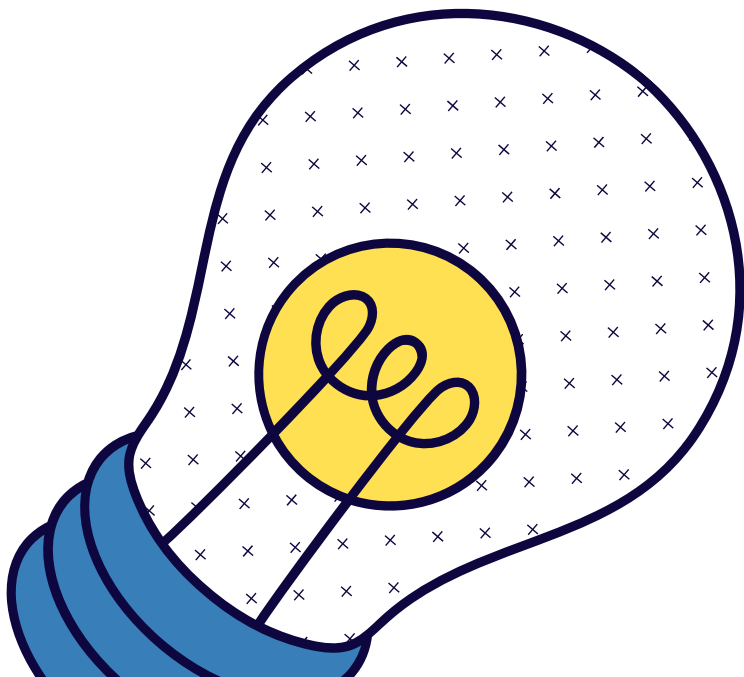




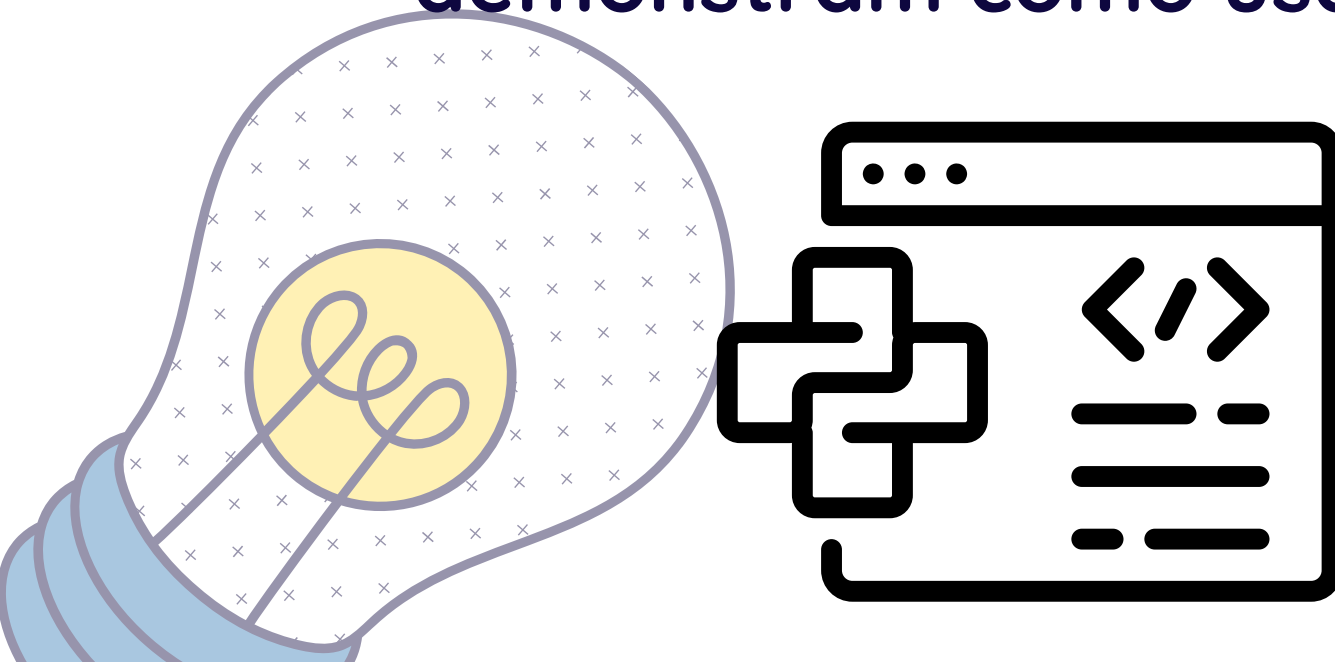
Testes Unitários com Pytest:

Garantindo a Qualidade do Seu Código



Por que Realizar Testes Unitários?

