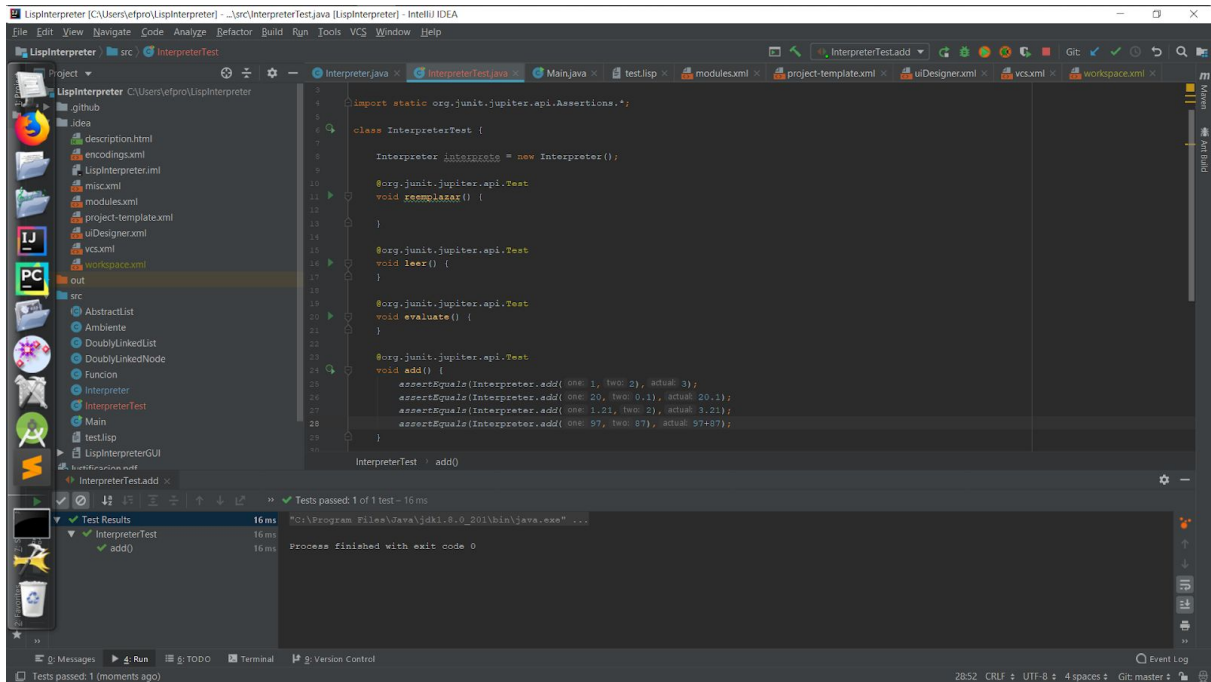


Pruebas JUnit

Interpreter Add



```
import static org.junit.jupiter.api.Assertions.*;

class InterpreterTest {

    Interpreter interpreter = new Interpreter();

    @org.junit.jupiter.api.Test
    void leer() {
    }

    @org.junit.jupiter.api.Test
    void evaluar() {
    }

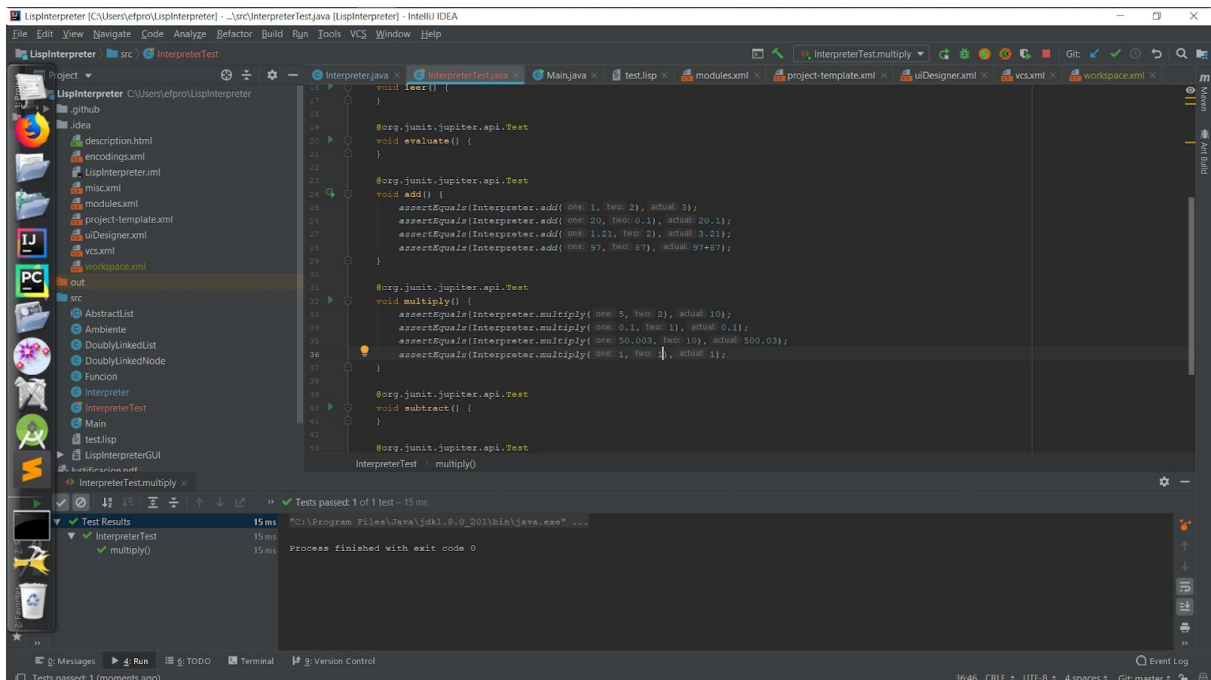
    @org.junit.jupiter.api.Test
    void add() {
        assertEquals(interpreter.add(1, 2), 3);
        assertEquals(interpreter.add(20, 0.1), 20.1);
        assertEquals(interpreter.add(1.21, 2), 3.21);
        assertEquals(interpreter.add(97, 0.03), 97.03);
    }
}
```

