

# Programação Front-end – AULA 22

## ReactJS

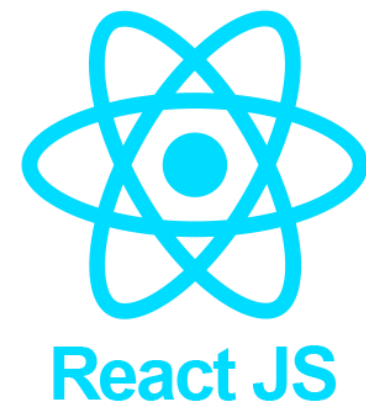
Matheus Moresco

Engenharia de Software - 3º Período

2025/01

# Por que usar React?

- Permite criar interfaces de forma modular e reutilizável.
- Atualiza apenas o que muda na tela (Virtual DOM), melhorando performance.
- Usa JSX, que facilita escrever HTML dentro do JavaScript.
- É flexível e pode ser combinado com diversas bibliotecas.
- Muitas soluções como **React Native** (apps mobile), **Next.js** (SSR e SSG), e **Remix** são baseadas em React.
- Facilita testes e manutenção com componentes independentes.



# Mercado e Aplicações Reais

- Um dos frameworks/bibliotecas mais usados no mercado frontend.
- Utilizado por empresas como Meta, Netflix, Airbnb, iFood, Nubank.
- Base para tecnologias como React Native e Next.js.
- Grande comunidade, muitos recursos de aprendizado e vagas disponíveis.

# Conceitos Fundamentais do React

**JSX (JavaScript XML):** JSX é uma extensão de sintaxe que permite escrever HTML dentro do JavaScript. Facilita a construção de interfaces visuais de forma declarativa.

**Componentes:** Componentes são funções ou classes que retornam JSX. Eles representam partes da interface, como botões, cabeçalhos ou seções inteiras.

```
function App() {  
  const mensagem = "Hello World!"  
  return (  
    <div className="App">  
      <h1>{mensagem}</h1>  
    </div>  
  );  
}  
  
function App2() {  
  return (  
    <div className="App2">  
      <App />  
    </div>  
  );  
}
```

# Conceitos Fundamentais do React

- **Props e State**

- **Props (Propriedades):** Dados que são passados de um componente pai para um componente filho. São **imutáveis** dentro do componente que os recebe.
- **State (Estado):** Dados **internos do componente** que podem mudar com o tempo. Controla o comportamento dinâmico da interface.

```
function App(props) {  
  const [contador, setContador] = useState(0);  
  
  return (  
    <div className="App">  
      <h2>Olá, {props.nome}!</h2>  
      <p>Você clicou {contador} vezes</p>  
      <button onClick={()  
        => setContador(contador + 1)  
      }>  
        Clique aqui  
      </button>  
    </div>  
  );  
}
```

# Conceitos Fundamentais do React

- **Eventos:** Eventos são funções que respondem a ações do usuário, como cliques, digitação ou rolagem. Usa-se o padrão **camelCase**, como **onClick**, **onChange** etc.

```
function App(props) {  
  const [contador, setContador] = useState(0);  
  
  function handleClick() {  
    setContador(contador + 1);  
  }  
  
  return (  
    <div className="App">  
      <h2>Olá, {props.nome}!</h2>  
      <p>Você clicou {contador} vezes</p>  
      <button onClick={handleClick}>  
        Clique aqui  
      </button>  
    </div>  
  );  
}
```

# Conceitos Fundamentais do React

- **useEffect:** é um hook do React que permite executar efeitos colaterais em componentes funcionais.
- Ele serve para lidar com operações que acontecem **fora do fluxo normal de renderização**, como:
  - Buscar dados em uma API
  - Assinar eventos (ex: evento de scroll, resize)
  - Atualizar o DOM manualmente
  - Configurar timers ou intervalos

```
function MeuComponente() {  
  useEffect(() => {  
    // Código a ser executado após a renderização  
    console.log('Componente montado ou atualizado');  
  }, [/* dependências */]);  
  
  return <div>Exemplo useEffect</div>;  
}
```

# Ambiente de Desenvolvimento

- Instalar o **Node.js** (vem com o npm):
  - Baixe em: <https://nodejs.org/>
- Criação de um projeto com create-react-app ou Vite
  - 'npx create-react-app meu\_projeto'
  - 'npm create vite'



## Exemplo Prático

1. Criar um projeto React
2. Criar um componente
3. Criar um botão de contador usado use state
4. Criar um formulário de cadastro de informações de pessoas.
5. Listar as pessoas cadastradas.

# Trabalho Final

- Formar os grupos para o trabalho Final, até 5 pessoas.
- Pensar em uma projeto Frontend para desenvolver com a equipe
- Requisitos do projeto:
  - Usar ReactJS
  - Ao menos 3 telas
  - Fazer acesso a uma API. (fake API, API publica ou API própria)
- Enviar Grupos no form: <https://forms.gle/rmosu3jDBZWdNZVc9>