

React

Introdução ao React



Francisco Costa

- *Freelancer Software Engineer & External Consultant*
- Formador de linguagens de programação
- Adepto de boas práticas e metodologias de desenvolvimento de software



Francisco Costa

- jfcpcosta@gmail.com
- <http://franciscocosta.net>
- <http://twitter.com/jfcpcosta>
- <http://linkedin.com/in/jfcpcosta>
- <https://github.com/jfcpcosta>



A photograph of three white computer keyboard keys standing upright. The keys are arranged horizontally and spell out the word "YOU". The key on the left contains the letter "Y", the middle key contains "O", and the key on the right contains "U". The letters are printed in a dark purple or black font. The background is a plain, light-colored surface.

Y O U

Programa

- Introdução a biblioteca **React**, seu enquadramento no universo de desenvolvimento em Aplicações “web-based”.
- Pequena revisão de conceitos bases em **React**, criação de projetos com “**creat-react-app**”, estrutura de uma aplicação, criação de **componentes**, **Virtual DOM**, **JSX**, **Props**, **State**, **Redux**...
- Bases de **Routing** em **React** com “**React Router**”
- Evolução da biblioteca, utilização de “**Funcional Components**” com “**React Hooks**”, utilização de “**useState**, **useReducer**, **useEffect**, ...” e “third party” **Hooks**



Requisitos

- Confortáveis com **HTML5**
- Confortáveis com **Javascript** (intro)
 - Funções **ES5** como **map**, **sort**, **filter**, ...
 - Alguns conhecimentos **ES6**, **ES7**, **ES8**, ...
 - (iremos ver e usar alguns conceitos)
- **Node.js** e **NPM**



Preparação do ambiente de trabalho

- Instalar **Node.js** e **GIT**
- Abrir (instalar) **Chrome**, e **extensões para React**:
 - **React Developer Tools**
- Escolher Editor de Código e instalar
- **VSCODE**, extensões: **es6**, **react**, **live server**, ...



