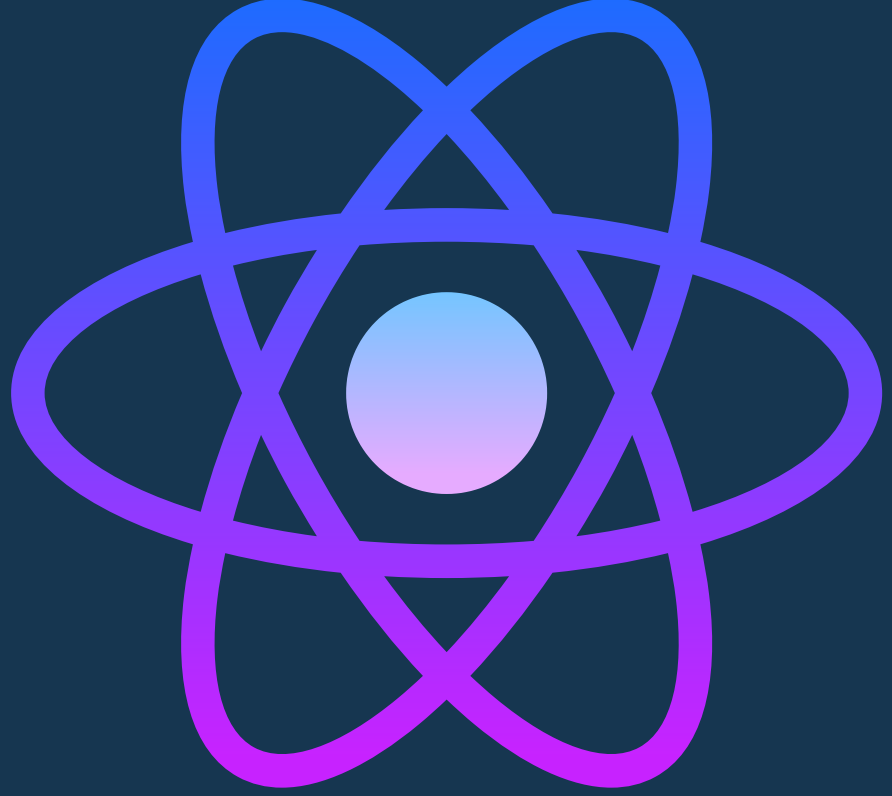
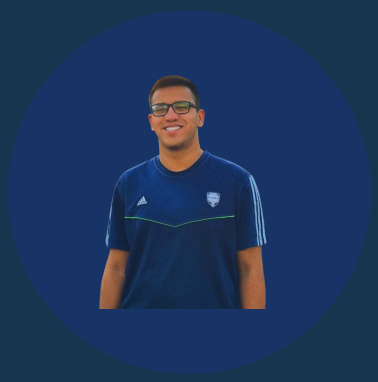
#ReactJS #Dicas #Desenvolvimento

PASSO-A-PASSO: O QUE É E COMO USAR useReducer?



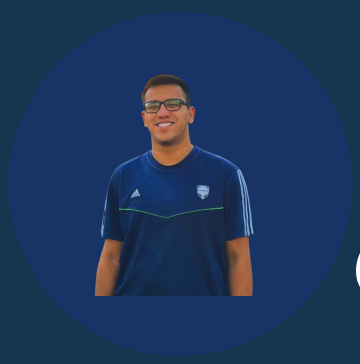


Victor Oliveira
@victoroliveira.dev

O que é o useReducer?

useReducer, assim como o useState, é uma alternativa para trabalharmos com estados da nossa aplicação. Usamos o useReducer quando temos uma lógica de estado um pouco mais complexa (como quando precisamos alterar vários estados ao mesmo tempo) ou quando a lógica de um estado, depende de um estado anterior.

Vamos entender como funciona na pratica:



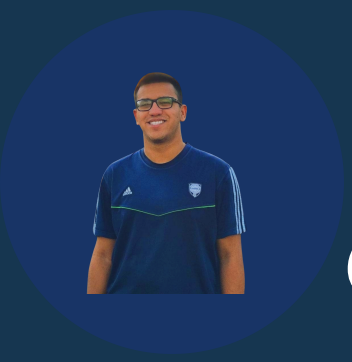
Victor Oliveira
@victoroliveira.dev

Antes de partimos para o exemplo, vamos entender como o Reducer é estruturado.
O useReducer, recebe dois argumentos:

1. Uma função reducer, que recebe um state (estado) e uma action (dispatch).
2. Recebe um estado inicial das informações que você quer tratar.

Ficando da seguinte maneira:

```
const [state, dispatch] = useReducer(reducer, initialState);
```

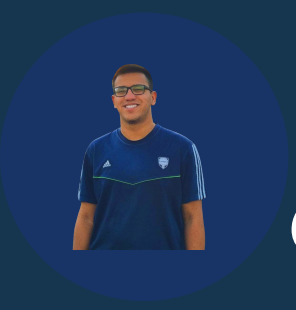


Victor Oliveira
@victoroliveira.dev

Show, agora vamos criar um simples botão que incrementa e decrementa uma contagem qualquer.
Vamos utilizar o useReducer, como no slide anterior e atribuir nossos onClicks aos botões:

```jsx
import React, {useReducer} from 'react'

function Counter() {
  const [state, dispatch] = useReducer(reducer, initialState);
  return (
    <>
      Count: {state.count}
      <button onClick={() => dispatch({type: 'decrement'})}>-</button>
      <button onClick={() => dispatch({type: 'increment'})}>+</button>
    </>
  );
}
```



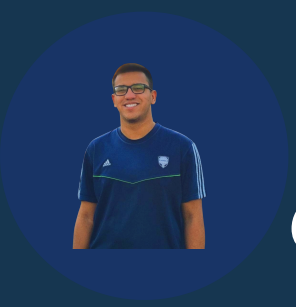
Victor Oliveira
@victoroliveira.dev

Por fim, basta criarmos as funções que passamos como parametros para nosso hook, não se esqueça que o reducer sempre deve retonar algo, que será nosso estado.

```
import React, {useReducer} from 'react'

const initialState = {count: 0};

function reducer(state, action) {
  switch (action.type) {
    case 'increment':
      return {count: state.count + 1};
    case 'decrement':
      return {count: state.count - 1};
    default:
      throw new Error();
  }
}
```



Victor Oliveira
@victoroliveira.dev

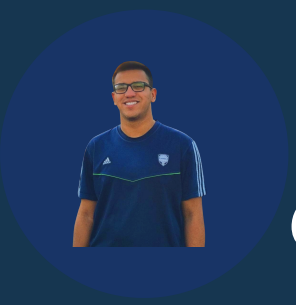
Boa! Simples não é ? Fique a vontade para implementar ainda mais estados ao nosso componente!

```jsx
import React, {useReducer} from 'react'

const initialState = {count: 0};

function reducer(state, action) {
  switch (action.type) {
    case 'increment':
      return {count: state.count + 1};
    case 'decrement':
      return {count: state.count - 1};
    default:
      throw new Error();
  }
}

function Counter() {
  const [state, dispatch] = useReducer(reducer, initialState);
  return (
    <>
      Count: {state.count}
      <button onClick={() => dispatch({type: 'decrement'})}>-</button>
      <button onClick={() => dispatch({type: 'increment'})}>+</button>
    </>
  );
}
```



Victor Oliveira
@victoroliveira.dev

Victor Oliveira

@victoroliveira.dev

CURTIU ? COMPARTILHE!