



DESENVOLVIMENTO FULL STACK

Eventos

Caio Eduardo do Prado Ireno
caio.ireno@faculdadeimpacta.com.br

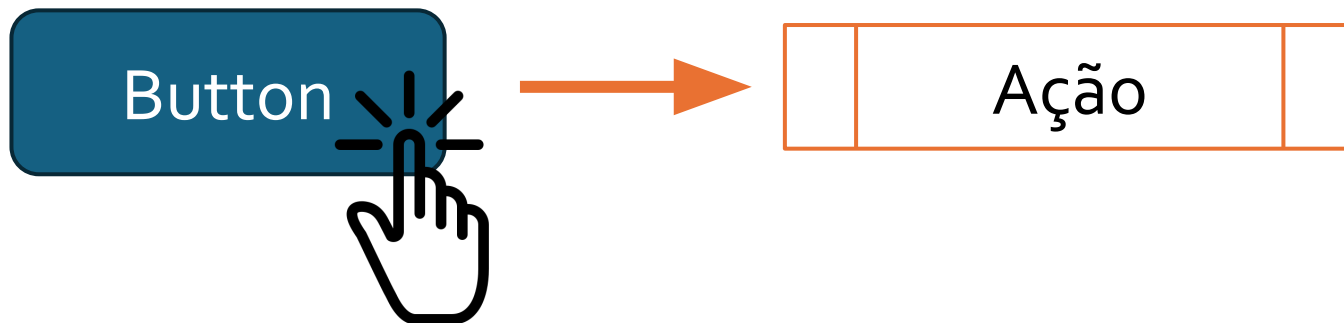
Adicionando interatividade

Alguns elementos na tela são atualizados de acordo com as ações do usuário. Por exemplo, ao clicar em uma galeria de imagens, a imagem exibida é alterada. No React, informações que mudam dinamicamente são chamadas de **estado**.

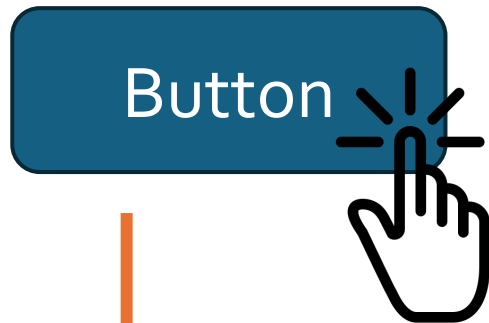
É possível adicionar estado a qualquer componente e modificá-lo conforme necessário.

Eventos

O React permite que você adicione **manipuladores de eventos** diretamente ao seu JSX. Esses manipuladores de eventos são **funções** que serão executadas em resposta às interações do usuário, como cliques, movimentos do mouse, foco em campos de formulário, entre outros.



Eventos



Nativo

Os componentes integrados do React, como **<button>**, oferecem suporte apenas a eventos de navegador nativos, como onClick



Personalizado

Criar seus próprios componentes personalizados e definir manipuladores de eventos com nomes específicos para as necessidades do seu aplicativo.

Eventos

Para adicionar um manipulador de eventos, você precisa primeiro definir uma função e, em seguida, passá-la como uma propriedade para o elemento JSX correspondente.

```
export default function Button() {  
  return <button>Não faço nada ainda</button>;  
}
```



```
export default function Button() {  
  function handleClick() {  
    alert("Você clicou em mim");  
  }  
  
  return <button onClick={handleClick}>Click me</button>;  
}
```

Eventos

Por convenção, os manipuladores de eventos geralmente são nomeados começando com **"handle"**, seguido pelo nome do evento. Isso ajuda a manter o código organizado e legível.

Exemplos comuns incluem:

- `onClick={handleClick}`
- `onMouseEnter={handleMouseEnter}`
- `onChange={handleChange}`

Seguir essa convenção torna o código mais fácil de entender.

```
<button onClick={() => {  
  alert('You clicked me!');  
}}>
```



CUIDADO

As funções usadas como manipuladores de **eventos** no React devem ser **passadas**, e não **chamadas** diretamente.

Passando uma função (correto)	Chamando uma função (incorreto)
<code><button onClick={handleClick}></code>	<code><button onClick={handleClick()}></code>

Se você deseja executar uma ação diretamente no JSX, pode usar uma **função anônima**.

Passando uma função (correto)	Chamando uma função (incorreto)
<code><button onClick={() => alert('...')}></code>	<code><button onClick={alert('...')}> </code>



CUIDADO

Vamos fazer um teste...

<https://react.dev/learn/adding-interactivity>

```
6   />
7   );
8 }
9
10 function Toolbar({ onPlayMovie, onUploadImage }) {
11   return (
12     <div>
```

'onPlayMovie' is missing in props validation eslint([react/prop-types](#))

(parameter) onPlayMovie: any

[View Problem \(Alt+F8\)](#) [Quick Fix... \(Ctrl+.\)](#)

É um
erro?

O ESLint **não interfere** diretamente no funcionamento do **React** ou do **JavaScript** em si; ele apenas fornece avisos ou erros durante o desenvolvimento para ajudar a seguir boas práticas e evitar bugs.

O **onPlayGame is missing in props validation**, é um aviso de linting gerado por uma regra específica do plugin do React no ESLint, que verifica se os seus componentes estão validando as props que recebem.

Sem ESLint

1. **O JavaScript é dinâmico:** O React não exige nativamente que você valide as props para que o código funcione.
2. **ESLint não é necessário para a execução:** O ESLint é apenas uma **ferramenta de análise estática de código**.
3. **A validação de props é opcional no React:** Embora a validação seja recomendada para evitar erros (como passar valores incorretos ou esquecer props obrigatórias), o React não força você a usá-la.

Os manipuladores de eventos podem ter efeitos colaterais?

Efeitos colaterais são mudanças feitas pelo código que **afetam algo fora da função onde ele está sendo executado.**

Sim! Os manipuladores de eventos são o melhor lugar para efeitos colaterais.

Exemplo: Alterar o valor de uma entrada em resposta à digitação ou alterar uma lista em resposta ao pressionamento de um botão.

No entanto, para alterar algumas informações, primeiro você precisa de alguma maneira de armazená-las. No React, isso é feito usando **STATE**, a memória de um componente.



Obrigado!