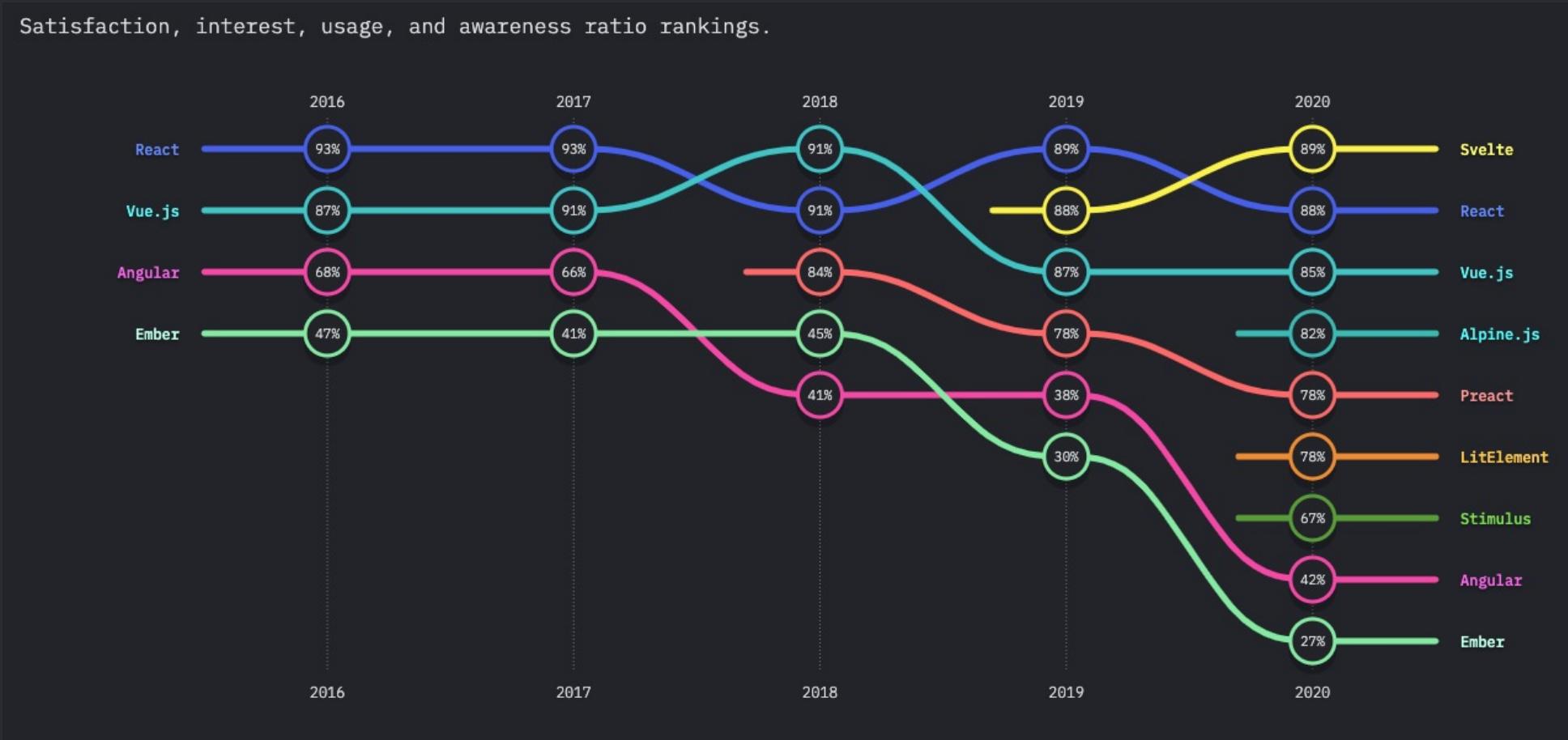
História



História



State Of Js



Introdução



Introdução

Com **React** ou **Vue**, temos de escolher outras bibliotecas para **routing, dependency injection, forms, bundling e deploying**



Introdução

O **Angular** é uma framework que inclui tudo o que precisamos para **desenvolver e fazer deploy** de uma Web app



React.js



14

React.js

- Biblioteca open source de Javascript desde **2013**
- Desenvolvida pelo **Facebook** e **Instagram** com intenção de ser o “**V**” no **MVC**
- Excelente para desenvolvimento de **SPA** (Single Page Application)



React.js

- Virtual DOM
- JSX
 - (sintaxe de XML para Javascript, opcional)



React.js

- **JS Objects** são mais rápidos que **DOM Objects**
- **React Virtual DOM** é um **JS Object**
- React **nunca** lê do “real” **DOM**
- React só escreve no “real” DOM **se necessitar**
- Atualiza **eficientemente** o **DOM**



