



python

# Testes unitários automatizados





# O que são testes unitários

- Testes unitários, como o nome sugere, devem testar unidades de trabalho isoladas, afim de mostrar que funcionam individualmente.

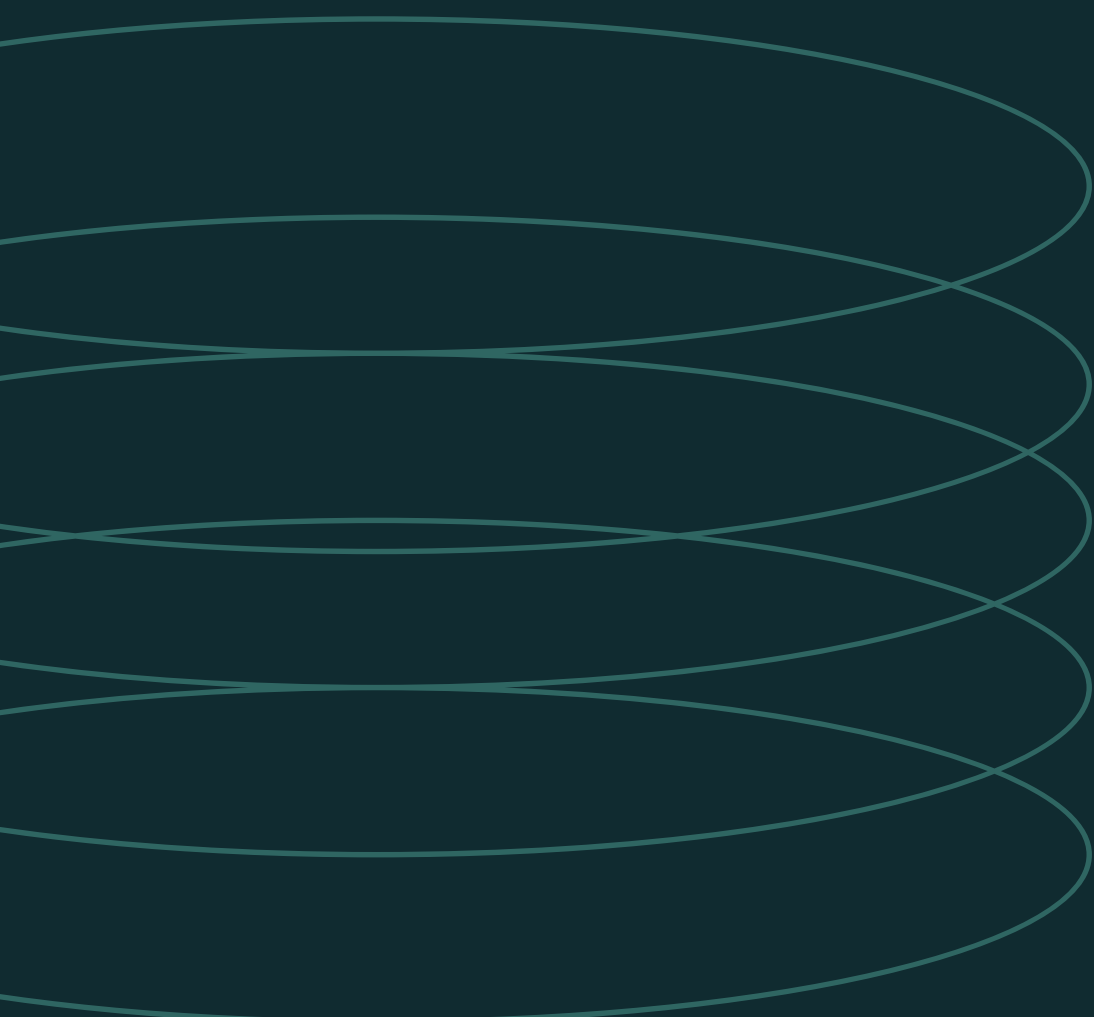




# Vantagens Teste unitário

- Previne contra o aparecimento de 'bugs' oriundos de códigos mal escritos;
- Código testado é mais confiável;
- Permite alterações sem medo (coragem);
- Testa situações de sucesso e de falha;
- Garantir maior manutenção do código.

# Teste automatizado X manual



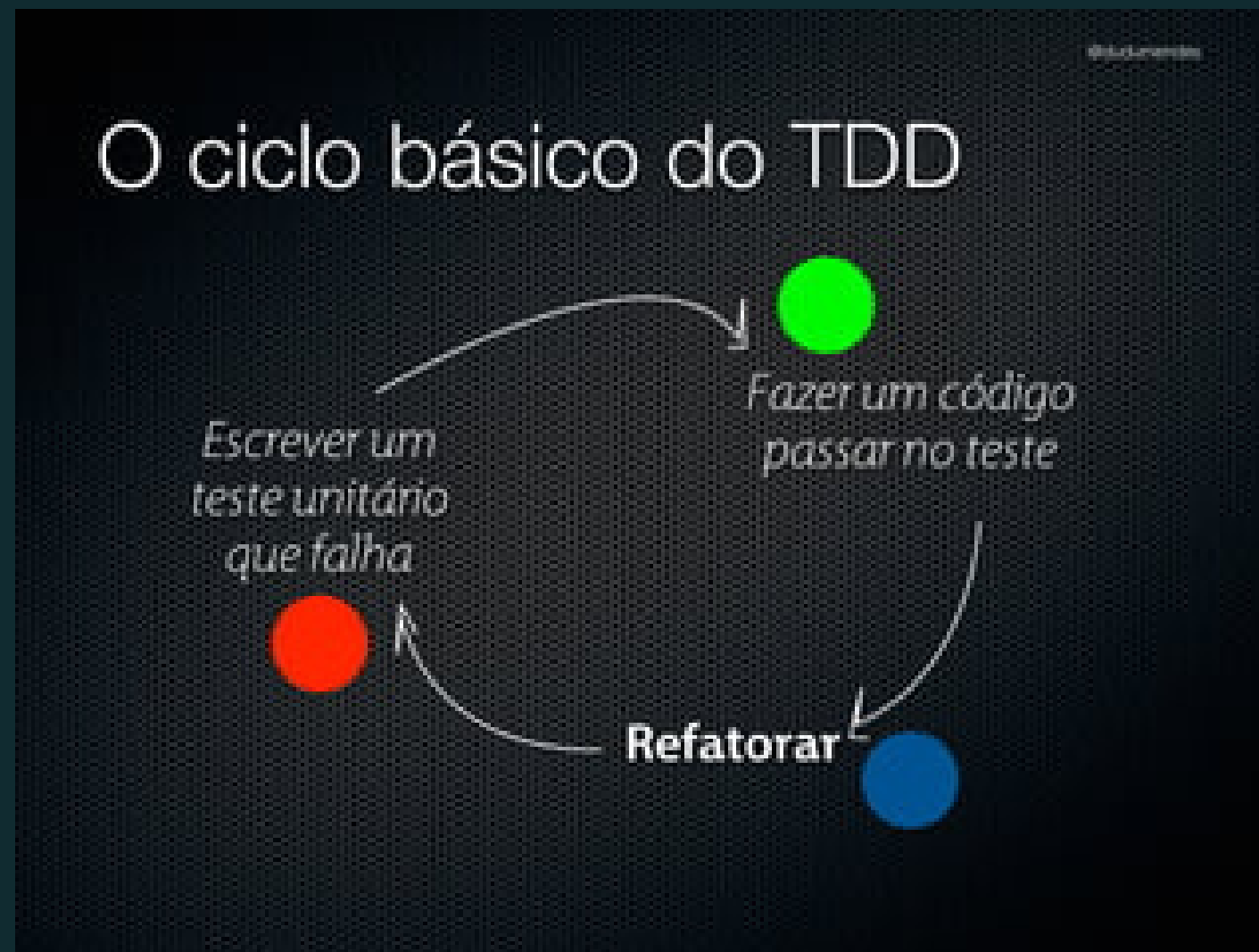
Existem duas maneiras de realizar testes em python - testes automatizados e testes manuais. No teste manual, o código é testado manualmente por um humano. O testador precisa fazer listas manualmente de todas as entradas possíveis e da saída esperada. É um trabalho tedioso e também demorado.

É aqui que os testes automatizados entram em jogo. O python consiste em estruturas integradas que facilitam o teste de código. Como o teste automatizado é baseado em script, ele é preciso. Python tem ferramentas e bibliotecas - como unittest e pytest

# TDD

## (Teste-Driven Development)

Desenvolvimento orientado a testes





# Qual é melhor - pytest ou unittest?

- Embora ambas as estruturas sejam ótimas para realizar testes em python, o pytest é mais fácil de trabalhar. O código em pytest é simples, compacto e eficiente.
- Para unittest, teremos que importar módulos, criar uma classe e definir as funções de teste dentro dessa classe. Mas para o pytest, só precisamos definir a função de teste. O Pytest também é rápido e eficiente.



O *pytest* é uma framework de teste para **python** que provê soluções para executar testes e fazer validações diversas, com a possibilidade de estender com plugins e até rodar testes do próprio *unittest* do python.

- Link
  - <https://docs.pytest.org/en/7.3.x/>

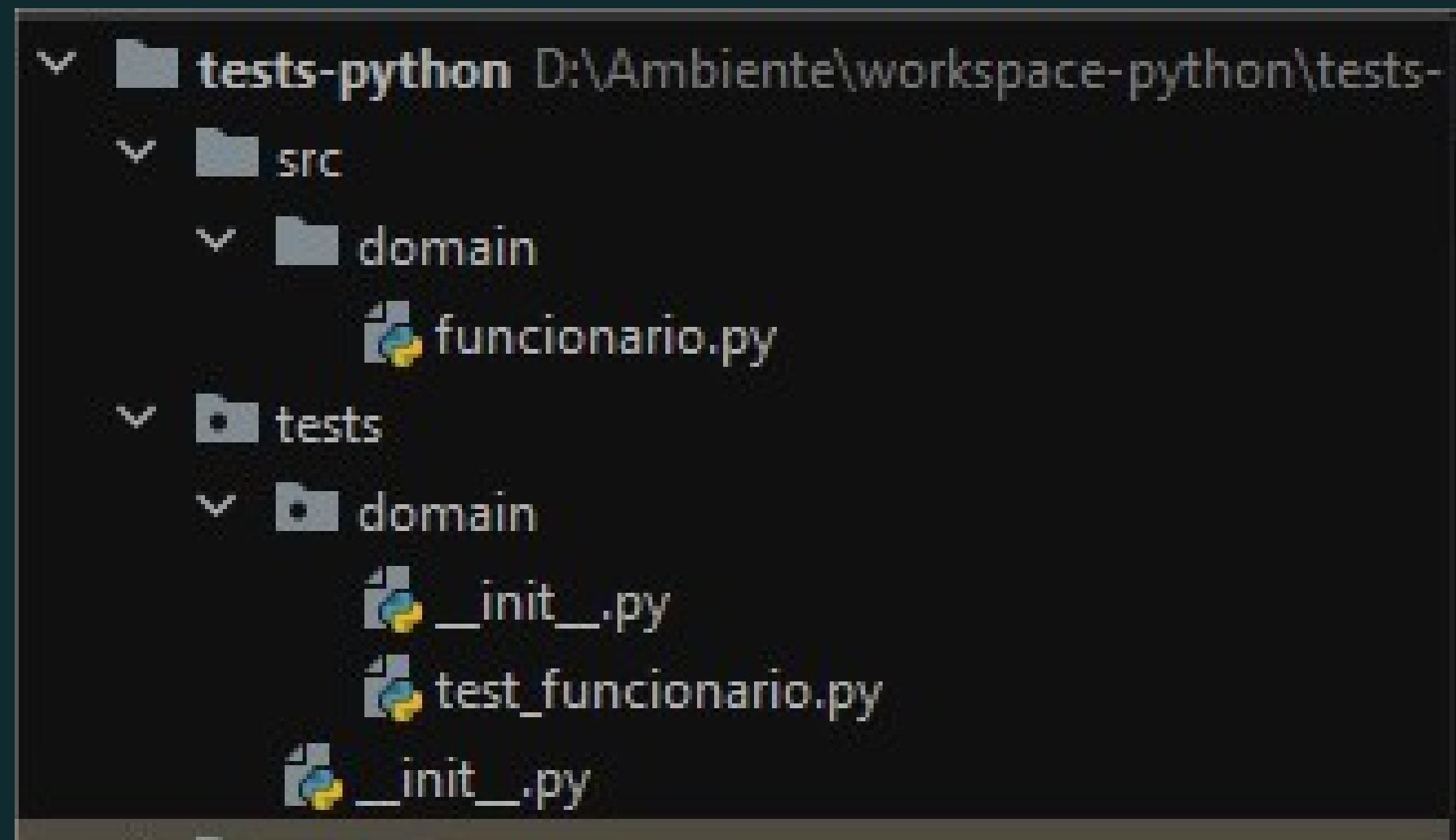
# Entendendo como `pytest` funciona:

- Quando você executa o comando `pytest` dentro do seu ambiente virtual python, ele vai fazer um scan nos diretórios e subdiretórios do seu repositório procurando por arquivos que respeitem o formato de nomenclatura `test_*.py` ou `*_test.py`.
- Como o processo de descoberta é amplo e existe mais de uma forma de chamar o `pytest` para rodar (tanto comando direto `pytest` mencionado anteriormente, quanto com o comando `python -m pytest`), é altamente recomendado usar o padrão `__init__.py` em cada diretório para o `pytest` reconhecê-los como módulos e evitar colisão de nome nos arquivo de teste.



# Exemplo estrutura de pastas:

- Na raiz do projeto crie a pasta tests e adicione um arquivo em branco `__init__.py`
- Crie também a pasta domain com dois arquivos `__init__.py` ( em branco ) e o arquivo `test_funcionario.py`





# Como instalar pytest:

- Link:
  - <https://pypi.org/project/pytest/>

```
pip install pytest
```

# Exemplo unittest:

- usando unittest biblioteca nativa do próprio python
- link
  - <https://docs.python.org/3/library/unittest.html>

```
from unittest import main, TestCase

def square(x):
    return x ** 2

class TestSquare(TestCase):
    def test_if_returns_square_of_2(self):
        result = square(2)
        expected = 4

        self.assertEqual(result, expected)

if __name__ == '__main__':
    main()
```

# Exemplo pytest:

You, há 1 segundo | 1 author (You)

```
from fastapi import status
from fastapi.testclient import TestClient
from src.main import app, auth
```

You, há 1 segundo • Uncommitted changes

```
client=TestClient(app)
```

```
def test_index_returns_correct():
    response = client.get('/')
    assert response.status_code == status.HTTP_200_OK
    assert response.json() == {"message": "service GOALS OK"}
```





# Links pesquisados:

- <https://medium.com/assertqualityassurance/tutorial-de-pytest-para-iniciantes-cbdd81c6d761>
- [https://engenharia.pagseguro.com.br/post/escrevendo\\_testes\\_com\\_pytest\\_e\\_mock/index.html](https://engenharia.pagseguro.com.br/post/escrevendo_testes_com_pytest_e_mock/index.html)
- <https://www.devmedia.com.br/test-driven-development-tdd-simples-e-pratico/18533>
- <https://vepo.medium.com/tdd-na-pr%C3%A1tica-1362fe5bfd43>
- <https://www.devmedia.com.br/teste-unitario-com-pyunit/41233>
- <https://pt.linux-console.net/?p=5939#gsc.tab=0>