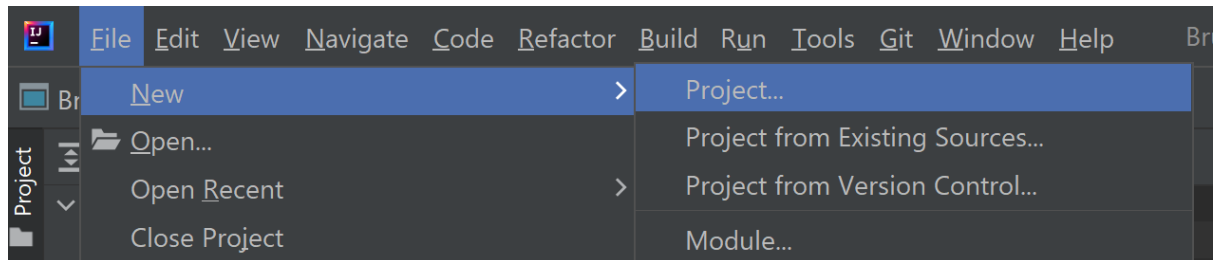


Entornos de Desarrollo

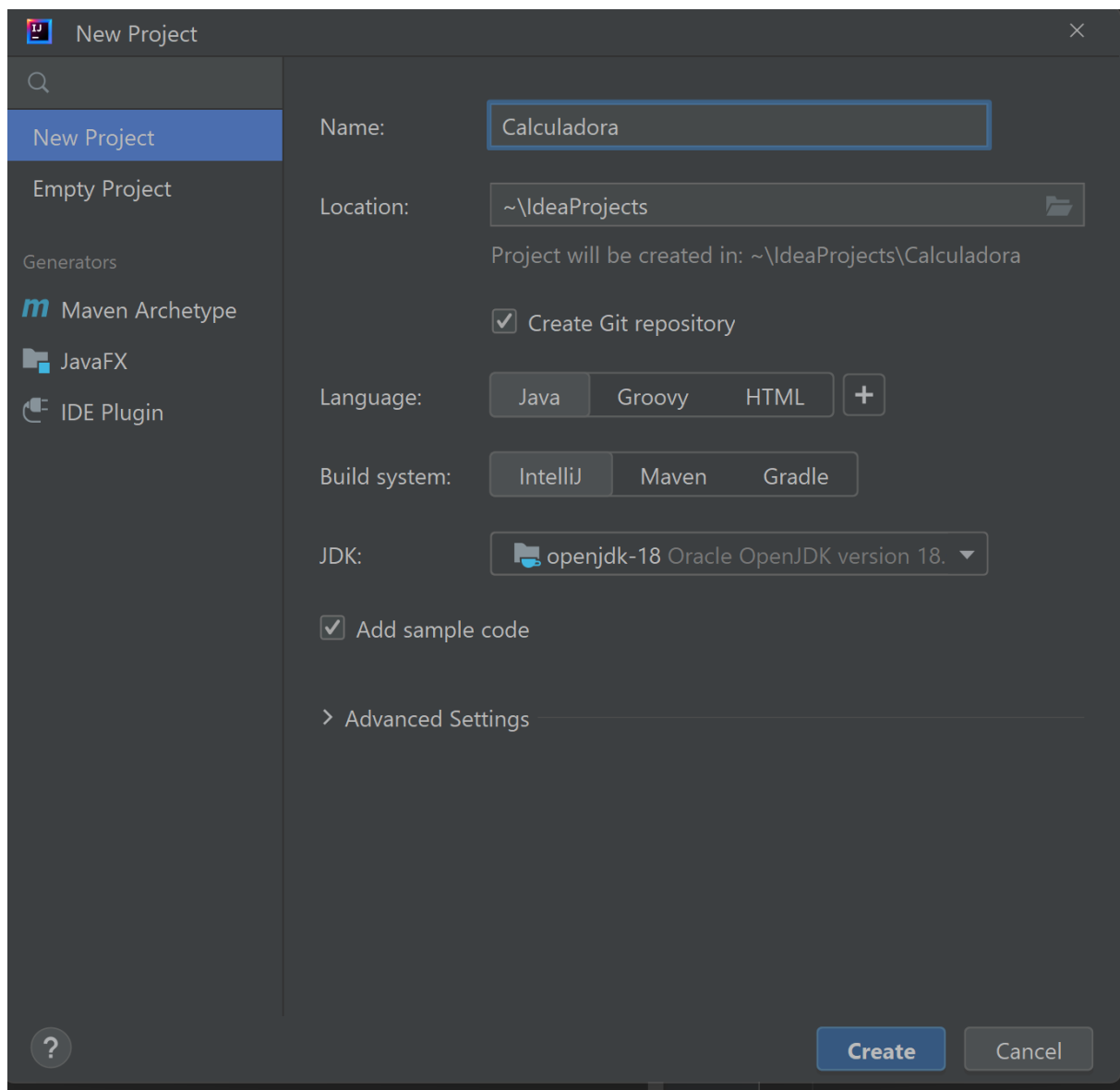
- uso de JUnit



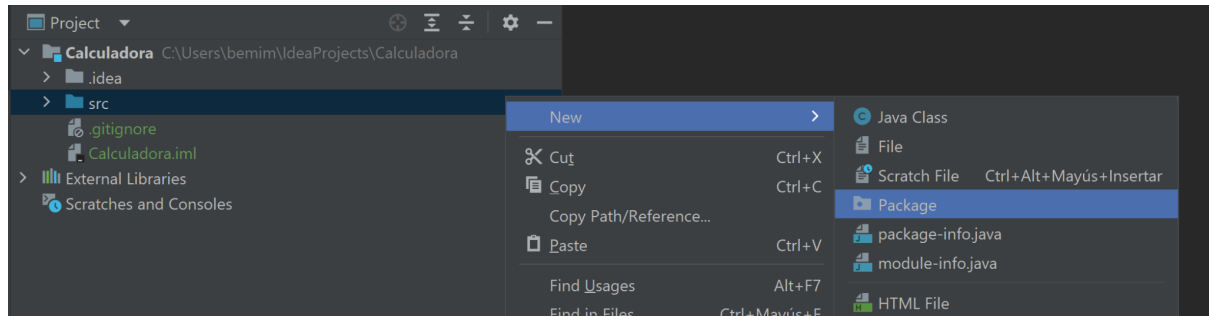
