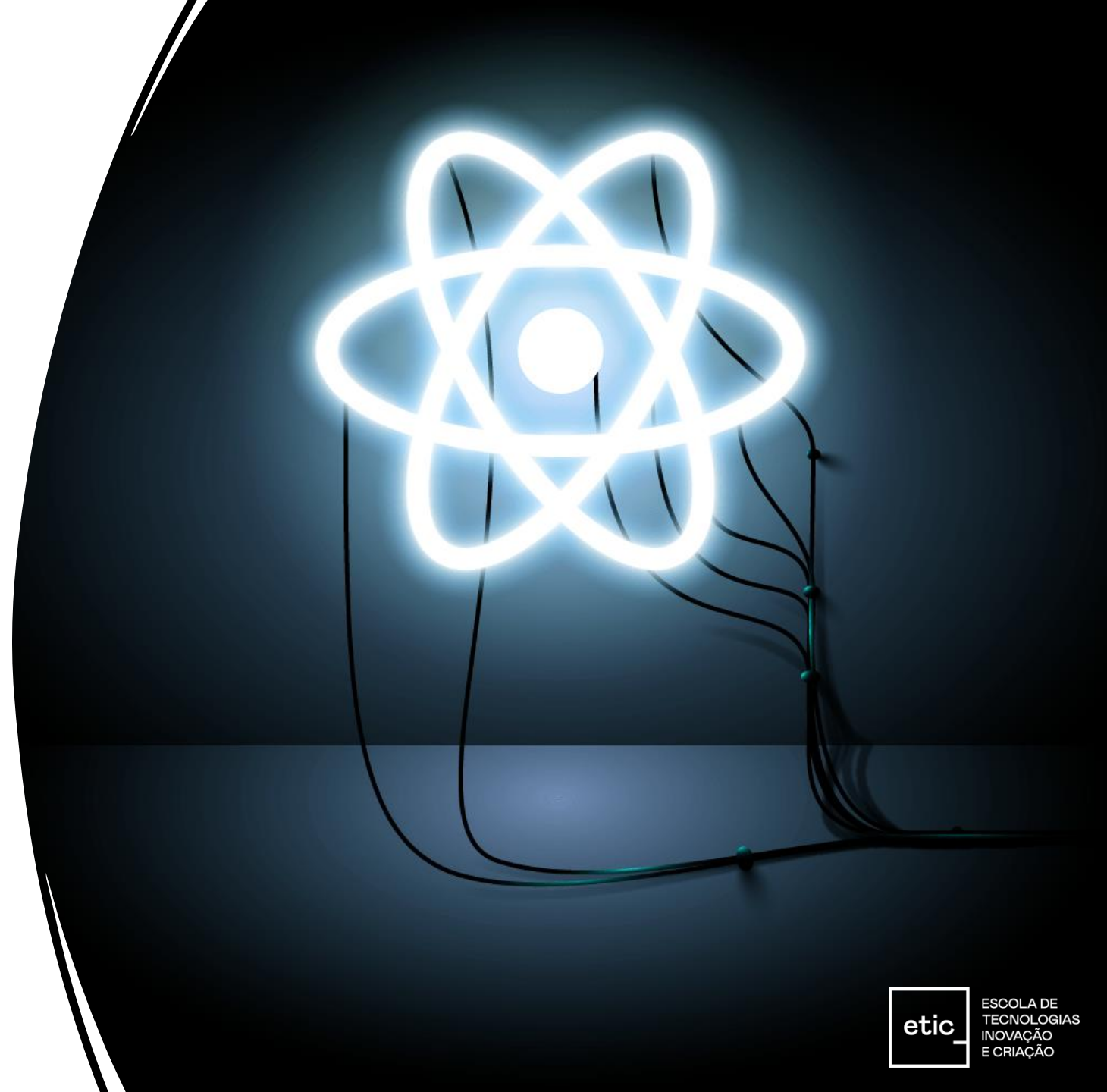


# REACT

## Introdução ao React

Formador: Bruno Deilhot





## Preparação do ambiente de trabalho

- Instalar Node.js e GIT
- Instalar Chrome e extensão React Developer Tools



# Preparação do ambiente de trabalho

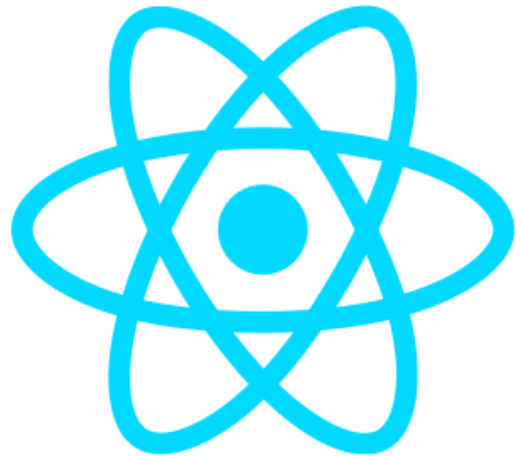


## Extensões de VSCode recomendadas:

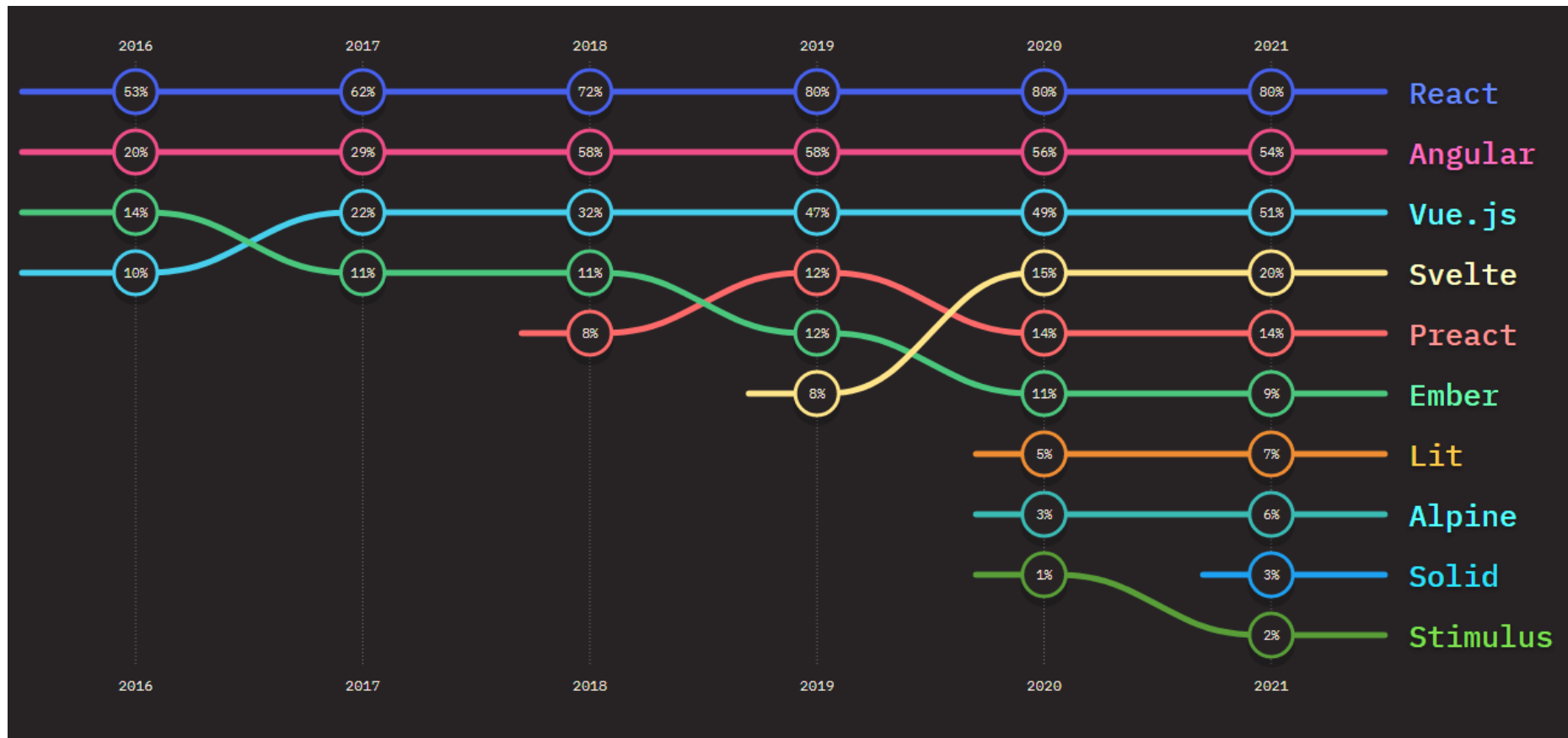
- ESLint
- Prettier ESLint
- Auto Rename Tag
- Auto Close Tag
- Path Intellisense
- Instalar fonte com ligatures (exemplo JetBrains Mono)