React.js

DOM

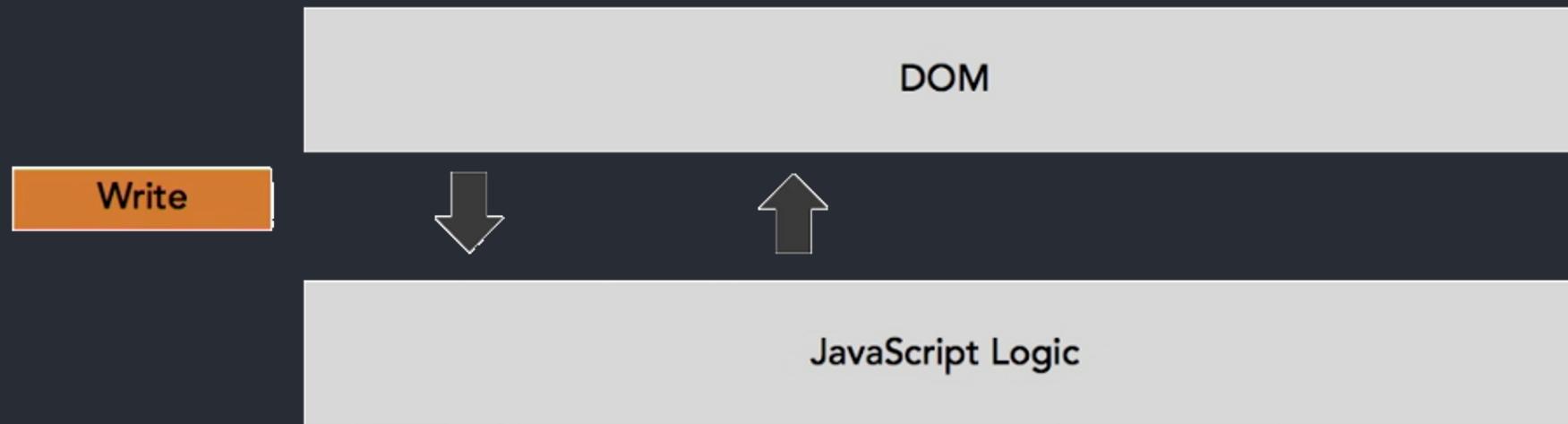
JavaScript Logic



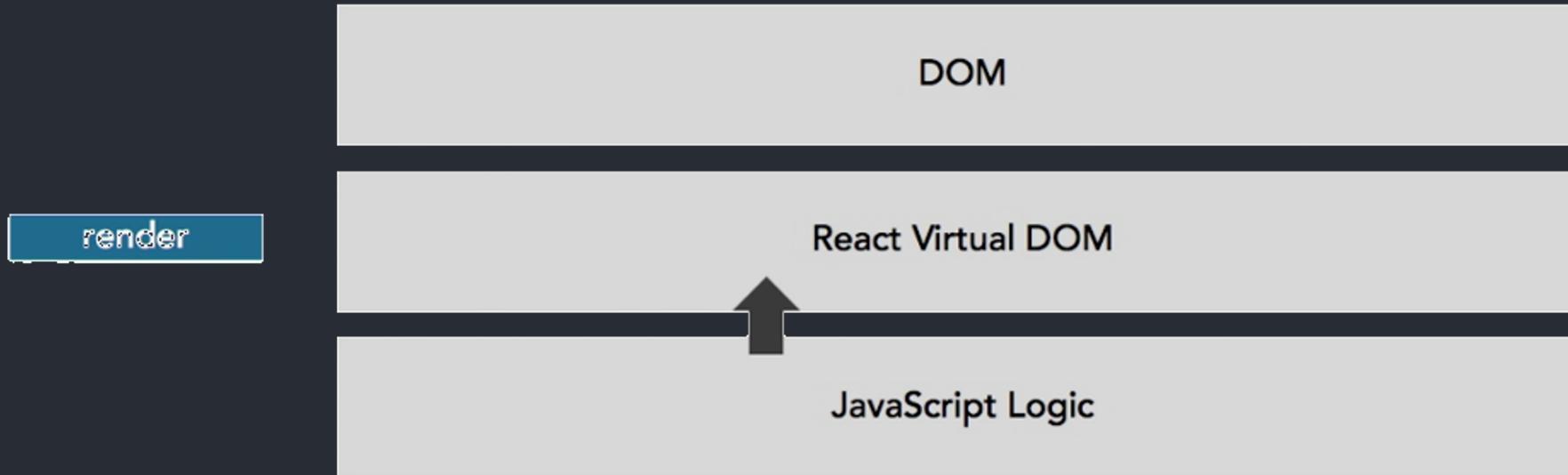
React.js



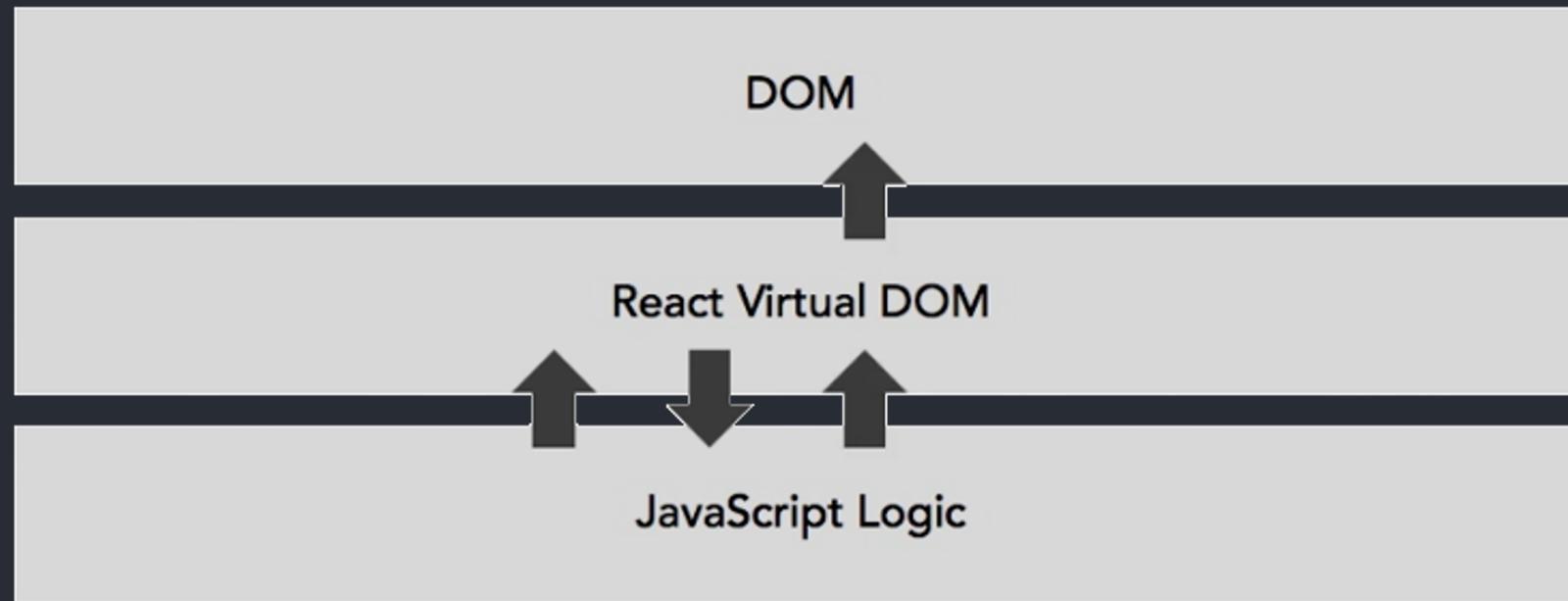
React.js



React.js



React.js



Exemplo



React.js

- Focado em **Componentes**
- **One-Way Data Flow**
 - States
 - Props



ES6



ECMAScript

ECMAScript é um standard para linguagens de client-side scripting



ECMAScript

Primeira edição da especificação do **ECMAScript** foi lançada em
1997

ES5 (standard) foi lançado em **2009**



ECMAScript

Sexta edição finalizada em 2015

Chamado de **ES6 ou ES2015**

ES7 em 2016, ES8 em 2017, ...