Para crear un proyecto java:



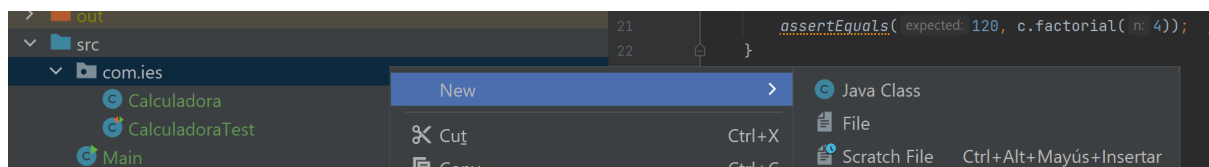
Le damos un nombre.



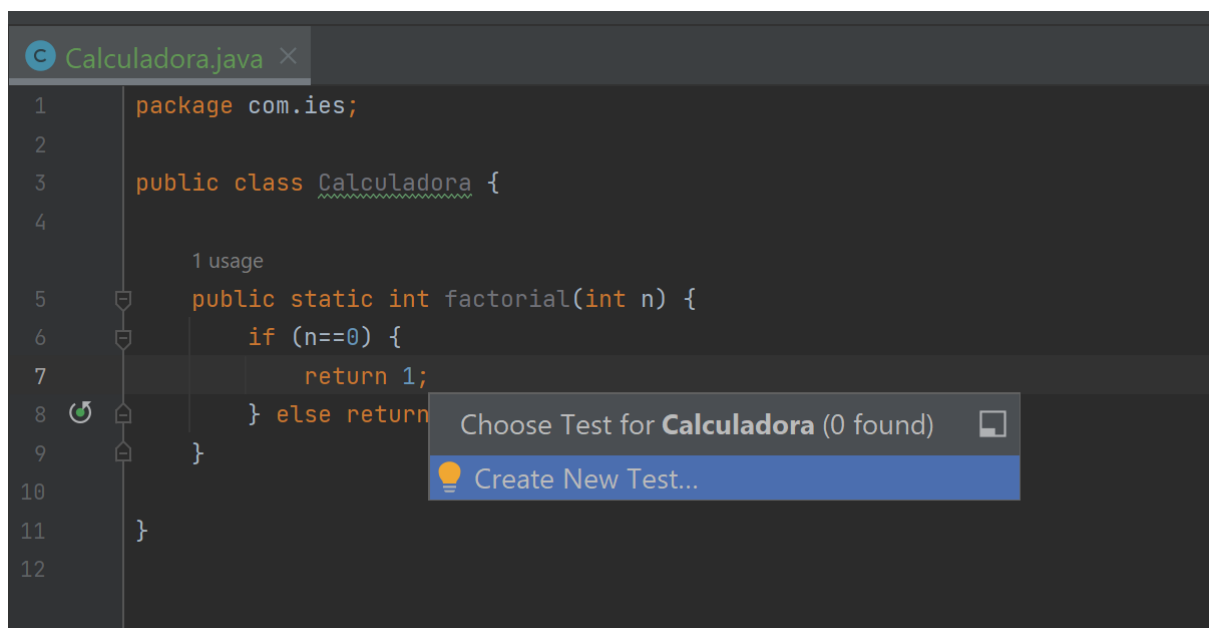
Para crear un paquete dentro del proyecto.



