

# Testes unitários com JUnit

Willyan Guimarães Caetano

Programador

# Mais sobre mim

- Atua há 10 anos com Java
- Maior experiência com backend
- Passatempos: Jogos, fotografia e quando pode viaja por aí
- <https://www.linkedin.com/in/willyancaetanodev>



# Objetivo do curso

- Entender a motivação de escrever testes unitários
- Conhecer o que é JUnit, configurar a ferramenta para a utilizar em seus projetos Java e aprofundar nos recursos que ele provê para construir testes unitários efetivos.

# Percurso

## Aula 1

Por que escrever testes unitários ?

## Aula 2

Hello World, JUnit!

## Aula 3

Aprofundando nos recursos

# Percurso

## Aula 4

Recursos de testes nas IDEs

## Aula 5

Boas práticas

## Aula 6

Conclusão

# Dúvidas durante o curso?

- > Fórum do curso
- > Comunidade online (Discord)



SCAN ME

## Aula 1

# Por que escrever testes unitários?

// Testes unitários com JUnit

# Objetivos

- Entender o que é testes unitários e sua importância



# O que são testes unitários?

- Também chamado de testes de unidade
- Testar a menor unidade de código possível
- Unidade: função, método, classes
- Testar uma aplicação na sua menor parte
- Geralmente escrito em tempo de desenvolvimento

# Exemplo simples

```
1  class Pessoa {
2
3      //construtor, atributos e outros métodos
4
5      public boolean ehMaiorDeIdade() {
6          return idade > 18;
7      }
8  }
9
10 class PessoaTeste {
11
12     @Test
13     void validaVerificacaoDeMaioridade() {
14         Pessoa joaozinho = new Pessoa("João", LocalDate.of(2004, 1, 1));
15         Assertions.assertTrue(joaozinho.ehMaiorDeIdade());
16     }
17 }
```

# Por quê escrever testes unitários ?

Situações no desenvolvimento de software que mostram a importância de testes unitários

- Compreender o código fonte
- Corrigir bugs com segurança
- Refatorar código sem introduzir bugs
- Entregar com segurança uma nova feature

# Por quê escrever testes unitários ?

Podemos ainda falar sobre:

- Pirâmide de Testes
- Testes unitários como métrica de Qualidade (confiabilidade)
- Cobertura de Testes
- Automação na execução de testes
- Cultura de agilidade

# Referências

- <https://imasters.com.br/devsecops/testes-unitarios-qual-a-importancia>
- <https://dayvsonlima.medium.com/entenda-de-uma-vez-por-todas-o-que-s%C3%A3o-testes-unit%C3%A1rios-para-que-servem-e-como-faz%C3%AA-los-2a6f645bab3>
- <https://medium.com/cesar-update/aprendendo-a-promover-uma-cultura-de-qualidade-no-desenvolvimento-de-s-oftware-%C3%A1gil-f5a3444d88d1>

# Dúvidas durante o curso?

- > Fórum do curso
- > Comunidade online (Discord)



SCAN ME

## Aula 2

# Hello world, JUnit!

// Testes unitários com JUnit

# Objetivos

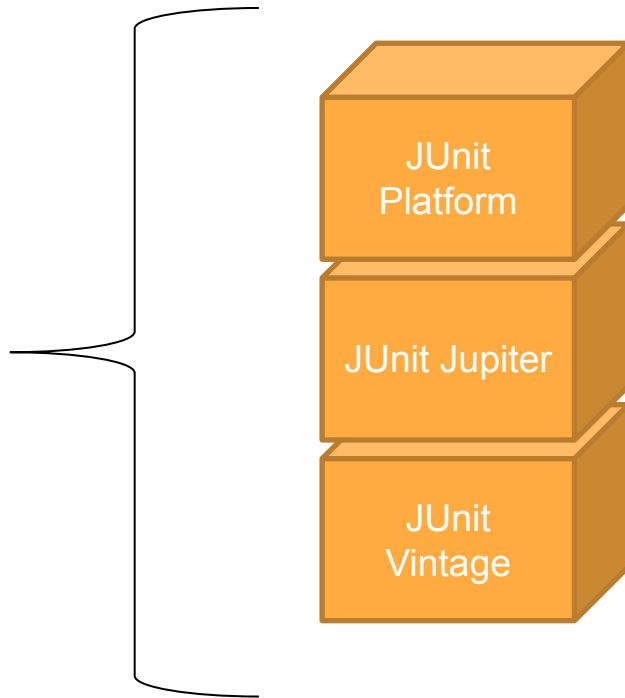
- Falar um pouco sobre a história do JUnit
- Falar um pouco sobre a versão atual
- Configurar em projeto Maven e Gradle a ferramenta