# Frameworks JavaScript - FrontEnd

---



# Frameworks JavaScript - FrontEnd





# História

---

- Biblioteca fundada em 2011 por engenheiro de software do Facebook
- Desenvolvido para ser o “V” no MVC
- Passa a open source em 2013
- Atualmente mantido pelo Facebook e comunidade

# Porquê React?

- Reduzir a complexidade
- Modularidade e reutilização de componentes
- Melhor desempenho através do **Virtual DOM**

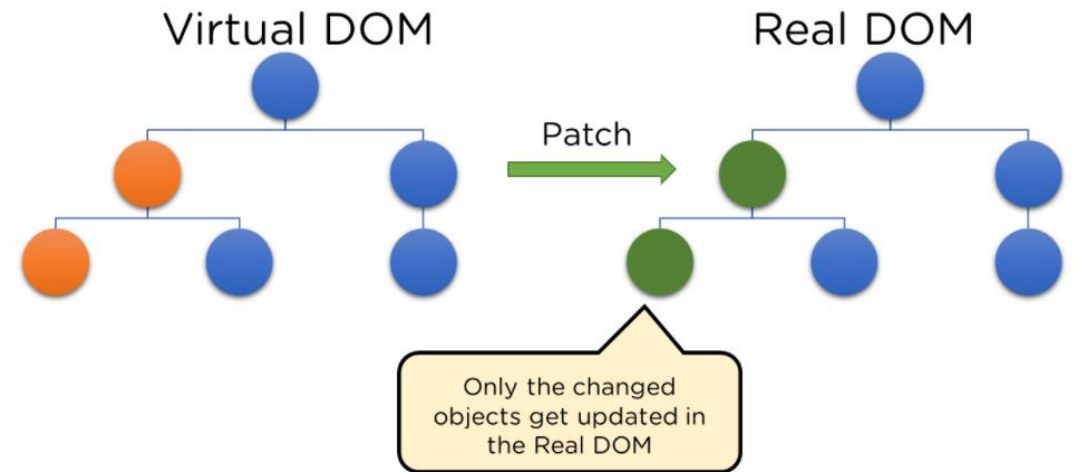


# Virtual DOM

Guarda informação em memória

Manipulação é mais rápida

Só atualiza no Real DOM os objetos em que o estado foi alterado





Exemplo



# JSX

---

- Possibilita escrever estrutura HTML em JavaScript

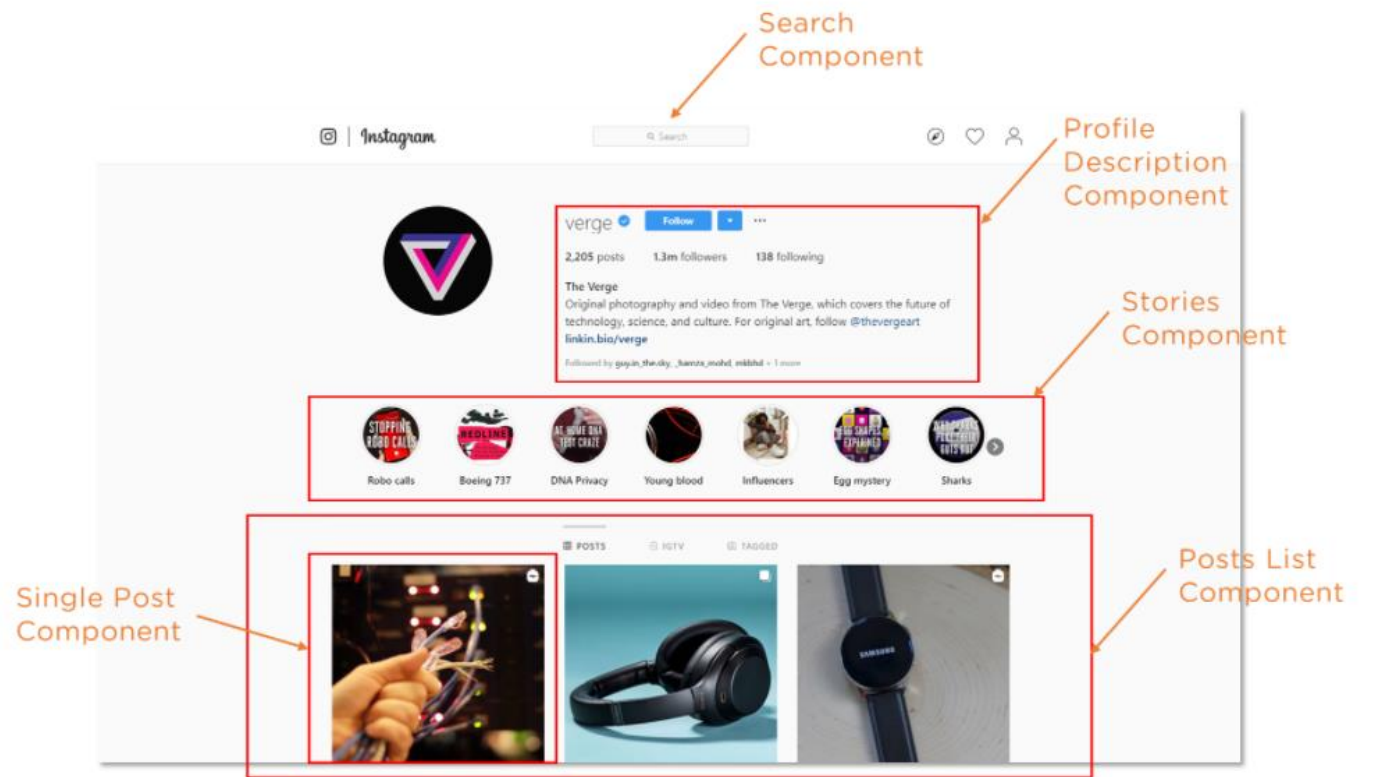


```
1 //JS
2 React.createElement('div', { className: 'ui items' }, 'Hello friend! I am a basic React component.');
```

