

06

## Boas práticas de código

Entre as boas práticas de código, está a divisão do código de uma função grande em funções pequenas.

Pensando nisso, todas as afirmativas abaixo são verdadeiras, **exceto uma**. Qual?

**A**

Dividindo o código de uma função grande em funções pequenas, evitamos o problema das múltiplas responsabilidades em única grande função, afinal uma função só deve ter uma única responsabilidade.

**B**

O código fica mais fácil de testar, pois com diversas pequenas funções, é muito mais fácil testá-las individualmente em busca de erros.



 A legibilidade do código fica prejudicada, visto que temos de ler diversas funções, em vez de apenas uma.



Essa afirmativa é falsa! Quebrar o código em pequenas partes **aumenta** a legibilidade. Mas é preciso tomar cuidado com a criação de funções, pois criar funções desnecessariamente pode aumentar a complexidade do código.

**D**

Melhora a manutenção do código, já que é mais fácil você cuidar de vários pequenos blocos simples do que um grande bloco complexo.



Sabemos que quebrar uma grande função complexa em pequenas funções é uma boa prática por causa de diversos fatores, mas podemos citar como os principais deles:

- Dar manutenção ao código fica muito mais fácil, visto que agora podemos examinar vários pequenos blocos, que são muito mais fáceis de compreender do que um grande bloco de código.
- Ao quebrar uma grande função, também estamos deixando ela com menos responsabilidades, com a meta de atingir o ideal de que cada função tenha apenas uma única responsabilidade.
- O código também fica muito mais fácil de testar, pois se temos diversas funções pequenas, conseguimos testar uma a uma em busca de erros no código.
- E por último, a legibilidade do código aumenta muito, pois dando nomes semânticos a cada uma das funções menores, conseguimos deixar bem claro o que aquela parte do código deve fazer e facilita o entendimento do todo como um geral.

[PRÓXIMA ATIVIDADE](#)