



# ASSERTJ

Grupo:

Augusto Noronha, Gabriel Lourenço e Pedro Máximo



# O que é o AssertJ?

- Framework de asserções para testes em Java;
- API rica e legível para a escrita de testes;
- Suporte a objetos Java comuns;
- Desenvolvida para simplificar as assertivas nos testes.

# Vantagens:

## **Legibilidade:**

Sintaxe fluente e encadeada, próxima da linguagem natural.

## **Mensagens de erro detalhadas:**

Indicam claramente o valor esperado e o obtido. Suporte a objetos Java comuns;

## **Integração simples:**

Funciona com JUnit, Mockito e outros frameworks de teste.

# Categorização:

Perspectiva	Classificação	Justificativa
Técnica de Teste	Caixa Preta c/ elementos de Caixa Branca	Testa métodos públicos sem ver a implementação, mas acessa atributos com extracting()
Nível de Teste	Unidade	Foca em UserService isolado usando mocks (@Mock UserRepository)
Tipo de Teste	Funcional	Verifica se cada função cumpre o comportamento esperado

# Como utilizar?

Adicionar dependência no projeto Maven/Gradle

```
<dependency>  
  <groupId>org.assertj</groupId>  
  <artifactId>assertj-core</artifactId>  
  <version>3.27.3</version>  
  <scope>test</scope>  
</dependency>
```

```
testImplementation("org.assertj:assertj-core:3.27.3")
```

# Assertivas

```
import static org.assertj.core.api.Assertions.assertThat; // main one
import static org.assertj.core.api.Assertions.atIndex; // for List assertions
import static org.assertj.core.api.Assertions.entry; // for Map assertions
import static org.assertj.core.api.Assertions.tuple; // when extracting several properties at once
import static org.assertj.core.api.Assertions.fail; // use when writing exception tests
import static org.assertj.core.api.Assertions.failBecauseExceptionWasNotThrown; // idem
import static org.assertj.core.api.Assertions.filter; // for Iterable/Array assertions
import static org.assertj.core.api.Assertions.offset; // for floating number assertions
import static org.assertj.core.api.Assertions.anyOf; // use with Condition
import static org.assertj.core.api.Assertions.contentOf; // use with File assertions
```

```
// entry point for all assertThat methods and utility methods (e.g. entry)
import static org.assertj.core.api.Assertions.*;

// basic assertions
assertThat(frodo.getName()).isEqualTo("Frodo");
assertThat(frodo).isNotEqualTo(sauron);

// chaining string specific assertions
assertThat(frodo.getName()).startsWith("Fro")
    .endsWith("do")
    .isEqualToIgnoringCase("frodo");

// collection specific assertions (there are plenty more)
// in the examples below fellowshipOfTheRing is a List<TolkienCharacter>
assertThat(fellowshipOfTheRing).hasSize(9)
    .contains(frodo, sam)
    .doesNotContain(sauron);

// as() is used to describe the test and will be shown before the error message
assertThat(frodo.getAge()).as("check %s's age", frodo.getName()).isEqualTo(33);

// Java 8 exception assertion, standard style ...
assertThatThrownBy(() -> { throw new Exception("boom!"); }).hasMessage("boom!");
// ... or BDD style
Throwable thrown = catchThrowable(() -> { throw new Exception("boom!"); });
assertThat(thrown).hasMessageContaining("boom");
```

# ASSERTIVAS

# Estratégias de Teste Utilizadas

## ■ Estratégia 1: Particionamento por Equivalência

Divide entradas em grupos de comportamento esperado.

Exemplo aplicado ao método `findAll()`:

- Lista com usuários → retorno com elementos.
- Lista vazia → retorno vazio, sem erro.

## ■ Estratégia 2: Análise de Valor Limite

Testa extremos e exceções.

Exemplo no método `findById()`:

- ID válido → retorna usuário.
- ID inválido → lança exceção. ID inválido → lança exceção.