# Um pouco de história

- Framework open source para criação de testes unitários criado por Erich Gamma e Kent Beck
- Kent Beck (TDD)
- Erich Gamma (Gang of Four, Design Patterns)

# Versão atual



# Configurando JUnit nos projetos



# Referências

- <https://junit.org/junit5/>
- <https://github.com/junit-team/junit5/>
- <https://www.baeldung.com/junit-5-gradle>
- <https://maven.apache.org/surefire/maven-surefire-plugin/examples/junit-platform.html>

# Dúvidas durante o curso?

- > Fórum do curso
- > Comunidade online (Discord)



SCAN ME

## Aula 3

# Aprofundando nos recursos

// Testes unitários com JUnit

# Objetivos

- Conhecer em detalhes os principais recursos que o JUnit provê

## Aula 3 . Etapa 1

# O básico para testar

// Testes unitários com JUnit



# Estrutura básica

```
class PessoaTeste {  
    @Test //--> Anotação é primordial para testar  
    void validaVerificacaoDeMaioridade() {  
        Pessoa joaozinho = new Pessoa("João", LocalDate.of(2004, 1, 1)); //--> cria um cenário  
        Assertions.assertTrue(joaozinho.ehMaiorDeIdade()); //--> Executa as validações  
    }  
}
```

# Referências

- <https://junit.org/junit5/docs/current/api/org.junit.jupiter.api/org/junit/jupiter/api/Assertions.html>
- [https://www.tutorialspoint.com/junit/junit\\_using\\_assertion.htm](https://www.tutorialspoint.com/junit/junit_using_assertion.htm)

# Dúvidas durante o curso?

- > Fórum do curso
- > Comunidade online (Discord)



SCAN ME

Aula 3 . Etapa 2

# Mais algumas asserções

// Testes unitários com JUnit

# Referências

- <https://junit.org/junit5/docs/current/api/org.junit.jupiter.api/org/junit/jupiter/api/Assertions.html>
- [https://www.tutorialspoint.com/junit/junit\\_using\\_assertion.htm](https://www.tutorialspoint.com/junit/junit_using_assertion.htm)

# Dúvidas durante o curso?

- > Fórum do curso
- > Comunidade online (Discord)



SCAN ME

## Aula 3 . Etapa 3

# After e Before

// Testes unitários com JUnit

# Referências

- <https://www.baeldung.com/junit-before-beforeclass-beforeeach-beforeall>



# Dúvidas durante o curso?

- > Fórum do curso
- > Comunidade online (Discord)



SCAN ME

Aula 3 . Etapa 4

# Assumptions e Testes condicionais

// Testes unitários com JUnit

# Referências

- <https://junit.org/junit5/docs/current/api/org.junit.jupiter.api/org/junit/jupiter/api/Assumptions.html>
- <https://www.baeldung.com/junit-5#2-assumptions>
- <https://mkyong.com/junit5/junit-5-assumptions-examples/>
- <https://junit.org/junit5/docs/current/user-guide/#writing-tests-conditional-execution>

# Dúvidas durante o curso?

- > Fórum do curso
- > Comunidade online (Discord)



SCAN ME

## Aula 3 . Etapa 5

# Testando exceptions

// Testes unitários com JUnit

# Dúvidas durante o curso?

- > Fórum do curso
- > Comunidade online (Discord)



SCAN ME

## Aula 3 . Etapa 6

# Ordenando testes

// Testes unitários com JUnit

# Referências

- <https://junit.org/junit5/docs/current/user-guide/#writing-tests-test-execution-order>



# Dúvidas durante o curso?

- > Fórum do curso
- > Comunidade online (Discord)



SCAN ME

## Aula 4

# Recursos de testes nas IDEs

// Testes unitários com JUnit

# Objetivos

- Conhecer o básico para executar testes unitários nas IDEs mais utilizadas pelo mercado

## Aula 4 . Etapa 1

# Visual Studio Code

// Testes unitários com JUnit

# Referências

- <https://code.visualstudio.com/docs/java/java-testing>

# Dúvidas durante o curso?

- > Fórum do curso
- > Comunidade online (Discord)



SCAN ME

## Aula 4 . Etapa 2

# Eclipse

// Testes unitários com JUnit

# Referências

- <https://wiki.eclipse.org/Eclipse/Testing>
- <https://nglauber.medium.com/junit-no-eclipse-ebd134fcf6d4>
- [https://edisciplinas.usp.br/pluginfile.php/5768433/mod\\_resource/content/0/Utilizando%20JUnit%20no%20Eclipse.pdf](https://edisciplinas.usp.br/pluginfile.php/5768433/mod_resource/content/0/Utilizando%20JUnit%20no%20Eclipse.pdf)