- **Código Confiável:** Garantem que cada parte do seu software funcione como esperado.
- **Detecção Precoce de Erros:** Encontre problemas no início do desenvolvimento, quando são mais fáceis de corrigir.
- **Refatoração Segura:** Faça mudanças no código sem medo de quebrar algo.
- **Documentação Viva:** Seus testes demonstram como usar seu código.



Pytest: Seu Aliado nos Testes

Se você, assim como eu, trabalha com o Python, uma ótima ferramenta para testes unitários é o PyTest.
Mas porque usar ele?

- **Simplicidade:** Escreva testes de forma clara e concisa.
- **Flexibilidade:** Adapta-se a diferentes estilos de teste.
- **Poder:** Recursos avançados para cenários complexos.

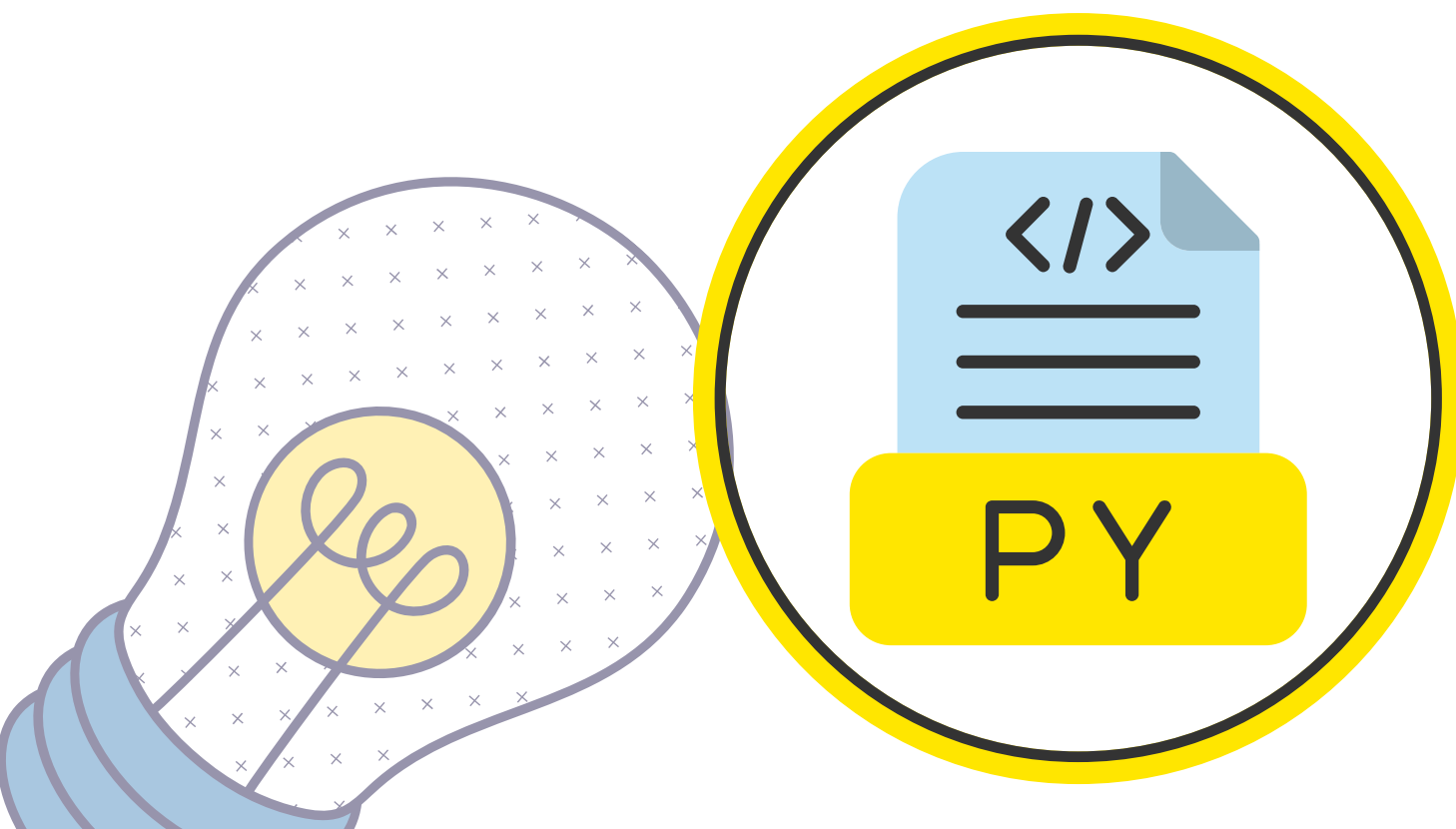


Como Escrever Testes com Pytest

1. Estrutura:

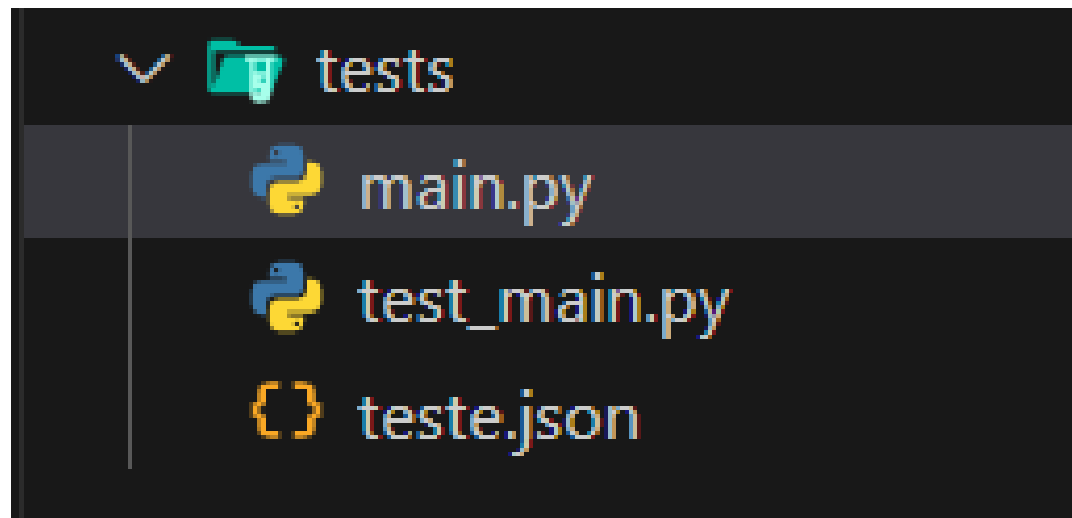
- a. Crie um arquivo com o nome `test_nome_do_seu_modulo.py`.
- b. Defina funções de teste com o prefixo `test_`.

Veja um exemplo a seguir:



Como Escrever Testes com Pytest

Aqui temos o main.py, logo, o arquivo de teste irá chamar “test_main.py”