Tests passed: 1 of 1 test - 16 ms

Test Results

Test	Time	Exit Code
InterpreterTest	16 ms	0
add()	16 ms	0

Interpreter multiply



```
void leer() {
}

@org.junit.jupiter.api.Test
void evaluar() {
}

@org.junit.jupiter.api.Test
void add() {
    assertEquals(interpreter.add(1, 2), 3);
    assertEquals(interpreter.add(20, 0.1), 20.1);
    assertEquals(interpreter.add(1.21, 2), 3.21);
    assertEquals(interpreter.add(97, 0.03), 97.03);
}

@org.junit.jupiter.api.Test
void multiply() {
    assertEquals(interpreter.multiply(5, 2), 10);
    assertEquals(interpreter.multiply(0.1, 2), 0.2);
    assertEquals(interpreter.multiply(50.003, 2), 100.006);
    assertEquals(interpreter.multiply(1, 1), 1);
}

@org.junit.jupiter.api.Test
void subtract() {
}

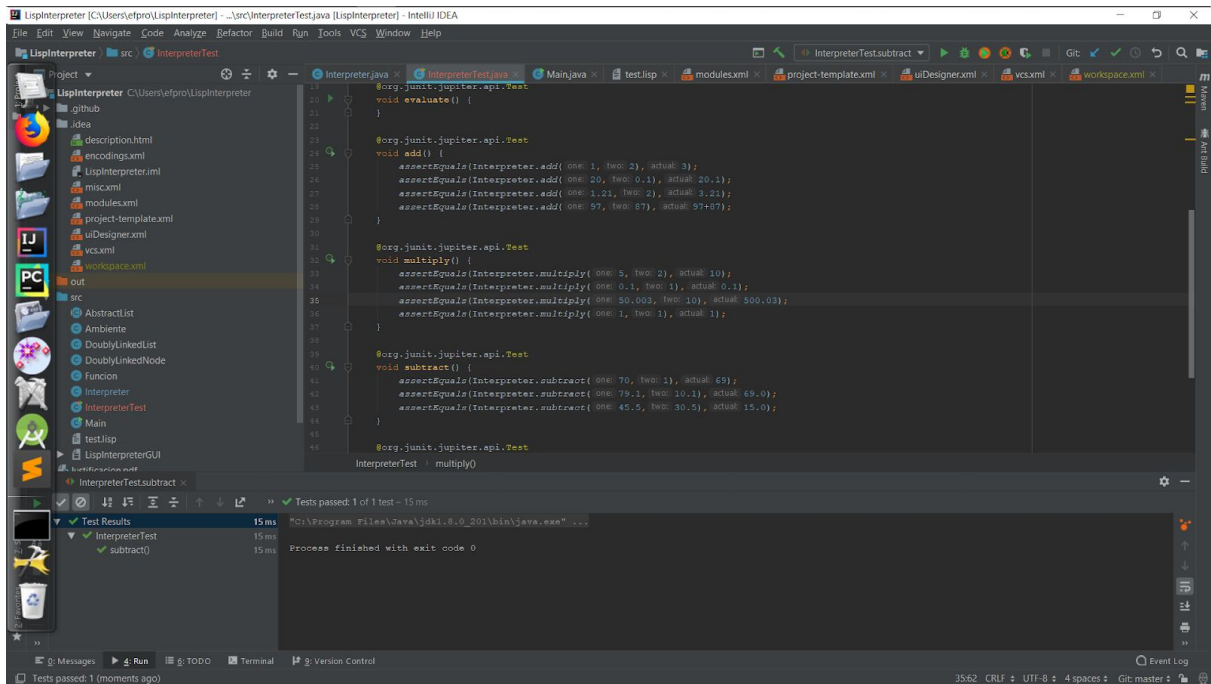
@org.junit.jupiter.api.Test
void divide() {
}
```

