

Desacoplando seus testes com HTTPretty





Quem sou eu

- Patrick Mazulo - 28 anos
- Trabalho nessa empresa do GIF bonitinho
- Diretor Presidente - APyB
 - <https://python.org.br/apyb/>
- @ericleribertson
- /mazulo



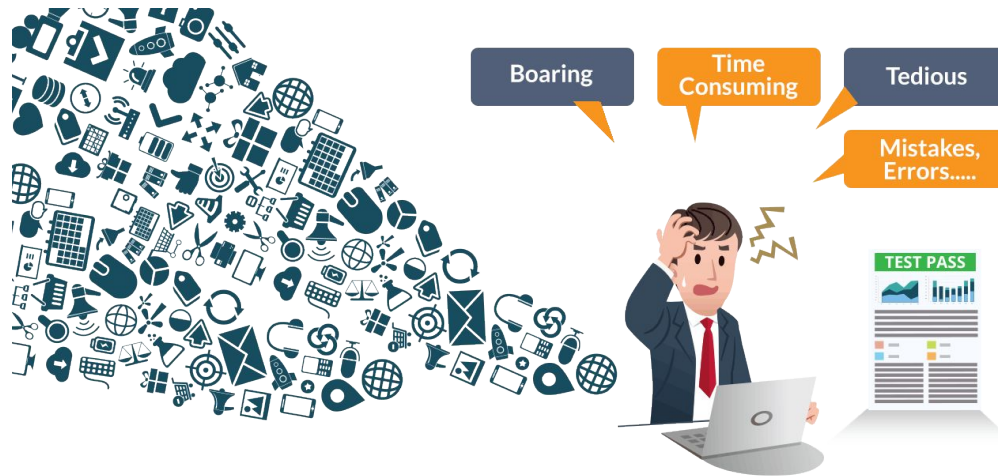


O que são esses testes?

- Teste é um processo dentro do desenvolvimento de software que avalia a(s) funcionalidade(s) de uma aplicação, tendo como objetivo confirmar se o software atendeu os requisitos especificados
- Desde épocas imemoráveis, testes fazem parte do processo de entrega de software

A importância dos testes

- Testes manuais



A importância dos testes

- Testes automatizados
 - Manter ritmo de entrega
 - CI/CD



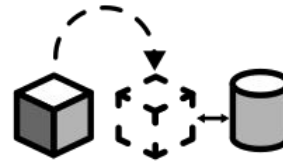
Build



Test



**Provide
Infraestrutura**



Deploy

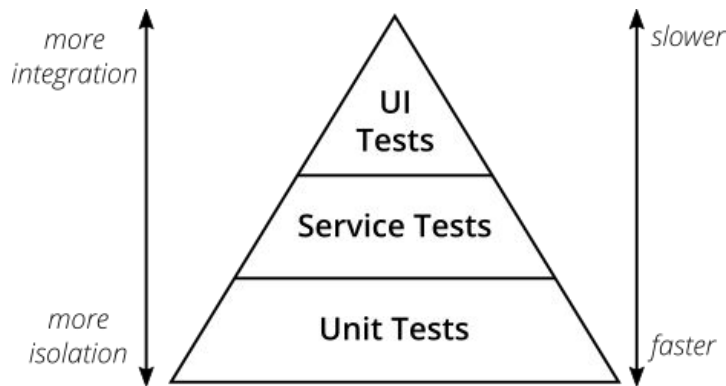


**Test
More**



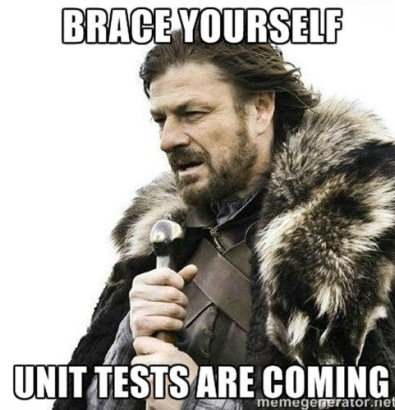
Pirâmide de testes

- Conceito apresentado por Mike Cohn no livro *Succeeding with Agile*
- Ilustração para ajudar a pensar sobre as diferentes camadas de testes



Unit tests - O que são? Onde vivem? Do que se alimentam?

- Alicerce da nossa suíte de testes
- Eles nos certificam que uma certa unidade do nosso projeto funciona como esperado
- Mas o que seria uma unidade?