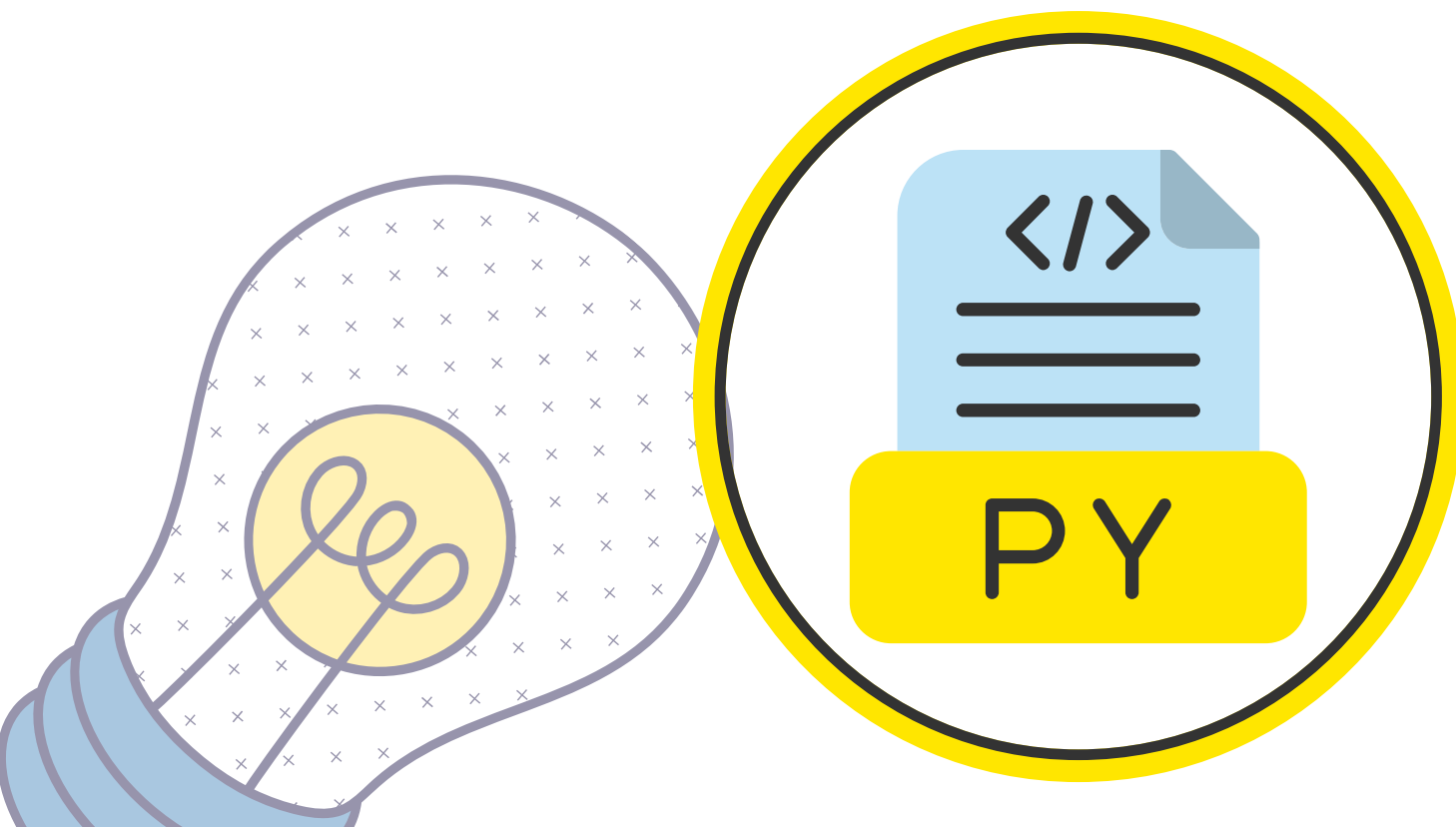


Como Escrever Testes com Pytest

1. Assertions:

- a. Use assert para verificar se o resultado esperado é igual ao obtido.

Veja mais um exemplo a seguir:



Como Escrever Testes com Pytest

Meu main.py é uma automação que analisa uma resposta .json e verifica se á duplicadas, desejo verificar se a seguinte função está funcionando como deve:

```
1  #...
2  def encontrar_duplicados(json_data, chave):
3      duplicados = {}
4      for item in json_data:
5          if chave in item: # Verifica se a chave existe no item
6              valor_chave = item[chave]
7              if valor_chave in duplicados:
8                  duplicados[valor_chave].append(item)
9              else:
10                 duplicados[valor_chave] = [item]
11     return {chave: itens for chave, itens in duplicados.items() if len(itens) > 1}
12  #...
```



Como Escrever Testes com Pytest

Para isso, no meu arquivo `test_main.py`, criei uma função, com mocked data, que validar se minha função irá mesmo encontrar duplicadas.

```
1  #...
2  def test_encontrar_duplicados():
3      # Dados de teste:
4      dados_teste = {
5          "data": [
6              {"id": 1, "nome": "Alice"},
7              {"id": 2, "nome": "Bob"},
8              {"id": 3, "nome": "Alice"},
9              {"id": 4, "nome": "Charlie"},
10             {"id": 5, "nome": "Bob"}
11         ]
12     }
13
14     # Resultado esperado:
15     resultado_esperado = {
16         "Alice": [
17             {"id": 1, "nome": "Alice"},
18             {"id": 3, "nome": "Alice"}
19         ],
20         "Bob": [
21             {"id": 2, "nome": "Bob"},
22             {"id": 5, "nome": "Bob"}
23         ]
24     }
25
26
27     # Execução da função:
28     duplicados = encontrar_duplicados(dados_teste["data"], "nome")
29
30     # Verificações (asserções):
31     assert duplicados == resultado_esperado
32     #...
```

