

REACT



Bruno Junqueira

CTO na Roverpix
Consultor de Tecnologia na Omnize
Empreendedor

<https://github.com/brunobra>

<https://www.linkedin.com/in/bjunqueira/>

Introdução

Assunto

React - Uma biblioteca de JavaScript para construir interfaces de usuários

Objetivo

Aprender os conceitos e aplicá-los no desenvolvimento de aplicações Web

Temas

- Introdução
- Faça você mesmo
- E mais...

Introdução

Como surgiu



XHP

O que é

React é uma biblioteca de JavaScript, declarativa, eficiente e flexível, para construir interfaces de usuários. Permite você construir UI's complexas a partir de pedaços de códigos pequenos e isolado chamados "componentes".

Benefícios

- Mantido pelo Facebook
- Fácil de aprender, fácil de usar
- Componentes reutilizáveis
- Performance
- Flexibilidade
- Ferramentas
- Bibliotecas
- Comunidade
- SPA (Single page application)

Quem usa



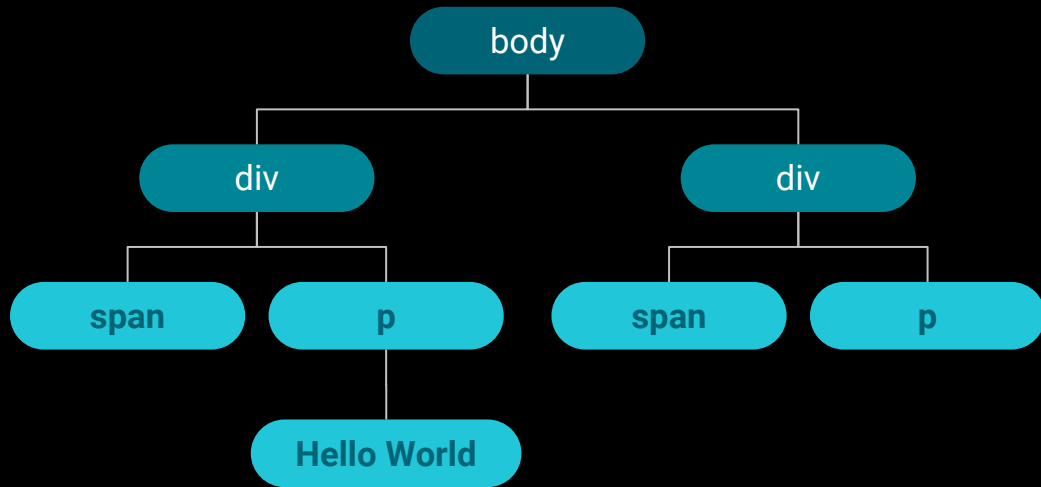
Uber



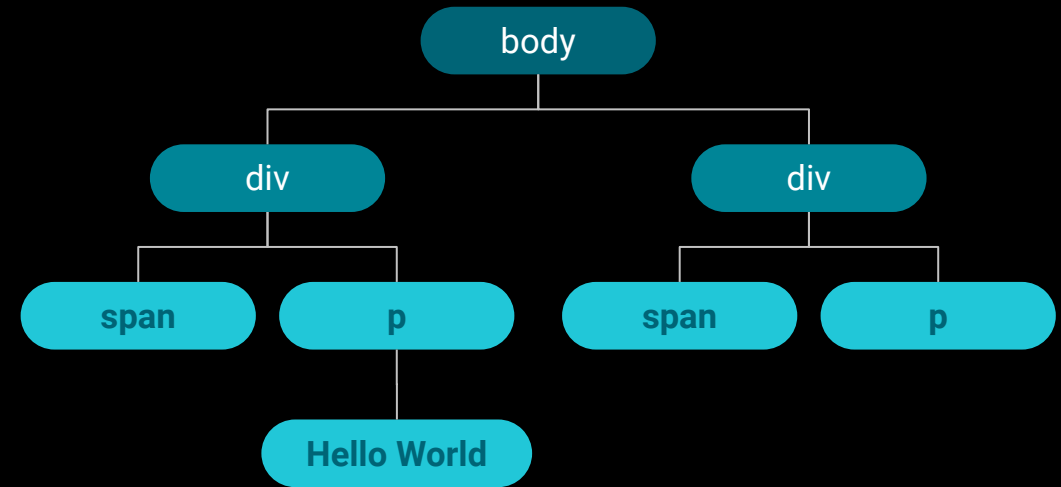
...