# Dúvidas durante o curso?

- > Fórum do curso
- > Comunidade online (Discord)



SCAN ME

## Aula 4 . Etapa 3

# IntelliJ IDEA

// Testes unitários com JUnit

# Referências

- <https://www.jetbrains.com/help/idea/tests-in-ide.html>

# Dúvidas durante o curso?

- > Fórum do curso
- > Comunidade online (Discord)



SCAN ME

## Aula 5

# Boas práticas

// Testes unitários com JUnit

# Objetivos

- Conhecer algumas boas práticas adotadas ao escrever testes unitários

# Busque simplicidade

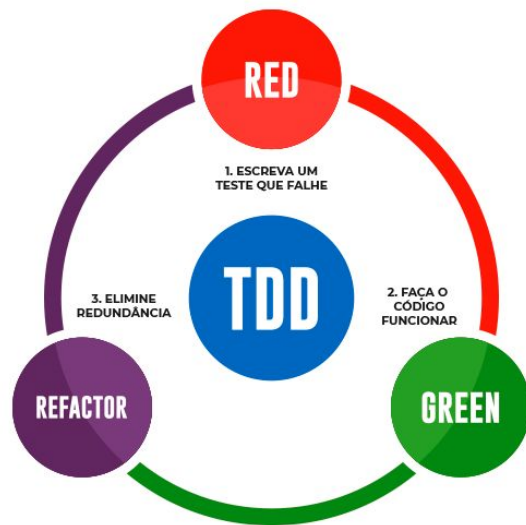
- Preocupe-se com os nomes
- Preocupe-se com a facilidade de leitura

# Comece a testar cedo

- Procure escrever o código de teste o mais próximo possível do código de execução.
- Veja um pouco sobre TDD



# Comece a testar cedo



# Busque padronização

- Nomenclatura

# Testes precisam ser determinísticos



# Economize tempo automatizando

- Ferramentas de cobertura de código  
(Jacoco)
- Automatize a execução dos seus testes

# Referências

- <https://www.testim.io/blog/unit-testing-best-practices/>
- <https://devporai.com.br/5-dicas-para-escrever-bons-testes-unitarios/>
- <https://jeziellago.medium.com/testes-boas-pr%C3%A1ticas-e-patterns-6bfe0925040>

# Dúvidas durante o curso?

- > Fórum do curso
- > Comunidade online (Discord)



SCAN ME

## Aula 6

# Conclusão

// Testes unitários com JUnit

# O que vimos até aqui ?

- Motivação
- Configuração
- Principais recursos
- Usando os recursos das IDEs
- Boas práticas



# Para saber mais

The screenshot shows the GitHub repository page for `willyancaetano/junit5-exemplos`. The repository is public and has a README file. The commit history shows a recent change to improve tests and fixes. The README file content is visible, describing the repository as a collection of examples for using JUnit 5 in a daily development workflow.

**Repository Details:**

- Repository: `willyancaetano/junit5-exemplos` (Public)
- Branches: 1 branch (master)
- Tags: 0 tags
- Commits: 9 commits (last commit: 01e024c, 2 days ago)

**Files:**

File	Commit	Time
<code>src</code>	change: improve tests and fixes	2 days ago
<code>.gitignore</code>	fix : improve gitignore	2 days ago
<code>CONFIGURACAO.md</code>	feat: remove .idea and add new test	2 days ago
<code>LICENSE</code>	Initial commit	12 days ago
<code>README.md</code>	refac: rename packages and add link to configurations md	2 days ago
<code>pom.xml</code>	feat: remove .idea and add new test	2 days ago

**README.md Content:**

## Testes unitários com JUnit

Repositório com roteiros e exemplos de como utilizar JUnit no seu dia a dia como dev. Foi utilizado para os exemplos a versão 5.8.2, mais atual até o momento (29/01/2022).

### Roteiros

**About:** Repositório com exemplos utilizando JUnit 5

- Readme
- MIT License
- 0 stars
- 1 watching
- 0 forks

**Releases:** No releases published. [Create a new release](#)

**Packages:** No packages published. [Publish your first package](#)

**Languages:**

willyancaetano/junit5-exemplos

# O que vimos até aqui ?

- <https://junit.org/junit5>
- Boas práticas
- Princípios FIRST
- Pesquisar sobre automatizar testes

# Dúvidas durante o curso?

- > Fórum do curso
- > Comunidade online (Discord)



SCAN ME

