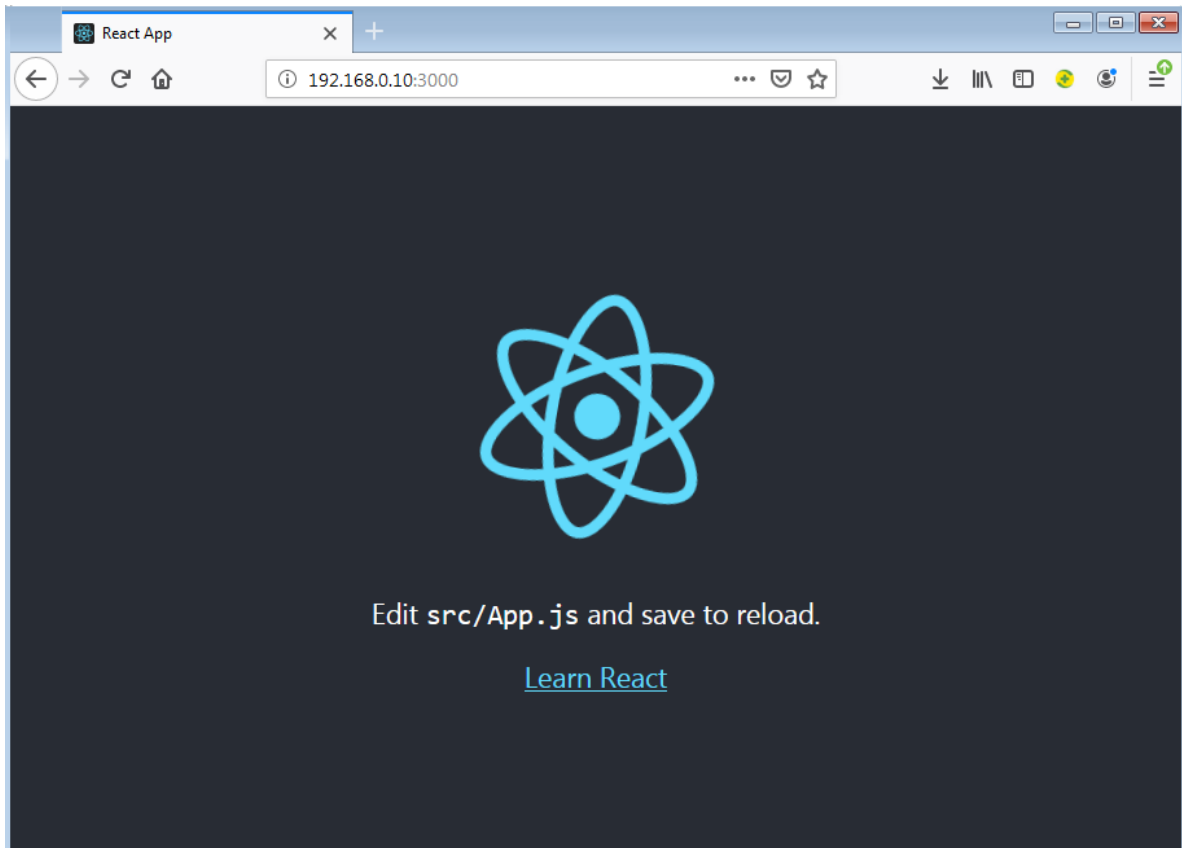
  Local:            http://localhost:3000/
  On Your Network:  http://192.168.0.10:3000/

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

|
```



**Acessando de qualquer dispositivo**



# Componentes

Um componente serve para renderizar HTML no browser.

## index.js

```
import React, { Component } from 'react';
import ReactDOM from 'react-dom';

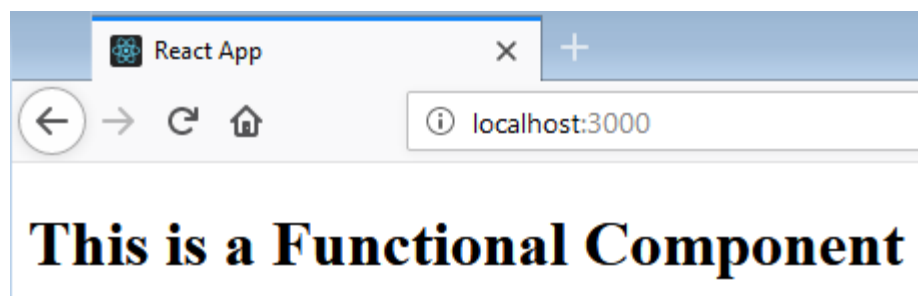
// Functional Components

const FunctionalComponent = () => (
  <h1>This is a Functional Component</h1>
)