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

# Componentes em React

- Fluxo de dados unidirectional




# ECMAScript

- Linguagem script standard que forma a base de JavaScript browser e Node.js
- Adiciona várias funcionalidades novas ao JavaScript



# Block scope

---



```
1  const name = 'Joao';  
2  
3  function getName() {  
4      const name = 'Ana';  
5  
6      console.log(name);  
7  }  
8  
9  getName();
```

# Default parameters

---

```
1  const tema = {
2      id: 215,
3      nome: 'planeta'
4  };
5
6  function getTema(id = 1, nome = 'default', cor = 'branco') {
7      console.log(id, nome, cor);
8  };
9
10 getTema(tema.id, tema.nome);
```

# Spread & Rest operators




```
1 // Spread operator
2 const array = ['joao', 'ana', 'miguel', 'maria'];
3
4 const array2 = ['paulo', 'teresa', ...array];
5
6 console.log(array2);
```



```
1 // Rest operator
2
3 function myFunction(a, b, ...args) {
4     console.log(args);
5 };
6
7 myFunction('joao', 'ana', 'miguel', 'maria');
```

# Destructuring



```
1  const array = ['joao', 'ana', 'miguel', 'maria', 'telma'];
2
3  const [a, b, ...c] = array;
4
5  console.log(a, b, c);
6
7
8
9  const object = {
10     firstName: 'Telma',
11     lastName: 'Serra',
12     age: 26,
13 };
14
15 const {firstName, lastName, age} = object;
16
17 console.log(firstName);
18
19
20 const {age: old} = object;
21
22 console.log(old);
```



# Arrow function

```
1  const arrowFunction = () => {  
2    console.log('Arrow Function');  
3  };  
4  
5  
6  const arrowFunctionParams = (a = 'nome', b, c) => {  
7    console.log(a, b, c);  
8  };  
9  
10 arrowFunctionParams();  
11  
12  
13  
14 const sum = (a, b) => a + b;  
15  
16 const result = sum(2, 2);  
17  
18 console.log(result);
```

# Template literal & Multiline strings

---



```
1  const person = {  
2    firstName: 'Maria',  
3    age: 25,  
4  };  
5  
6  const {firstName, age} = person;  
7  
8  const string = `Hoje a ${firstName} fez ${age} anos.`;  
9  
10 console.log(string);  
11
```

# Async Await

---

```
1  const asyncFunction = async () => {  
2    return "I am an async function";  
3  };  
4  
5  console.log(asyncFunction());  
6  
7  (async () => {  
8    console.log(await asyncFunction());  
9  })();
```

# Criar o projeto

---

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



# Exemplo

---



▲ 59 ▼

[Tinfoil: Tailored tinfoil hats](#)

We already have your measurements and shipping address.

Submitted by:

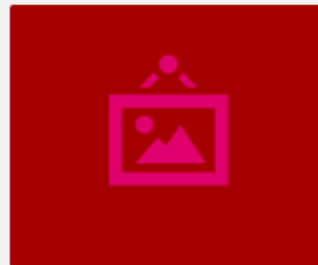


▲ 45 ▼

[Yellow Pail](#)

On-demand sand castle construction expertise.

Submitted by:

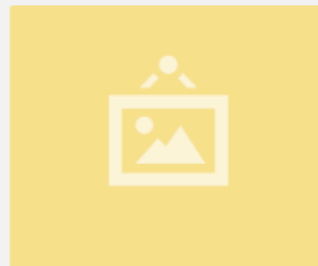


▲ 24 ▼

[Supermajority: The Fantasy Congress League](#)

Earn points when your favorite politicians pass legislation.

Submitted by:



▲ 23 ▼

[Haught or Naught](#)

High-minded or absent-minded? You decide.

Submitted by:



**git**

# Git – Sistema Open Source Distribuído de Controlo de Versões

## Sistema de controlo

- Rastrear e guardar o nosso código

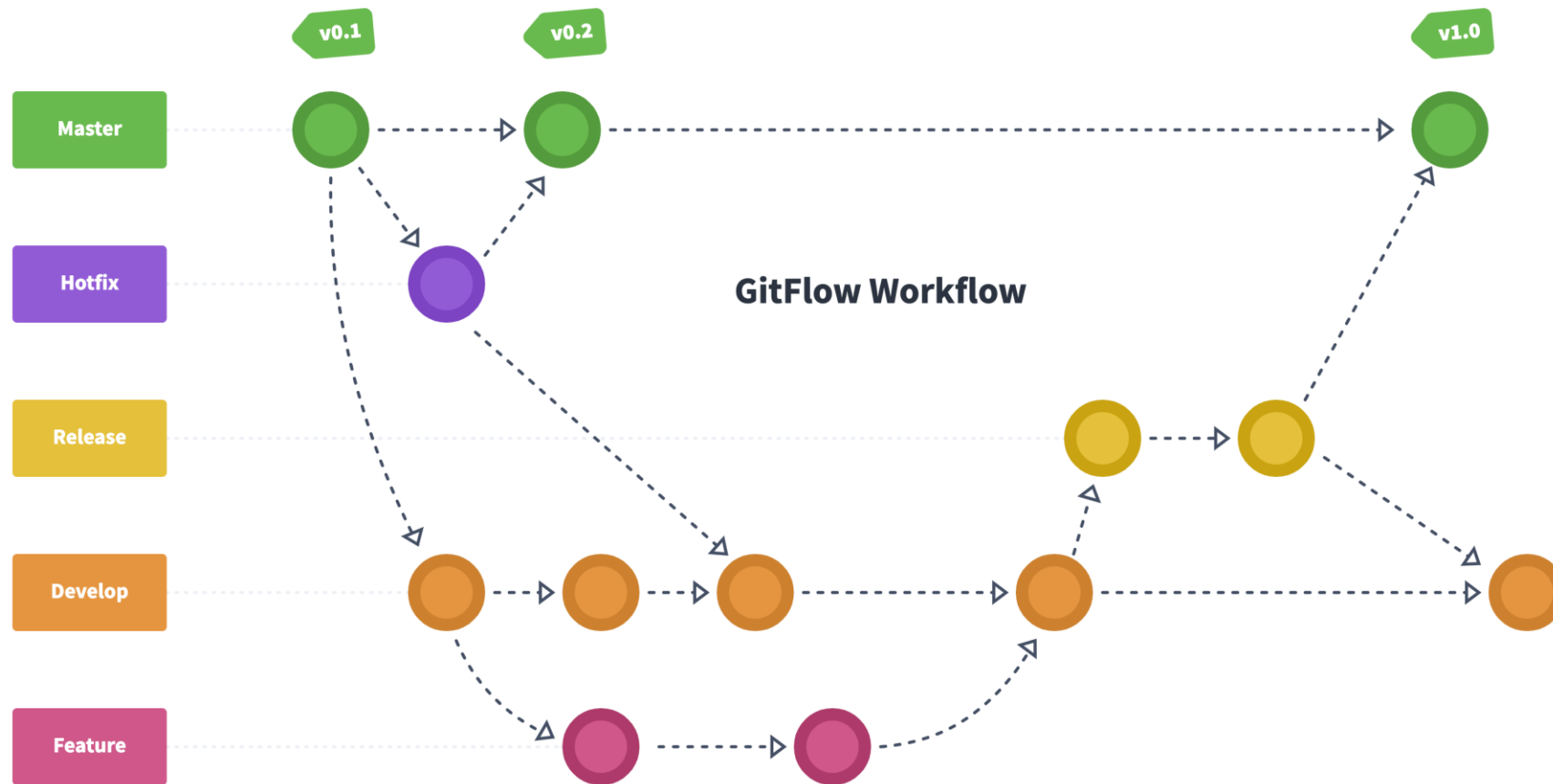
## Sistema de controlo de versões

- O código que é guardado está em constante modificação e pode ser desenvolvido por vários programadores em simultâneo
- Através de controlo de versões é mantido um histórico de todas as alterações efetuadas

## Sistema distribuído de controlo de versões

- O código é guardado localmente no computador dos programadores e também nos repositórios em servidores remotos

