

Testing unitario: JUnit

Pruebas parametrizadas

Las pruebas parametrizadas en JUnit 5 permiten ejecutar la misma prueba con diferentes conjuntos de datos de entrada, lo cual es útil para verificar el comportamiento de un método o función con diversos valores. Esto garantiza que el código sea robusto y funcione correctamente en diferentes escenarios.

Para crear una prueba parametrizada en JUnit 5, se utiliza la anotación `@ParameterizedTest` en lugar de `@Test`. Luego, se proporcionan los datos de entrada utilizando una de las siguientes anotaciones de origen de datos: `@ValueSource`, `@MethodSource`, `@CsvSource`, `@CsvFileSource`, entre otras.

Anotación	Descripción
<code>@ValueSource</code>	Especifica una matriz de valores literales de un solo tipo para ser utilizados como argumentos en las pruebas.
<pre>@ParameterizedTest @ValueSource (ints = {2, 3, 5, 7, 11, 13, 17, 19}) void testEsPrimo(int number) { assertEquals(Primos.esPrimo(number), "El número debería ser primo"); }</pre>	
<code>@MethodSource</code>	Hace referencia a un método que proporcionará los parámetros para la prueba parametrizada.
<pre>@ParameterizedTest @MethodSource("provideStrings") void testWithExplicitLocalMethodSource(String argument) { assertNotNull(argument); } // Este método debe ser estático y no debe tener parámetros. static String[] provideStrings() { return new String[]{"manzana", "banana"}; }</pre>	

@CsvSource

Permite proporcionar valores para una prueba parametrizada en formato de cadena de texto CSV (Valores separados por comas).

```
class CalculadoraTest {  
    @ParameterizedTest  
    @CsvSource({"1, 2, 3", "2, 3, 5", "3, 5, 8"})  
    void testSuma(int num1, int num2, int expectedResult) {  
        assertEquals(expectedResult, Calculadora.suma(num1, num2), "La suma debería ser correcta");  
    }  
}
```

@CsvFileSource

Similar a @CsvSource, pero los valores se leen de un archivo CSV en lugar de ser proporcionados directamente en la anotación

```
@ParameterizedTest  
@CsvFileSource(resources = "/test/day-week-data.csv", numLinesToSkip = 1) //  
suponiendo que tienes un archivo llamado day-week-data.csv en tu carpeta de  
test  
void testWithCsvFileSource(String dayWeek, Integer expectedResult) {  
    Integer result = getNumberDay(dayWeek);  
    assertEquals(expectedResult, result);  
}
```



💡 DOCUMENTACIÓN OFICIAL:

Existen más formas de realizar pruebas parametrizadas que involucran el uso de objetos, las cuales se pueden explorar en la documentación de JUnit 5 en el apartado de pruebas parametrizadas: [Documentación de JUnit 5 – Pruebas parametrizadas](#)