

The background of the slide is decorated with various hand-drawn illustrations of fruits and vegetables. These include a blueberry cluster in the top left, an orange slice at the top center, a green leaf at the top center-right, a watermelon slice at the top right, a lime at the top right, a lemon at the middle left, a green leaf at the middle left, an avocado at the middle right, a lemon slice at the middle right, a strawberry at the bottom left, a banana at the bottom center, a lime slice at the bottom center-right, an orange at the bottom right, and a cherry at the bottom right. The text is centered in the middle of the slide.

Saindo do zero
em testes



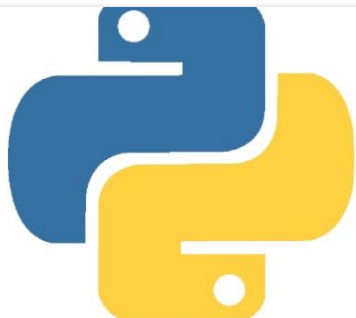
Antes de tudo



Olar bebês!

Sou Eduardo Mendes (@dunossauro)

Apto fazedor de lambdas, pythonista, apaixonado por software **livre** e ciência.



Live de Python



Eduardo Mendes

12,8 mil inscritos

PERSONALIZAR O CANAL

YOUTUBE STUDIO

INÍCIO

VÍDEOS

PLAYLISTS

COMUNIDADE

CANAIS

SOBRE



Atividades recentes

▶ REPRODUZIR TODOS



Live de Python #124 -
Trabalhando com FOSS co...



Selenium com Python
#Plantão - Dúvidas e...

