

Universidad del Valle de Guatemala

Departamento de Computación

Algoritmos y Estructuras de Datos



-Hoja de Trabajo 2-

Javier Ramírez | 21600

UML

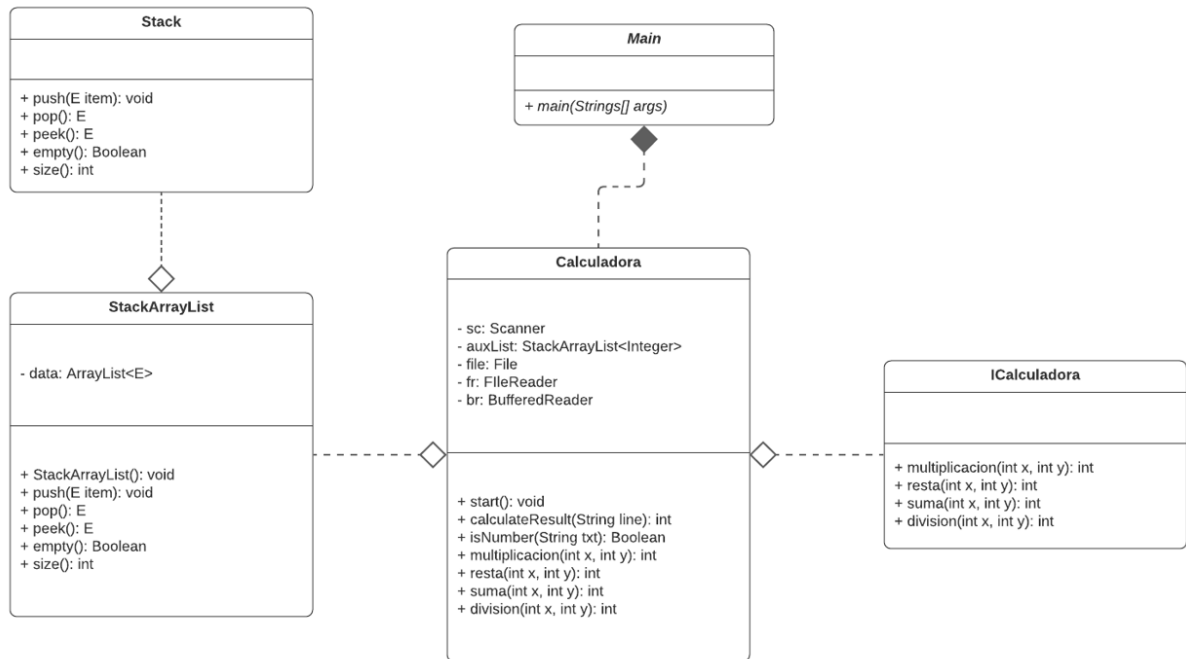
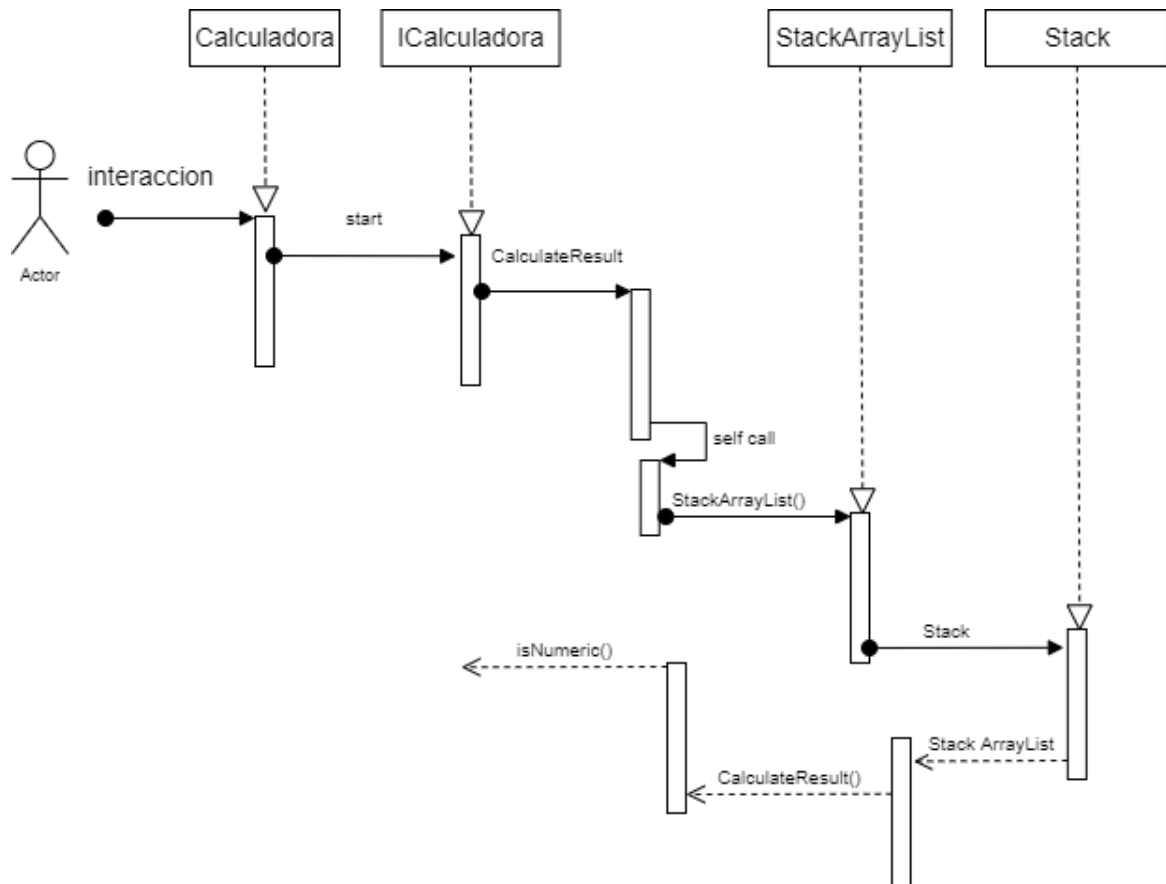