Como funciona *Virtual DOM*

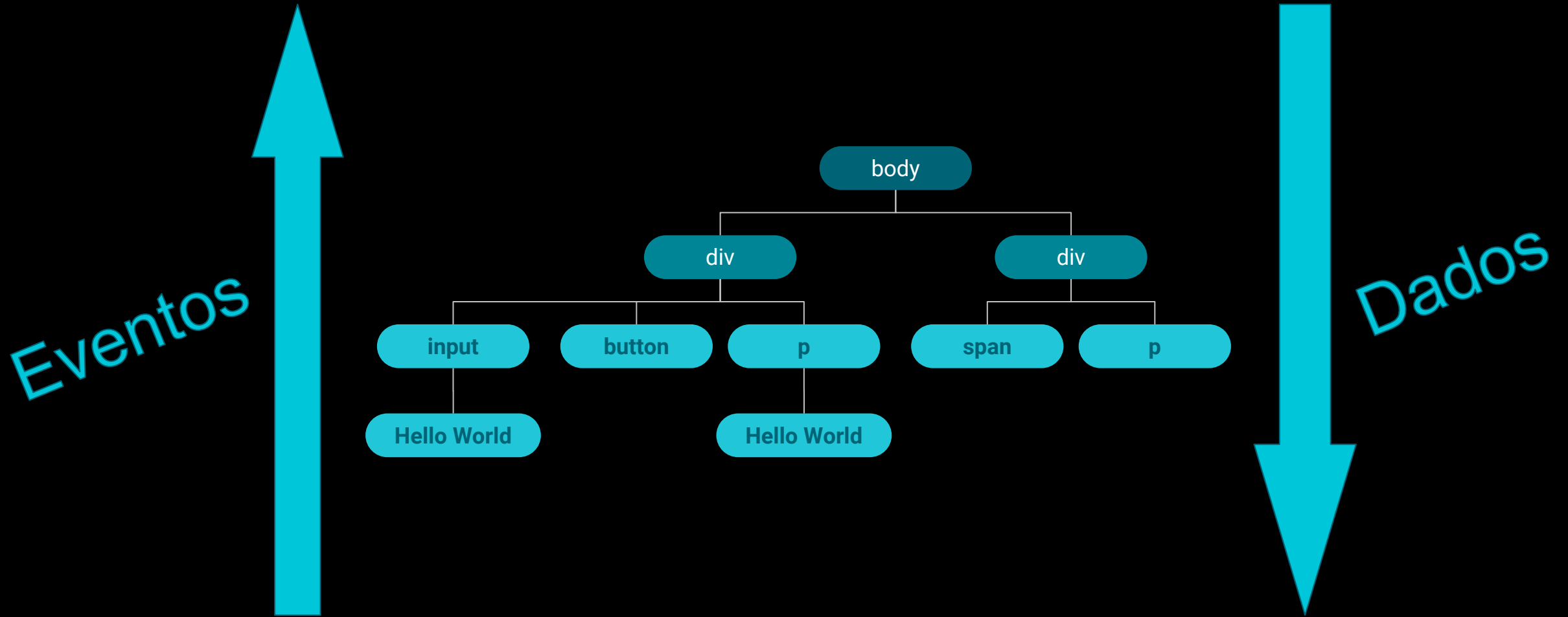
Virtual DOM



DOM



Como funciona *One-way Data flow*



Como funciona Componentes (JSX)

```
class HelloMessage extends React.Component {  
  render() {  
    return (  
      <p>  
        {this.props.name}  
      </p>  
    );  
  }  
}
```

```
<HelloMessage name="Bruno"/>
```

Como começar *Método 1*

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

  <script src="app.js"></script>
</body>
</html>
```

Como começar? *Método 2*

Instalar e criar projeto básico:

npm < 5.2

```
$ npm install -g create-react-app
```

```
$ create-react-app my-app
```

npm >= 5.2

```
$ npx create-react-app my-app
```

Executar projeto:

```
$ cd my-app
```

```
$ npm start
```



Welcome to React

To get started, edit `src/App.js` and save to reload.

Antes de começar



webpack

src/index.js

```
import bar from './bar';  
  
bar();
```

src/bar.js

```
export default function bar() {  
  //  
}
```

<https://webpack.js.org/>

Antes de começar



webpack

webpack.config.js

```
const path = require('path');

module.exports = {
  entry: './src/index.js',
  output: {
    path: path.resolve(__dirname, 'dist'),
    filename: 'bundle.js'
  }
};
```

page.html

```
<!doctype html>
<html>
  <head>
    ...
  </head>
  <body>
    ...
    <script src="dist/bundle.js"></script>
  </body>
</html>
```

<https://webpack.js.org/>

Antes de começar