CANAIS EM DESTAQUE

CodeShow

INSCRITO

Spacedevs

INSCRITO

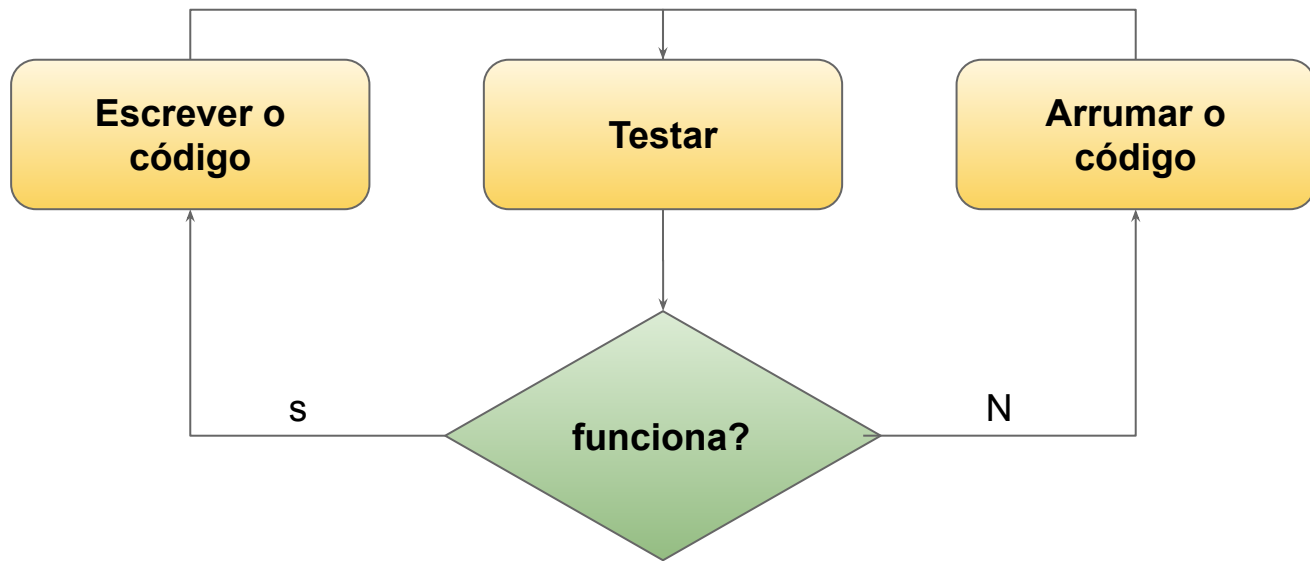
youtube.com/eduardomendes

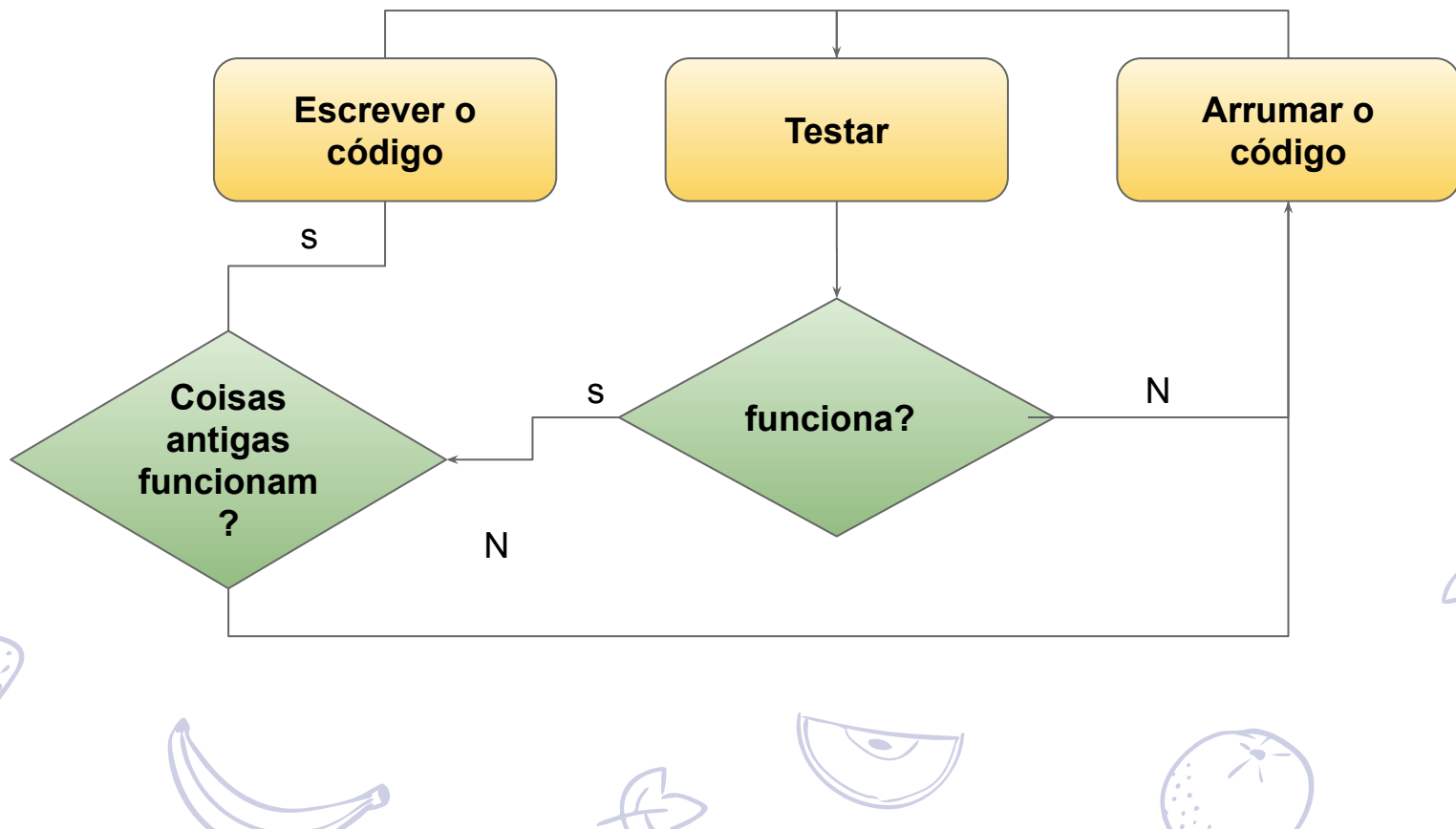
Roteiro

- ✗ Testes, para que testes?
- ✗ Tipos de teste
- ✗ Bibliotecas de teste
- ✗ Testes de unidade









The background of the slide is decorated with various hand-drawn illustrations of fruits and vegetables. These include a blueberry cluster in the top left, an orange slice at the top center, a green leafy vegetable at the top right, a watermelon slice on the right, a lemon on the left, a green leafy vegetable on the left, a strawberry at the bottom left, a banana at the bottom left, a green leafy vegetable at the bottom center, a green vegetable slice at the bottom center, a cherry at the bottom right, and a lemon slice on the right. A green circle with two black exclamation marks is positioned above the main text.

