**Introdução**

Pirâmide

>”e2e - End to End” ou teste de ponta a ponta \* mais caro e complexo\*

Testa todo fluxo do sistema, de uma ponta À outras, desde o front end até o último nível de front end

>”integration” ou teste de integração \*preço mediano, amplitude maior,

Testa várias unidades

>”unit test” ou teste unitário \*mais barato,\*

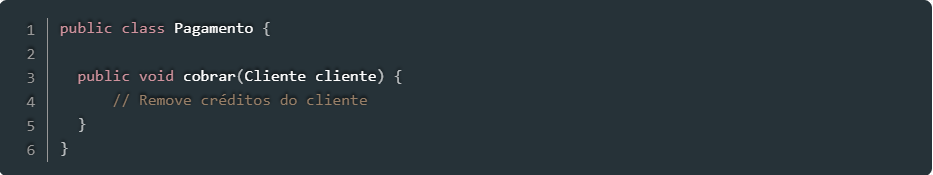
Testa apenas uma única unidade pequena, geralmente de algorítimos, como uma função ou método

A pirâmide mostra que a base “unit test” é maior, por ser mais barato, sendo assim mais usado que os demais.

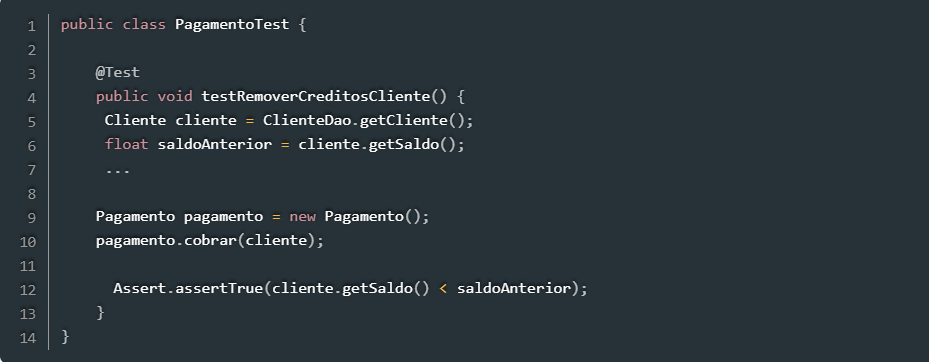
**O que são testes unitários?**

O teste unitário é uma verificação feita com uma pequena porção de código, uma unidade de um software. Ou seja, é diferente do teste geral, que se dedica a atestar o fluxo do sistema, com as funcionalidades principai

No código abaixo temos como exemplo uma classe de Pagamento, que possui o método cobrar. Note que cobrar recebe como parâmetro um Cliente.



Sendo assim, poderíamos escrever um teste para esse método, verificando se o saldo do cliente é menor após sua chamada. Como resultado teríamos uma classe de teste, como demonstrado abaixo:



#### **Quando fazer Teste Unitário?**

#### **Porque usar Testes Unitários?**

**Abaixo temos alguns dos principais fatores que motivam o uso sistemático da prática de testes unitários:**

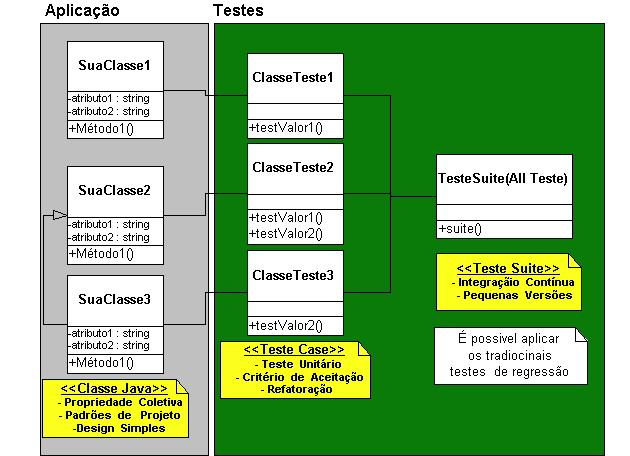
* **Prevenção de BUGS**
* **Código testado é mais confiável;**
* **Permite alterações sem medo;**
* **Testa situações de sucesso e de falha;**
* **Redução de custos.**

#### **Quem faz o Teste Unitário?**

**Test Case**

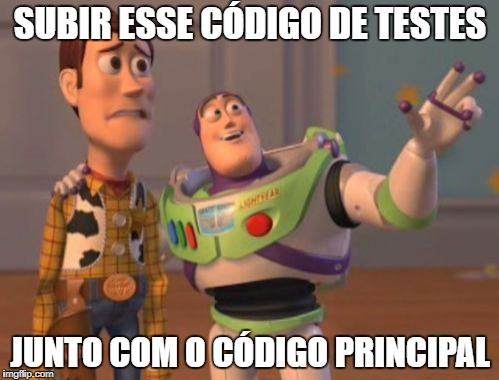
**Test Suite**

**Teste de aceitação (homologação)**

****

#### **Teste Unitário: O que testar?**

* **A principal regra para saber o que testar é: “Tenha criatividade para imaginar as possibilidades de testes”;**
* **Comece pelas mais simples e deixe os testes “complexos“ para o final;**
* **Use apenas dados suficientes (não teste 10 condições se três forem suficientes);**
* **Não teste métodos triviais, tipo get e set;**
* **No caso de um método set, só faça o teste caso haja validação de dados;**
* **Achou um bug? Não conserte sem antes escrever um teste que o pegue (se você não o fizer, ele volta).**



**Ferramentas para testes unitários**

-NUnit e xUnit

-PHPUnit

-JUnit

**JUnit para Java**

**O que é JUnit**

#### **Por que usar JUnit?**

* **Facilita a criação, execução automática de testes e a apresentação dos resultados;**
* **É Orientado a Objeto;**
* **É gratuito**

#### **Analisando o resultado no JUnit**