```
var sum = function (a, b) {  
  return a + b;  
};
```

```
const sum = (a, b) => a + b;
```



Antes de começar

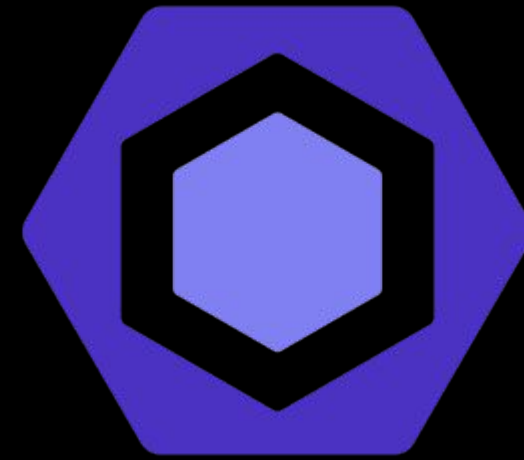
```
class HelloWorld extends React.Component {  
  render() {  
    return (  
      <p>  
        Hello World  
      </p>  
    );  
  }  
}
```



```
const HelloWorld = React.createElement(  
  'p',  
  {},  
  'Hello World'  
);
```

<https://babeljs.io>

Antes de começar



ESLint

```
var foo = bar;
```

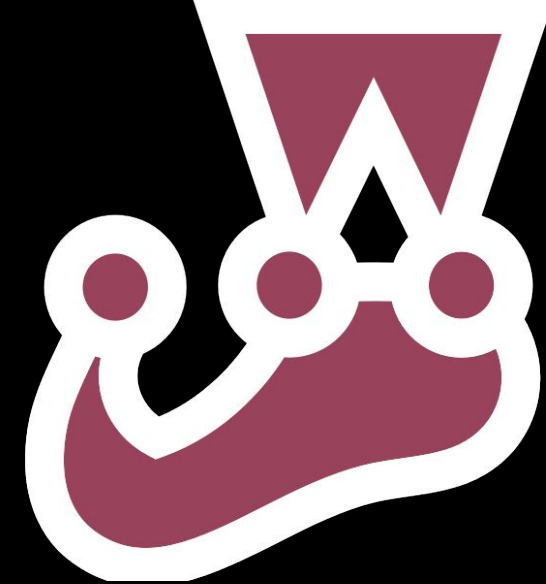
1:5 - 'foo' is assigned a value but never used. (no-unused-vars)

1:11 - 'bar' is not defined. (no-undef)

<https://eslint.org/>

Antes de começar

Jest



```
2. node
PASS packages/jest-matchers/src/__tests__/matchers-test.js
PASS packages/jest-editor-support/src/__tests__/parsers/TypeScriptParser-test.js
PASS packages/jest-cli/src/reporters/__tests__/VerboseReporter-test.js
PASS packages/eslint-plugin-jest/src/rules/__tests__/no-focused-tests-test.js
PASS packages/jest-matchers/src/__tests__/spyMatchers-test.js
PASS packages/jest-runtime/src/__tests__/Resolve-browser-test.js
PASS packages/jest-runtime/src/__tests__/Runtime-internalModule-test.js
PASS packages/jest-snapshot/src/__tests__/utils-test.js
PASS packages/jest-editor-support/src/__tests__/parsers/BabylonParser-test.js
PASS packages/jest-diff/src/__tests__/diff-test.js
PASS packages/jest-cli/src/lib/__tests__/setWatchMode-test.js
PASS packages/pretty-format/src/__tests__/AsymmetricMatcher-test.js

RUNS packages/jest-matcher-utils/src/__tests__/index-test.js
RUNS packages/jest-config/src/__tests__/normalize-test.js
RUNS packages/jest-runtime/src/__tests__/Runtime-genMockFromModule-test.js
RUNS packages/jest-haste-map/src/__tests__/index-test.js
RUNS packages/jest-editor-support/src/__tests__/Runner-test.js
RUNS packages/jest-matchers/src/__tests__/asymmetric-matchers-test.js
RUNS packages/jest-snapshot/src/__tests__/matcher-test.js

