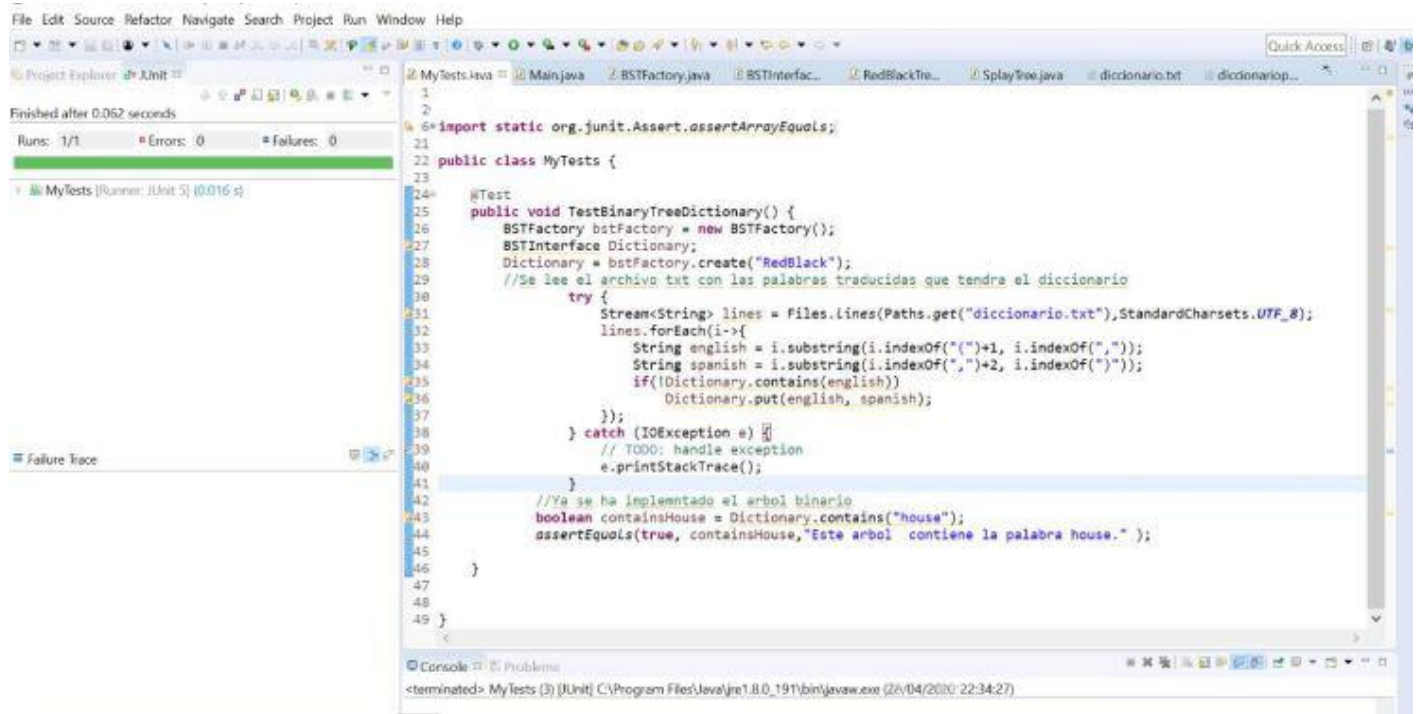
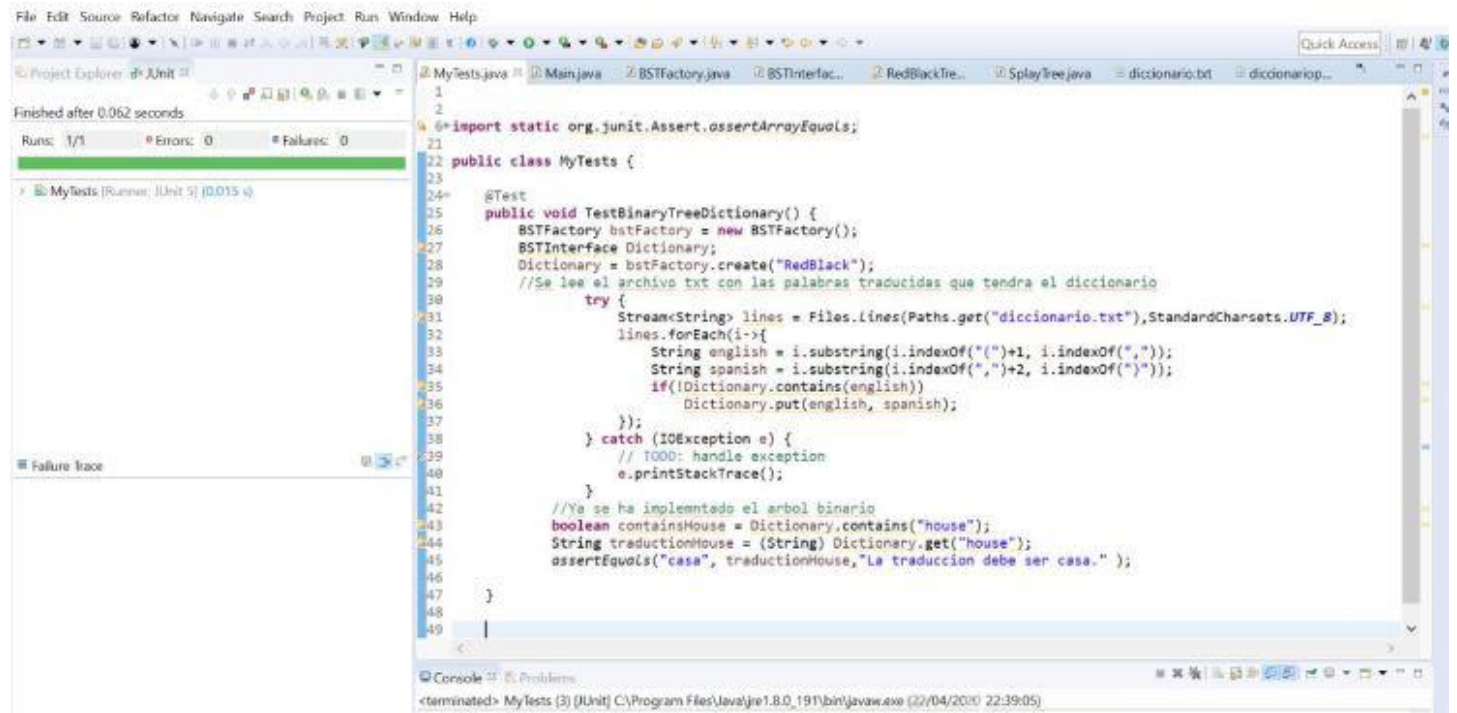


Arbol inserta los datos de traducción



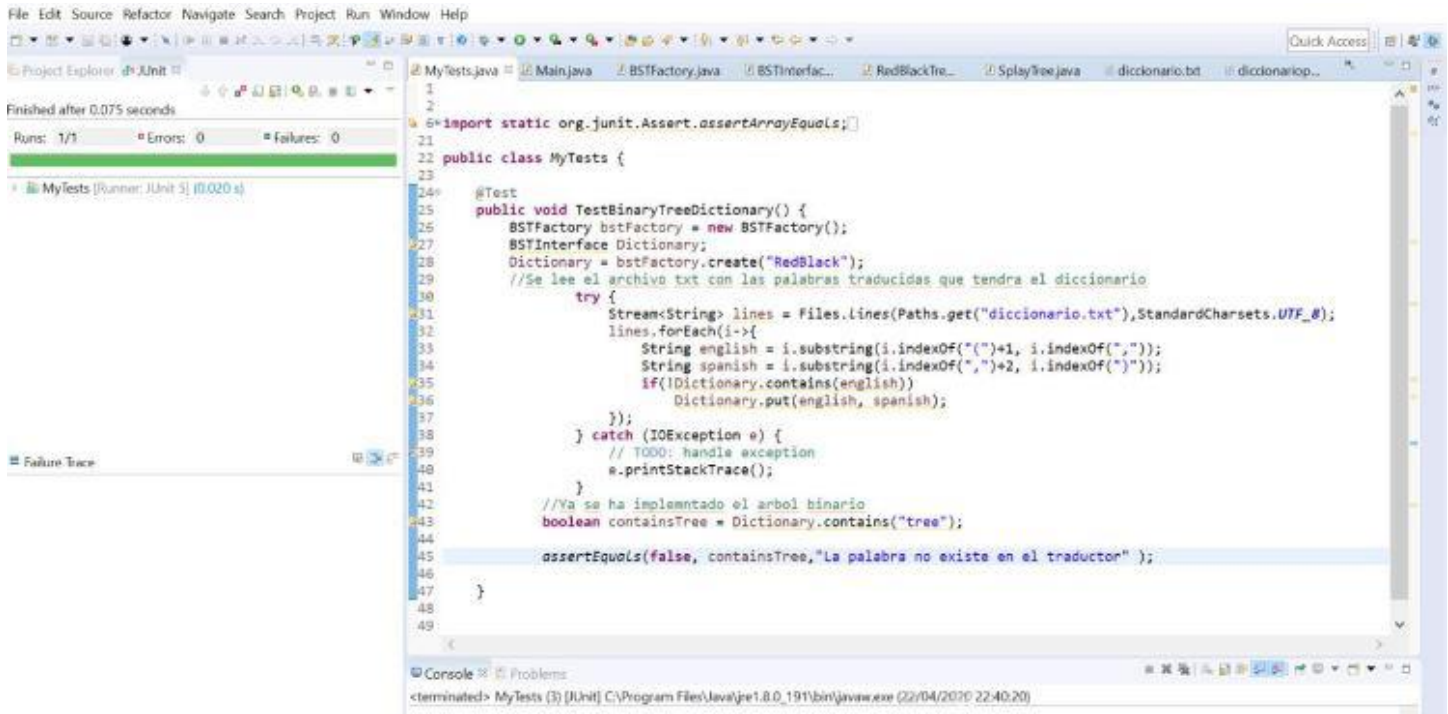
```
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer JUnit
Finished after 0.062 seconds
Runs: 1/1 Errors: 0 Failures: 0
MyTests [Runner: JUnit 5] (0.016 s)
MyTests.java Main.java BSTFactory.java BSTInterfac... RedBlackTre... SplayTree.java diccionario.bt diccionarioop...
1
2
3
4 import static org.junit.Assert.assertEquals;
5
6 public class MyTests {
7
8     @Test
9     public void TestBinaryTreeDictionary() {
10         BSTFactory bstFactory = new BSTFactory();
11         BSTInterface Dictionary;
12         Dictionary = bstFactory.create("RedBlack");
13         //Se lee el archivo txt con las palabras traducidas que tendra el diccionario
14         try {
15             Stream<String> lines = Files.lines(Paths.get("diccionario.txt"), StandardCharsets.UTF_8);
16             lines.forEach(i->{
17                 String english = i.substring(i.indexOf("(")+1, i.indexOf(","));
18                 String spanish = i.substring(i.indexOf(",")+2, i.indexOf(")"));