// Class Components

class ClassComponent extends Component{
  state = {
    text: 'CLASS COMPONENT'
  }
  render(){
    return (
      <h1>This is a {this.state.text}</h1>
    )
  }
}

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



Não é HTML, é JSX

## index.js

```
import React, { Component } from 'react';
import ReactDOM from 'react-dom';

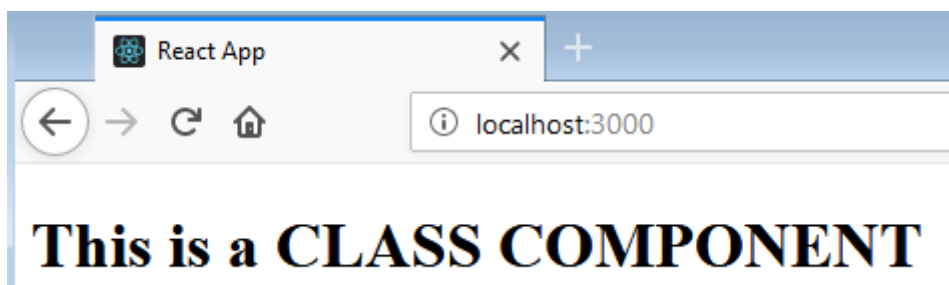
// Functional Components

const FunctionalComponent = () => (
  <h1>This is a Functional Component</h1>
)

// Class Components

class ClassComponent extends Component{
  state = {
    text: 'CLASS COMPONENT'
  }
  render(){
    return (
      <h1>This is a {this.state.text}</h1>
    )
  }
}

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



# Babel

## Como o react compila as informações para enviar para a aplicação

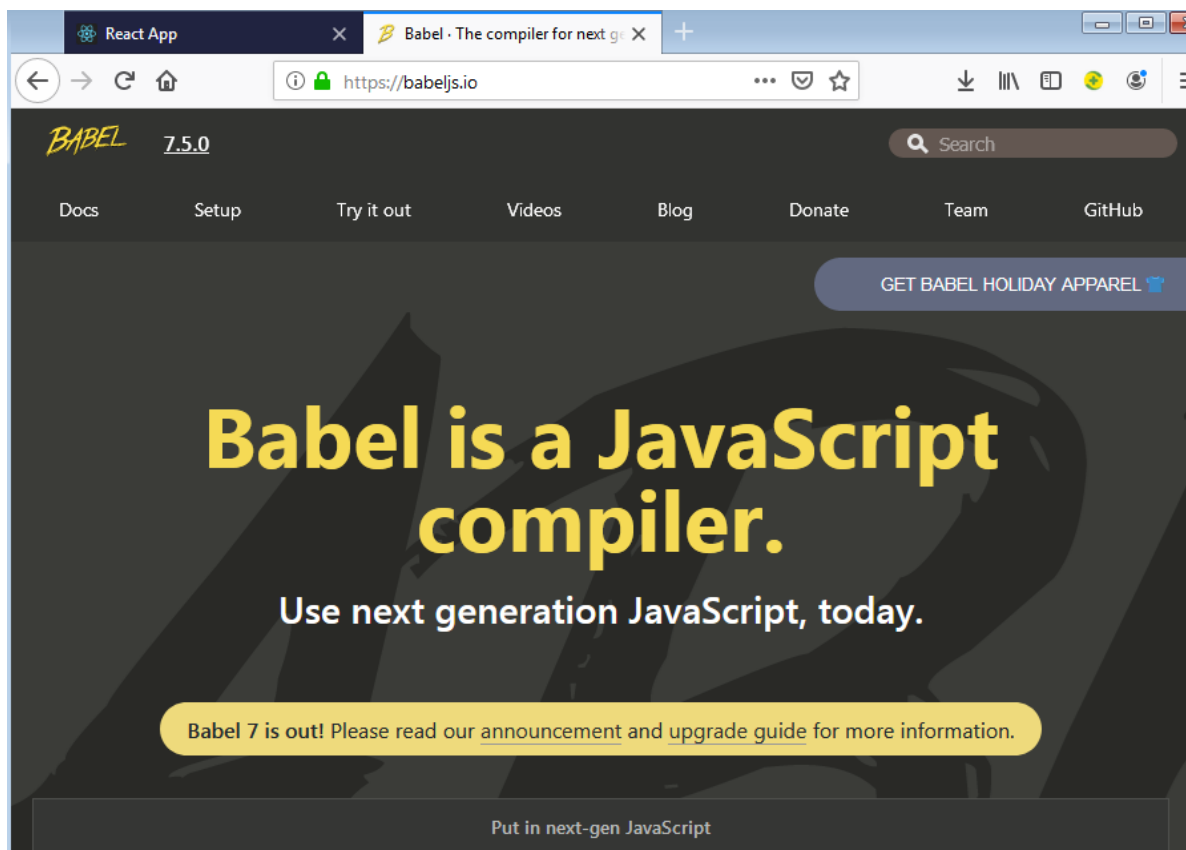
Estamos utilizando o javascript s6 ou ECMAScript 6.

Não é muito comum o suporte nativo ao s6 nos browsers mais antigos.

Babel traduz o javascript novo (s6) para o mais antigo (s5).

Acesse a URL:

<https://babeljs.io/>