Así se crearía una clase dentro del paquete. La clase la pasaremos a llamar Calculadora en este caso.

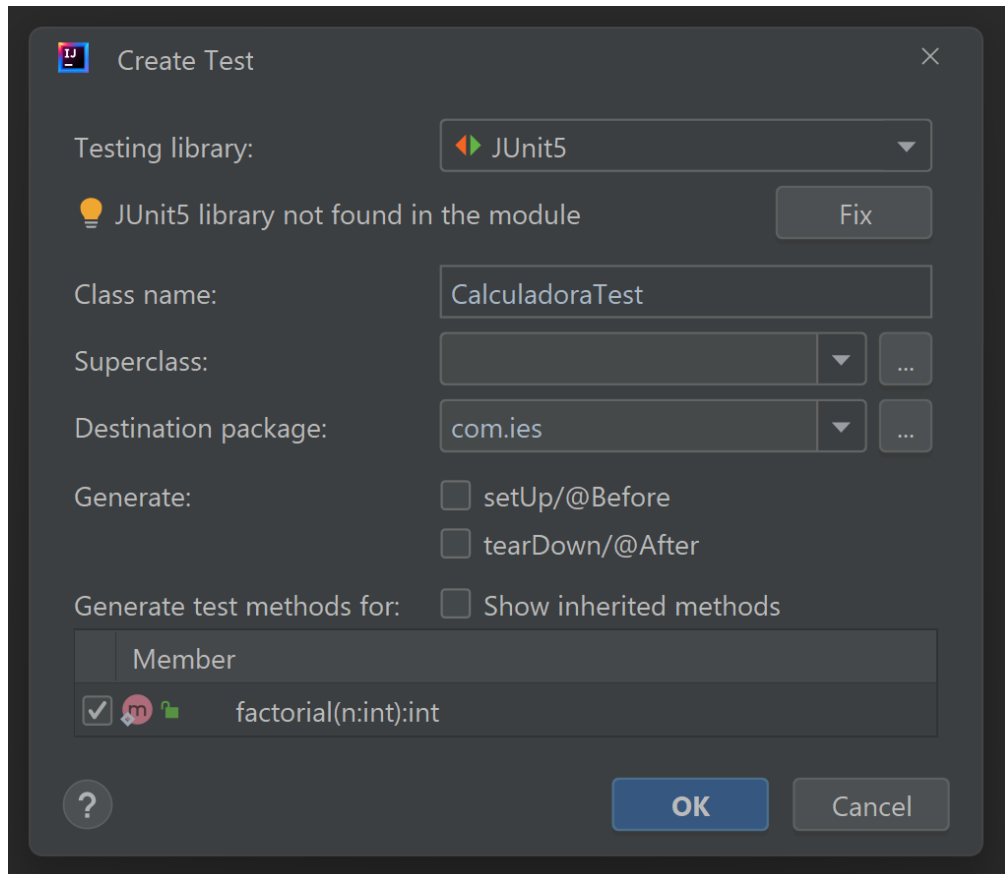


Para crear una clase test de una clase hacemos control+mayus+t sobre la clase objetivo, en este caso en Calculadora.

