

■ Switch case ainda vale a pena? Alternativas modernas em JavaScript

O switch/case sempre foi uma estrutura muito utilizada em JavaScript para lidar com múltiplas condições. Entretanto, com a evolução da linguagem e do ecossistema, surgiram alternativas mais elegantes, legíveis e fáceis de manter.

■ O clássico switch/case

```
function getStatusMessage(status) {  
    switch (status) {  
        case "loading":  
            return "Carregando...";  
        case "success":  
            return "Operação concluída!";  
        case "error":  
            return "Algo deu errado.";  
        default:  
            return "Status desconhecido.";  
    }  
}
```

Funciona, mas pode ficar verboso conforme os casos aumentam.

■ Alternativa 1: Objeto literal (lookup table)

```
const statusMessages = {  
    loading: "Carregando...",  
    success: "Operação concluída!",  
    error: "Algo deu errado.",  
};  
  
function getStatusMessage(status) {  
    return statusMessages[status] ?? "Status desconhecido.";  
}
```

■ Mais conciso ■ Fácil de estender ■ Ótimo para buscas diretas por string ou número

■ Alternativa 2: Funções em objeto literal

```
const actions = {  
    start: () => console.log("Iniciando..."),  
    stop: () => console.log("Parando..."),  
    pause: () => console.log("Pausando..."),  
};  
  
function execute(action) {  
    const fn = actions[action] ?? (() => console.log("Ação inválida"));  
    fn();  
}
```

■ Alternativa 3: Map

```
const actions = new Map([
  ["start", () => console.log("Iniciando...")],
  ["stop", () => console.log("Parando...")],
  ["pause", () => console.log("Pausando...")],
]);

function execute(action) {
  const fn = actions.get(action) ?? (() => console.log("Ação inválida"));
  fn();
}
```

■ Quando ainda usar switch/case?

Apesar das alternativas modernas, o switch ainda pode ser útil em alguns cenários:

- Casos muito pequenos (2 ou 3 opções).
- Quando há fall-through (mesmo código para vários casos).
- Situações em que clareza semântica importa mais do que concisão.

■ Conclusão

- Para tabelas de decisão grandes, prefira objetos literais ou Map.
- Para lógica curta, um if/else já basta.
- O switch/case continua válido, mas tende a ser substituído em código moderno por estruturas mais simples e diretas.

■ Em resumo: quanto mais declarativo e menos verboso for seu código, mais legível e sustentável ele será para o time.