

# Testes de Integração de API Rest



# Rafael Souza da Silva

 Backend Developer

- Graduado em Análise e Desenvolvimento de Sistemas
- 8 Anos trabalhando com desenvolvimento WEB

 <https://linkedin.com/in/arkanister>

 <https://github.com/arkanister>

# Repositório

```
$ git clone git@github.com:arkanister/pybr2020-integration-tests-django.git
```

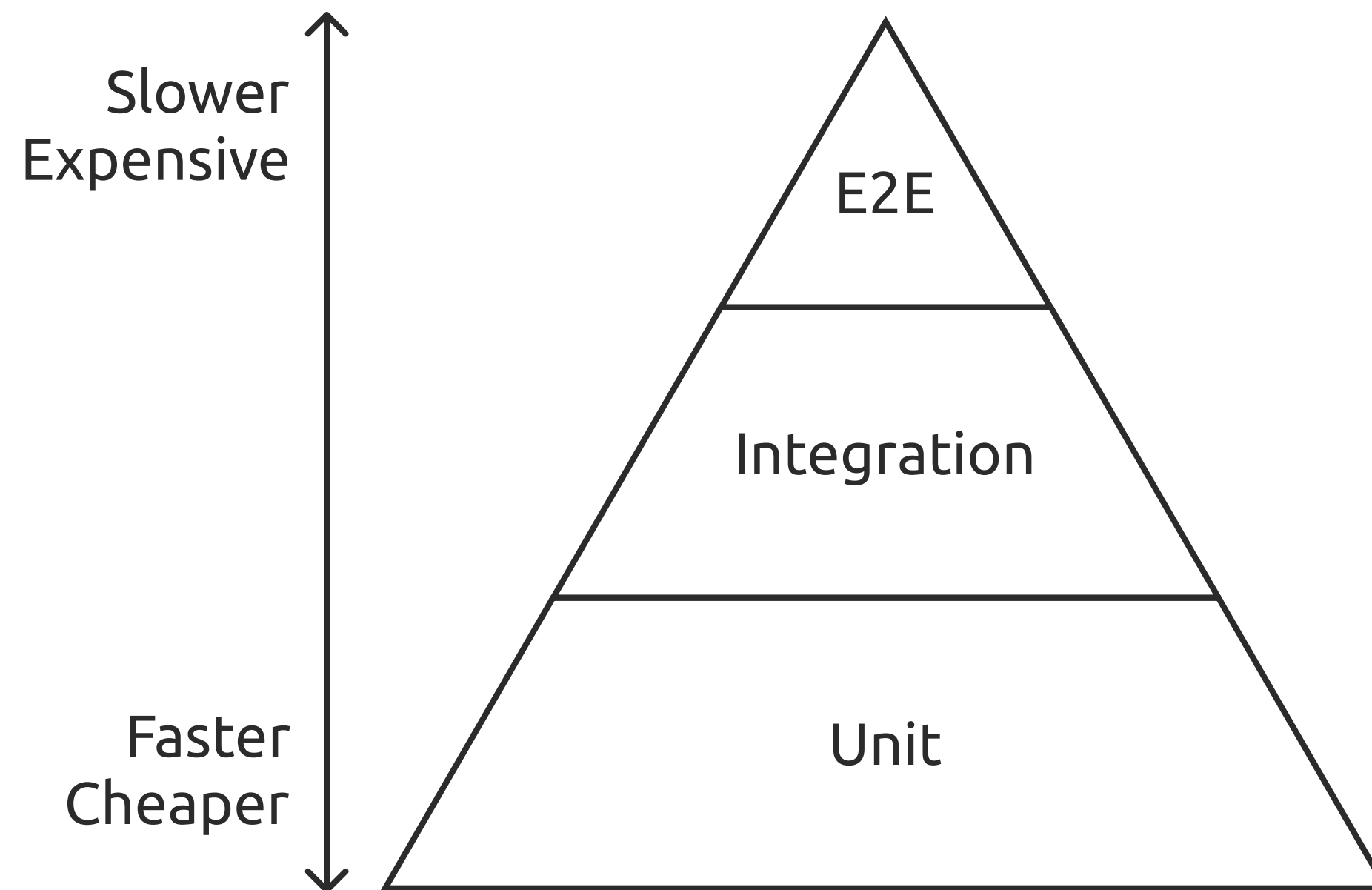
# Princípios

- Deve ser integrado ao ambiente de desenvolvimento.
- Deve rodar o mais rápido possível.
- Deve prover o maior valor com menor esforço.
- Deve prover manentabilidade.

# Por Que?

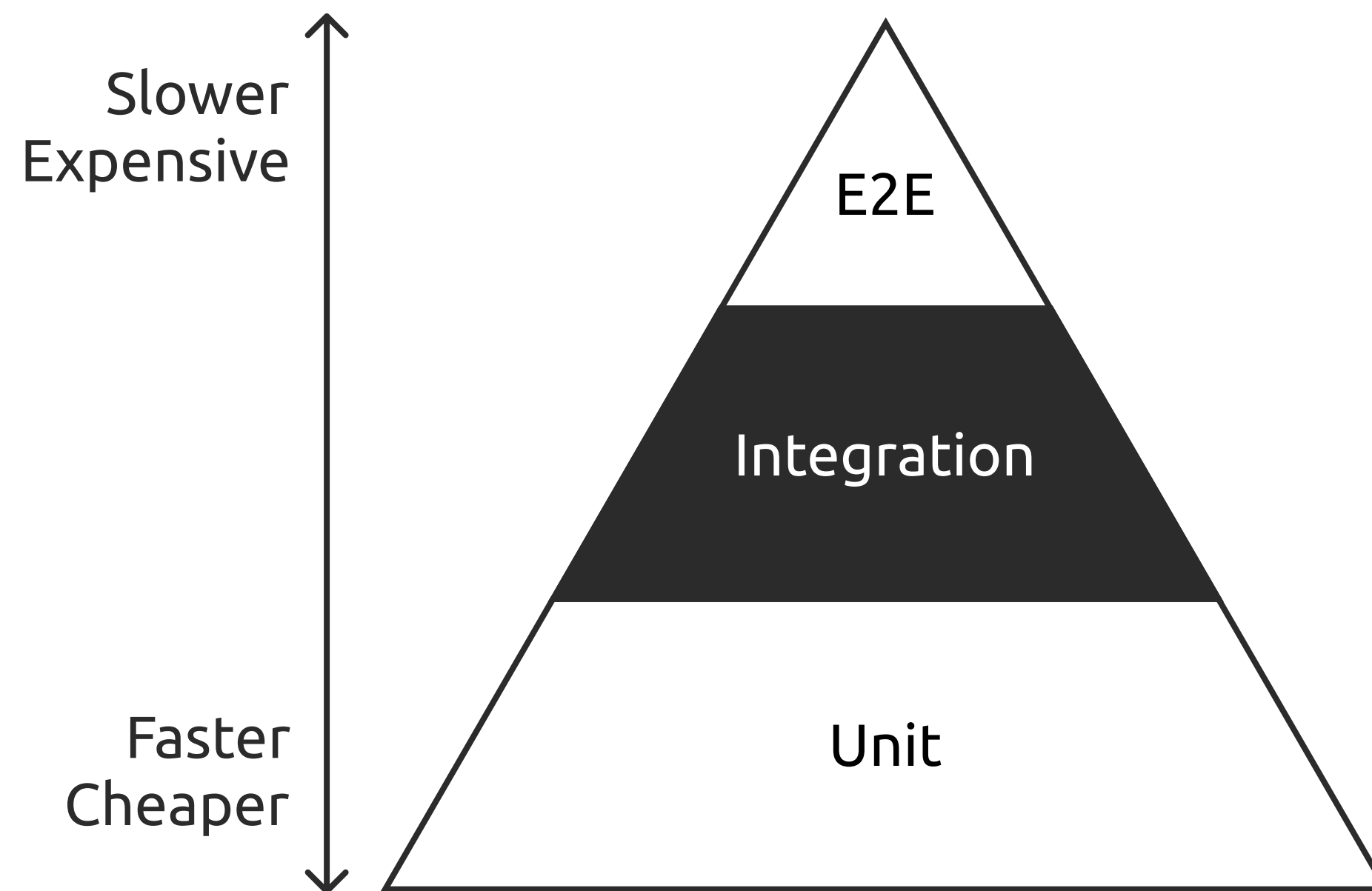
- Garantia de código.
- Prevenir bugs.
- Melhora a manutenção da aplicação.

# O Que?



<https://martinfowler.com/bliki/TestPyramid.html>

# O Que?



<https://martinfowler.com/bliki/TestPyramid.html>

# O Que?

- Unidade de comportamento.
- Partes mais Importantes do código.
- Partes que trazem maior valor para o projeto.



# Como?

## Arrange

## Act

## Assert

```
class Operator:

    def sum(a, b):
        """ Returns the sum of two numbers """
        return a + b

def test_should_sum_two_numbers():

    # Arrange
    operator = Operator()

    # Act
    result = operator.sum(1, 1)

    # Assert
    assert 2 == result, "%d != %d" % (result, 2)
```

# Como?

A nomenclatura deve refletir exatamente o objetivo do teste.

```
def test_should_sum_two_numbers():  
    ...
```

```
def test_should_not_sum_strings():  
    ...
```

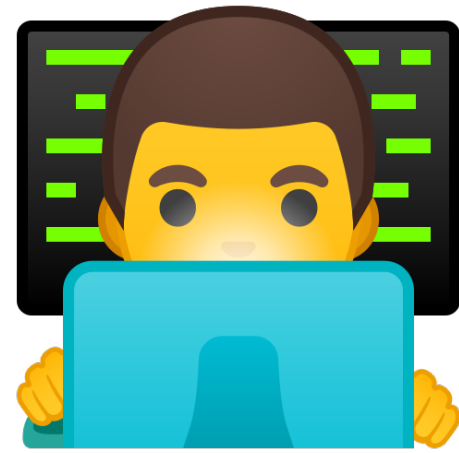
```
def test_should_retrieve_by_category():  
    ...
```

# Como?

Cada teste deve ter apenas um objetivo.

Multiplos blocos de ação devem ser evitados.

```
class NoteAPITestCase(APITestCase):  
    def test_should_list(self):  
        ...  
  
    def test_should_retrieve(self):  
        ...
```



Let's  
Code