

PRO 5.2. Junit II

1. Métodos Test

assertEquals()

```
@Test
public void testResta() {
    System.out.println(x: "resta");

    Calculadora calc1 = new Calculadora(num1: 20, num2: 10);
    int resultado = calc1.resta();
    int resultadoEsperado = 10;
    assertEquals(expected: resultadoEsperado, actual: resultado);
}
```

assertTrue()

```
@Test
public void testResta2() {

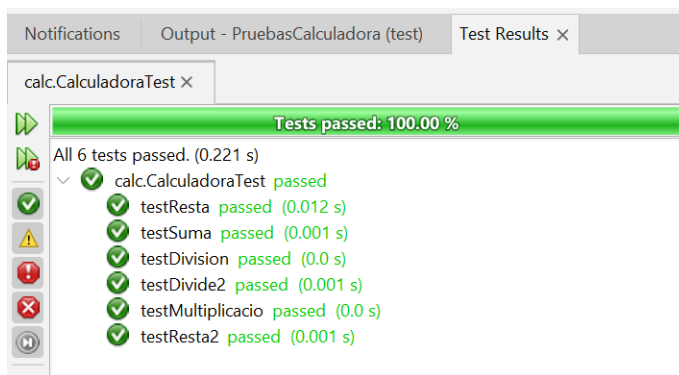
    Calculadora calc = new Calculadora(num1: 4, num2: 2);
    boolean result = calc.resta2();
    assertTrue(condition: result);

}
```

assertNull()

```
@Test
public void testDivide2() {
    Calculadora calc = new Calculadora(num1: 4, num2: 0);
    String resultado = calc.divide2();
    assertNull(object: resultado);
}
```

Resultado de los Tests



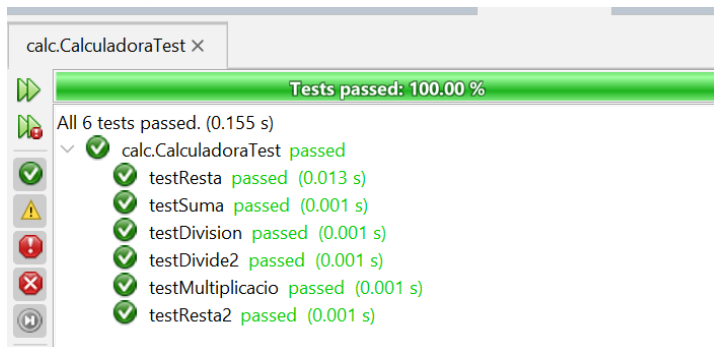
Para usar `assertNotNull()` y `assertFalse()` los códigos serían:

```
@Test
public void testResta2() {

    Calculadora calc = new Calculadora(num1:2, num2:4);
    boolean result = calc.resta2();
    assertFalse(condition: result);
}

@Test
public void testDivide2() {

    Calculadora calc = new Calculadora(num1:4, num2:2);
    String resultado = calc.divide2();
    assertNotNull(object: resultado);
}
```



2 - Fecha

Pruebas para probar la clase Fecha:

```
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62

@Test
public void testDevuelveFecha() {
    System.out.println(x: "DevuelveFecha");
    int tipo = 1;
    Fecha instance = new Fecha();
    String expectedResult = "2024/02";
    String result = instance.DevuelveFecha(tipo);
    assertEquals(expected: expectedResult, actual: result);
}

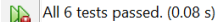
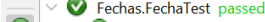
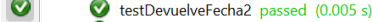
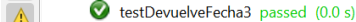
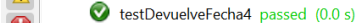
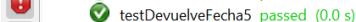
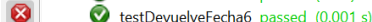
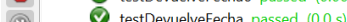
@Test
public void testDevuelveFecha2() {
    System.out.println(x: "DevuelveFecha");
    int tipo = 2;
    Fecha instance = new Fecha();
    String expectedResult = "02/2024";
    String result = instance.DevuelveFecha(tipo);
    assertEquals(expected: expectedResult, actual: result);
}
```

```

64     @Test
65     public void testDevuelveFecha3() {
66         System.out.println(x: "DevuelveFecha");
67         int tipo = 3;
68         Fecha instance = new Fecha();
69         String expectedResult = "02/24";
70         String result = instance.DevuelveFecha(tipo);
71         assertEquals(expected: expectedResult, actual: result);
72     }
73
74
75     @Test
76     public void testDevuelveFecha4() {
77         System.out.println(x: "DevuelveFecha");
78         int tipo = 4;
79         Fecha instance = new Fecha();
80         String expectedResult = "ERROR";
81         String result = instance.DevuelveFecha(tipo);
82         assertEquals(expected: expectedResult, actual: result);
83     }
84
85
86     @Test
87     public void testDevuelveFecha5() {
88         System.out.println(x: "DevuelveFecha");
89         int tipo = 0;
90         Fecha instance = new Fecha();
91         String expectedResult = "ERROR";
92         String result = instance.DevuelveFecha(tipo);
93         assertEquals(expected: expectedResult, actual: result);
94     }
95
96
97     @Test
98     public void testDevuelveFecha6() {
99         System.out.println(x: "DevuelveFecha");
100        int tipo = -2;
101        Fecha instance = new Fecha();
102        String expectedResult = "ERROR";
103        String result = instance.DevuelveFecha(tipo);
104        assertEquals(expected: expectedResult, actual: result);
105    }
106