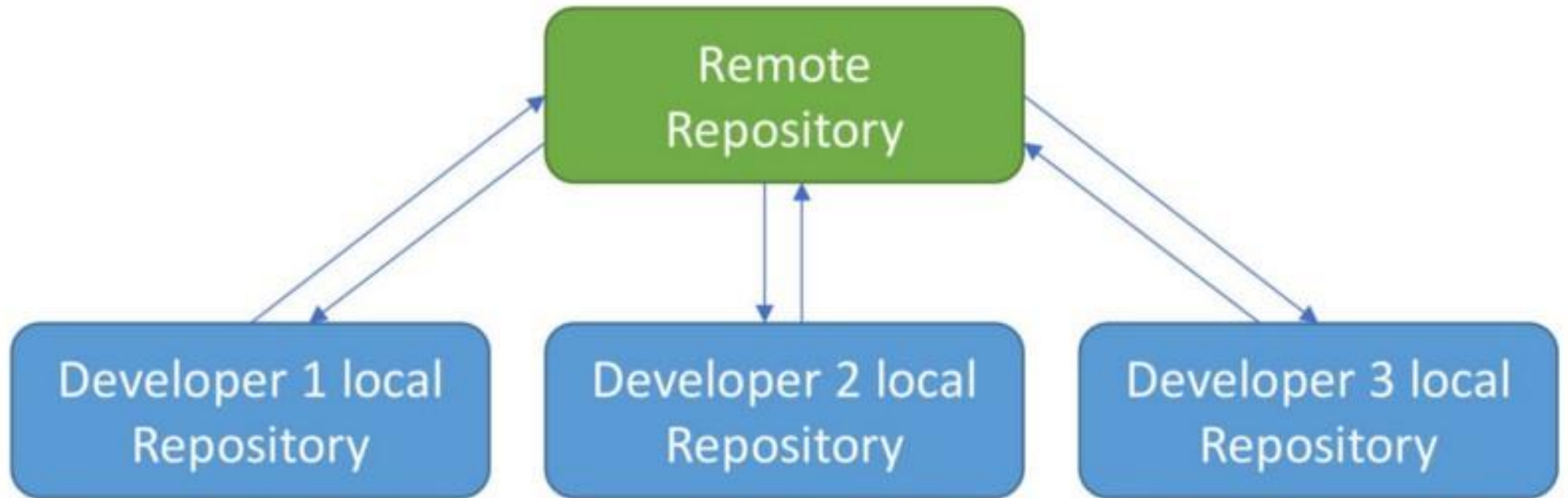
# Git Flow





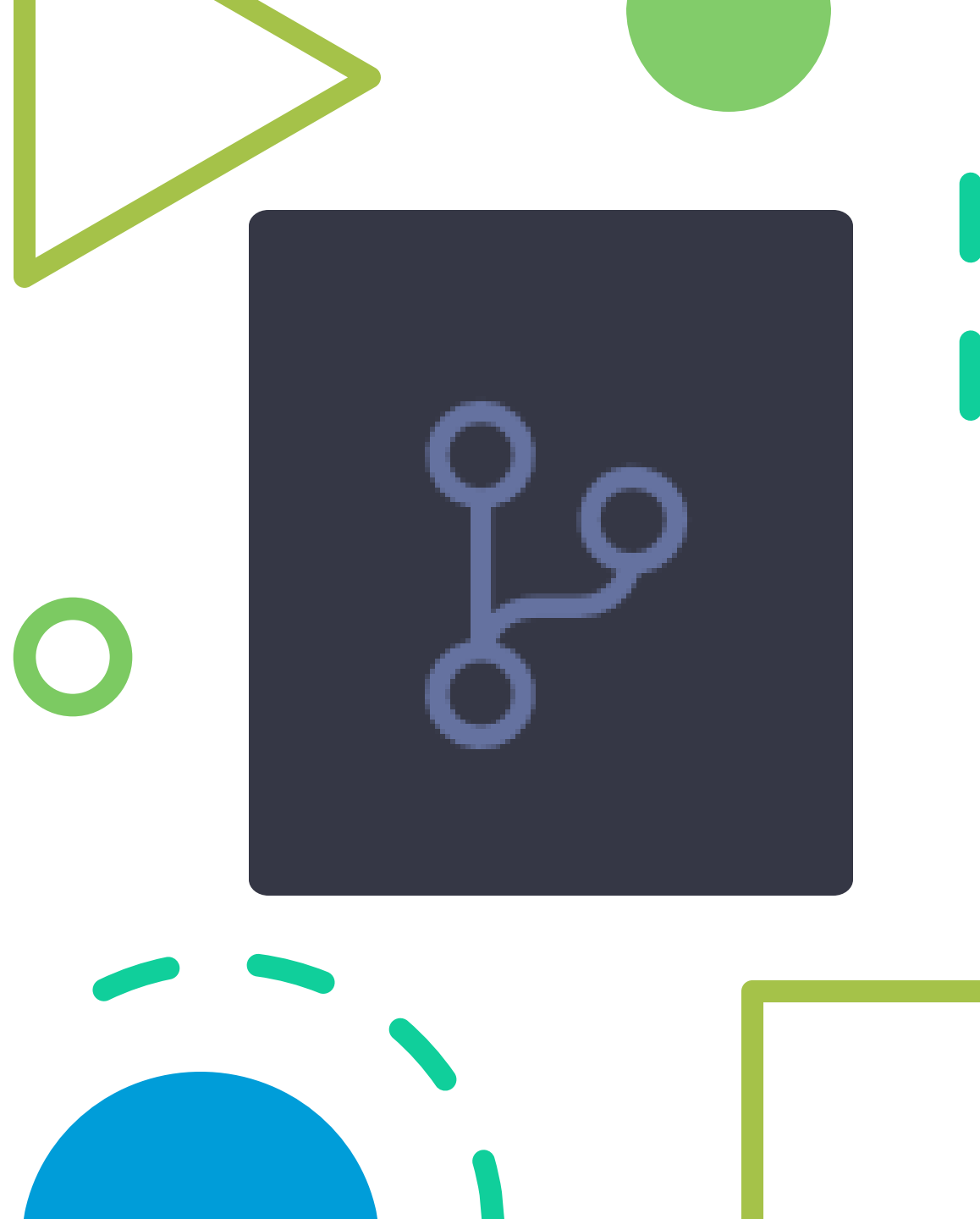
# GitHub

---



# GitHub

1. Criar conta no GitHub
2. Associar a conta ao VSCode
3. Abrir a pasta voting-app no VSCode
4. Abrir a tab Source Control
5. Publicar para o GitHub

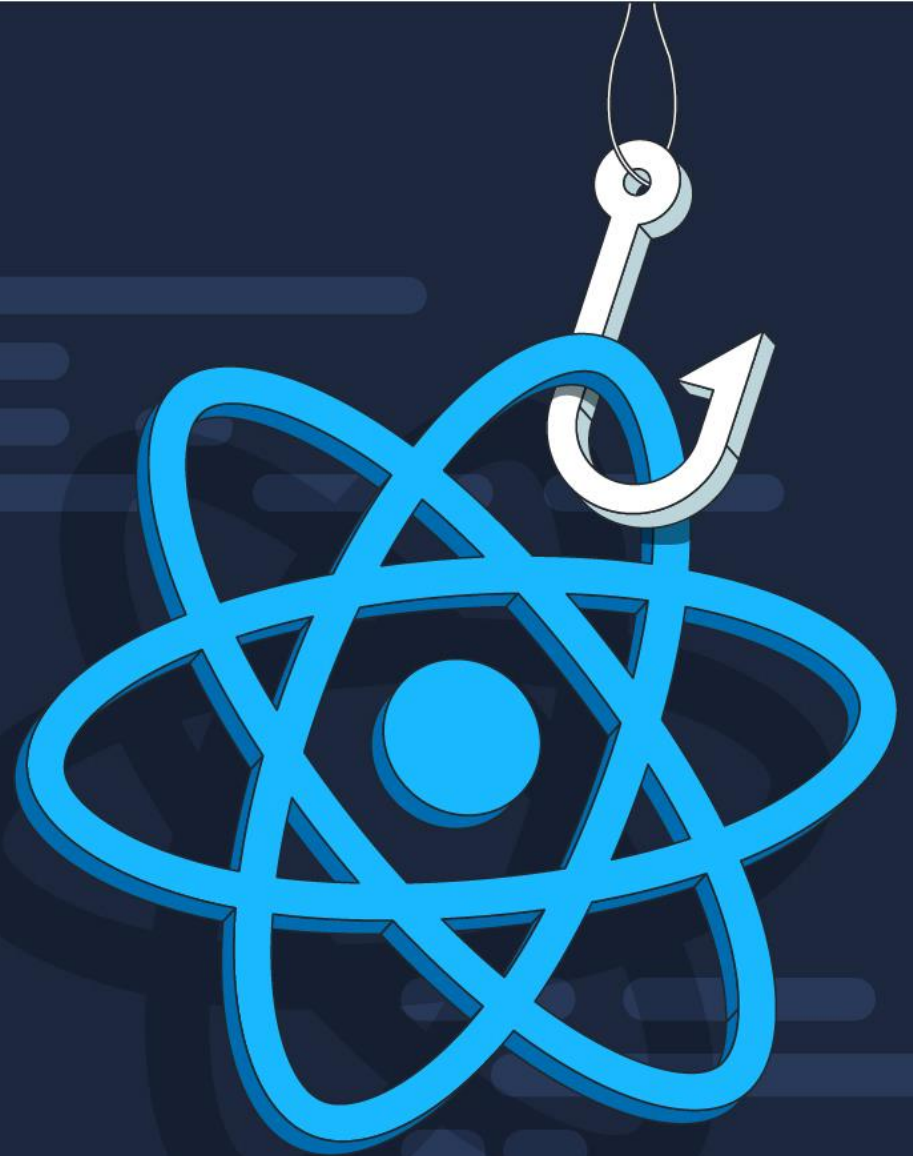


# Git Config

- `git config --global user.name "Bruno Deilhot"`
- `git config --global user.email "bruno_deilhot@professores.etic.pt"`

# React Hooks

---





# React Hooks

- Foram a resposta para substituir a necessidade de usar classes e construtores para gerir estados nos componentes
- Ao longo dos anos o seu desenvolvimento tornou a programação funcional a mais utilizada

# React Hooks

## useState

---

Função retorna dois elementos:

- name => estado atual
- setName => setter (atualiza o estado)

```
1  import { useState } from 'react';
2
3  const App = () => {
4
5      const [name, setName] = useState();
6
7      const handleChange = (event) => {
8          setName(event.target.value);
9      };
10
11     return (
12         <div>
13             {name ? `O meu nome é ${name}.` : 'Eu estou anônimo!'}
14             <hr />
15             <input type="text" value={name} onChange={handleChange} />
16         </div>
17     )
18
19 };
20
21 export default App;
```

# Lightbulb Challenge

**Nível 1** – Uma lâmpada controlada por dois botões (ON e OFF)

**Nível 2** – Uma lâmpada controlada por um botão ( ON / OFF )

**Nível 3** – Três lâmpadas com um botão para controlar cada uma individualmente e outro botão para ligar / desligar todas

**Nível 4** – (Opcional / TPC) Criar uma constante `nrOfLightbulbs` que pode aceitar qualquer número e a partir deste criar um array de booleanos que será o estado inicial das lâmpadas (true or false). Deste estado inicial renderizar os componentes `Lightbulb` com botão ON/OFF



