

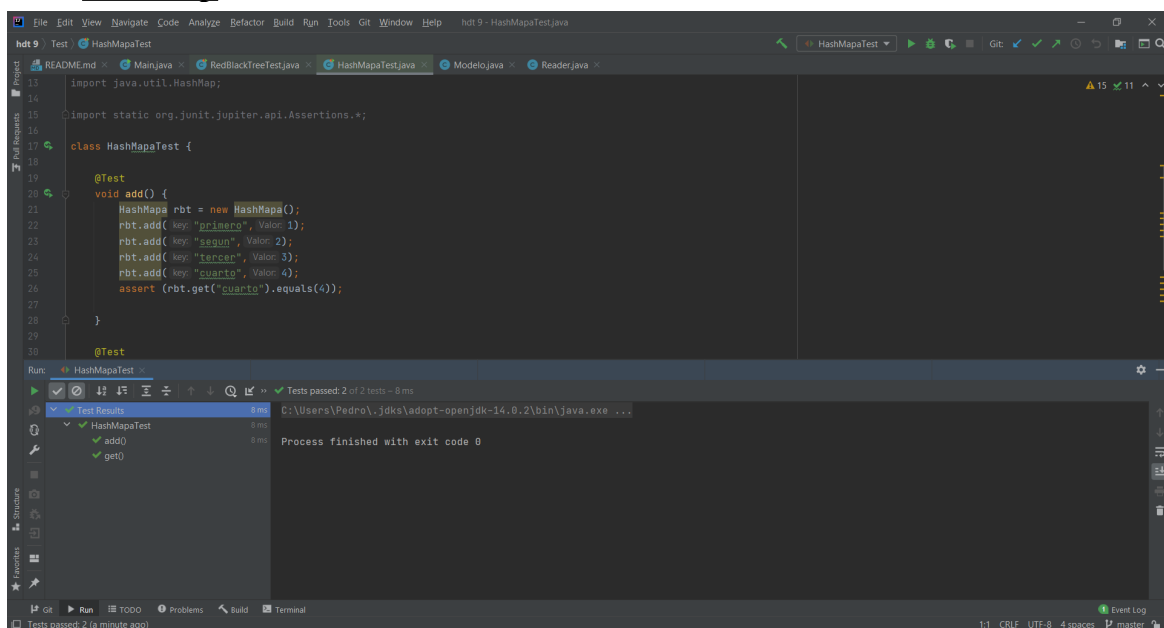
Hoja de trabajo #9

Link al repositorio de Github

<https://github.com/Jack200133/Hoja-de-trabajo-9.git>

Pruebas unitarias

- HashMap



```
import java.util.HashMap;

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

class HashMapTest {

    @Test
    void add() {
        HashMap rbt = new HashMap();
        rbt.add( key: "primero", Value: 1);
        rbt.add( key: "segun", Value: 2);
        rbt.add( key: "tercer", Value: 3);
        rbt.add( key: "cuarto", Value: 4);
        assert (rbt.get("cuarto").equals(4));
    }

    @Test
    void get() {
    }
}
```

