

# O que é Redux Toolkit? (RTK)

**Redux Toolkit (RTK)** é a forma **oficial e recomendada** de usar Redux hoje.

Ele resolve o maior problema do Redux "clássico":

- ➔ muita verbosidade
- ➔ muita repetição de código
- ➔ configuração confusa

RTK deixa tudo mais fácil, rápido e moderno.

---

## ✓ Vantagens do Redux Toolkit

### 1 Menos código

Com RTK, você cria **reducers + actions automaticamente** usando `createSlice()`.

### 2 Já vem configurado com:

- Redux DevTools
- Thunk (para requisições assíncronas)
- Middlewares úteis

### 3 Imutabilidade automática

Você pode escrever código "mutável", mas ele gera alterações imutáveis por baixo dos panos usando *Immer*.

## Exemplo usando Redux Toolkit

### 👉 Criando um Slice

```
import { createSlice } from "@reduxjs/toolkit";

const contadorSlice = createSlice({
  name: "contador",
  initialState: { valor: 0 },
  reducers: {
    incrementar: (state) => { state.valor += 1 },
```

```
    decrementar: (state) => { state.valor -= 1 },
    adicionar: (state, action) => { state.valor += action.payload },
  },
});
```

```
export const { incrementar, decrementar, adicionar } =
contadorSlice.actions;
export default contadorSlice.reducer;
```

### 👉 Criando a Store

```
import { configureStore } from "@reduxjs/toolkit";
import contadorReducer from "../contadorSlice";
```

```
export const store = configureStore({
  reducer: {
    contador: contadorReducer,
  },
});
```

### 👉 Usando no React

```
import { useSelector, useDispatch } from "react-redux";
import { incrementar } from "../contadorSlice";

function Contador() {
  const valor = useSelector((state) => state.contador.valor);
  const dispatch = useDispatch();

  return (
    <>
      <p>{valor}</p>
      <button onClick={() => dispatch(incrementar())}>+</button>
    </>
  );
}
```

## Qual usar: Redux “puro” ou Redux Toolkit?

Redux Tradicional	Redux Toolkit
Muito código	Pouco código
Verboso	Simples
Difícil para iniciantes	Recomendado oficialmente
Mais configuração manual	Já vem com tudo pronto

**Conclusão:**

👉 Se você for começar um projeto hoje, use **Redux Toolkit**.  
Redux puro só é usado em projetos antigos.

## Quando usar Redux / Redux Toolkit?

Use quando:

- Muitos componentes precisam do mesmo estado
- Há requisições assíncronas complexas
- Estados precisam ser previsíveis e auditáveis
- Aplicação grande (equipe grande)

Não use quando:

- A aplicação é pequena
- O React Context resolve o problema
- Você não precisa de logs, middlewares, etc.