# React Hooks useEffect

---

- Permite-nos criar efeitos secundários no componente
- Geralmente é invocados com base numa ou mais dependências

```
1  import { useEffect, useState } from "react";
2
3  const UsersList = () => {
4
5      const [user, setUser] = useState('John');
6
7      useEffect(() => {
8          const storedUser = localStorage.getItem('user');
9          if (storedUser) {
10             setUser(storedUser);
11          }
12      }, []);
13
14      useEffect(() => localStorage.setItem('user', user), [user]);
15
16      return (
17          <div>
18              <select value={user} onChange={(e) => setUser(e.target.value)}>
19                  <option>John</option>
20                  <option>Mary</option>
21                  <option>Jane</option>
22                  <option>Lou</option>
23              </select>
24
25              <p>Selected user: {user}</p>
26          </div>
27      );
28  }
29
30  export default UsersList;
```



# React Hooks - useContext

---

```
1 import { createContext, useState } from "react";
2
3 export const AppContext = createContext();
4
5 const AppProvider = ({children}) => {
6
7     const [user, setUser] = useState({
8         name: "Francisca Pardal",
9         age: 21,
10    });
11
12    return (
13        <AppContext.Provider value={{user, setUser}}>
14            {children}
15        </AppContext.Provider>
16    )
17 };
18
19
20 export default AppProvider;
```

```
1 import { useContext } from "react";
2 import { AppContext } from "../contexts/AppContext";
3
4 const Greeting = () => {
5
6     const { user } = useContext(AppContext);
7
8     const { name, age } = user;
9
10    return (
11        <p style={{ backgroundColor: "aliceblue" }}>
12            Olá, o meu nome é {name} e tenho {age} anos
13        </p>
14    );
15 };
16
17 export default Greeting;
```

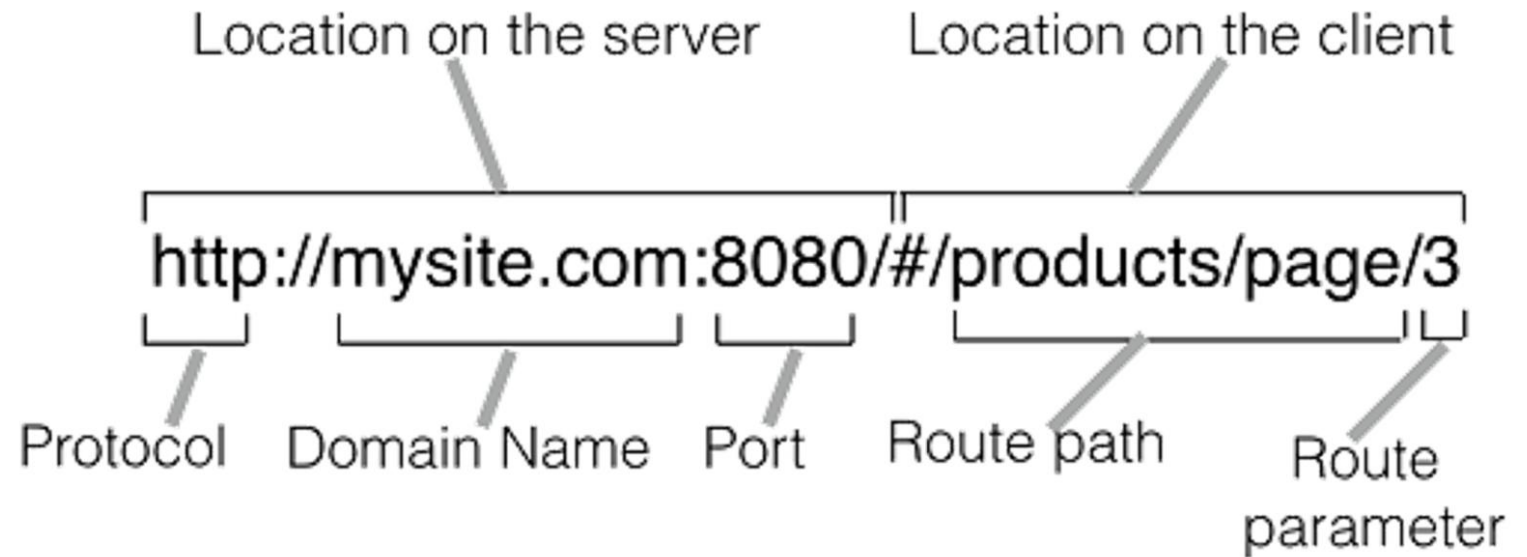
# React Hooks - useReducer

- Substitui useEffect para gestão de estados mais complexos
- Utiliza por base o método reduce

```
1 import { useReducer } from "react";
2 import { reducer } from "../reducer";
3
4 const Counter = () => {
5   const initialState = 0;
6
7   const [state, dispatch] = useReducer(reducer, initialState);
8
9   const handleIncrement = () => {
10     dispatch("INCREMENT");
11   };
12
13   const handleDecrement = () => {
14     dispatch("DECREMENT");
15   };
16
17   return (
18     <div>
19       <h2>{state}</h2>
20       <button onClick={handleIncrement}>Increment</button>
21       <button onClick={handleDecrement}>Decrement</button>
22     </div>
23   );
24 };
25
26 export default Counter;
```

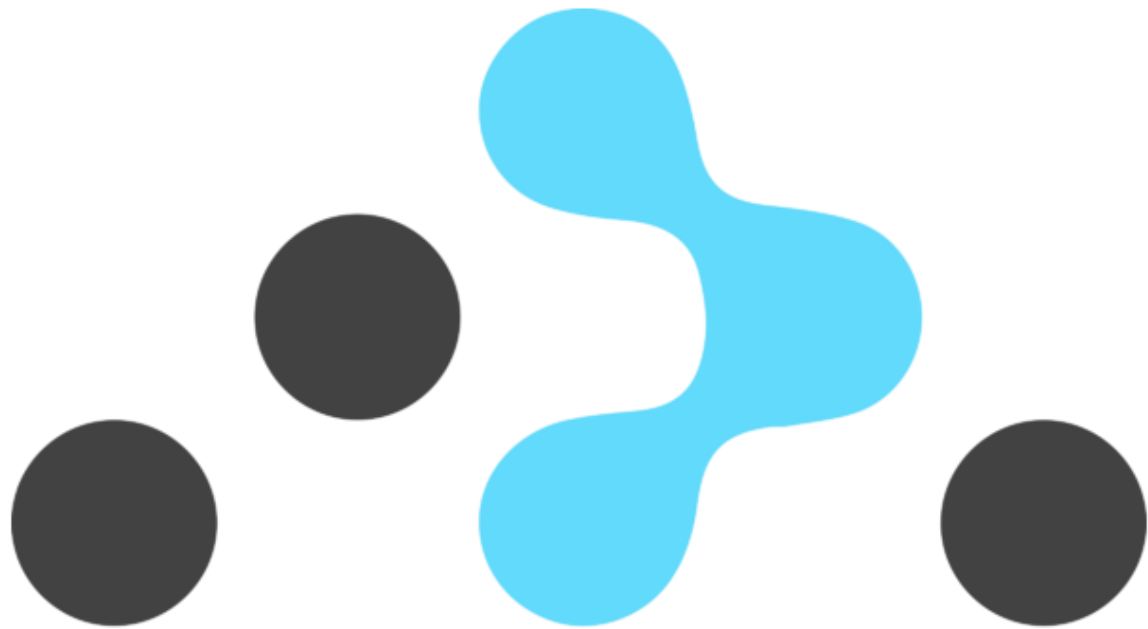
```
1 export const reducer = (state, action) => {
2   switch (action) {
3     case "INCREMENT":
4       return state + 1;
5     case "DECREMENT":
6       return state - 1;
7     case "TOGGLE":
8       return !state;
9     default:
10      return state;
11   }
12 };
13
```