!!

Cada testes executado
manualmente leva 1 minuto.

30 testes.....

The background of the slide is decorated with various hand-drawn illustrations of fruits and vegetables. These include a blueberry cluster in the top left, a lemon slice in the top center, a green leafy vegetable in the top right, a watermelon slice in the top right, a lime in the middle right, a lemon slice in the middle right, a strawberry in the bottom left, a banana in the bottom left, a green leafy vegetable in the bottom center, a lime slice in the bottom center, a cherry in the bottom right, and a lemon slice in the bottom right.

“

Agora, e o que foi feito
antes?

Toda feature nova leva mais
uma carga de testes?

Então vamos
aprender testes

Fi: 76% | 151.6 GiB | Wi: 52% at Intel® wireless | 140.4 MHz / 2.4 GHz | 370.48.0.0 | BAT: 66.66% 07:26 07:13 | 2019-08-12 23:53:36

An abstract geometric pattern composed of numerous overlapping squares of various sizes. The primary colors used are a vibrant blue, a bright yellow, a warm orange, and a deep purple. The squares are arranged in a non-repeating, organic fashion, creating a complex, layered visual effect. The pattern fills the entire frame, with some squares appearing more prominent than others due to their position and size.

Tipos de teste

- ✗ Testes unitários
- ✗ Testes de Integração
- ✗ Testes de sistema
- ✗ Testes de regressão
- ✗ Testes de regressão visual
- ✗ Testes de caixa preta/branca/cinza
- ✗ Testes de mutação



Tipos de teste

- ✗ Testes unitários
- ✗ Testes de Integração
- ✗ Testes de sistema
- ✗ Testes de regressão
- ✗ Testes de regressão visual
- ✗ Testes de caixa preta/branca/cinza
- ✗ Testes de mutação



Bibliotecas de teste



Bibliotecas

- ✗ Unittest
 - Nativo
- ✗ Pytest
 - <https://docs.pytest.org/>
- ✗ Mamba
 - <https://nestorsalceda.com/mamba>
- ✗ Behave
 - <https://github.com/behave/behave>

