

```
[INFO] --- compiler:3.11.0:testCompile (default-testCompile) @ JUnitDemo3 ---
[INFO] Changes detected - recompiling the module! :dependency
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 1 source file with javac [debug target 11] to target\test-classes
[WARNING] system modules path not set in conjunction with -source 11
[INFO]
[INFO] --- surefire:2.22.1:test (default-test) @ JUnitDemo3 ---
[INFO]
[INFO] -----
[INFO]  T E S T S
[INFO] -----
[INFO] Running FactorialTest
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.064 s - in FactorialTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.894 s
[INFO] Finished at: 2023-11-16T20:23:38+01:00
[INFO] -----
```

✓	✓	FactorialTest	36 ms
✓	✓	testFactorial(int, int)	36 ms
	✓	1: factorial de 0=1	32 ms
	✓	2: factorial de 1=1	1 ms
	✓	3: factorial de 2=2	1 ms
	✓	4: factorial de 3=6	
	✓	5: factorial de 4=24	1 ms
	✓	6: factorial de 5=120	1 ms

Finished after 0,147 seconds

Runs: 6/6 Errors: 0 Failures: 0

FactorialTest [Runner: JUnit 5] (0,001 s)

- testFactorial(int, int) (0,001 s)
 - 1: factorial de 0=1 (0,001 s)
 - 2: factorial de 1=1 (0,000 s)
 - 3: factorial de 2=2 (0,000 s)
 - 4: factorial de 3=6 (0,000 s)
 - 5: factorial de 4=24 (0,000 s)
 - 6: factorial de 5=120 (0,000 s)

```

1 import org.junit.jupiter.params.ParameterizedTest;
2
3
4
5
6 class FactorialTest {
7     //Instanciamos la clase que tiene el método a testear
8     Factorial facto = new Factorial();
9
10    //Creamos el método "valores" que genera el par de valores (entrada, esperado) del test
11    private static Integer[][] valores() {
12        return new Integer[][]{
13            {0, 1}, {1, 1}, {2, 2}, {3, 6}, {4, 24}, {5, 120}
14        };
15    }
16
17    //=====Método que testea al método calcula de la clase Factorial=====//
18    // La forma de indicarle que es un test parametrizado
19    @ParameterizedTest(name = "{index}: factorial de {0}={1}")
20    //La forma de indicarle que los valores vendrán inyectados a través de un método
21    @MethodSource("valores")
22    public void testFactorial(int entrada, int esperado) {
23        assertEquals(esperado, facto.calcula(entrada));
24    }
25 }
26

```

```
id testFactorial(int entrada, int esperado) { assertEquals(esperado, facto.
```

surefire:test

> Dependencies

[INFO]

[INFO] -----

[INFO] T E S T S

[INFO] -----

[INFO] Running FactorialTest

[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.065 s - in FactorialTest

[INFO]

[INFO] Results:

[INFO]

[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0

[INFO]

[INFO] -----

[INFO] BUILD SUCCESS

[INFO] -----

[INFO] Total time: 2.156 s

[INFO] Finished at: 2023-11-16T20:15:51+01:00

[INFO] -----