

# Testes com Python

## Ideias chave

O testing de software ou GQ de software é uma disciplina em engenharia de software que permite ter processos de execução de um programa ou aplicação e uma metodologia de trabalho com o objetivo de localizar erros de software.

O processo de testing envolve:

- Economize tempo e esforço.
- Verificar se os erros do passado não se repetem.
- Verificar se o produto é funcional para o usuário.
- Verificar se não há nenhuma funcionalidade quebrada na 'build'.

Existem diferentes tipos de testes, dependendo do aspecto particular de nosso programa que queremos testar:

- Testes unitários
- testes de integração
- teste de ponta a ponta (E2E)
- testes de aceitação
- teste da caixa branca
- teste da caixa preta
- teste da caixa cinza
- testes manuais
- testes estáticos
- testes dinâmicos
- teste da interface do usuário
- teste de fumaça
- teste de regressão
- teste de carga
- teste de inserção

Os testes unitários realizam testes em partes do código de forma independente.

Os testes de integração se tornam necessários quando queremos testar sistematicamente um projeto como um todo.

Python inclui bibliotecas nativas para testing, outras bibliotecas de terceiros incluem:

- Robot
- PyTest
- Teste de unidade
- DocTest
- Nariz2
- Testify
- Unittest

Com unittest vamos fazer um processo de automatização da verificação do nosso código.

No unittest não só temos que verificar se a função está retornando os resultados corretos, como também temos que nos certificar de que as exceções sejam disparadas corretamente de acordo com os casos. Também podemos verificar se o tipo de dados que a função está recebendo por parâmetro está correto ou não.

Estruturas de teste como o PyTest podem trabalhar diretamente com afirmações para formar UnitTests totalmente funcionais.