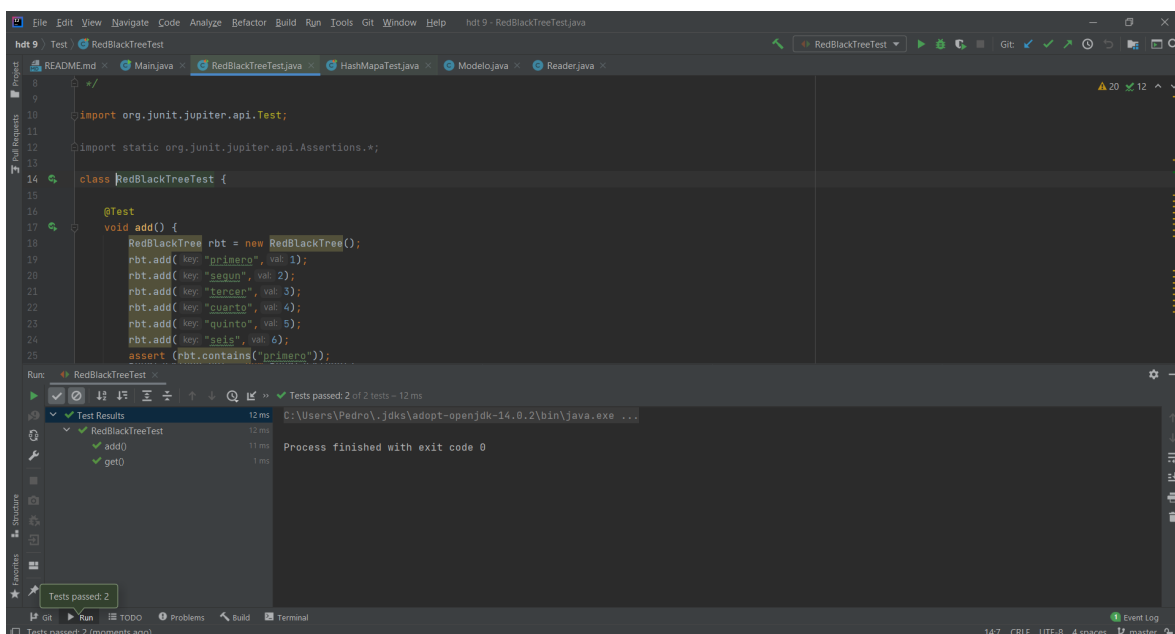
Run: HashMapTest

Test Results: 2 of 2 tests - 8 ms

- HashMapTest
- add() 8 ms
- get() 8 ms

Process finished with exit code 0

- Red Black Tree



```
import org.junit.jupiter.api.Test;

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

class RedBlackTreeTest {

    @Test
    void add() {
        RedBlackTree rbt = new RedBlackTree();
        rbt.add( key: "primero", val: 1);
        rbt.add( key: "segun", val: 2);
        rbt.add( key: "tercer", val: 3);
        rbt.add( key: "cuarto", val: 4);
        rbt.add( key: "quinto", val: 5);
        rbt.add( key: "seis", val: 6);
        assert (rbt.contains("primero"));
    }

    @Test
    void get() {
    }
}
```

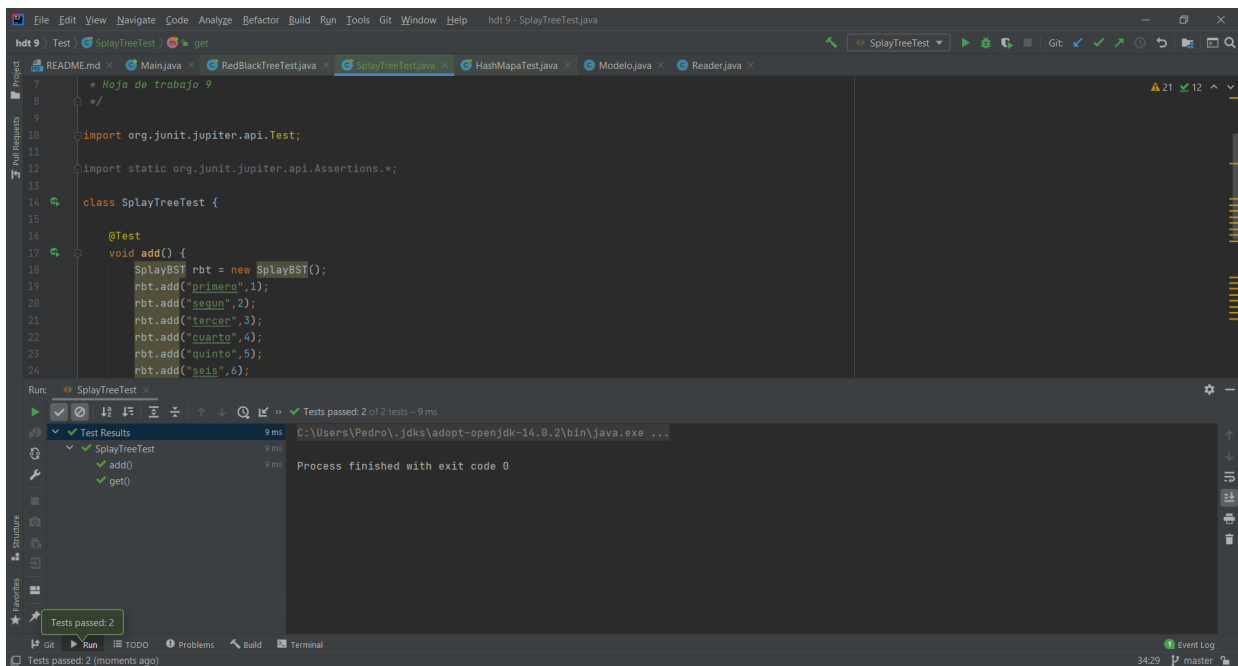
Run: RedBlackTreeTest

Test Results: 2 of 2 tests - 12 ms

- RedBlackTreeTest
- add() 12 ms
- get() 1 ms

Process finished with exit code 0

• Splay Tree



Medición de tiempo con Profiler

Para llevar a cabo la medición del tiempo de ejecución del traductor con las diferentes implementaciones utilizadas, se utilizó JProfiler para poder realizar esto:

