

JUnit - Anotaciones y Métodos de Asserts Explicados

Anotaciones de JUnit

1. **@Test**

- Define un método de prueba en JUnit.
- Ejemplo:

```
```java
@Test
void testSuma() {
 assertEquals(4, 2 + 2);
}
```
```

2. **@BeforeEach**

- Se ejecuta antes de cada prueba.
- Se usa para inicializar datos antes de cada test.
- Ejemplo:

```
```java
@BeforeEach
void setUp() {
 systemUnderTest = new MiClase();
}
```
```

3. ****@AfterEach****

- Se ejecuta después de cada prueba.
- Se usa para limpiar recursos después de cada test.
- Ejemplo:

```
```java
@AfterEach
void tearDown() {
 systemUnderTest = null;
}
```
```

4. ****@BeforeAll****

- Se ejecuta ****una sola vez**** antes de todas las pruebas.
- Se usa para configurar recursos globales.
- Ejemplo:

```
```java
@BeforeAll
static void init() {
 database.connect();
}
```
```

5. ****@AfterAll****

- Se ejecuta ****una sola vez**** después de todas las pruebas.
- Se usa para limpiar recursos globales.
- Ejemplo:

```
```java
@AfterAll
static void cleanup() {
 database.disconnect();
}
```
```

6. ****@Disabled****

- Desactiva una prueba temporalmente.
- Ejemplo:

```
```java
@Disabled
@Test
void testNoEjecutado() {
 fail("Este test está deshabilitado");
}
```
```

7. ****@RepeatedTest****

- Ejecuta una prueba varias veces.
- Ejemplo:

```
```java
@RepeatedTest(5)
void testRepetido() {
 assertEquals(4, 2 + 2);
}
```
```

8. **@ParameterizedTest**

- Permite ejecutar una prueba con diferentes valores.

- Ejemplo:

```
```java
@ParameterizedTest
@ValueSource(ints = {1, 2, 3, 4})
void testConParametros(int numero) {
 assertTrue(numero > 0);
}
```
```

Métodos de Assertions en JUnit

1. **assertEquals(expected, actual)**

- Compara si dos valores son iguales.

- Ejemplo:

```
```java
assertEquals(5, 2 + 3);
```
```

2. **assertNotEquals(unexpected, actual)**

- Verifica que dos valores no sean iguales.

- Ejemplo:

```
```java
```

```
assertNotEquals(4, 2 + 3);
```

```
...
```

### 3. **assertTrue(condition)**

- Verifica que una condición sea verdadera.

- Ejemplo:

```
```java
```

```
assertTrue(5 > 2);
```

```
...
```

4. **assertFalse(condition)**

- Verifica que una condición sea falsa.

- Ejemplo:

```
```java
```

```
assertFalse(2 > 5);
```

```
...
```

### 5. **assertNull(object)**

- Verifica que un objeto sea `null`.

- Ejemplo:

```
```java
```

```
String texto = null;
```

```
assertNull(texto);
```

```
...
```

6. **assertNotNull(object)**

- Verifica que un objeto **NO** sea `null`.

- Ejemplo:

```
```java
String texto = "Hola";

assertNotNull(texto);
```
```

7. **assertSame(expected, actual)**

- Verifica que dos objetos sean el mismo en memoria.

- Ejemplo:

```
```java
String a = "Texto";

String b = a;

assertSame(a, b);
```
```

8. **assertNotSame(expected, actual)**

- Verifica que dos objetos ****NO**** sean el mismo en memoria.

- Ejemplo:

```
```java
String a = new String("Texto");

String b = new String("Texto");

assertNotSame(a, b);
```
```

9. **assertArrayEquals(expectedArray, actualArray)**

- Compara si dos arrays son iguales.

- Ejemplo:

```
```java  

int[] expected = {1, 2, 3};

int[] actual = {1, 2, 3};

assertArrayEquals(expected, actual);

```
```

10. **assertThrows(exception.class, executable)****

- Verifica que una excepción específica sea lanzada.
- Ejemplo:

```
```java  

assertThrows(ArithmeticException.class, () -> {

 int resultado = 10 / 0;

});

```
```

Métodos Adicionales en JUnit

1. **fail(message)****

- Fuerza la falla de una prueba.
- Ejemplo:

```
```java  

fail("Este test debe fallar");

```
```

2. **assumeTrue(condition)****

- Solo ejecuta la prueba si la condición es verdadera.

- Ejemplo:

```
```java
assumeTrue(System.getProperty("os.name").contains("Windows"));
```
```

3. **assumeFalse(condition)**

- Solo ejecuta la prueba si la condición es falsa.

- Ejemplo:

```
```java
assumeFalse(System.getProperty("os.name").contains("Linux"));
```
```