Test Suites: 39 passed, 39 of 76 total
Tests: 709 passed, 709 total
Snapshots: 418 passed, 418 total
Time: 14s, estimated 19s
```

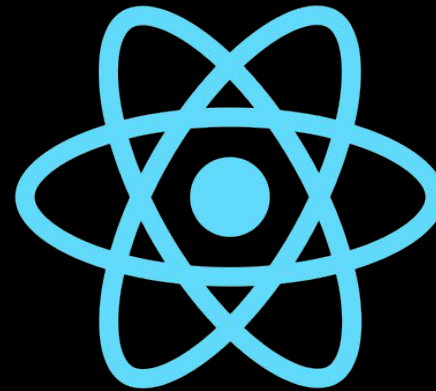
```
2. zsh
[cpojer: ~/Projects/jest/examples/getting_started (master)]$ jest --coverage
PASS ./sum.test.js
  ✓ adds 1 + 2 to equal 3 (4ms)

-----|-----|-----|-----|-----|-----|
File    | % Stmts | % Branch | % Funcs | % Lines | Uncovered Lines |
-----|-----|-----|-----|-----|-----|
All files|     100 |       100 |     100 |     100 |                  |
sum.js  |     100 |       100 |     100 |     100 |                  |
-----|-----|-----|-----|-----|-----|

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 1.133s
Ran all test suites.
[cpojer: ~/Projects/jest/examples/getting_started (master)]$
```

<https://jestjs.io/>

Antes de começar

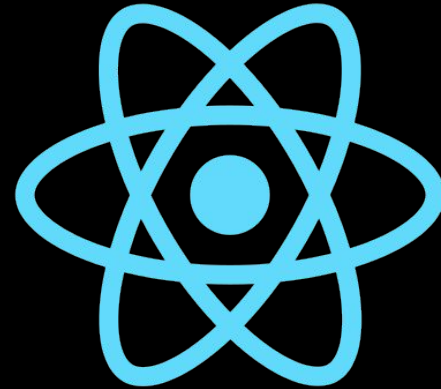


Create React App



<https://facebook.github.io/create-react-app/>

Antes de começar



Create React App

- `npm start`
- `npm test`
- `npm run build`
- `npm run eject`

<https://facebook.github.io/create-react-app/>

Faça você mesmo

Proposta de aplicação

Carrinho de compras, onde será possível:

- Exibir lista dos produtos
- Exibir valor total dos produtos
- Remover produtos
- Adicionar produtos

IDE



 ATOM

Dependências



Pasta do projeto

```
$ mkdir aula-react
```

```
$ cd aula-react
```

Criando

```
$ npm install create-react-app
```

```
$ create-react-app cart
```

ou

```
$ npx create-react-app cart
```

Executando

```
$ npm start
```

Acessando

<http://localhost:3000>

Testando

```
$ npm test
```

Estrutura de pastas e arquivos

Separar pastas de
assets e componentes.

Tipos de Componentes *Stateless*

```
const HelloWorld = () => <p>'Hello World'</p>;
```


Tipos de Componentes *Statefull*

```
class NameInput extends React.Component {  
  state = {name: 'Bruno'}  
  
  onChange (e) {  
    this.setState({  
      name: e.target.value  
    });  
  }  
  
  render() {  
    return (  
      <input onChange={this.onChange.bind(this)} />  
    );  
  }  
}
```

State

State é um objeto, contendo informações que podem ser acessadas e manipuladas pelos os componentes.

State

```
class NameInput extends React.Component {  
  state = {name: 'Bruno'}  
  
  onChange (e) {  
    this.setState({  
      name: e.target.value  
    });  
  }  
  
  render() {  
    return (  
      <input onChange={this.onChange.bind(this)} />  
    );  
  }  
}
```

Criação dos componentes

Nome: **Cart**

Propósito: **Container dos componentes do carrinho.**

Nome: **CartHeader**

Propósito: **Exibir título do carrinho e o valor total de todos os produtos.**

Nome: **Product**

Propósito: **Exibir o nome e o valor do produto.**

Props

São propriedades definidas pelos componentes, que podem receber valores ou funções.

Props

```
const HelloMessage = ({name}) => `Hello ${name}`;
```

```
<HelloMessage name="Bruno" />
```

PropTypes

Conjunto de regras para
validar o tipo de dado
recebido por uma propriedade
de um componente.

PropTypes

```
const HelloMessage = ({name}) => `Hello ${name}`;
```

```
HelloMessage.propTypes = {  
  name: PropTypes.string  
};
```

```
<HelloMessage name="Bruno" />
```


Adicionando props aos componentes

Nome: **CartHeader**

Props: **title (string), total (number)**

Nome: **Product**

Props: **name (string), price (number)**

Lista

```
const List = ({items}) => (  
  <ul>  
    {items.map((item) => (  
      <li key={item}>{item}</li>  
    ))}  
  </ul>  
);
```

```
<List items={[1,2,3]} />
```

Adicionando props aos componentes

Nome: **Cart**

Props: **products (array)**

Exemplo de lista de produtos:

```
[  
  {  
    id: 1,  
    name: 'Produto',  
    price: 20.0  
  }  
]
```

Componente Lifecycle

Cada componente contém diversos métodos de "lifecycle", que você pode sobrescrever e executar códigos em um determinado momento do processo.

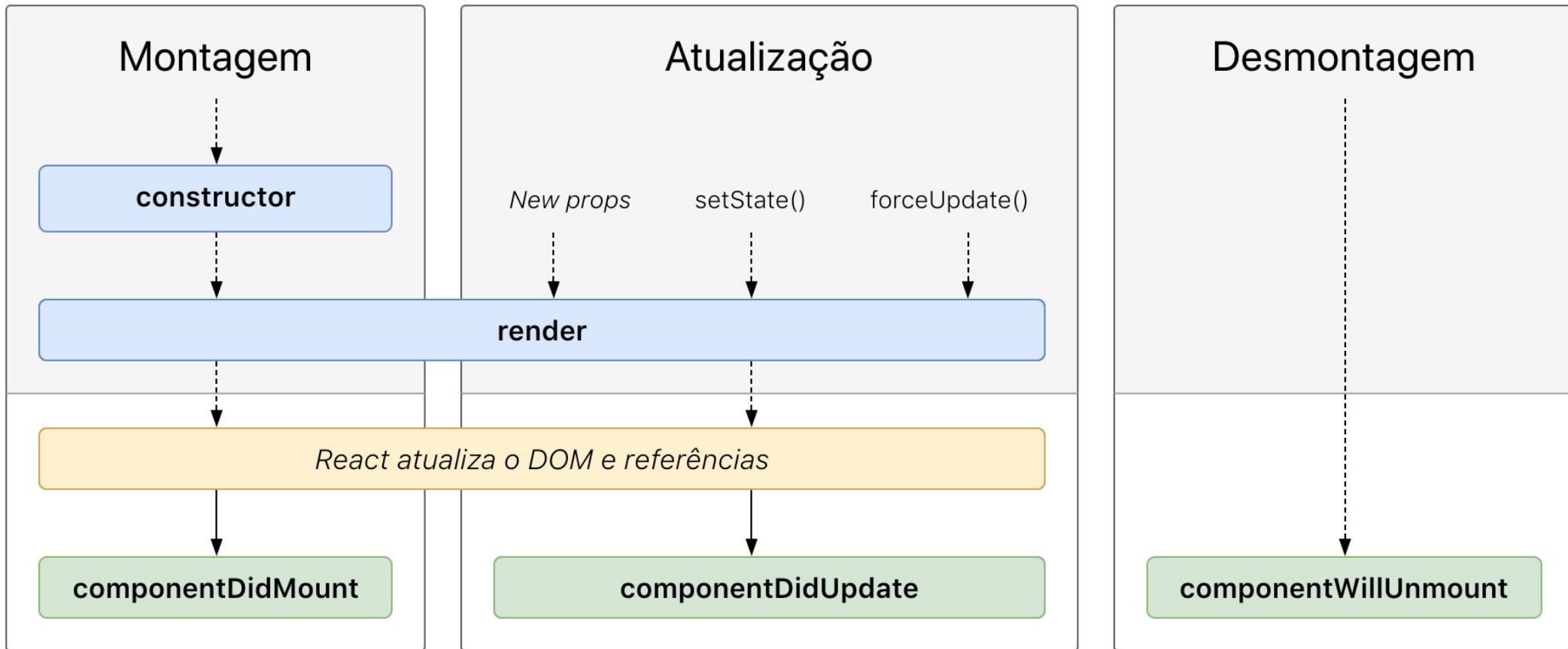
Componente Lifecycle

"Fase Render"

Pura e sem efeitos colaterais. Pode ser pausada, abortada ou reiniciada pelo React.

"Fase Commit"

Pode operar o DOM, executar efeitos colaterais, agendar atualizações.



Funções dentro de componentes

Calculando o valor total dos produtos no carrinho.

Eventos

Eventos são executados quando algo é feito pelo o usuário no navegador.
Como por exemplo um clique de um botão, ou ao pressionar uma tecla do teclado.

Eventos

```
class NameInput extends React.Component {  
  state = {name: 'Bruno'}
```

```
  onChange (e) {  
    this.setState({  
      name: e.target.value  
    });  
  }
```

```
  render() {  
    return (  
      <input ref={c => this.input = c} onChange={this.onChange.bind(this)} />  
    );  
  }  
}
```


Excluindo produtos

Adicionar nova prop no componente Product.

Nome: **Product**

Props: **name (string), price (number), onDelete (func)**

Ref

Referência ao objeto DOM do
elemento HTML.

Ref