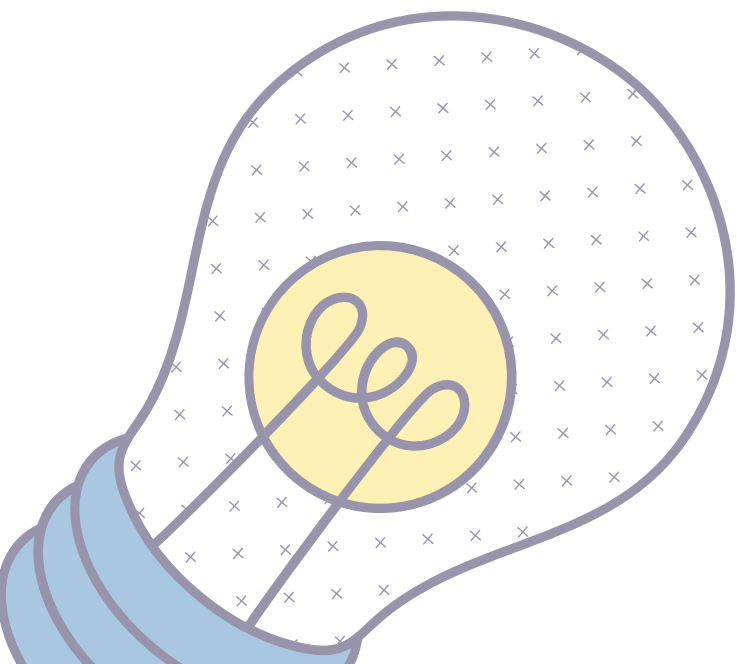

Como Escrever Testes com Pytest

1. Executando Testes

a. No terminal, execute: `pytest`

```
Desktop\verifica duplicatas no json> pytest
```

Ao Executar, basta aguardar. O PyTest irá executar os teste, e te informar o resultado, conforme os exemplos a seguir:



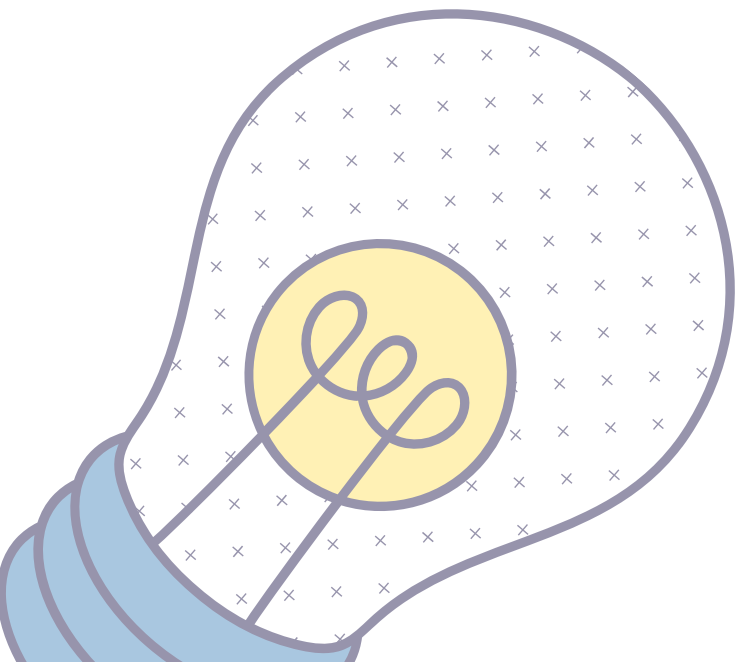
Como Escrever Testes com Pytest

Testes concluídos com sucesso:

```
PS C:\Users\Thalles\Desktop\verifica duplicatas no json> pytest
===== test session starts =====
platform win32 -- Python 3.11.3, pytest-8.2.2, pluggy-1.5.0
rootdir: C:\Users\Thalles\Desktop\verifica duplicatas no json
collected 3 items

tests\test_main.py ... [100%]

===== 3 passed in 0.03s =====
PS C:\Users\Thalles\Desktop\verifica duplicatas no json> |
```



