

Actividad PRO 5.1 Junit

Código de la clase “miCalculadora”

```
Source History
package calc;

public class miCalculadora {

    private int num1;
    private int num2;
    private int num3;

    public miCalculadora(int num1, int num2, int num3){
        this.num1 = num1;
        this.num2 = num2;
        this.num3 = num3;
    }

    public int cartesiano(){
        return num1*num2*num3;
    }

    public int sumatriple(){
        return num1+num2+num3;
    }

    public float media(){
        return (float) sumatriple()/3;
    }

    public int mayorDeLosTres(){
        int max = num1;

        if (num2 > num1){
            max = num2;
        }

        if (num3 > num2){
            max = num3;
        }

        return max;
    }
}
```

Código de la clase Junit



```
1
2  package calc;
3
4  import org.junit.After;
5  import org.junit.AfterClass;
6  import org.junit.Before;
7  import org.junit.BeforeClass;
8  import org.junit.Test;
9  import static org.junit.Assert.*;
10
11
12  public class miCalculadoraTest {
13
14      public miCalculadoraTest() {
15      }
16
17      @BeforeClass
18      public static void setUpClass() {
19      }
20
21      @AfterClass
22      public static void tearDownClass() {
23      }
24
25      @Before
26      public void setUp() {
27
28
29      @After
30      public void tearDown() {
31      }
32  }
```

```

33  /**
34   * Test of cartesiano method, of class miCalculadora.
35   */
36  @Test
37  public void testCartesiano() {
38      System.out.println(x: "cartesiano");
39      /*miCalculadora instance = null;
40      int expResult = 0;
41      int result = instance.cartesiano();
42      assertEquals(expResult, result);
43      // TODO review the generated test code and remove the default call to fail.
44      fail("The test case is a prototype.");*/

45
46      miCalculadora calc1 = new miCalculadora(num1: 2, num2: 3, num3: 4);
47      int resultado = calc1.cartesiano();
48      int resEsperado = 24;
49      assertEquals(expected: resEsperado, actual: resultado);
50  }
51

```

```

52  /**
53   * Test of sumatriple method, of class miCalculadora.
54   */
55  @Test
56  public void testSumatriple() {
57      System.out.println(x: "sumatriple");
58      /*miCalculadora instance = null;
59      int expResult = 0;
60      int result = instance.sumatriple();
61      assertEquals(expResult, result);
62      // TODO review the generated test code and remove the default call to fail.
63      fail("The test case is a prototype.");*/

64
65      miCalculadora calc1 = new miCalculadora(num1: 2, num2: 3, num3: 4);
66      int resultado = calc1.sumatriple();
67      int resEsperado = 9;
68      assertEquals(expected: resEsperado, actual: resultado);
69  }

```

```

70  /**
71   * Test of media method, of class miCalculadora.
72   */
73  @Test
74  public void testMedia() {
75      System.out.println(x: "media");
76      /*miCalculadora instance = null;
77      float expResult = 0.0F;
78      float result = instance.media();
79      assertEquals(expResult, result, 0);
80      // TODO review the generated test code and remove the default call to fail.
81      fail("The test case is a prototype.");*/

82
83      miCalculadora calc1 = new miCalculadora (num1: 2, num2: 3, num3: 4);
84      float resultado = calc1.media();
85      float resEsperado = 3;
86      assertEquals(expected: resEsperado, actual: resultado, delta: 1);
87  }
88
89

```

```

93      @Test
94      public void testMayorDeLosTres() {
95          System.out.println(x: "mayorDeLosTres");
96          /*miCalculadora instance = null;
97          int expResult = 0;
98          int result = instance.mayorDeLosTres();
99          assertEquals(expResult, result);
100          // TODO review the generated test code and remove the default call to fail.
101          fail("The test case is a prototype.");*/
102
103          miCalculadora calc1 = new miCalculadora(num1: 2, num2: 3, num3: 4);
104          int resultado = calc1.mayorDeLosTres();
105          int resEsperado = 4;
106          assertEquals(expected: resEsperado, actual: resultado);
107      }
108
109  }

```

Resultado del Test

Test Results

calc.CalculadoraTest ×

calc.miCalculadoraTest ×

▶▶

▶▶

✓

✓

⚠

✖

⏮

⏭

>>

Tests passed: 100.00 %

All 4 tests passed. (0.1 s)

✓ calc.miCalculadoraTest passed

✓ testMedia passed (0.0 s)

✓ testMayorDeLosTres passed (0.0 s)

✓ testSumatriple passed (0.0 s)

✓ testCartesiano passed (0.0 s)

media

mayorDeLosTres

sumatriple

cartesiano

Output

Notifications

Test Results