```
class NameInput extends React.Component {  
  state = {name: 'Bruno'}  
  
  onChange (e) {  
    this.setState({  
      name: e.target.value  
    });  
  }  
  
  render() {  
    return (  
      <input ref={c => this.input = c} onChange={this.onChange.bind(this)} />  
    );  
  }  
}
```

Inserindo um novo produto

Adicionar um novo componente.

Nome: **CartFooter**

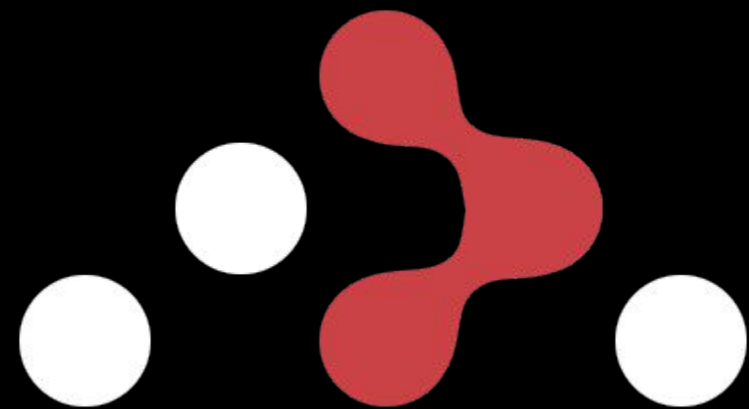
Props: **onCreateProduct (func)**

Projeto completo

<https://github.com/brunobra/aula-react>

E mais...

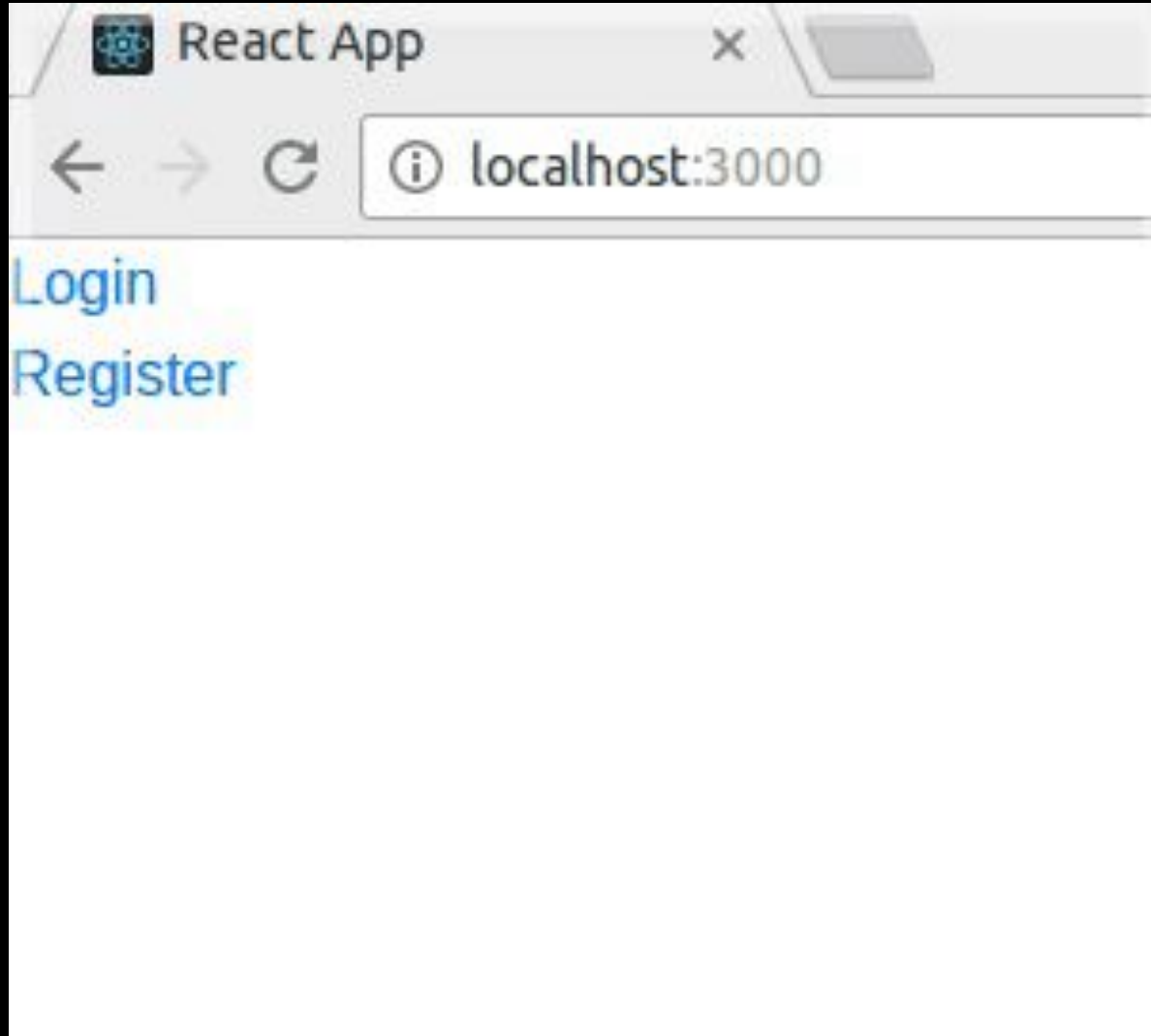
React router



Coleção de componentes de navegação.

<https://reacttraining.com/react-router/>

React router