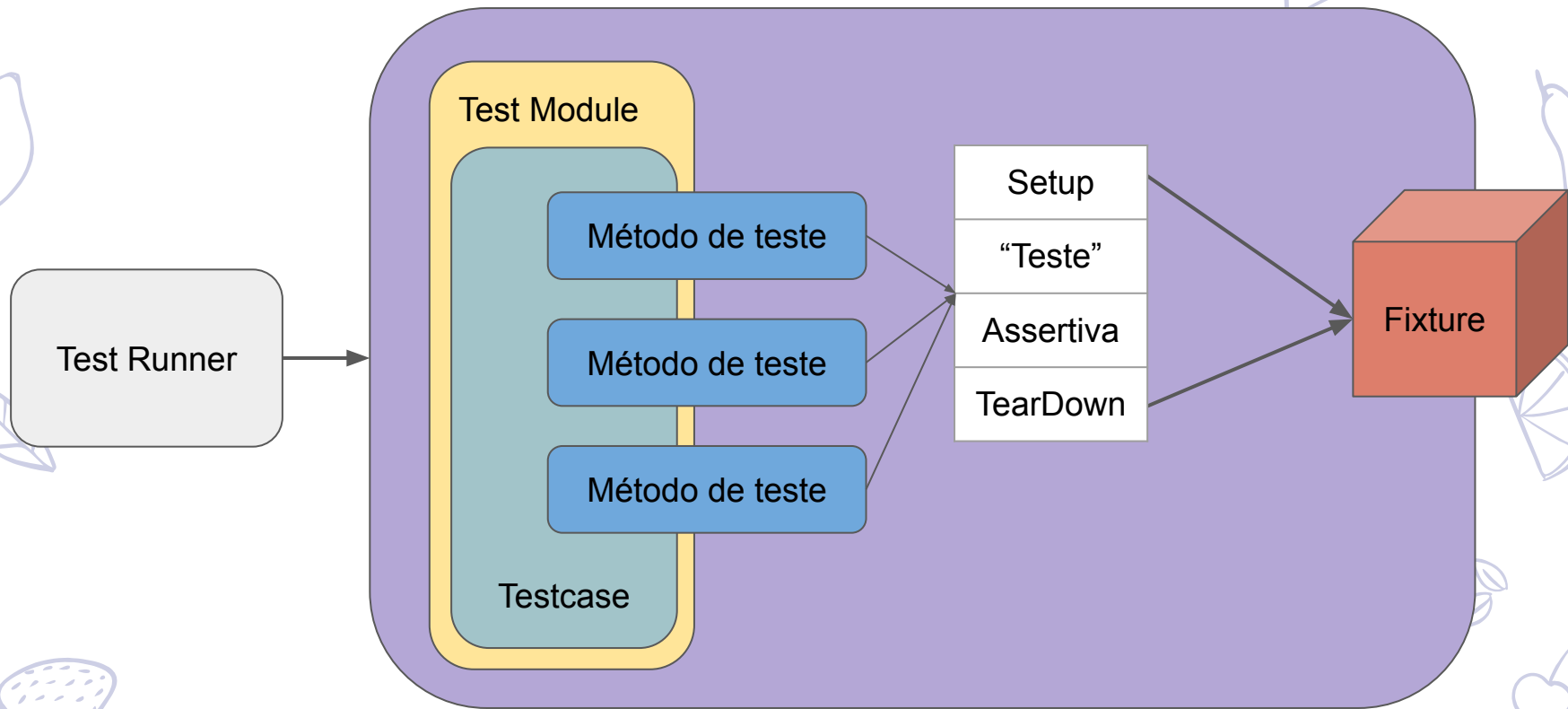


Bibliotecas

- ✗ Unittest
- ✗ Pytest
- ✗ Mamba
- ✗ Behave



Um belo dia com
X-Unit



Test Runner

```
~/live75 >>> python -m unittest -v
```

21:33:00

```
-----  
-----  
Ran 0 tests in 0.000s  
  
OK
```

Testcase

```
class TestCase(unittest.TestCase):  
    def setUp(self):  
        print('No setUp')  
  
    def tearDown(self):  
        print('No tearDown')  
  
    def test_01(self):  
        print('No test_01')  
        self.assertEqual(1, 1, msg='1 != 1')
```

Fixtures

Método de teste

Assertiva

Nosso primeiro desafio

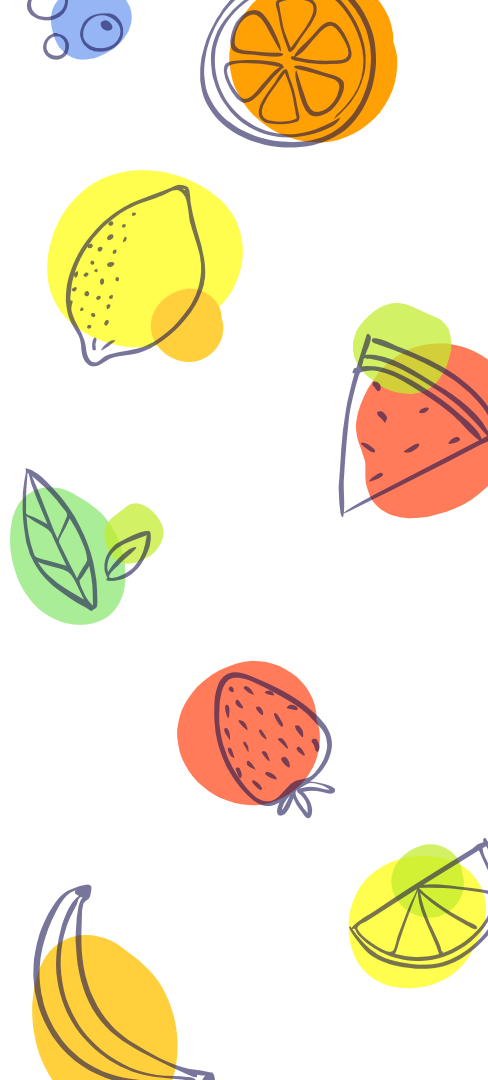
```
def soma(x, y):  
    """Retorna uma soma de 2 números."""  
    return x + y
```



SUT (System Under Test)