ECMAScript

Scope, let, const, template literal, multiline string, optional parameters, arrow functions, rest, spread, generator functions, destructuring, classes...



let (blocked scoped)

```
// scope

let a = 12; // accessible everywhere
function myFunction() {
  console.log(a); // alerts 12
  let b = 13;
  if(true) {
    let c = 14; // this is ONLY accessible inside if block
    alert(b); // alerts 13
  }
  alert(c); // alerts undefined
}
myFunction();
alert(b); // alerts undefined
```



Redeclaring variables

```
// redeclaring variables

let a = 0;
let a = 1; // SyntaxError: Identifier 'a' has already been declared
function myFunction() {
  let b = 2;
  let b = 3; // SyntaxError: Identifier 'b' has already been declared
  if(true) {
    let c = 4;
    let c = 5; // SyntaxError: Identifier 'c' has already been declared
  }
}
myFunction();
```



const (blocked scoped)

```
// const

const pi = 3.141;
pi = 4; // not possible in this universe, or in other terms,
         // throws Read-only error

// Referencing objects using constant variables
const a = {
  name: "Joao"
};
console.log(a.name);
a.name = "Jose";
console.log(a.name);
a = {};
```

//throws read-only exception



Default parameters

```
● ● ●  
  
// Default parameters ES5  
function myFunction(x, y, z) {  
    x = x === undefined ? 1 : x; // ternary operator  
    y = y || 2; // or :  
    z = z || 3;  
    console.log(x, y, z); //Output "6 7 3"  
}  
myFunction(6, 7);  
  
// Default parameters ES6  
function myFunction(x = 1, y = 2, z = 3) {  
    console.log(x, y, z);  
}  
myFunction(6,7); // Outputs 6 7 3
```



Spread & rest operators

```
● ● ●  
// spread operator  
let array1 = [2,3,4];  
let array2 = [1, ...array1, 5, 6, 7];  
console.log(array2); //Output "1, 2, 3, 4, 5, 6, 7"
```

```
// REst operator  
function myFunction(a, b, ...args) {  
    console.log(args); //Output "3, 4, 5"  
}  
myFunction(1, 2, 3, 4, 5);
```



Destructuring Arrays & Objects

```
● ● ●  
  
// destructuring Arrays  
  
let myArray = [1, 2, 3];  
let a, b, c;  
[a, b, c] = myArray; //array destructuring assignment syntax  
  
// or  
let [a, b, c] = [1, 2, 3];  
  
//with rest operator  
let [a, ...b] = [1, 2, 3, 4, 5, 6];  
  console.log(a); // 1  
  console.log(Array.isArray(b)); // true  
  console.log(b); // 2,3,4,5,6  
  
// destructuring Objects  
let object = {"name" : "John", "age" : 23};  
let name, age;  
({name, age} = object); //object destructuring assignment syntax
```



Arrow functions

```
● ● ●  
//arrow functions  
  
var circumference = function(pi, r) {  
  var area = 2 * pi * r;  
  return area;  
}  
var result = circumference(3.141592, 3);  
console.log(result); //Output 18.84952  
  
  
let circumference = (pi, r) => 2 * pi * r;  
let result = circumference(3.141592, 3);  
console.log(result); //Output 18.84952
```



Template Literals, Multiline Strings

```
//template literals

const a = 20;
const b = 10;
const c = "JavaScript";
const str = `My age is ${a+b} and I love ${c}`;
console.log(str);
```