# Routing



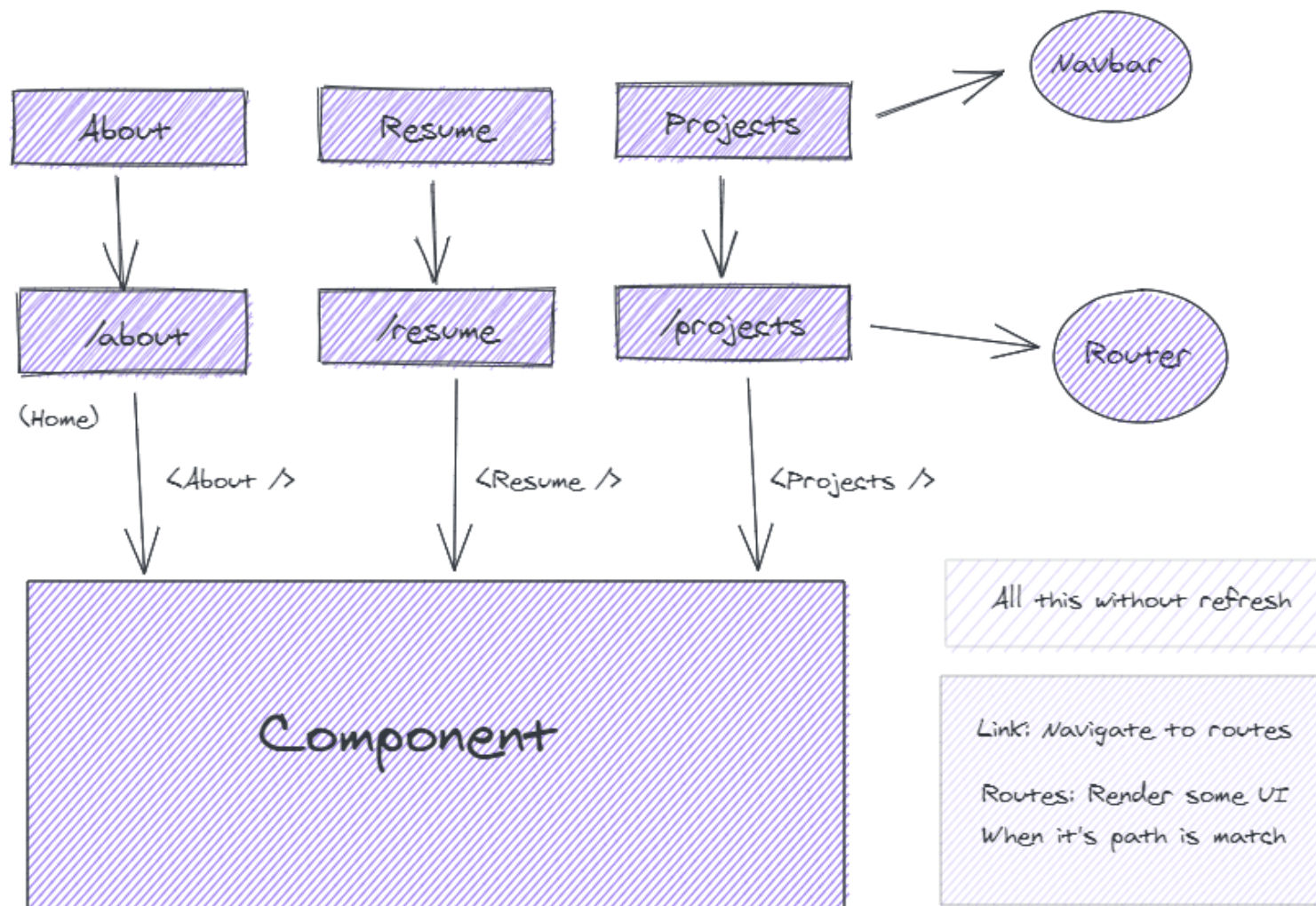
# React Router

---



# React Router

---



# TypeScript

---



# TypeScript

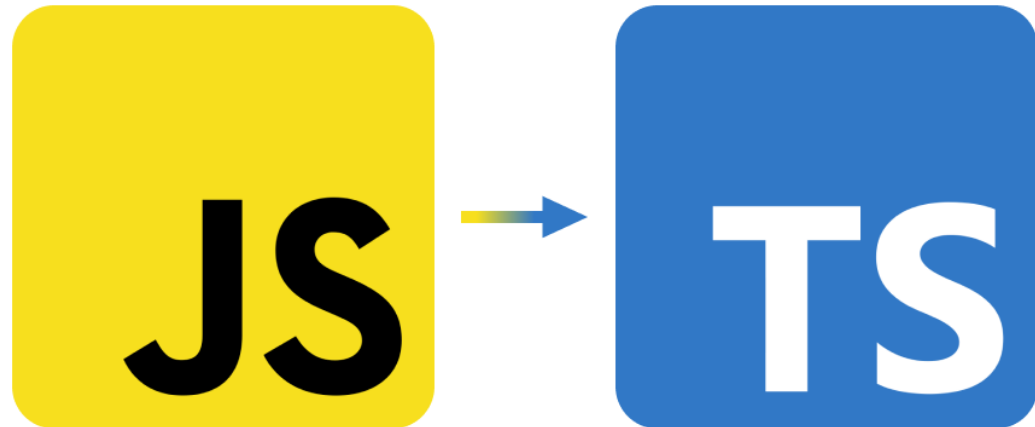
- Superset de JavaScript
- Adiciona tipagem ao JavaScript
- Tipagem adiciona regras que nos ajuda a detetar erros antes da compilação