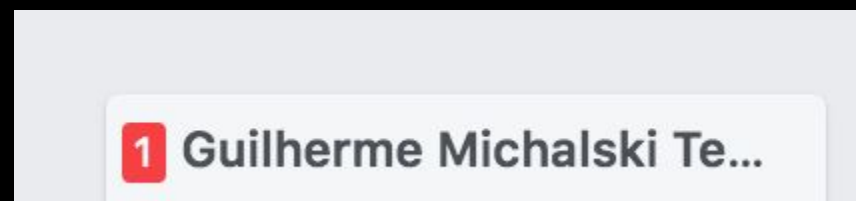


<https://reacttraining.com/react-router/>



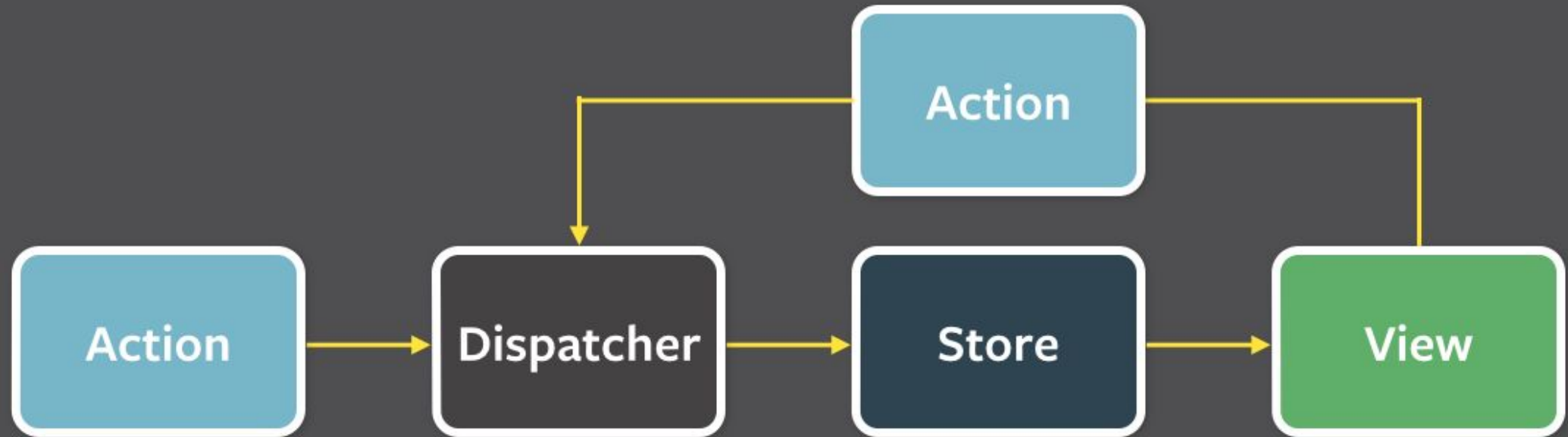
Padrão criado para complementar os componentes React, utilizando "unidirectional data flow".

Flux - Como surgiu



<https://facebook.github.io/flux/>

Flux



<https://facebook.github.io/flux/>

Redux



Redux é uma biblioteca construída seguindo alguns padrões do Flux, contendo diversas APIs que auxiliam no desenvolvimento, consistência, testes e debugs.

<https://redux.js.org/>

Redux



<https://github.com/brunobra/aula-react>

Branch: redux

<https://redux.js.org/>

Storybook



Storybook é uma "UI" de um ambiente de desenvolvimento, testes e documentação de componentes.