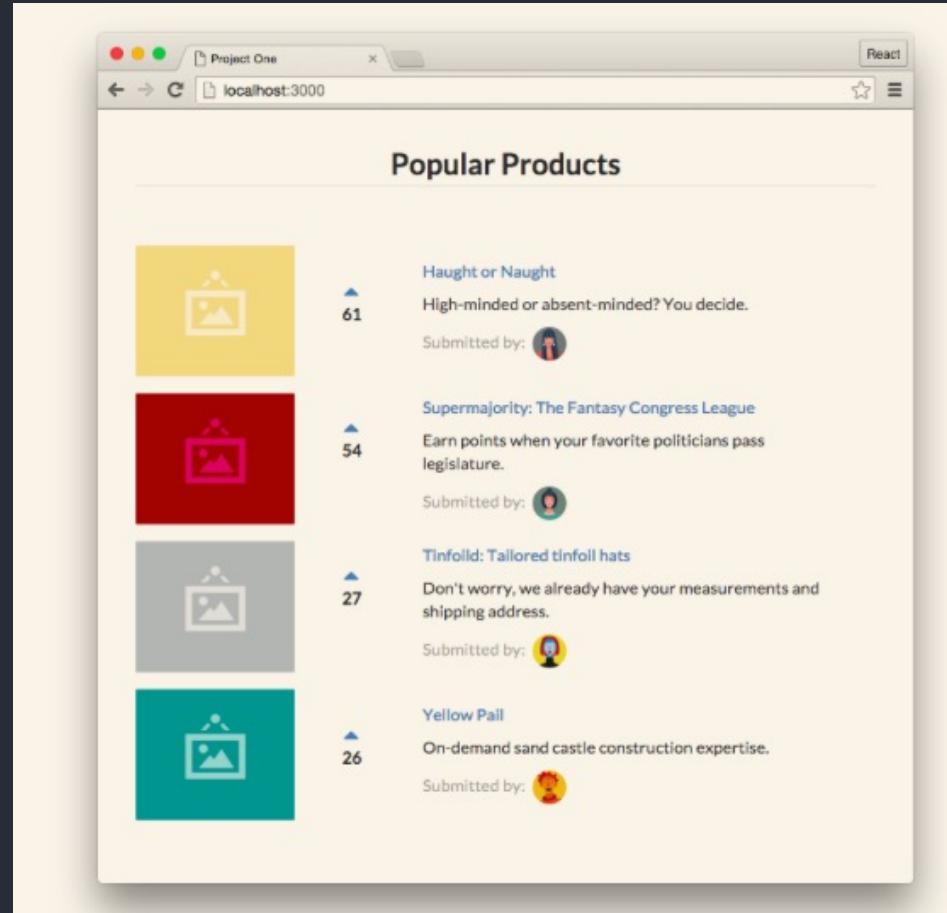


Exemplo

Voting App



Voting App



Voting App

Popular Products

	61	Haught or Naught High-minded or absent-minded? You decide.	Submitted by:
	54	Supermajority: The Fantasy Congress League Earn points when your favorite politicians pass legislature.	Submitted by:
	27	Tinfoild: Tailored tinfoil hats Don't worry, we already have your measurements and shipping address.	Submitted by:
	26	Yellow Pail On-demand sand castle construction expertise.	Submitted by:

ES6 Classes – React Component

```
● ● ●  
  
class HelloWorld extends React.Component {  
  render() {  
    return(Hello, world!)  
  }  
}  
  
const HelloWorld = React.createClass({  
  render() {  
    return(Hello, world!)  
  }  
})
```



JS vs JSX

```
// JS
React.createElement('div', { className: 'ui items' }, 'Hello, friend! I am a basic React component.')

// JSX
<div className='ui items'>
  Hello, friend! I am a basic React component.
</div>
```