```
1 const FunctionalComponent = () => {  
2   return <h1>This is a Functional Component</h1>  
3 }
```

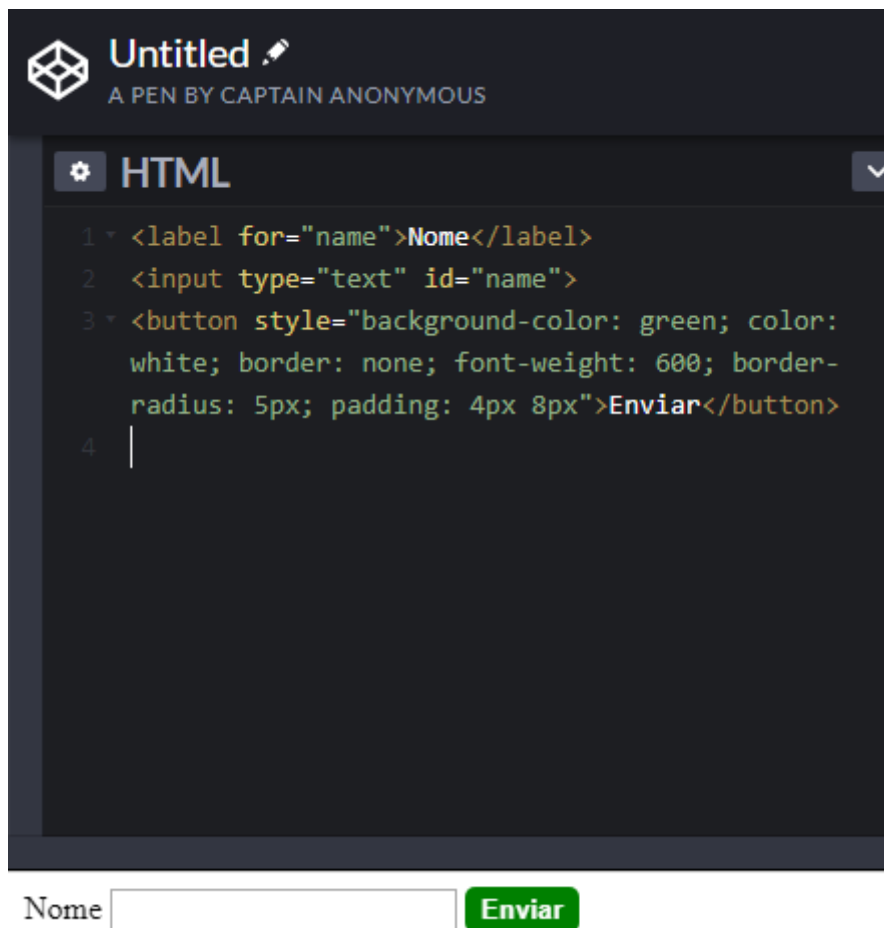
```
1 "use strict";  
2  
3 var FunctionalComponent = function FunctionalComponent() {  
4   return React.createElement("h1", null, "This is a Functional  
   Component");  
5 };
```

Portanto sempre baixe a dependência do Babel no seu projeto para evitar problemas de compatibilidade.

# Criando um formulário em HTML

- Acesse a URL:

<https://codepen.io/>