Diagrama de Secuencia



Test

The screenshot displays an IDE with a Java file named `TestProgram.java` and its test results. The code defines a `TestProgram` class with two test methods: `testCalculadora()` and `testStackPush()`. Both tests are annotated with `@Test` and `Run Test | Debug Test | ✓`. The `testCalculadora()` method creates a `Calculadora` object, calculates the result of `"6 2 3 + *"`, and asserts it equals 30. The `testStackPush()` method creates a `StackArrayList<Integer>` object, pushes integers 1 through 5, and asserts its size is 5. The test results panel on the right shows that all tests passed successfully.

```
1 import static org.junit.jupiter.api.Assertions.assertEquals;
2
3 import org.junit.Test;
4
5
6 Run Test | Debug Test | ✓
7 public class TestProgram {
8
9     @Test
10    Run Test | Debug Test | ✓
11    public void testCalculadora(){
12        Calculadora calc = new Calculadora();
13        int result = calc.calculateResult("6 2 3 + *");
14        assertEquals(result, 30);
15    }
16
17    @Test
18    Run Test | Debug Test | ✓
19    public void testStackPush(){
20        StackArrayList<Integer> stack = new StackArrayLis
21        stack.push(1);
22        stack.push(2);
23        stack.push(3);
24        stack.push(4);
25        stack.push(5);
26        assertEquals(stack.size(), 5);
27    }
28
29    @Test
```

TestProgram

Test	Status	Time	Icon
> testStackPeek	Passed	0.01s	📄
> testStackPush	Passed	0s	📄
> testCalculadora	Passed	0.02s	📄
> testStackPop	Passed	0s	📄

Repositorio

<https://github.com/J4v1er-502/Hoja-de-Trabajo-2>