

Formar grupos de (no máximo) 3 integrantes  
Instalar o framework JUnit  
O Desafio “Mão na Massa”  
Baixar o programa  
(Opcional) Criar casos de teste no papel  
Implementar os casos de teste usando JUnit

# Introdução ao Teste de Software

## Desafio “Mão na Massa”

Vinicius H. S. Durelli

✉ [durelli@ufsj.edu.br](mailto:durelli@ufsj.edu.br)

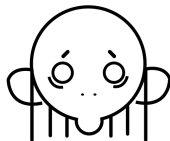
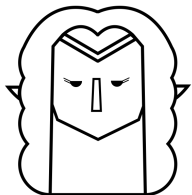


## Desafio “Mão na Massa”: etapas

- 1 Formar grupos de (no máximo) 3 integrantes
- 2 Instalar o framework JUnit
- 3 O Desafio “Mão na Massa”
- 4 Baixar o programa
- 5 (Opcional) Criar casos de teste no papel
- 6 Implementar os casos de teste usando JUnit

Um, dois, três, valendo...

## **Escolham os membros do grupo!** (The Fellowship of the Ring Desafio “Mão na Massa”)



- 1 Formar grupos de (no máximo) 3 integrantes
- 2 Instalar o framework JUnit**
- 3 O Desafio “Mão na Massa”
- 4 Baixar o programa
- 5 (Opcional) Criar casos de teste no papel
- 6 Implementar os casos de teste usando JUnit

## Antes de mais nada, é preciso instalar o framework JUnit

Para o desafio “Mão na Massa”, vamos usar o framework JUnit (apresentado na aula passada). **Configurar o framework na plataforma desejada também faz parte do desafio.**

Detalhes:

- Qualquer editor/IDE pode ser usado durante o desafio.
- Ao final do desafio, somente a classe contendo os testes de unidade deve ser enviada.<sup>1</sup>

➡ Em caso de dúvidas, mais informações sobre como configurar o framework JUnit em várias plataformas pode ser encontrado em: <https://junit.org/junit5/docs/current/user-guide/>.

---

<sup>1</sup> Ou as classes, caso o grupo tenha trabalhado de forma paralela.

- 1 Formar grupos de (no máximo) 3 integrantes
- 2 Instalar o framework JUnit
- 3 O Desafio “Mão na Massa”**
- 4 Baixar o programa
- 5 (Opcional) Criar casos de teste no papel
- 6 Implementar os casos de teste usando JUnit

O desafio consiste em testar um programa relativamente simples. Os integrantes de cada grupo devem criar e implementar casos de teste usando o framework JUnit.

## Descrição do Programa

O programa aceita três valores inteiros a partir da linha de comando. Os três valores representam os lados de um triângulo. O programa mostra uma mensagem indicando o tipo do triângulo (Myers et al. 2011).

- Equilátero: possui os três lados com medidas iguais.
- Isósceles: possui dois lados com medidas iguais.
- Escaleno: possui os três lados com medidas diferentes.

- 1 Formar grupos de (no máximo) 3 integrantes
- 2 Instalar o framework JUnit
- 3 O Desafio “Mão na Massa”
- 4 Baixar o programa**
- 5 (Opcional) Criar casos de teste no papel
- 6 Implementar os casos de teste usando JUnit



## Baixar o programa. . .

Uma implementação encontra-se disponível em [https://github.com/macro-mancer/slides\\_das\\_aulas](https://github.com/macro-mancer/slides_das_aulas).

- Criar um projeto na plataforma/editor/IDE de escolha;
- Entender a implementação;
- Executar o programa (se necessário).

- 1 Formar grupos de (no máximo) 3 integrantes
- 2 Instalar o framework JUnit
- 3 O Desafio “Mão na Massa”
- 4 Baixar o programa
- 5 (Opcional) Criar casos de teste no papel**
- 6 Implementar os casos de teste usando JUnit

## (Opcional) Criar casos de teste no papel

Alguns exemplos de como os casos de teste podem ser organizados:

Caso de Teste	Condição	Valores	Resultado Esperado
#1	Três lados válidos, iguais	7, 7, 7	"equilátero"
#2	Dois lados válidos, iguais	5, 5, 3	"isósceles"
#3	Dois lados válidos e iguais, um lado inválido (terceiro).	5, 5, -5	"inválido"

- 1 Formar grupos de (no máximo) 3 integrantes
- 2 Instalar o framework JUnit
- 3 O Desafio “Mão na Massa”
- 4 Baixar o programa
- 5 (Opcional) Criar casos de teste no papel
- 6 Implementar os casos de teste usando JUnit**

## Implementar casos de teste usando JUnit

Exemplo de caso de teste:

```
class TestTriangle {  
    @Test  
    void testEquilateral() {  
        //side1 = 7, side2 = 7, side3 = 7,  
        Triangle t = new Triangle(7, 7, 7);  
        //equilateral?  
        assertEquals(TriangleType.valueOf("EQUILATERAL"),  
            t.getType());  
    }  
}
```

## Avaliando os testes (1)

*“It’s a long way to the top  
If you wanna ~~rock~~ ‘n’ roll software testing”*

*It’s A Long Way To The Top (If You Wanna Rock ‘n’ Roll) by AC/DC*

1- Foi incluído um caso de teste que representa um triângulo escaleno válido? (Note que casos de teste como 1, 2, 3 e 2, 5, 10 não são válidos)

2- Foi incluído um caso de teste que representa um triângulo equilátero?

3- Foi incluído um caso de teste que representa um triângulo isósceles?

(Note que um caso de teste como 2, 2, 4 não é válido)

## Avaliando os testes (2)

4- Foram incluídos casos de teste para triângulos isósceles que testam todas as três possíveis permutações de dois estados iguais (e.g., 3, 3, 4; 3, 4, 3; e 4, 3, 3?

5- Foi incluído um caso de teste em que pelo menos um dos lados possui o valor 0?

6- Foi incluído um caso de teste em que um dos lados possui um valor negativo?

7- Foi incluído um caso de teste com três inteiros maiores que zero tal que a soma de dois lados é igual ao tamanho do terceiro lado?

(Note que o caso de teste 1, 2, 3 pode ser usado novamente para indicar um problema no programa, i.e., 1, 2, 3 não é um triângulo válido)

## Avaliando os testes (3)

8- Foram incluídos 3 casos de teste como os descritos no item 7 testando cada uma das permutações possíveis, e.g., 1, 2, 3; 1, 3, 2; e 3, 1, 2?

9- Foi incluído um caso de teste todos os lados tem tamanho 0?

10- Foi incluído um caso de teste com três inteiros maiores que zero tal que a soma de dois lados é menor que o tamanho do terceiro lado (e.g., 12, 15, 30)?

11- Foram incluídos 3 casos de teste como os descritos no item 10 testando cada uma das permutações possíveis, e.g., 1, 2, 4; 1, 4, 2; e 4, 1, 2?



Myers, Glenford J., Corey Sandler, & Tom Badgett (2011). *The Art of Software Testing*. 3rd. Wiley, p. 256.

"Elf", "Gollum", and "Hobbit" icon by Rodrigo Vidinich, from the Noun Project (<https://thenounproject.com/>).