```

Resultado de las pruebas:

calc.CalculadoraTest ×	edd_fecha.FechaTest ×	Fechas.FechaTest ×
Tests passed: 100.00 %		
<div>  </div>		
<div>  </div>		
<div>  </div>		
<div>  </div>		
<div>  </div>		
<div>  </div>		
<div>  </div>		
<div>  </div>		
		<div> DevuelveFecha 2 DevuelveFecha 3 DevuelveFecha error DevuelveFecha error DevuelveFecha error DevuelveFecha 1 </div>

3 - Factorial

Pruebas para probar la clase:

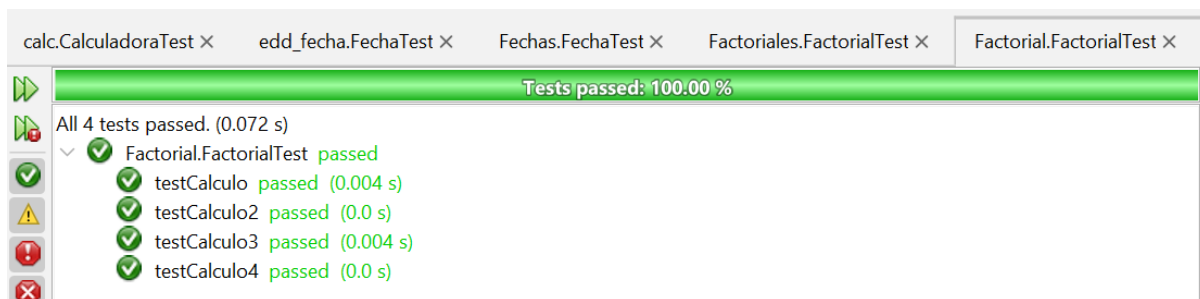
```
@Test
public void testCalculo() {
    System.out.println(x: "calculo");
    Factorial instance = new Factorial(num:3);
    long expResult = 6L;
    long result = instance.calculo();
    assertEquals(expected: expResult, actual: result);
}

@Test
public void testCalculo2() {
    System.out.println(x: "calculo");
    Factorial instance = new Factorial(num:6);
    long expResult = 720L;
    long result = instance.calculo();
    assertEquals(expected: expResult, actual: result);
}

@Test (expected = Exception.class)
public void testCalculo3() {
    System.out.println(x: "calculo");
    Factorial instance = new Factorial(num:-3);
    instance.calculo();
}

@Test
public void testCalculo4() {
    System.out.println(x: "calculo");
    Factorial instance = new Factorial(num:0);
    long expResult = 1L;
    long result = instance.calculo();
    assertEquals(expected: expResult, actual: result);
}
```

Resultado de los tests:



The screenshot shows an IDE's test runner interface. At the top, there are tabs for different test classes: 'calc.CalculadoraTest', 'edd_fecha.FechaTest', 'Fechas.FechaTest', 'Factoriales.FactorialTest', and 'Factorial.FactorialTest'. The 'Factorial.FactorialTest' tab is selected. Below the tabs, a green progress bar indicates 'Tests passed: 100.00 %'. Underneath the progress bar, it says 'All 4 tests passed. (0.072 s)'. A list of tests is shown with green checkmarks and the word 'passed' in green text:

- Factorial.FactorialTest passed
 - testCalculo passed (0.004 s)
 - testCalculo2 passed (0.0 s)
 - testCalculo3 passed (0.004 s)
 - testCalculo4 passed (0.0 s)