<https://storybook.js.org/>

Storybook



UI Component Development

<https://storybook.js.org/>

Storybook



<https://github.com/brunobra/aula-react>

Branch: storybook

<https://facebook.github.io/react-native/>

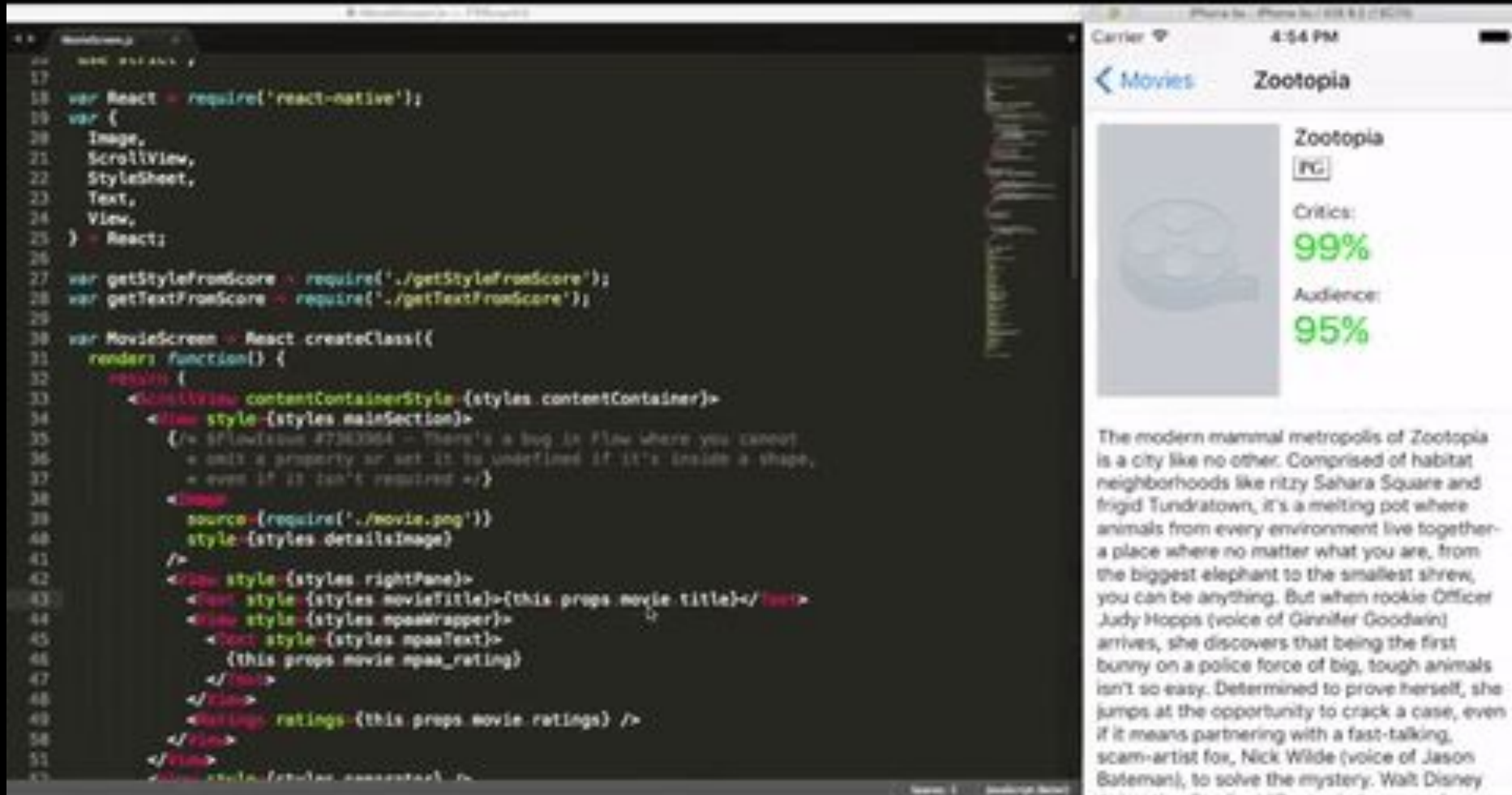
React Native



É um framework que permite escrever código nativo mobile, da mesma forma como desenvolvemos aplicações web, usando Javascript e React.

<https://facebook.github.io/react-native/>

React Native



<https://facebook.github.io/react-native/>

Obrigado!

brunootaviojunqueira@gmail.com

<https://github.com/brunobra>

<https://www.linkedin.com/in/bjunqueira/>