

Desafio Prático: Aprimorando o `useState`

Agora que você já compreendeu como funciona o `useState`, chegou o momento de colocar seus conhecimentos em prática! Você vai expandir o mesmo projeto adicionando **novas funcionalidades** para treinar ainda mais o uso de estado no React.

Cada desafio abaixo vai exigir que você **pense na lógica do estado, crie funções e modifique a interface** para que o comportamento desejado aconteça.

Desafio 1: Dobrar o contador

Crie um **novo botão** que, ao ser clicado, **dobre o valor do contador** atual. Se o contador estiver em `5`, ele deve passar para `10`. Se estiver em `8`, ele deve ir para `16`, e assim por diante.

 *Dica:* Você precisará modificar o estado do contador usando `setContador`!

Desafio 2: Inverter o texto da mensagem


Adicione um **botão ao lado do campo de texto**. Quando o usuário clicar nesse botão, o texto digitado deve ser **invertido**. Por exemplo:

- Se ele digitar `"React"`, deve aparecer `"tcaeR"`.
- Se ele digitar `"useState"`, deve aparecer `"etatSesu"`.

 *Dica:* Pesquise como inverter uma string em JavaScript!

Desafio 3: Desabilitar o botão "Diminuir" quando o contador estiver em zero

Modifique o botão "Diminuir" para que ele **fique desativado** (`disabled`) sempre que o contador estiver em `0`. Isso evita que o número fique negativo.

 *Dica:* Você vai precisar verificar o valor do contador antes de permitir que ele diminua!

Desafio 4: Criar um input para definir o incremento do contador

Adicione um **novo campo de entrada** (`input`) onde o usuário pode digitar de **quanto em quanto** o contador deve aumentar.
Por exemplo:

- Se ele digitar `3`, ao clicar no botão "Aumentar", o contador deve crescer de `3` em `3`.
- Se ele digitar `10`, o contador deve crescer de `10` em `10`.

 *Dica:* Esse novo valor do incremento também pode ser guardado em um estado usando `useState`!

Finalizando

Com esses desafios, você vai aprofundar seu conhecimento sobre `useState`, manipulando estados de diferentes formas: números, strings e inputs dinâmicos.

Boa prática! 😊🎉