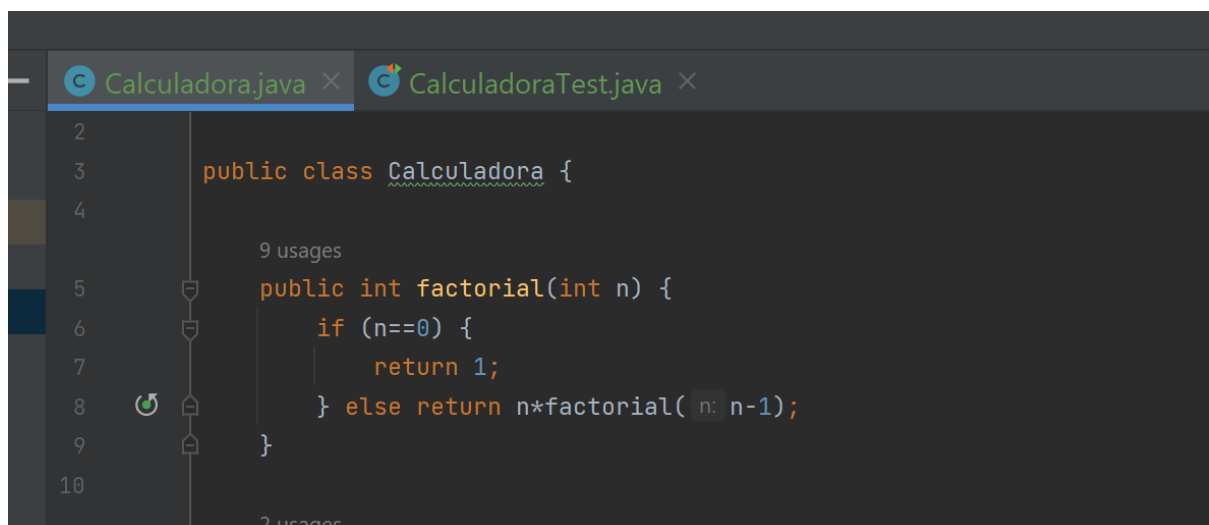


Seleccionamos una versión de JUnit, en este caso he elegido la 5.

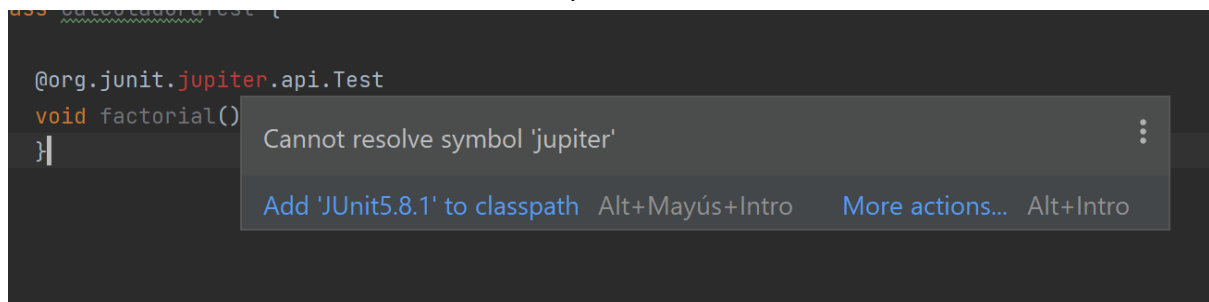
Le damos un nombre a la clase test (suele llamarse NombreDeLaClaseTest)



