

# JUnit

JUnit es un framework Java para implementar test en Java. JUnit 5 requiere Java 8 (o superior). JUnit se basa en [anotaciones](#):

- `@Test` : indica que el método que la contiene es un test: expected y timeout.
- `@Before` (JUnit4) / `@BeforeEach` (JUnit5): ejecuta el método que la contiene justo antes de cada test.
- `@After` (JUnit4) / `@AfterEach` (JUnit5): ejecuta el método que la contiene justo después de cada test.
- `@BeforeClass` (JUnit4) / `@BeforeAll` (JUnit5): ejecuta el método (estático) que la contiene justo antes del primer test.
- `@AfterClass` (JUnit4) / `@AfterAll` (JUnit5): ejecuta el método (estático) que la contiene justo después del último test.
- `@Ignore` / `@Disabled` : evita la ejecución del tests. No es muy recomendable su uso porque puede ocultar test fallidos. Si dudamos si el test debe estar o no, quizás borrarlo es la mejor de las decisiones.
- `@DisplayName("cadena")` (JUnit5): Declara un nombre de visualización personalizado para la clase de prueba o el método de prueba. En lugar de usar esta anotación es recomendable utilizar nombres para los métodos lo suficientemente descriptivos como para que no sea necesario usar esta anotación.

```
@DisplayName("Aserciones soportadas")
class StandardTests {

    @BeforeAll
    static void initAll() {
    }

    @BeforeEach
    void init() {
    }

    @Test
    void regular_testi_method() {
        // ...
    }

    @Test
    @DisplayName("Verdadero o falso?")
    void regular_testi_method() {
        // ...
    }

    @Test
    void failingTest() {
        fail("a failing test");
    }

    @Test
    @Disabled("este tests no se ejecuta")
    void skippedTest() {
        // not executed
    }

    @AfterEach
    void tearDown() {
    }

    @AfterAll
    static void tearDownAll() {
    }
}
```

Las condiciones de aceptación del test se implementa con las [aserciones](#). Las más comunes son los siguientes:

- **assertTrue/assertFalse (condición a testear)**: Comprueba que la condición es cierta o falsa.

- **assertEquals/assertNotEquals (valor esperado, valor obtenido)**: Es importante el orden de los valores esperado y obtenido.
- **assertNull/assertNotNull (object)**: Comprueba que el objeto obtenido es nulo o no.
- **assertSame/assertNotSame(object1, object2)**: Comprueba si dos objetos son iguales o no.
- **fail()**: Fuerza que el test termine con fallo. Se puede indicar un mensaje.

```
class AssertionsTest {

    @Test
    void standardAssertions() {
        assertEquals(2, 2);
        assertEquals(4, 4, "Ahora el mensaje opcional de la aserción es el último parámetro.");
        assertTrue(2 == 2, () -> "Al usar una lambda para indicar el mensaje, "
            + "esta se evalúa cuando se va a mostrar (no cuando se ejecuta el assert), "
            + "de esta manera se evita el tiempo de construir mensajes complejos innecesariamente.");
    }

    @Test
    void groupedAssertions() {
        // En un grupo de aserciones se ejecutan todas ellas
        // y ser reportan todas los fallos juntos
        assertAll("user",
            () -> assertEquals("Francisco", user.getFirstName()),
            () -> assertEquals("Pérez", user.getLastName())
        );
    }

    @Test
    void exceptionTesting() {
        Throwable exception = expectThrows(IllegalArgumentException.class, () -> {
            throw new IllegalArgumentException("a message");
        });
        assertEquals("a message", exception.getMessage());
    }
}
```

---

## Enlaces de interés

- [JUnit 5](#)
- <https://github.com/junit-team/junit5-samples>
- <https://www.baeldung.com/junit>
- <https://www.tutorialspoint.com/junit/index.htm>
- <https://www.vogella.com/tutorials/JUnit/article.html>

---

## Licencia



Esta obra está bajo una [licencia de Creative Commons Reconocimiento-Compartir Igual 4.0 Internacional](#).