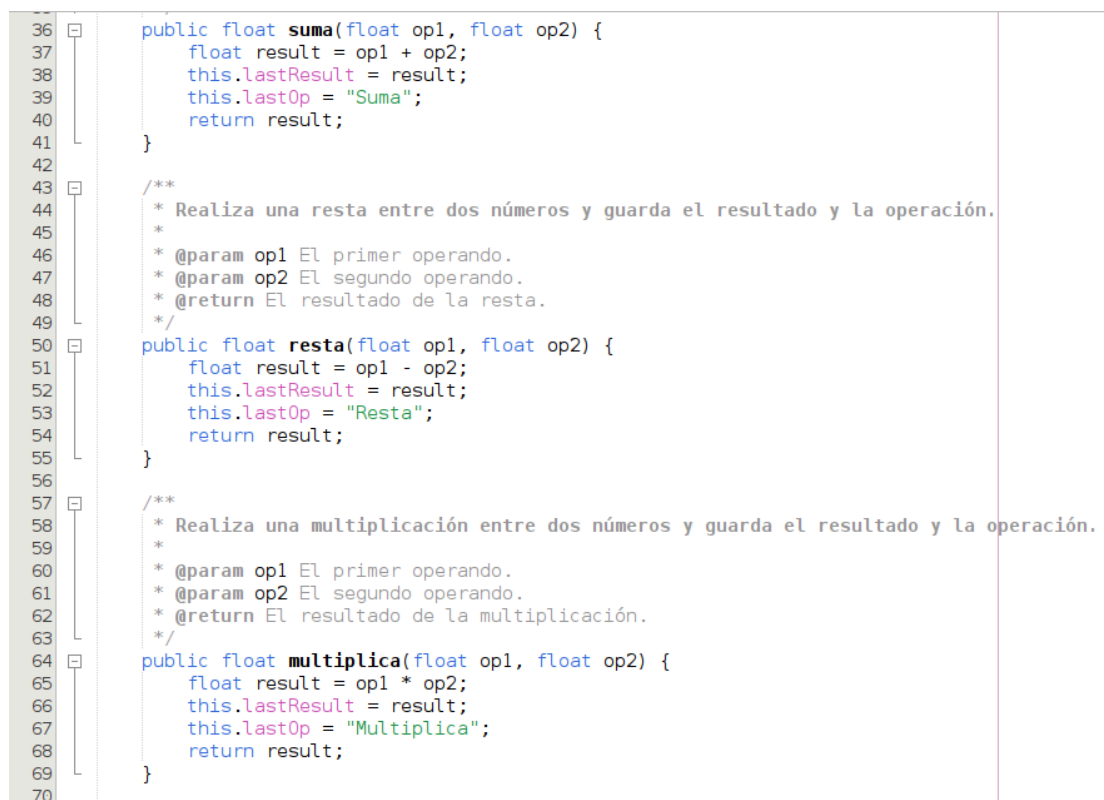


```

1  /**
2   * Esta clase representa una calculadora simple que realiza operaciones aritméticas básicas
3   * y guarda el resultado y la última operación realizada.
4   *
5   * @author Felip
6   */
7  public class Calculadora {
8      private float lastResult;
9      private String lastOp;
10
11      /**
12       * Obtiene el último resultado calculado.
13       *
14       * @return El último resultado calculado.
15       */
16      public float getLastResult() {
17          return this.lastResult;
18      }
19
20      /**
21       * Obtiene la última operación realizada.
22       *
23       * @return La última operación realizada.
24       */
25      public String getLastOp() {
26          return this.lastOp;
27      }
28
29      /**
30       * Realiza una suma entre dos números y guarda el resultado y la operación.
31       *
32       * @param op1 El primer operando.
33       * @param op2 El segundo operando.
34       * @return El resultado de la suma.
35       */

```

Captura archivo Calculadora comentado con javadoc



```

36      public float suma(float op1, float op2) {
37          float result = op1 + op2;
38          this.lastResult = result;
39          this.lastOp = "Suma";
40          return result;
41      }
42
43      /**
44       * Realiza una resta entre dos números y guarda el resultado y la operación.
45       *
46       * @param op1 El primer operando.
47       * @param op2 El segundo operando.
48       * @return El resultado de la resta.
49       */
50      public float resta(float op1, float op2) {
51          float result = op1 - op2;
52          this.lastResult = result;
53          this.lastOp = "Resta";
54          return result;
55      }
56
57      /**
58       * Realiza una multiplicación entre dos números y guarda el resultado y la operación.
59       *
60       * @param op1 El primer operando.
61       * @param op2 El segundo operando.
62       * @return El resultado de la multiplicación.
63       */
64      public float multiplica(float op1, float op2) {
65          float result = op1 * op2;
66          this.lastResult = result;
67          this.lastOp = "Multiplica";
68          return result;
69      }
70

```

Captura archivo Calculadora comentado con javadoc

```

71  |  /**
72  |  * Realiza una división entre dos números y guarda el resultado y la operación.
73  |  *
74  |  * @param op1 El primer operando (dividendo).
75  |  * @param op2 El segundo operando (divisor).
76  |  * @return El resultado de la división.
77  |  */
78  |  public float divideix(float op1, float op2) {
79  |      float result = op1 / op2;
80  |      this.lastResult = result;
81  |      this.lastOp = "Divideix";
82  |      return result;
83  |  }
84  |
85  |  /**
86  |  * Comprueba si el primer número es mayor que el segundo.
87  |  *
88  |  * @param op1 El primer número.
89  |  * @param op2 El segundo número.
90  |  * @return true si el primer número es mayor que el segundo, false de lo contrario.
91  |  */
92  |  public boolean mayorQue(float op1, float op2) {
93  |      return op1 > op2;
94  |  }
95  |
96  |  /**
97  |  * Restablece el último resultado y la última operación realizada.
98  |  */
99  |  public void restablecer() {
100 |      this.lastResult = 0;
101 |      this.lastOp = "Ninguna";
102 |  }
103 |  }
104 |