# Casos de Teste Implementados

## ■ Caso 1 — Lista com Usuários:

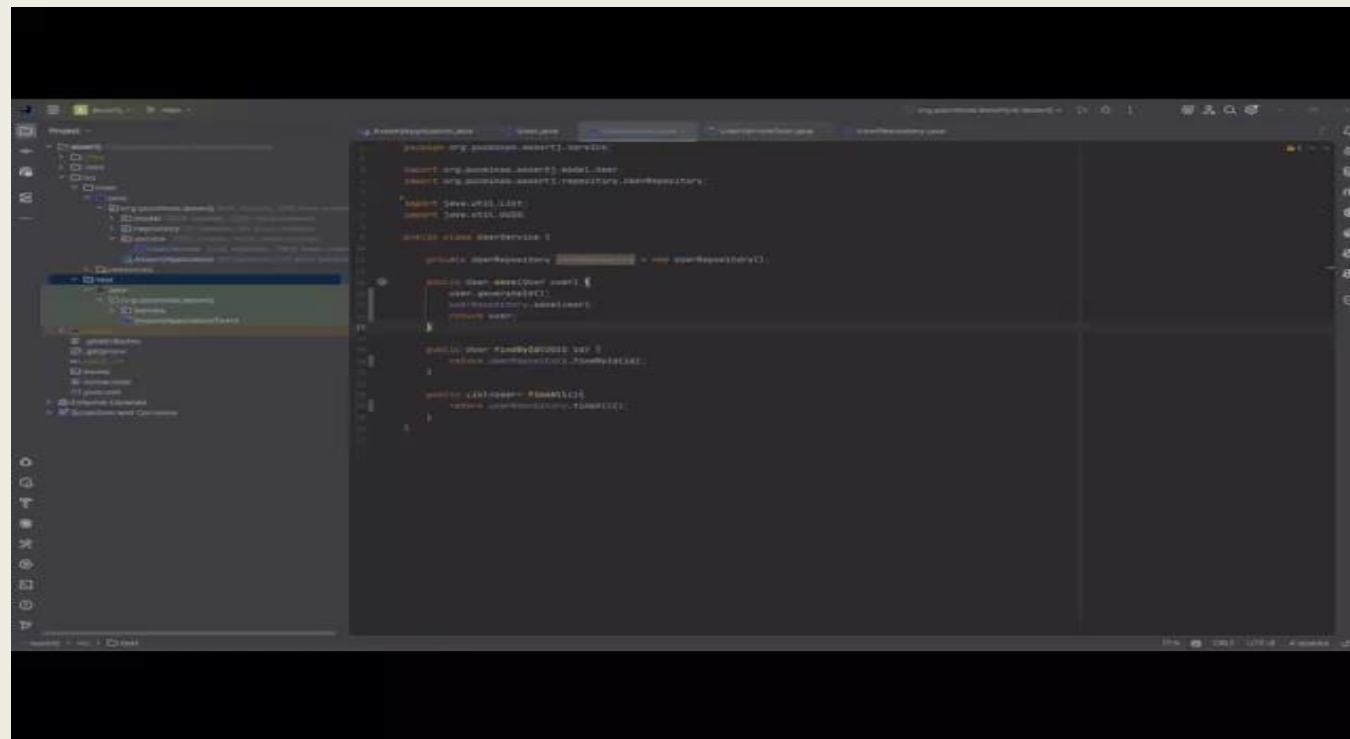
```
66
67  @Test  ▴ Gabriel Resende
68  ▶ void findAll_withSavedUsers_returnsAllUsers() {
69      var firstUser = createUser();
70      var secondUser = createUser();
71      when(userRepository.findAll()).thenReturn(List.of(firstUser, secondUser));
72
73      var result = userService.findAll();
74
75      assertThat(result)
76          .hasSize(expected: 2)
77          .containsExactlyInAnyOrder(firstUser, secondUser);
78  }
79
```

# Casos de Teste Implementados

## ■ Caso 2 — ID Inválido:

```
55
56  @Test  Gabriel Resende
57  void findById_withInvalidId_throwsIllegalArgumentException() {
58      var id = UUID.randomUUID();
59
60      when(userRepository.findById(id)).thenReturn(new IllegalArgumentException("Usuario nao encontrado para o ID: " + id));
61
62      assertThatExceptionOfType(IllegalArgumentException.class)
63          .isThrownBy(() -> userService.findById(id))
64          .withMessage("Usuario nao encontrado para o ID: " + id);
65
66  }
```

# Execução dos Testes



# Referências Bibliográficas

- Documentação Oficial: <https://assertj.github.io/doc/#assertj-overview>
- Tutorial de utilização: <https://www.baeldung.com/introduction-to-assertj>