

3.D. JUnit

2. Probando JUnit.

2.2. Etiquetas JUnit.

Las etiquetas o anotaciones permiten configurar las pruebas a realizar. En esta sección se describen las utilizadas con mayor frecuencia.

Tras una breve descripción de varias de las anotaciones en la siguiente tabla, se propone una nueva versión de la clase de prueba ClaseUnojUnitTest, que incluye ejemplos para que puedas ponerlas en práctica.

Etiqueta @Test	
Indica que el siguiente método será llamado para ser probado.	<pre>@Test void testDevuelveTrue() { System.out.println("Llamando a testDevuelveTrue"); }</pre>
Etiqueta Repeatedtest	
Indica que el siguiente método será llamado las veces que la etiqueta recibe como parámetro. Dos en este caso.	<pre>@RepeatedTest(2) void testRepiteTest() { System.out.println("Llamando a testRepiteTest"); }</pre>
Etiqueta @BeforeAll	
Indica que el siguiente método es llamado una sola vez en toda la ejecución del programa. A continuación son llamadas las pruebas etiquetadas con @Test, @Repeatedtest, @ParameterizedTest o @TestFactory.	<pre>@BeforeAll static void MetodoBeforeAll() { System.out.println("Llamando a MetodoBeforeAll"); }</pre>
Etiqueta @AfterAll	
Indica que el siguiente método es llamado una sola vez en toda la ejecución del programa. Antes habrán sido llamadas todas las pruebas etiquetadas con @Test, @Repeatedtest, @ParameterizedTest o @TestFactory.	<pre>@AfterAll static void MetodoAfterAll() { System.out.println("Llamando a MetodoAfterAll"); }</pre>
Etiqueta @BeforeEach	
Indica que el siguiente método es llamado antes de ejecutar cada una de las pruebas etiquetadas con @Test, @Repeatedtest, @ParameterizedTest o @TestFactory.	<pre>@BeforeEach void MetodoBeforeEach() { System.out.println("Llamando a MetodoBeforeEach"); }</pre>
Etiqueta @AfterEach	
Indica que el siguiente método es llamado después de ejecutar cada una de las pruebas etiquetadas con @Test, @Repeatedtest, @ParameterizedTest o @TestFactory.	<pre>@AfterEach void MetodoAfterEach() { System.out.println("Llamando a MetodoAfterEach"); }</pre>
Etiquetas @ParameterizedTest y @ValueSource	
Permite pasar al método de prueba parámetros a utilizar en la ejecución. Se lanza una vez por parámetro pasado.	<pre>@ParameterizedTest @ValueSource(strings = {"HOLA", "ADIOS"}) void testParameterizedTest(String sMiInput) { System.out.println("Llamando a testParameterizedTest " + sMiInput); }</pre>
Etiquetas @ParameterizedTest y @CsvSource	

<p>@ValueSource sólo permite el paso de un parámetro al método de prueba. Si necesitamos pasar varios se utiliza la etiqueta CsvSource. Se lanza una vez por cada tupla de parámetros pasada.</p>	<pre><code>@ParameterizedTest @CsvSource({"HOLA,1","ADIOS,2"}) void testParameterizedIntTest(String a, int b) { System.out.println("Llamando a testParameterizedIntTest : " + a + " --- " + b); }</code></pre>
<p>@Disabled</p>	
<p>Desactiva la llamada a un método de prueba.</p>	<pre><code>@Test @Disabled void testDesactivado() { System.out.println("Desactivada la llamada a testDesactivado"); }</code></pre>
<p>@DisplayName</p>	<p>Muestra un nombre personalizado al método bajo prueba en el reporte de pruebas. Más adelante se muestran ejemplos.</p>

El código de la ClaseUnoJUnitTest.java para esta prueba queda como sigue:

```
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Disabled;
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.RepeatedTest;
import org.junit.jupiter.api.RepetitionInfo;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.CsvSource;
import org.junit.jupiter.params.provider.ValueSource;

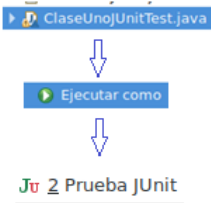
class ClaseUnoJUnitTest {
    // Llamada al metodo una sola vez antes de todos los test
    @BeforeAll
    static void MetodoBeforeAll() {
        System.out.println("Llamando a MetodoBeforeAll");
    }
    // Llamada al metodo una vez despues de todos los test
    @AfterAll
    static void MetodoAfterAll() {
        System.out.println("Llamando a MetodoAfterAll");
    }
    // Llamada al metodo antes de cada metodo de prueba
    @BeforeEach
    void MetodoBeforeEach() {
        System.out.println("Llamando a MetodoBeforeEach");
    }
    // Llamada al metodo despues de cada metodo de prueba
    @AfterEach
    void MetodoAfterEach() {
        System.out.println("Llamando a MetodoAfterEach");
    }
    // Metodo etiquetado para ser probado
    @Test
    void testDevuelveTrue() {
        System.out.println("Llamando a testDevuelveTrue");
    }
    // Metodo etiquetado para ser probado 2 veces
    @RepeatedTest(2)
    void testRepiteTest() {
        System.out.println("Llamando a testRepiteTest");
    }
    // El metodo de prueba recibe un parametro de entrada.
    // Se ejecuta tantas veces como parametros se indican.
    @ParameterizedTest
    @ValueSource(strings = {"HOLA","ADIOS"})
    void testParameterizedTest(String sMilInput) {
        System.out.println("Llamando a testParameterizedTest " + sMilInput );
    }
    // El metodo de prueba recibe varios parametros de entrada.
    // Se ejecuta tantas veces como tuplas de parametros se indican.
```

```

@ParameterizedTest
@CsvSource({"HOLA,1","ADIOS,2"})
void testParameterizedIntTest(String a, int b) {
    System.out.println("Llamando a testParameterizedIntTest : "
        + a + "--- " + b);
}
// Metodo de prueba desactivado.
@Test
@Disabled
void testDesactivado() {
    System.out.println("Desactivada la llamada a testDesactivado");
}
}

```

Las siguientes figuras muestran como lanzar las pruebas definidas en el fichero ClaseUnoJUnitTest.java y el resultado obtenido tras su ejecución.

<p>Botón derecho en la clase de test.</p> 	<pre> Llamando a MetodoBeforeAll Llamando a MetodoBeforeEach Llamando a testParameterizedIntTest : HOLA--- 1 Llamando a MetodoAfterEach Llamando a MetodoBeforeEach Llamando a testParameterizedIntTest : ADIOS--- 2 Llamando a MetodoAfterEach Llamando a MetodoBeforeEach Llamando a testParameterizedTest HOLA Llamando a MetodoAfterEach Llamando a MetodoBeforeEach Llamando a testParameterizedTest ADIOS Llamando a MetodoAfterEach Llamando a MetodoBeforeEach Llamando a testRepiteTest Llamando a MetodoAfterEach Llamando a MetodoBeforeEach Llamando a testRepiteTest Llamando a MetodoAfterEach Llamando a MetodoBeforeEach Llamando a testDevuelveTrue Llamando a MetodoAfterEach Llamando a MetodoAfterAll </pre>
---	--