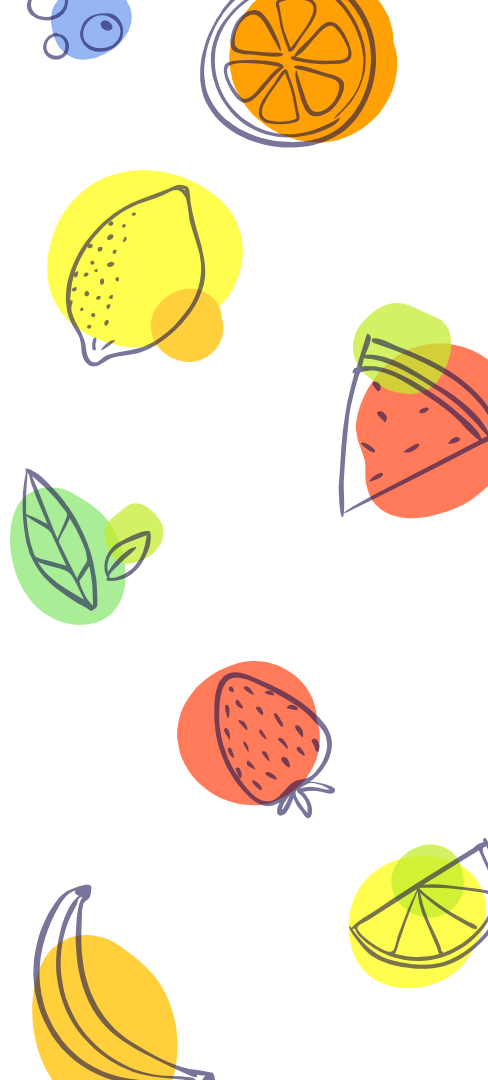
Geralmente, “o que está sendo testado” na bibliografia aparece como SUT, mas pode aparecer com nomes diferentes, como:

- SUT: Sistema em teste (A coisa em si)
- DOC: Componente de quem o SUT depende

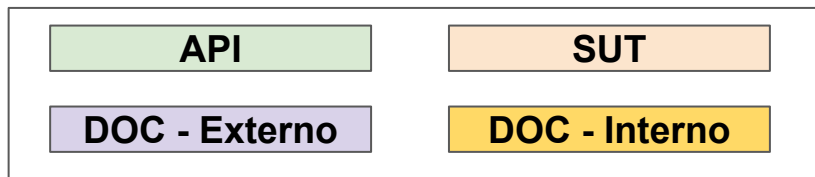


Nosso segundo desafio

```
def expr(x, y, z):  
    """  
    Resolve a expressão.  
  
    (x + y) - z  
    """  
    return subtracao(soma(x, y), z)
```

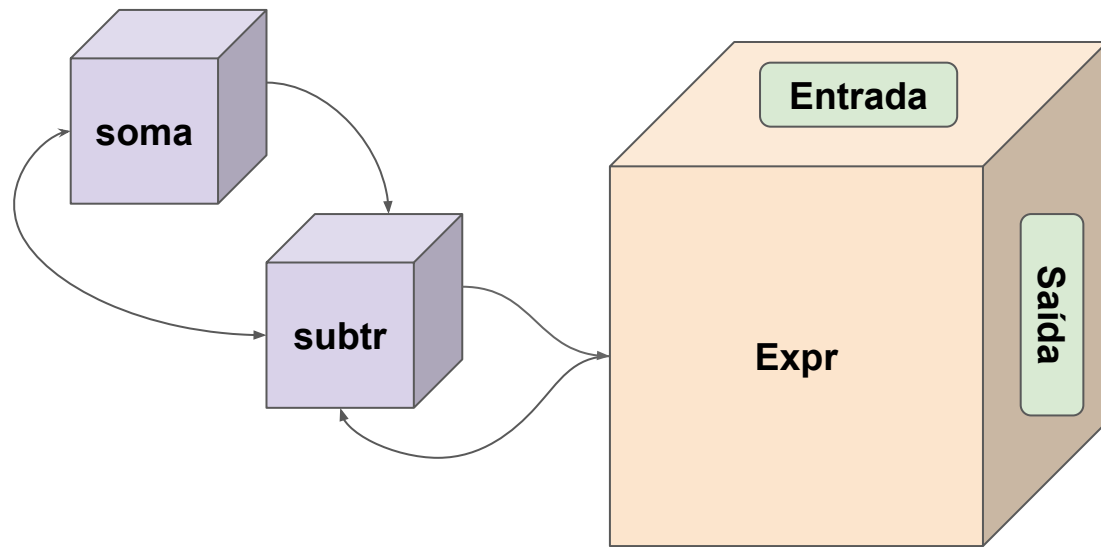


Acoplamento



Não é tão complicado entender o sentido de acoplamento.

O quanto uma coisa depende de outra.

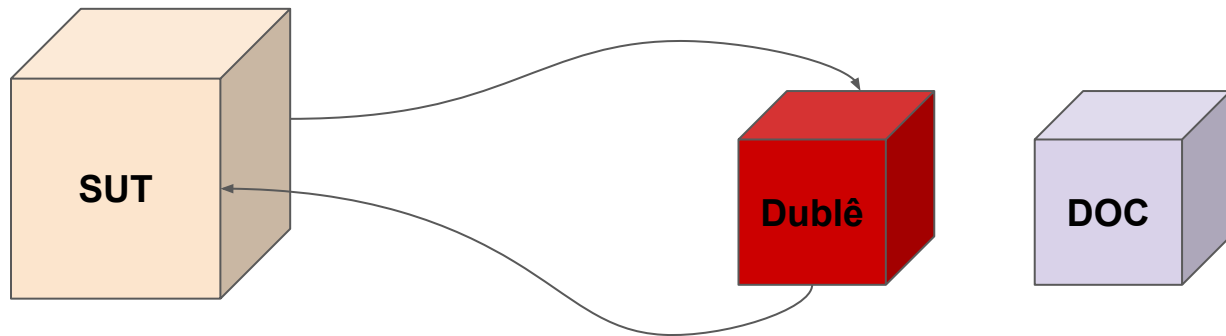




Dublês de teste

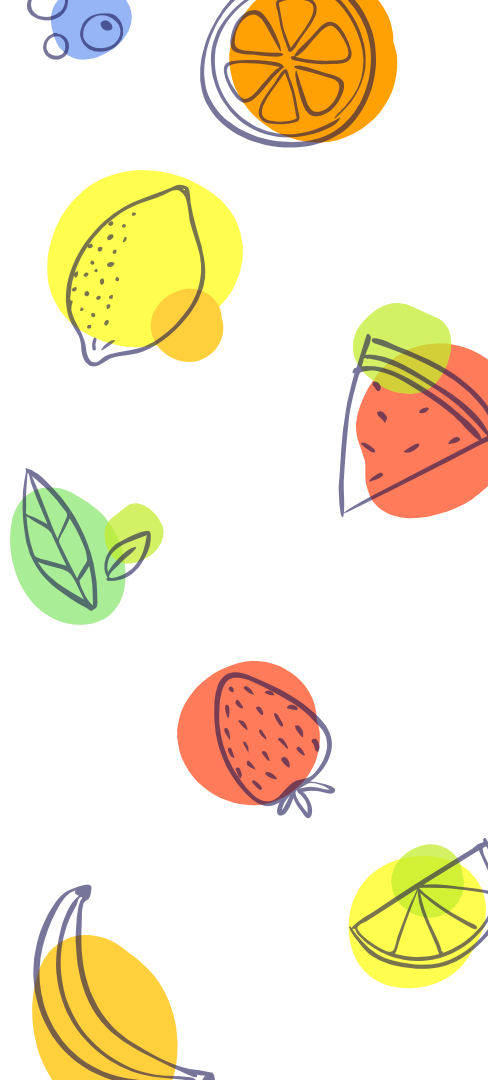
Dublês

Dublês tornam seus testes mais **determinísticos**, pois eles vão remover as dependências do jogo.



Dublês

```
def teste_expressao_com_1_1_1_deve_retonar_1(self):  
    x = 1  
    y = 1  
    z = 1  
    esperado = 1  
    with patch('code.subtracao', return_value=1):  
        self.assertEqual(expr(x, y, z), esperado)
```



Aprenda comigo

- ✕ Live de python #75
- ✕ Live de python #76
- ✕ Live de python #79
- ✕ Live de python #80
- ✕ Live de python #81
- ✕ Live de python #83





XOXO

Perguntas?

gist.github.com/dunossauro/ff9c2d346ba05ee66887346ccd36c20a

Você pode me achar em

@dunossauro

mendesxeduardo@gmail.com

youtube.com/eduardomendes