



Teste de Software

Professor: Euclides Paim
euclidespaim@gmail.com



Testes Unitários

Professor: Euclides Paim
euclidespaim@gmail.com



Teste de Software

Testes Unitários

Sumário

- Objetivos
- Conceitos Fundamentais
- Exemplos
- Etapas
- Aplicação





Teste de Software

Testes Unitários

Objetivos da Aula

- Compreender o que são testes unitários.
- Entender a importância dos testes unitários no ciclo de desenvolvimento.
- Aplicar testes unitários simples com exemplos em Python.

.



Teste de Software

Testes Unitários

Recapitulando o que já vimos

- Fundamentos: garantir qualidade, detectar erros cedo, reduzir custo.
- Aspectos psicológicos: viés do desenvolvedor, resistência a testes.
- Mandamentos do testador: pensar como o usuário, buscar falhas, repetir testes.
- Tipos de testes: caixa branca/preta, unitário, integração, sistema, aceitação etc.

.



Teste de Software

Testes Unitários

O que é um Teste Unitário?

É um teste que verifica se **uma unidade isolada de código** (como uma função) está funcionando corretamente.



Teste de Software

Testes Unitários

Unidade de código?

- Funções
- Métodos
- Classes





Teste de Software

Testes Unitários

Qual o objetivo do teste unitário?

- Garantir que **cada parte isolada do código** funciona como esperado, antes de juntar tudo (integração), testamos as peças individualmente.



Teste de Software

Testes Unitários

Por que fazer teste unitário?

- Detecta erros cedo
- Facilita manutenção e refatoração
- Ajuda a entender o comportamento do código
- Garante que o código continua funcionando após mudanças



Teste de Software

Testes Unitários

Características dos testes unitários

- ✓ Simples
- ✓ Rápidos
- ✓ Repetíveis
- ✓ Independentes
- ✓ Automatizados



Teste de Software

Testes Unitários

Quando testar unidades?

- Idealmente, **logo após escrever a função.**
- Também depois de alterar/refatorar código.





Teste de Software

Testes Unitários

Boas práticas ao testar unidades

- Teste **um comportamento por vez**
- Testes devem ser **claros e curtos**
- Evite depender de banco de dados ou internet
- Nomeie os testes de forma descritiva





Teste de Software

Testes Unitários

Exemplo de unidade: função `soma`

```
def soma(a, b):  
    return a + b
```

Essa é uma **unidade de código** que pode ser testada isoladamente.



Teste de Software

Testes Unitários

O que testar na função **soma**?

- ☒ Soma de dois positivos
- ☒ Soma com zero
- ☒ Soma de número negativo
- ☐ Soma com texto (erro esperado?)

```
def soma(a, b):  
    return a + b
```



Teste de Software

Testes Unitários

Ferramentas para teste unitário

 Em Python:

- `assert` (teste simples e direto)
- `unittest` (módulo da linguagem)
- `pytest` (mais avançado, para projetos maiores)



Teste de Software

Testes Unitários

Estrutura de um teste unitário

1. Entrada (dados de teste)
2. Ação (executa a função)
3. Verificação (compara com o resultado esperado)



Teste de Software

Testes Unitários

Exemplo dessa estrutura:

```
entrada = (2, 3)          # entrada
resultado = soma(*entrada) # ação
assert resultado == 5      # verificação
```



Referências

Referências Básicas

CORMEN, Thomas H.; LEISERSON, Charles E.; RIVEST, Ronald L.; STEIN, Clifford. ***Algoritmos: teoria e prática***. 3. ed. Rio de Janeiro: Elsevier, 2013.

SEBESTA, Robert W. ***Conceitos de linguagens de programação***. 10. ed. São Paulo: Pearson, 2018.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. ***Algoritmos: lógica para desenvolvimento de programação de computadores***. 27. ed. São Paulo: Érica, 2016.

DOWNEY, Allen B. ***Pense em Python: como pensar como um cientista da computação***. Tradução de Cássio de Souza Costa. 2. ed. São Paulo: Novatec, 2016.

Referências Complementares

IEPSEN, Edécio Fernando. ***Lógica de programação e algoritmos com JavaScript***. 2. ed. São Paulo: Novatec, 2022.

Referências na Internet

<https://www.w3schools.com/python/>