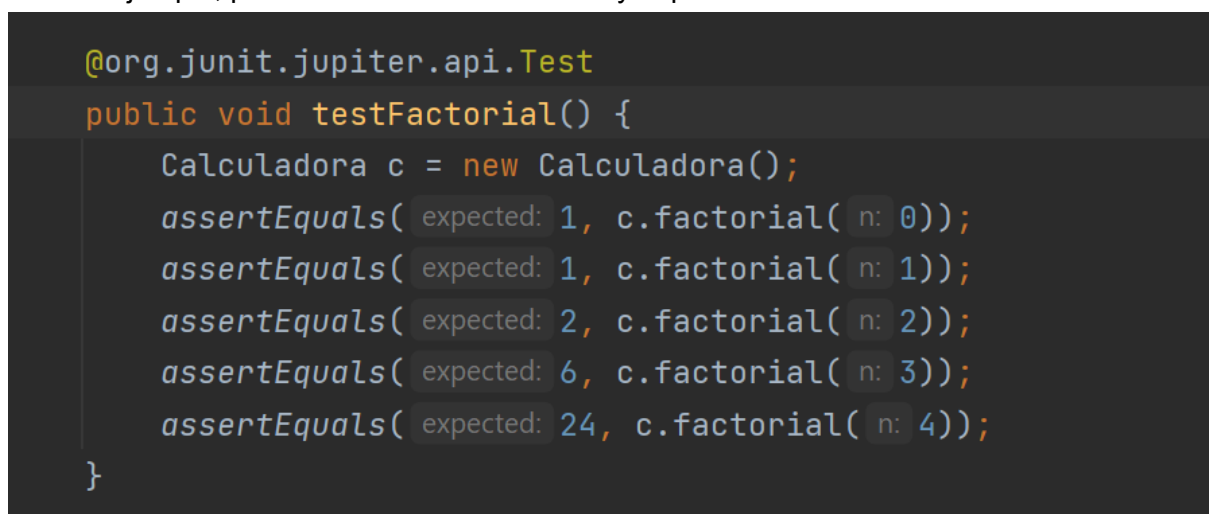
Este es el método a testear. Como se puede observar, nos ha creado la clase CalculadoraTest.



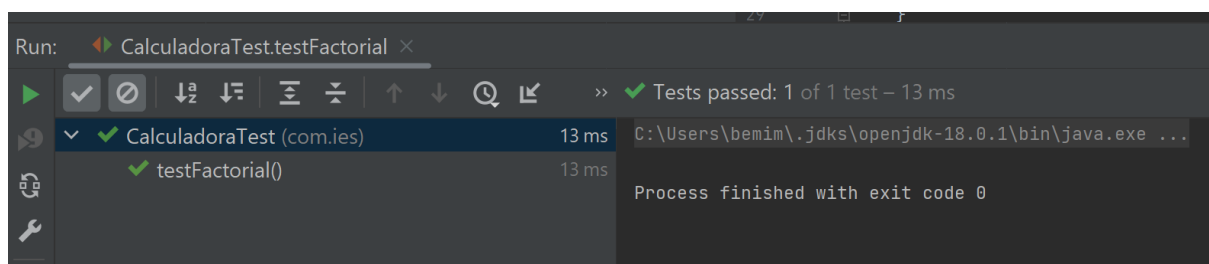
A continuación añadimos JUnit5.8.1 a classpath.



En este ejemplo, probamos el método factorial y esperamos un resultado exitoso.



Y así es.

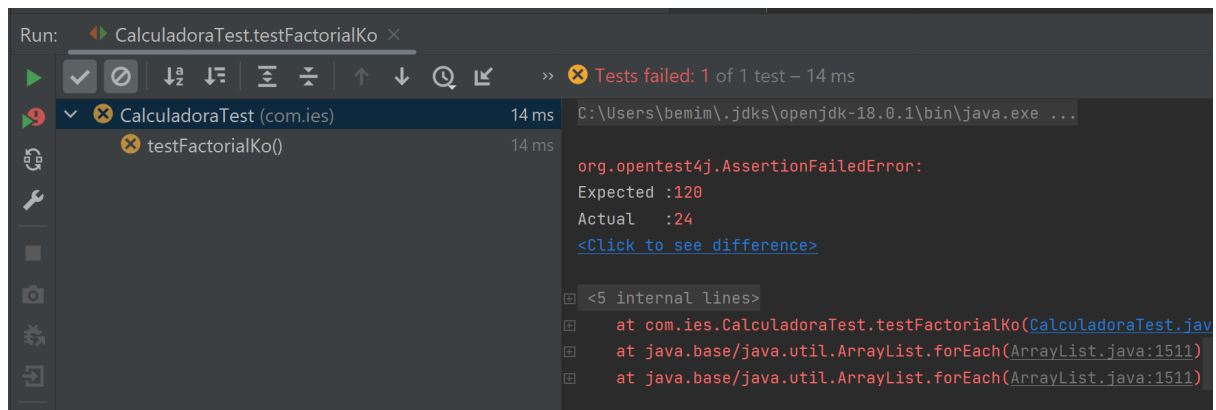


Caso de prueba incorrecto intencional

```
@org.junit.jupiter.api.Test
public void testFactorialKo() {
    Calculadora c = new Calculadora();

    assertEquals( expected: 120, c.factorial( n: 4)); // Caso de prueba incorrecto intencional
}
```

