

# Exercício - Teste

**Questão 1.** Crie um teste unitário para uma função `multiplicar(a, b)` que retorna o produto de dois números. Garanta que a função retorne os resultados corretos para valores positivos, negativos e zero.

**Questão 2.** Usando o TDD, Teste uma função `filtrarPares(arr)` que recebe um array de números e retorna um novo array contendo apenas os números pares. Verifique se o array retornado contém apenas números pares e se arrays vazios retornam outro array vazio.

**Questão 3.** Usando o Cypress ou Selenium, crie um teste E2E que abra o Google, pesquise o termo "JavaScript testing", e verifique se o primeiro link da pesquisa contém

**Questão 4.** Crie um teste de benchmark para medir o tempo de execução de uma função recursiva `fibonacci(n)` que retorna o n-ésimo número da sequência de Fibonacci. Teste o desempenho para valores de `n = 10`, `n = 20` e `n = 30`, garantindo que o tempo de execução seja razoável (por exemplo, abaixo de 100ms para `n = 30`).

```
function fibonacci(n) {  
  if (n <= 1) {  
    return n;  
  }  
  return fibonacci(n - 1) + fibonacci(n - 2);  
}
```

**Questão 5.** Usando a API <https://github.com/yanncarvalho/NoteBookManager>, crie um teste que verifica se qualquer um dos endpoints está retornando o valor esperado, conforme documentação swagger.