Como Escrever Testes com Pytest

Testes concluídos sem sucesso:

```
PS C:\Users\Thalles\Desktop\verifica duplicatas no json> pytest
===== test session starts =====
platform win32 -- Python 3.11.3, pytest-8.2.2, pluggy-1.5.0
rootdir: C:\Users\Thalles\Desktop\verifica duplicatas no json
collected 3 items

tests\test_main.py F.. [100%]

===== FAILURES =====
_____ test_encontrar_duplicados _____

def test_encontrar_duplicados():
    # Dados de teste:
    dados_teste = {
        "data": [
            {"id": 1, "nome": "Alice"},
            {"id": 2, "nome": "Bob"},
            {"id": 3, "nome": "Charlie"},
            {"id": 4, "nome": "Charlie"},
            {"id": 5, "nome": "Bob"}
        ]
    }

    # Resultado esperado:
    resultado_esperado = {
        "Alice": [
            {"id": 1, "nome": "Alice"},
            {"id": 3, "nome": "Alice"}
        ],
        "Bob": [
            {"id": 2, "nome": "Bob"},
            {"id": 5, "nome": "Bob"}
        ]
    }

    # Execução da função:
    duplicados = encontrar_duplicados(dados_teste["data"], "nome")

    # Verificações (asserções):
    > assert duplicados == resultado_esperado
E   AssertionError: assert {'Bob': [{'id...: 'Charlie'}]} == {'Alice': [{'...ome': 'Bob'}]}
E
E   Omitting 1 identical items, use -vv to show
E   Left contains 1 more item:
E   {'Charlie': [{'id': 3, 'nome': 'Charlie'}, {'id': 4, 'nome': 'Charlie'}]}
E   Right contains 1 more item:
E   {'Alice': [{'id': 1, 'nome': 'Alice'}, {'id': 3, 'nome': 'Alice'}]}
E   Use -v to get more diff

tests\test_main.py:34: AssertionError

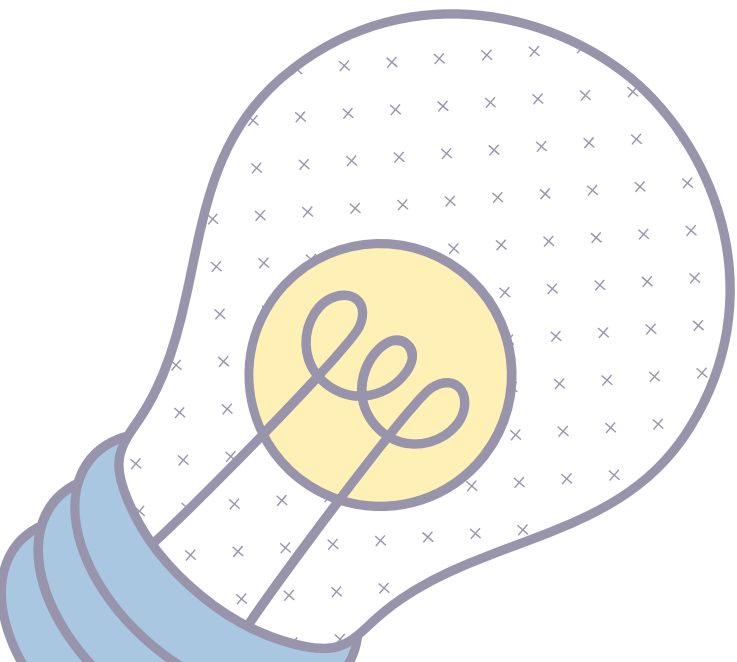
===== short test summary info =====
FAILED tests/test_main.py::test_encontrar_duplicados - AssertionError: assert {'Bob': [{'id...: 'Charlie'}]} == {'Alice': [{'...ome': 'Bob'}]}
===== 1 failed, 2 passed in 0.18s =====
PS C:\Users\Thalles\Desktop\verifica duplicatas no json>
```

Dicas Para os Testes

Dicas:

- Mantenha os testes simples e focados.
- Teste um comportamento por função.
- Use nomes descritivos para seus testes.
- Execute seus testes regularmente.
- Integração Contínua:
 - Automatize seus testes para rodar a cada mudança no código.

Com testes unitários e Pytest, você garante a qualidade do seu código e entrega um software mais confiável e fácil de manter.



FIM!

Gostou do conteúdo??

Deixa o like e me segue!!! 😊😊

NÃO ESQUEÇA DE COMPARTILHAR! 😊💧

Também me siga no insta!
[@thalles_webdeveloper](#)

Obrigado pela atenção 🦏😊