Resultado:

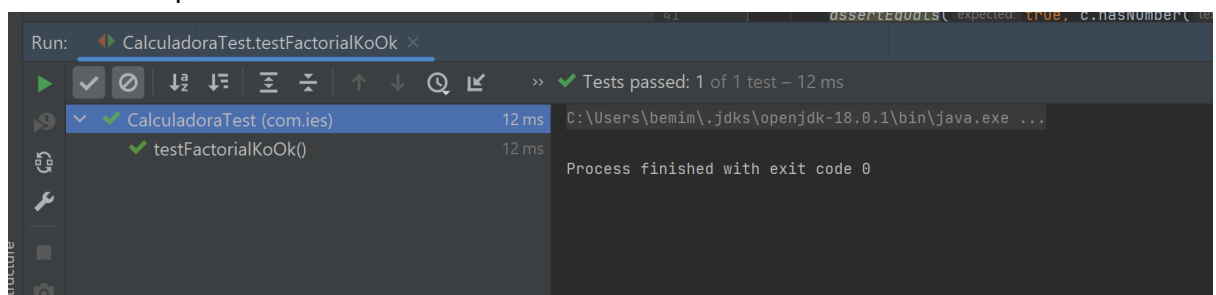


Caso de prueba incorrecto intencional. Le indicamos que la diferencia será de 2 -> (3! - resultado esperado = 2)

```
@org.junit.jupiter.api.Test
public void testFactorialKoOk() {
    Calculadora c = new Calculadora();

    assertEquals( expected: 4, c.factorial( n: 3), delta: 2);
}
```

Resultado esperado.

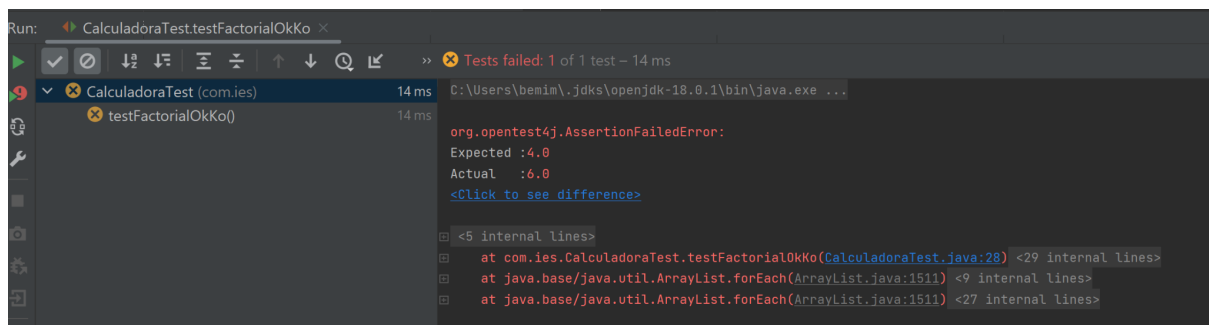


Le indicamos que el siguiente caso devolverá un valor correcto. Spoiler: será incorrecto.

```
@org.junit.jupiter.api.Test
public void testFactorialOkKo() {
    Calculadora c = new Calculadora();

    assertEquals(expected: 4, c.factorial( n: 3), delta: 0);
}
```

Resultado esperado. El resultado es 6, no 4.



```
Run: CalculadoraTest.testFactorialOkKo
Tests failed: 1 of 1 test - 14 ms
CalculadoraTest (com.ies) 14 ms
testFactorialOkKo() 14 ms
org.opentest4j.AssertionFailedError:
Expected :4.0
Actual   :6.0
<Click to see difference>
<5 internal lines>
at com.ies.CalculadoraTest.testFactorialOkKo(CalculadoraTest.java:28) <29 internal lines>
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511) <9 internal lines>
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511) <27 internal lines>
```