```
<label for="name">Nome</label>
<input type="text" id="name">
<button style="background-color: green; color: white; border: none; font-
weight: 600; border-radius: 5px; padding: 4px 8px">Enviar</button>
```

## index.js

```
import React, { Component } from 'react';
import ReactDOM from 'react-dom';

// Functional Components

const FunctionalComponent = () => (
  <div>
    <label for="name">Nome</label>
    <input type="text" id="name">
    <button style="background-color: green; color: white;
border: none; font-weight: 600; border-radius: 5px; padding: 4px
8px">Enviar</button>
  </div>
)

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

### Failed to compile

```
./src/index.js
Line 11: Parsing error: Expected corresponding JSX closing tag for <input>

   9 |         <input type="text" id="name">
  10 |         <button style="background-color: green; color: white; border: none; font-weight: 600; border-radius: 5px; padding: 4px 8px">Enviar</button>
> 11 |     </div>
      |     ^
  12 |   )
  13 |
  14 | ReactDOM.render(<FunctionalComponent />, document.getElementById('root'));
```

This error occurred during the build time and cannot be dismissed.



# Convertendo HTML para JSX

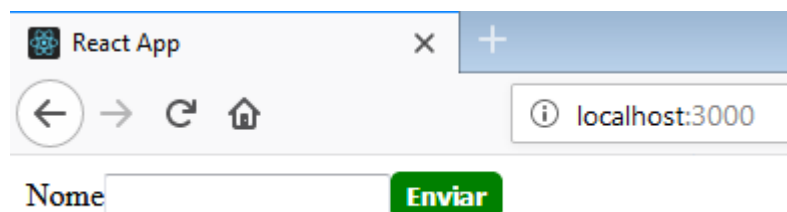
## index.js

```
import React, { Component } from 'react';
import ReactDOM from 'react-dom';

// Functional Components

const FunctionalComponent = () => (
  <div>
    <label for="name" className="texto">Nome</label>
    <input type="text" id="name"/>
    <button style={{backgroundColor: 'green', color: 'white',
border: 'none', fontWeight: '600', borderRadius: '5px', padding:
'4px 8px'}}>Enviar</button>
  </div>
)

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



```
indexjs x index.html X
1 import React, { Component } from 'react'
2 import ReactDOM from 'react-dom'
3
4 const hintText = () => (
5   'Digite o seu nome'
6 )
7
8 //FUNCTIONAL COMPONENTS
9 const FunctionalComponent = () => {
10   const textButton = 'Send'
11   const style = {backgroundColor: 'red', color: 'white', border: 'none', fontWeight: '600', borderRadius: '5px', padding: '5px'}
12
13   return(
14     <div>
15       <label for="name" className="texto">Nome: </label>
16       <input type="text" id="name" placeholder={hintText()} />
17       <button style={style}>{textButton}</button>
18     </div>
19   )
20 }
21
22 ReactDOM.render(<FunctionalComponent />, document.querySelector('#root'))
```

# Referenciando variáveis no JSX

index.js

```
import React, { Component } from 'react';
import ReactDOM from 'react-dom';

const hintText = () => (
  'Digite o seu nome'
)

// Functional Components

const style = {backgroundColor: 'green', color: 'white', border:
'none', fontWeight: '600', borderRadius: '5px', padding: '4px 8px'}

const FunctionalComponent = () => (
  <div>
    <label for="name" className="texto">Nome</label>
    <input type="text" id="name" placeholder={hintText()} />
    <button style={style}>Enviar</button>
  </div>
)

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