Tests passed: 1 of 1 test - 15 ms

Test Results

Test	Time	Exit Code
InterpreterTest	15 ms	0
multiply()	15 ms	0

Interpreter Subtract



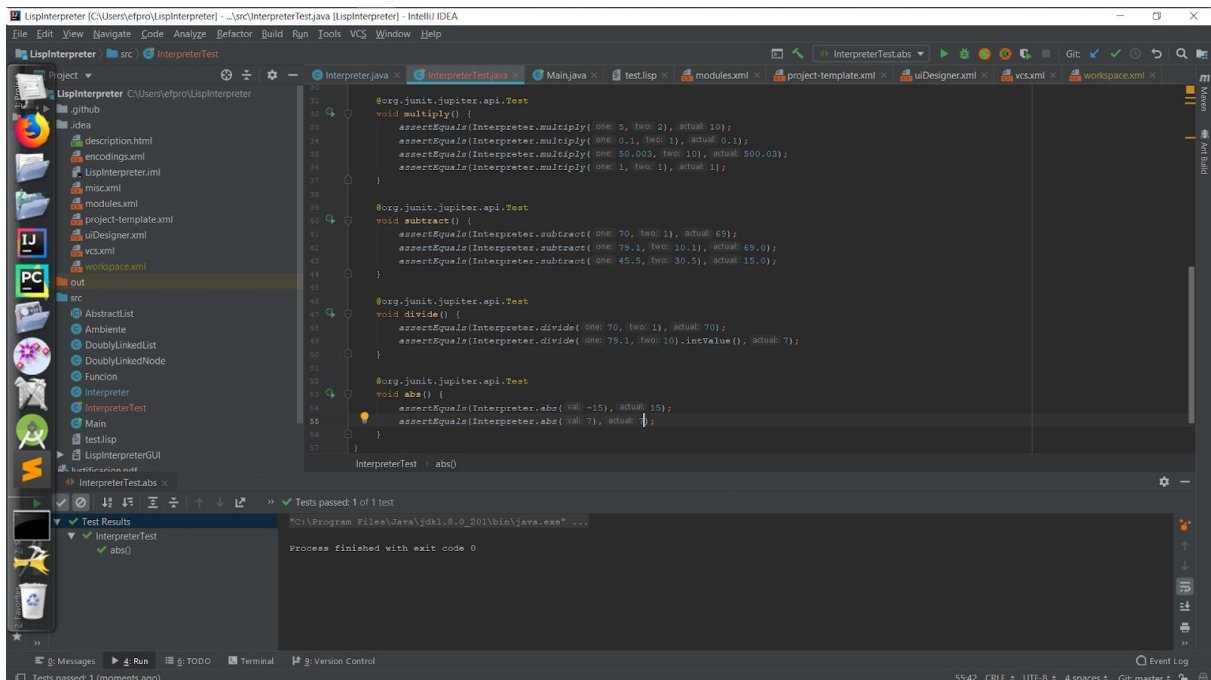
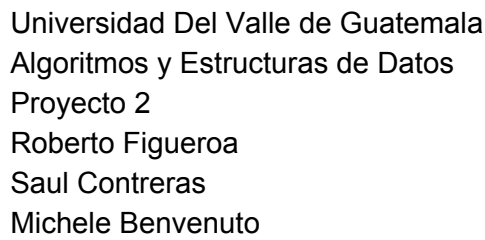
```
18  @org.junit.jupiter.api.Test
19  void evaluate() {
20  }
21
22
23  @org.junit.jupiter.api.Test
24  void add() {
25      assertEquals(Interpreter.add(1, two: 2), actual: 3);
26      assertEquals(Interpreter.add(20, two: 0.1), actual: 20.1);
27      assertEquals(Interpreter.add(1.21, two: 2), actual: 3.21);
28      assertEquals(Interpreter.add(97, two: 87), actual: 97+87);
29  }
30
31  @org.junit.jupiter.api.Test
32  void multiply() {
33      assertEquals(Interpreter.multiply(5, two: 2), actual: 10);
34      assertEquals(Interpreter.multiply(0.1, two: 1), actual: 0.1);
35      assertEquals(Interpreter.multiply(30.003, two: 10), actual: 300.03);
36      assertEquals(Interpreter.multiply(1, two: 1), actual: 1);
37  }
38
39  @org.junit.jupiter.api.Test
40  void subtract() {
41      assertEquals(Interpreter.subtract(70, two: 1), actual: 69);
42      assertEquals(Interpreter.subtract(75.1, two: 10.1), actual: 69.0);
43      assertEquals(Interpreter.subtract(45.5, two: 30.5), actual: 15.0);
44  }
45
46  @org.junit.jupiter.api.Test
47  void multiply0() {
48  }
49  }
```

Test Results

Test	Duration	Status
InterpreterTest	15 ms	Passed
subtract()	15 ms	Passed

Process finished with exit code 0

Interpreter divide



Interpreter interpretar:

```

1  import java.util.function.BiFunction;
2  import java.util.function.UnaryOperator;
3
4  import static org.junit.jupiter.api.Assertions.*;
5
6  class InterpreterTest {
7
8      Interpreter interpreter = new Interpreter();
9
10     @org.junit.jupiter.api.Test
11     void reemplazar() {
12         String[] newcode = {"(", "+", "2", "1", ")"};
13         String code = "(+ 2 1)";
14         assertEquals(interpreter.reemplazar(code)[0], newcode[0]);
15         assertEquals(interpreter.reemplazar(code)[1], newcode[1]);
16         assertEquals(interpreter.reemplazar(code)[2], newcode[2]);
17         assertEquals(interpreter.reemplazar(code)[3], newcode[3]);
18         assertEquals(interpreter.reemplazar(code)[4], newcode[4]);
19     }
20
21     @org.junit.jupiter.api.Test
22     void leer() {
23     }
24
25     @org.junit.jupiter.api.Test
26     void evaluar() {
27     }
28 }

```

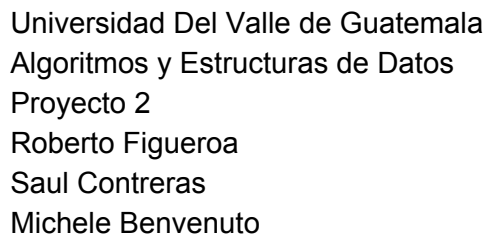
Tests passed: 1 of 1 test - 16 ms

Test Results

Test	Time
InterpreterTest	16 ms
reemplazar()	16 ms

Process finished with exit code 0

Interpreter read:





Universidad Del Valle de Guatemala

Algoritmos y Estructuras de Datos

Proyecto 2

Roberto Figueroa
Saul Contreras
Michele Benvenuto

The screenshot shows the IntelliJ IDEA interface with the file `AmbienteTest.java` open. The code defines a `public class AmbienteTest` with two methods: `get()` and `containsKey()`. Both methods create an `Ambiente` object and perform assertions. The `Run` tab at the bottom shows that two tests passed successfully in 2ms.

```
1 import ...
2
3
4 public class AmbienteTest {
5
6     @Test
7     public void get() {
8         Ambiente amb = new Ambiente();
9         String value = "value";
10        assertEquals(amb.get(value), actual: null);
11    }
12
13
14
15     @Test
16     public void containsKey() {
17         Ambiente amb = new Ambiente();
18         int value = 11122;
19         assertEquals(amb.containsKey(value), actual: false);
20    }
21 }
```

Run: AmbienteTest
Tests passed: 2 of 2 tests - 2 ms
get: 2ms
containsKey: 0ms
Process finished with exit code 0

Prueba para DoublyLinkedList:

The screenshot shows the IntelliJ IDEA interface with the file `DoublyLinkedListTest.java` open. The code defines a `public class DoublyLinkedListTest` with a `removeLast()` method. The method creates a `DoublyLinkedList` object, adds several elements, and then removes the last element. The `Run` tab at the bottom shows that two tests passed successfully in 48ms.

```
1 import ...
2
3
4 public class DoublyLinkedListTest {
5
6     @Test
7     public void removeLast() {
8
9         DoublyLinkedList<String> list = new DoublyLinkedList<>();
10        list.addFirst("First");
11        list.add("Second");
12        list.add("Third");
13        list.add("Fourth");
14        list.add("Fifth");
15        list.add("Sixth");
16        list.add("Seventh");
17        list.add("Eighth");
18        list.add("Ninth");
19        list.add("Tenth");
20        assertEquals(list.removeLast(), actual: "First");
21    }
22
23
24
25
26
27
28 }
```

Run: DoublyLinkedListTest
Tests passed: 2 of 2 tests - 48 ms
get: 48ms
removeLast: 0ms
Process finished with exit code 0