```

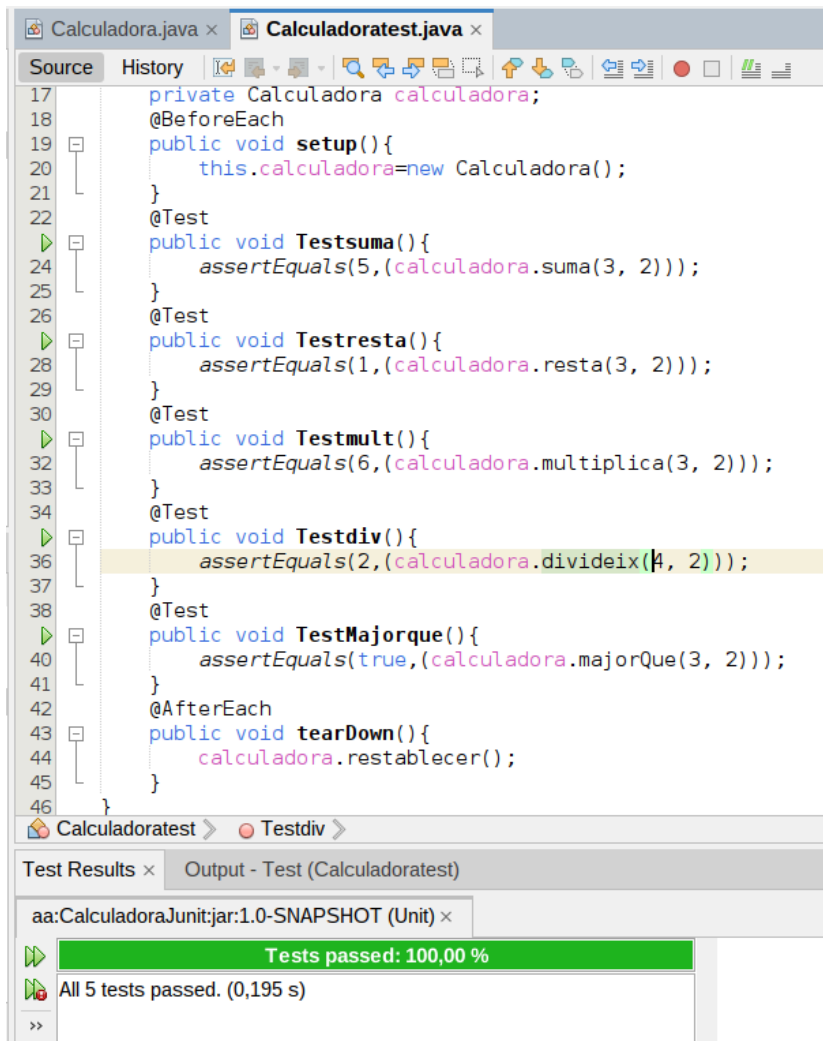
Captura archivo Calculadora comentado con javadoc

```

14  |  * @author Felip
15  |  */
16  |  public class Calculadortest {
17  |      private Calculadora calculadora;
18  |      @BeforeEach
19  |      public void setup(){
20  |          this.calculadora=new Calculadora();
21  |      }
22  |      @Test
23  |      public void Testsuma(){
24  |          assertEquals(5,(calculadora.suma(3, 2)));
25  |      }
26  |      @Test
27  |      public void Testresta(){
28  |          assertEquals(1,(calculadora.resta(3, 2)));
29  |      }
30  |      @Test
31  |      public void Testmult(){
32  |          assertEquals(6,(calculadora.multiplica(3, 2)));
33  |      }
34  |      @Test
35  |      public void Testdiv(){
36  |          assertEquals(2,(calculadora.divideix(4, 2)));
37  |      }
38  |      @Test
39  |      public void TestMayorque(){
40  |          assertEquals(true,(calculadora.mayorQue(3, 2)));
41  |      }
42  |      @AfterEach
43  |      public void tearDown(){
44  |          calculadora.restablecer();
45  |      }
46  |  }

```

Captura archivo testjava



```
17 private Calculadora calculadora;  
18 @BeforeEach  
19 public void setup(){  
20     this.calculadora=new Calculadora();  
21 }  
22 @Test  
23 public void Testsuma(){  
24     assertEquals(5,(calculadora.suma(3, 2)));  
25 }  
26 @Test  
27 public void Testresta(){  
28     assertEquals(1,(calculadora.resta(3, 2)));  
29 }  
30 @Test  
31 public void Testmult(){  
32     assertEquals(6,(calculadora.multiplica(3, 2)));  
33 }  
34 @Test  
35 public void Testdiv(){  
36     assertEquals(2,(calculadora.divideix(4, 2)));  
37 }  
38 @Test  
39 public void TestMajorque(){  
40     assertEquals(true,(calculadora.majorQue(3, 2)));  
41 }  
42 @AfterEach  
43 public void tearDown(){  
44     calculadora.restablecer();  
45 }  
46 }
```

Test Results × Output - Test (Calculadortest)

aa:CalculadoraJUnit:jar:1.0-SNAPSHOT (Unit) ×

Tests passed: 100,00 %

All 5 tests passed. (0,195 s)

>>

Ejecución test de Junit