

## **Bem Vindo ao Curso de React pela Step lecionado pelo professor Fabio Luiz**

Neste curso aprenderá fundamentos no React e algumas bibliotecas essenciais ao desenvolvimento Front end dinamico usando o framework JavaScript React. Todos os codigos de exemplo usados nas aulas estará no repositório: [github.com/devfabioluiz/aulasReact](https://github.com/devfabioluiz/aulasReact) na aplicação my-app, a qual será a base de todas as aulas, sendo dividida por cada topico de cada aula em uma pasta separada com o nome Aula (numero da aula).

Bons estudos!

Recomendações de ambiente: instalar VSCode com plugins Codeium, VS Icons, Simple React Snippets, ESLint e Prettier) ajudarão a melhorar sua performance.

### **Antes da aula começar – Preparando ambiente:**

#### **Passo 1) Instale o Node.js e o npm**

- Baixe e instale o [Node.js](https://nodejs.org/). O npm (Node Package Manager) será instalado automaticamente.

Passo 2) Criar aplicação React. Realize os passos a seguir:

- **Abra o VSCode, clique em Terminal e depois em New Terminal e digite e execute o comando para criar uma nova aplicação React**

```
npx create-react-app minha-aplicacao
```

Substitua minha-aplicacao pelo nome do seu projeto.

- Entre no diretório do projeto digitando e executando o codigo `cd 'minha-aplicacao'` (nome da aplicacao que colocou)
- Inicie o servidor de desenvolvimento executando o codigo `npm start`

- Pronto! Abrirá o seu navegador no endereço localhost:3000 rodando sua aplicação React, a qual estará com o código reproduzido na página dentro do arquivo App.js na raiz do projeto.

Passo 3) Visualizar e entender a estrutura do projeto:

minha-aplicacao/

```
|— public/
|   |— index.html
|— src/
|   |— App.css
|   |— App.js
|   |— App.test.js
|   |— index.css
|   |— index.js
|   |— logo.svg
|   └— reportWebVitals.js
|— .gitignore
|— package.json
|— README.md
└— node_modules/
```

- **src/**: Contém o código-fonte da aplicação.
- **public/**: Contém os arquivos públicos (não processados pelo React).
- **index.js**: O ponto de entrada do React.
- **App.js**: O componente principal da aplicação. É nele que estará o código que está sendo exibido no seu navegador.

## **Aula 1 - Introdução ao React e Tipos de Componentes**

### **Introdução ao React**

React é uma biblioteca JavaScript desenvolvida pelo Facebook para criar interfaces de usuário (UI) interativas e reutilizáveis. Ele facilita o desenvolvimento de aplicações web dinâmicas ao permitir a construção de componentes independentes e gerenciáveis. Uma de suas principais características é o uso do Virtual DOM, que otimiza atualizações na interface, garantindo alto desempenho.

### **Tipos de Componentes**

#### **Componentes Funcionais**

Componentes funcionais são funções JavaScript que retornam elementos React. Eles são recomendados para a maioria dos casos de uso modernos devido à sua simplicidade e eficiência. Com a introdução dos React Hooks na versão 16.8, os componentes funcionais ganharam ainda mais poder, permitindo gerenciar estado e ciclos de vida sem a necessidade de componentes de classe.

Na pasta Aula 1, no componente Saudacao vemos um exemplo de um componente funcional, utilizando uma const para declarar uma arrow function, tal como já visto nas aulas de JavaScript.

#### **Componentes de Classe**

Componentes de classe são uma forma mais antiga de criar componentes em React. Eles utilizam classes JavaScript e possuem um método especial chamado render(), que retorna os elementos React a serem exibidos. Embora ainda sejam suportados, os componentes de classe são menos utilizados em novos projetos devido à complexidade adicional em relação aos componentes funcionais com Hooks.

Na pasta Aula 1, no componente SaudacaoClasse vemos um exemplo de um componente de classe, utilizando uma classe tal como visto nas aulas anteriores de JavaScript.

## Comparativo entre Componentes Funcionais e de Classe

A seguir, apresentamos um comparativo entre os dois tipos de componentes:

Aspecto	Componentes Funcionais	Componentes de Classe
Complexidade	Mais simples e concisos	Mais complexos devido ao uso de classes
Estado	Gerenciado com Hooks	Gerenciado com this.state e setState
Ciclo de Vida	Hooks como useEffect	Métodos como componentDidMount