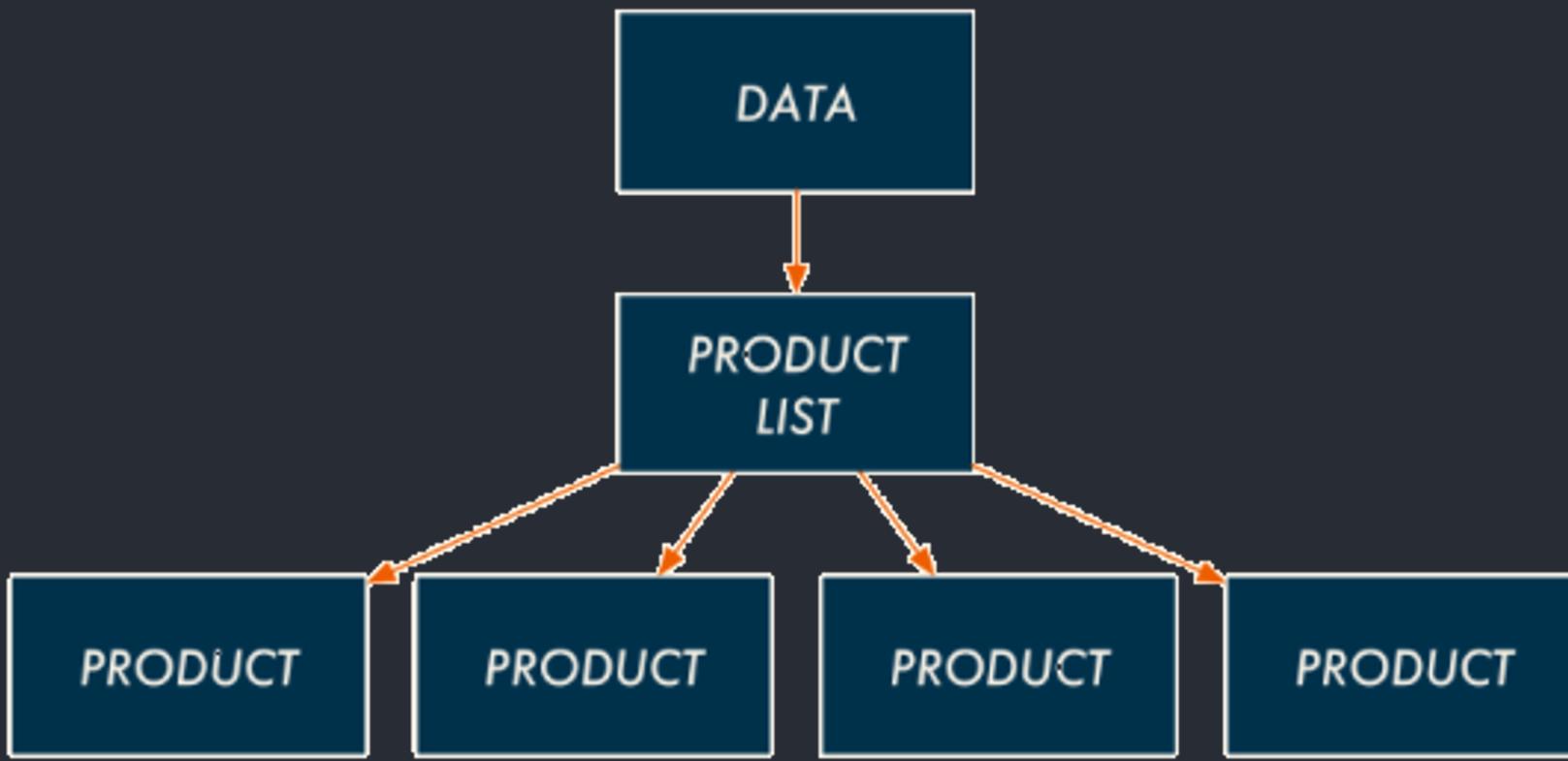


Código...

```
$ npx create-react-app voting-app  
$ cd voting-app  
$ npm start
```



React - Props



React – “key” attribute



React - state

- Enquanto as **props** são imutáveis e são detidas pelo **componente pai**
- **O state** é detido pelo **próprio componente**
- Atualizado com **this.setState()**



React - state

Quando o state ou as props de um componente atualizam, o componente faz re-render a ele próprio



Resumo



Resumo

- Organize your React App in Components
- JSX inside render method
- Data Flow from Parent to Child through props
- Event flow through functions
- State vs Props
- How to manipulate State
- React LifeCycle Methods



Exercícios práticos



Exercícios práticos

- Acrescentar botão para retirar votos
- Acrescentar um botão para alterar a visualização da ProductList alternadamente por ordem descendente e ascendente
- **Extra:** Criar formulário para adicionar produto á lista.