# TypeScript

---

- `npm install -g typescript`





# TypeScript

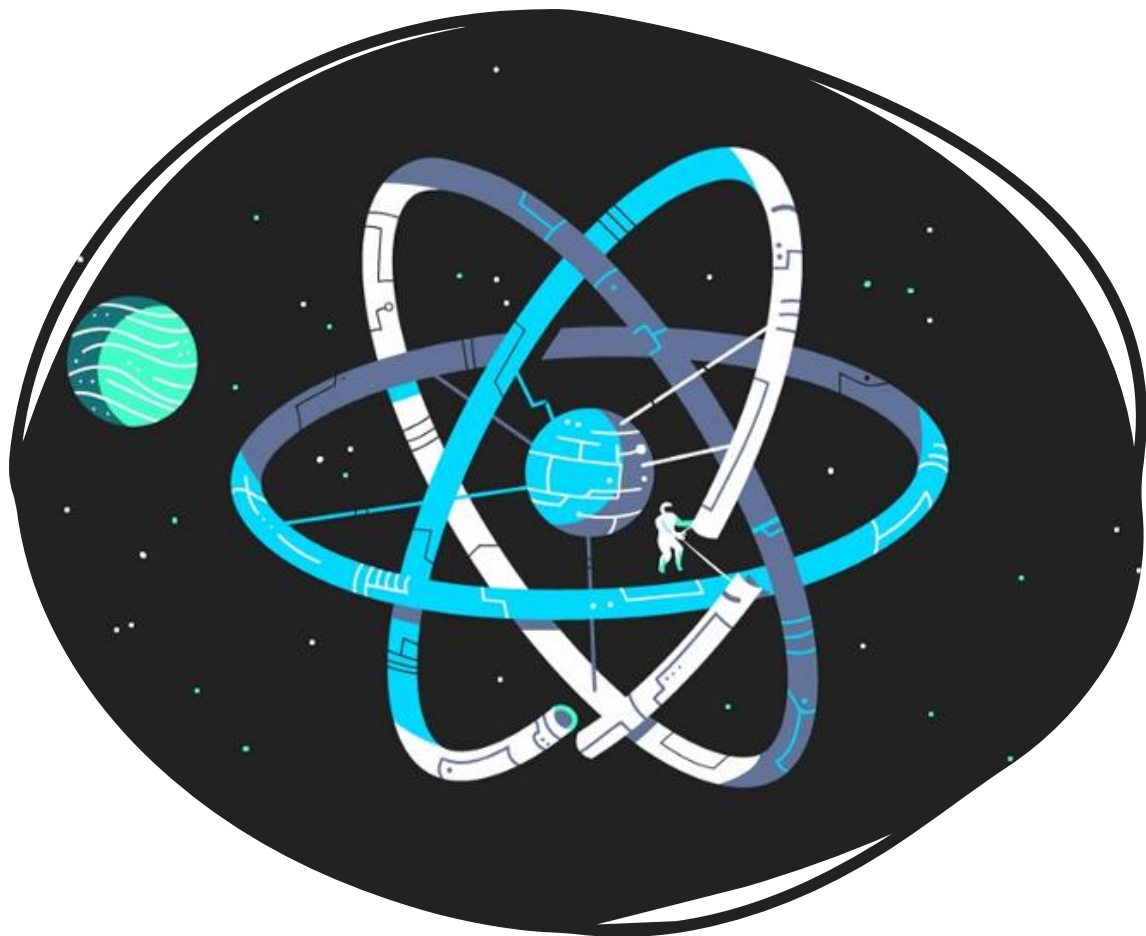
---

## **JavaScript:**

- `npx create-react-app app-name`

## **TypeScript:**

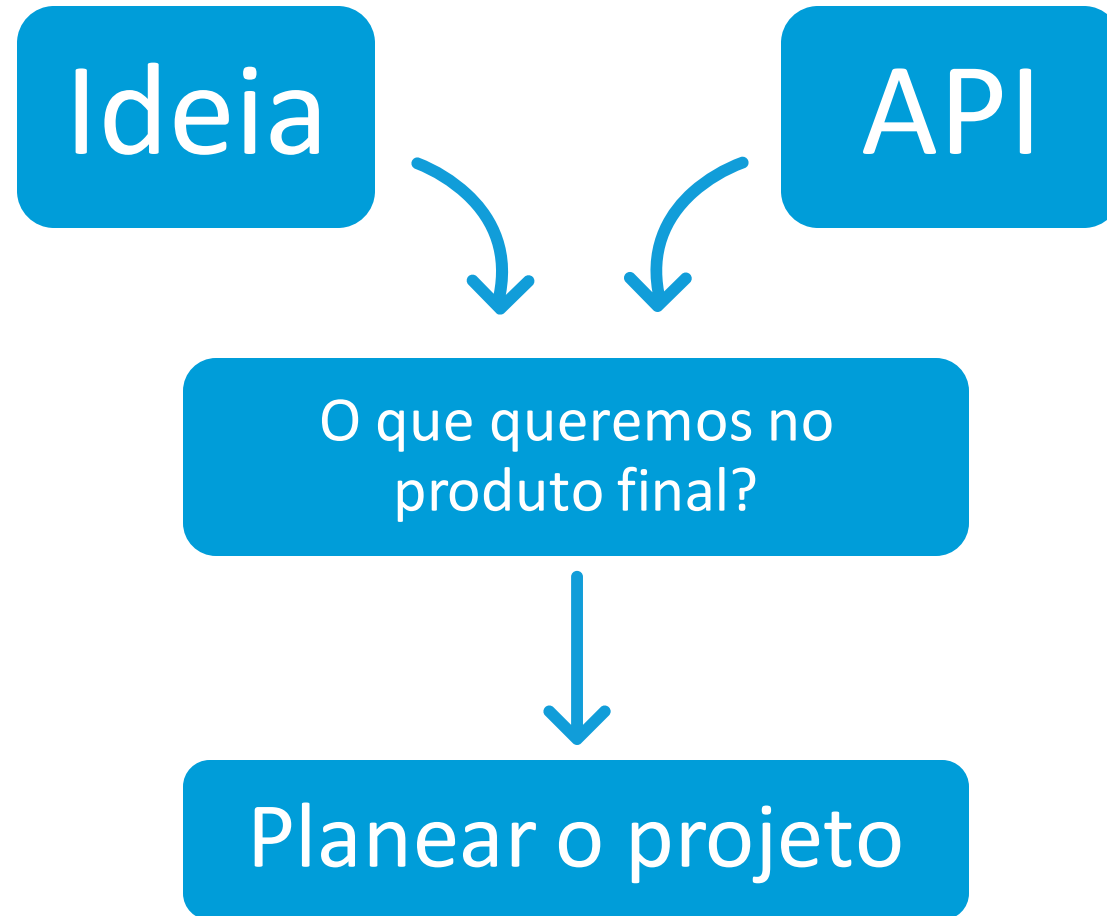
- `npx create-react-app app-name --template typescript`



# Projeto React

---

# Projeto React



# Projeto React - Setup

1. `npx create-react-app portfolio-example --template typescript`
2. `npm i react-router-dom @types/react-router-dom`
3. `npm i styled-components @types/styled-components`
4. `npm i axios`

# Requisitos e Avaliação do Projeto

## Requisitos:

- Consumir uma API externa, apresentar os dados e permitir manipular os mesmos, exemplo pesquisar ou favoritos;
- Gestão global de estado com useContext ou React Redux;
- Sistema de rotas com React Router;
- Utilizar Local Storage para guardar informação, exemplo Dark ou Light Mode;
- Ao correr o comando "npm start" o projeto é apresentado e encontra-se funcional;

Critério de Avaliação	Valor
Cumprir os requisitos	15
Estrutura e organização do projeto	2
Implementação de código DRY (Don't Repeat Yourself)	1
Projeto com TypeScript e tipagem adequada	1
Repositório no GitHub e projeto Live	1