Unit tests

- Se fizermos essa pergunta para 3 pessoas, teremos 4 respostas (levemente) diferentes
- Em uma linguagem funcional, uma unidade poderia ser uma função
- Em uma linguagem O.O, isso pode variar de um único método para uma classe inteira

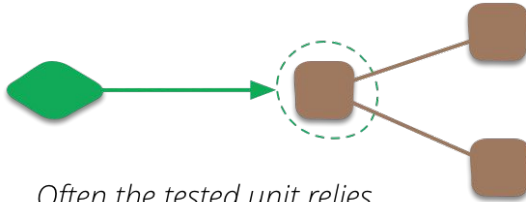




Unit tests

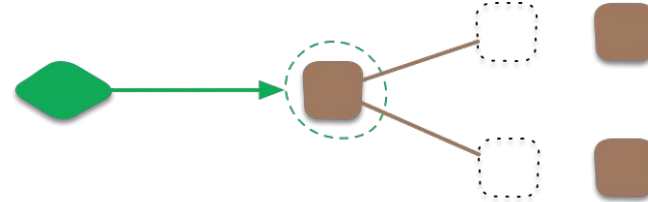
- Sociáveis ou Solitários?

Sociable Tests



Often the tested unit relies on other units to fulfill its behavior

Solitary Tests



Some unit testers prefer to isolate the tested unit

```
1  from product import Product
2  from customer import Customer
3
4
5  class Order:
6
7      def __init__(self):
8          self.product = Product()
9          self.customer = Customer()
10
11     def price(self):
12         product_price = self.get_product_price()
13         shipping_price = self.get_shipping_price()
14
15         # Here we have some crazy logic
16         # and that's what we want to test
17
18     def get_product_price(self):
19         return self.product.get_price()
20
21     def get_shipping_price(self):
22         return self.customer.calculate_shipping_price()
```



Características

- Unit tests devem cumprir alguns requisitos:
 - Devem testar uma **unidade**
 - Devem rodar isolados: sem banco de dados, sistema de arquivos ou **chamadas HTTP(!)**
 - Devem rodar **rápido**



Velocidade

- O quão rápido nós estamos falando?
 - Rápido o suficiente para que você não tenha problema em rodar os testes para cada mudança que você faz naquele pedaço de código
 - Rápido o suficiente para que seja possível você rodar a **suíte inteira** de testes antes de dar um *git push*

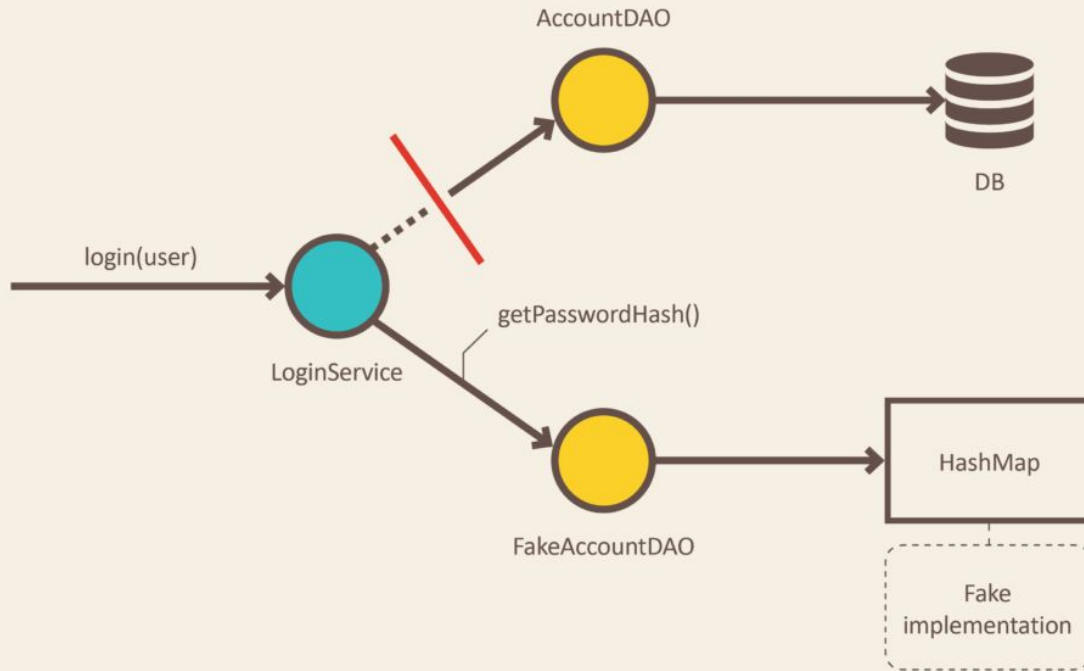




Mocks!

- **Mocking** é bastante utilizado testes unitários. Uma dada unidade sob testes pode ter algumas dependências que podem ser substituídas por outros objetos que simulam esse comportamento.
- Em resumo: mocking é criar um objeto que *imita* o comportamento de outro objeto.

Fake





Mock no Python





HTTPretty?

- Uma ferramenta que mocka clientes HTTP no Python. Fornece um mock completo do TCP socket
- Inspirado no [Faker](#) para Ruby





HTTPretty em ação









Pessoas > Tecnologia