

# TestNG Framework

Aluna: Marilia Caroline de Almeida Cardoso Silva

Professor: Breno Alexandro Ferreira de Miranda

# Sumário

- TestNG
  - Características Principais;
  - Vantagens e Desvantagens.
- Tutorial
  - Principais Trechos;
- Desafios

# TestNG

- É um framework, em Java e *open source*, para testes automatizados;
- Feito com base em NUnit e JUnit, porém com adição de algumas funcionalidades que o tornam melhor que suas inspirações;
  - Agrupamento, dependência, priorização, anotações, etc.
- Cobre uma variedade de tipos de teste: teste unitário, funcional, *end to end*, de integração, etc.;
- Disponível em: <https://testng.org/doc/> e no Marketplace da IDE Eclipse.

# TestNG

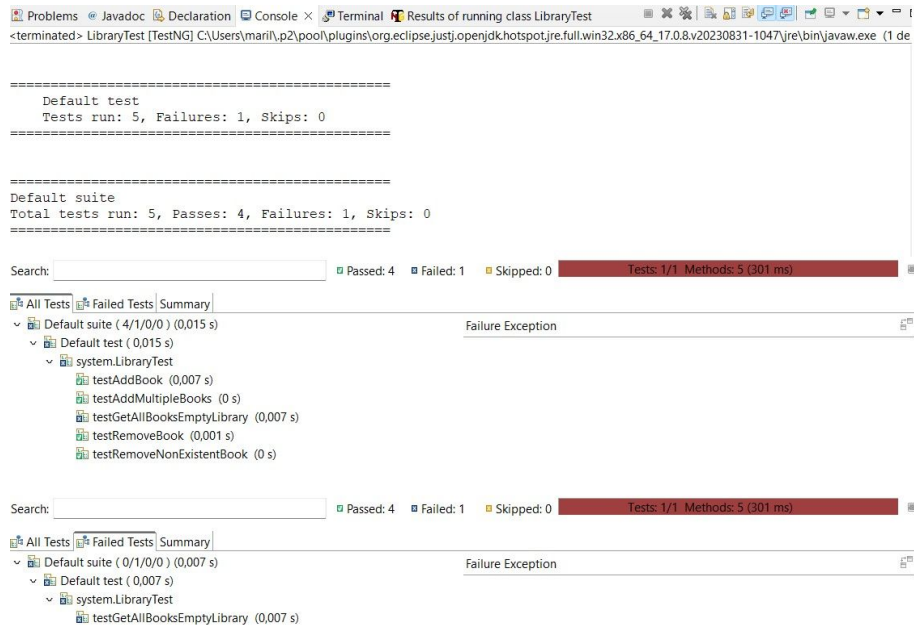
- Vantagens:
  - Configuração flexível;
  - Anotações;
  - Execução paralela;
  - Agrupamento e Priorização;
  - Integração.
- Desvantagens:
  - Algumas funcionalidades não são tão simples de aprender;
  - Tamanho da comunidade;
  - Código legado;
  - Desafios de integração.

# Tutorial

- Para o tutorial, foi considerado um sistema simples de cadastro de livros em uma biblioteca, cujo código completo está no repositório:  
<https://github.com/mcacs/testngproject>;
- Ele é composto das classes Book, Library e LibraryTest;
  - Book: responsável por representar uma instância de um livro;
  - Library: responsável por representar uma biblioteca com uma lista de livros;
  - LibraryTest: código de teste com cinco testes unitários, 'testAddBook', 'testRemoveBook', 'testGetAllBooksEmptyLibrary', 'testAddMultipleBooks' e 'testRemoveNonExistentBook'.

# Tutorial

- Após rodar os testes, o TestNG gera os seguintes *reports*:



```
=====
Default test
Tests run: 5, Failures: 1, Skips: 0
=====

=====
Default suite
Total tests run: 5, Passes: 4, Failures: 1, Skips: 0
=====

Search: [ ] Passed: 4 Failed: 1 Skipped: 0 Tests: 1/1. Methods: 5 (201 ms)

All Tests Failed Tests Summary
- Default suite ( 4/1/0/0 ) (0,015 s) Failure Exception
  - Default test ( 0,015 s)
    - system.LibraryTest
      - testAddBook (0,007 s)
      - testAddMultipleBooks (0 s)
      - testGetAllBooksEmptyLibrary (0,007 s)
      - testRemoveBook (0,001 s)
      - testRemoveNonExistentBook (0 s)

Search: [ ] Passed: 4 Failed: 1 Skipped: 0 Tests: 1/1. Methods: 5 (201 ms)

All Tests Failed Tests Summary
- Default suite ( 0/1/0/0 ) (0,007 s) Failure Exception
  - Default test ( 0,007 s)
    - system.LibraryTest
      - testGetAllBooksEmptyLibrary (0,007 s)
```

Figura 1 - *Reports* gerados após os testes

# Tutorial

- Nos *reports*, temos as informações da quantidade de testes que foram gerados, quantos e quais passaram ou falharam;
- Nesse caso, 4 testes passaram e 1 falhou:
  - Testes que passaram: 'testAddBook', 'testRemoveBook', 'testAddMultipleBooks' e 'testRemoveNonExistentBook';
  - Teste que falhou: 'testGetAllBooksEmptyLibrary'
- Geração de arquivos .xml para os testes que passaram e o que falhou.
  - Arquivos com informações para definir suítes de testes, testes individuais, classes de teste, grupos de teste e várias opções de execução.

# Desafios

- Para encerrar, alguns desafios que foram encontrados na construção do tutorial:
  - Falta de familiaridade com a linguagem Java;
  - Necessidade de baixar e configurar outra IDE para utilizar a ferramenta;
  - Além disso, há a necessidade de adicionar vários arquivos do tipo JAR à biblioteca do TestNG para que o arquivo de teste seja rodado de forma correta, o que não é bem esclarecido nos materiais disponíveis *online*.



# TestNG Framework

Aluna: Marilia Caroline de Almeida Cardoso Silva

Professor: Breno Alexandro Ferreira de Miranda