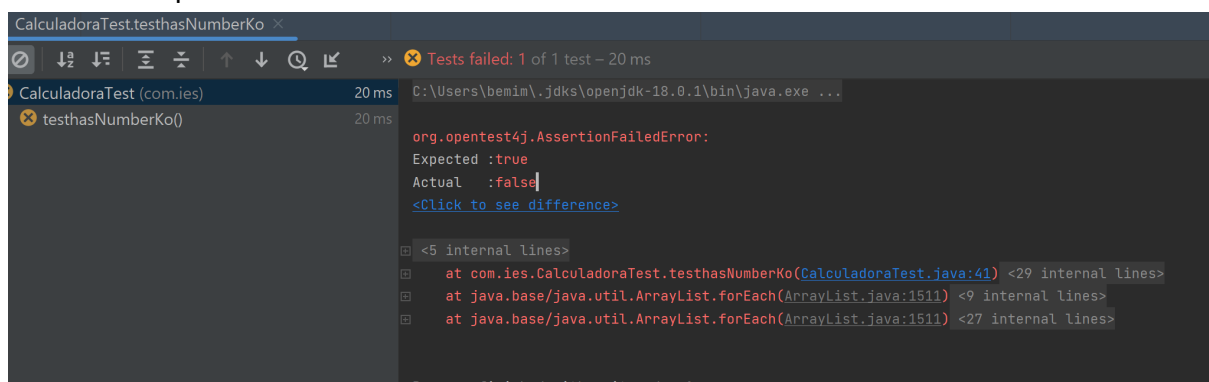
Creo otro método para testearlo. Este método devuelve true si el texto enviado por parámetro contiene un número.

```
1 usage
public boolean hasNumber(String text) {
    String numbers = "0123456789";
    for (int i = 0; i < text.length(); i++) {
        for (int j = 0; j < numbers.length(); j++) {
            if (text.charAt(i) == numbers.charAt(j)) {
                return true;
            }
        }
    }
    return false;
}
```

Caso de prueba para que de un resultado incorrecto. Debería devolver false ya que “hola” no contiene números

```
@org.junit.jupiter.api.Test
public void testhasNumberKo() {
    Calculadora c = new Calculadora();
    assertEquals( expected: true, c.hasNumber( text: "hola"));
}
```

Resultado esperado.

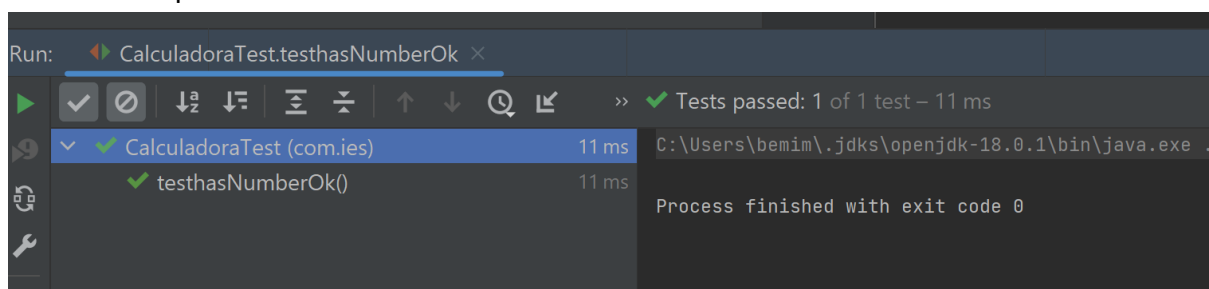


The screenshot shows the IDE's test results window. The test `CalculadoraTest.testhasNumberKo()` failed. The error message is `org.opentest4j.AssertionFailedError: Expected :true Actual :false`. The stack trace shows the test method `testhasNumberKo` at line 41 of `CalculadoraTest.java`, and the `assertEquals` method at line 1511 of `ArrayList.java`.

Caso de prueba para que el resultado sea correcto. “hola123” contiene letras, debería devolver true.

```
@org.junit.jupiter.api.Test
public void testhasNumberOk() {
    Calculadora c = new Calculadora();
    assertEquals( expected: true, c.hasNumber( text: "hola123"));
}
```

Resultado esperado.



The screenshot shows the IDE's test results window. The test `CalculadoraTest.testhasNumberOk()` passed. The status bar indicates `Tests passed: 1 of 1 test - 11 ms`. The stack trace shows the test method `testhasNumberOk` at line 11 of `CalculadoraTest.java`, and the `assertEquals` method at line 1511 of `ArrayList.java`.