19                 if(!Dictionary.contains(english))
20                     Dictionary.put(english, spanish);
21             });
22         } catch (IOException e) {
23             // TODO: handle exception
24             e.printStackTrace();
25         }
26         //Ya se ha implementado el arbol binario
27         boolean containsHouse = Dictionary.contains("house");
28         assertEquals(true, containsHouse, "Este arbol contiene la palabra house.");
29     }
30 }
31
32 Console Problems
33 <terminated> MyTests (3) [JUnit] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (22/04/2020 22:34:27)
```

Encuentro de palabras



```
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer JUnit
Finished after 0.062 seconds
Runs: 1/1 Errors: 0 Failures: 0
MyTests [Runner: JUnit 5] (0.015 s)
MyTests.java Main.java BSTFactory.java BSTInterfac... RedBlackTre... SplayTree.java diccionario.bt diccionarioop...
1
2
3
4 import static org.junit.Assert.assertEquals;
5
6 public class MyTests {
7
8     @Test
9     public void TestBinaryTreeDictionary() {
10         BSTFactory bstFactory = new BSTFactory();
11         BSTInterface Dictionary;
12         Dictionary = bstFactory.create("RedBlack");
13         //Se lee el archivo txt con las palabras traducidas que tendra el diccionario
14         try {
15             Stream<String> lines = Files.lines(Paths.get("diccionario.txt"), StandardCharsets.UTF_8);
16             lines.forEach(i->{
17                 String english = i.substring(i.indexOf("(")+1, i.indexOf(","));
18                 String spanish = i.substring(i.indexOf(",")+2, i.indexOf(")"));
19                 if(!Dictionary.contains(english))
20                     Dictionary.put(english, spanish);
21             });
22         } catch (IOException e) {
23             // TODO: handle exception
24             e.printStackTrace();
25         }
26         //Ya se ha implementado el arbol binario
27         boolean containsHouse = Dictionary.contains("house");
28         String traductionHouse = (String) Dictionary.get("house");
29         assertEquals("casa", traductionHouse, "La traduccion debe ser casa.");
30     }
31 }
32
33 Console Problems
34 <terminated> MyTests (3) [JUnit] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (22/04/2020 22:39:05)
```

Cuando la palabra no existe

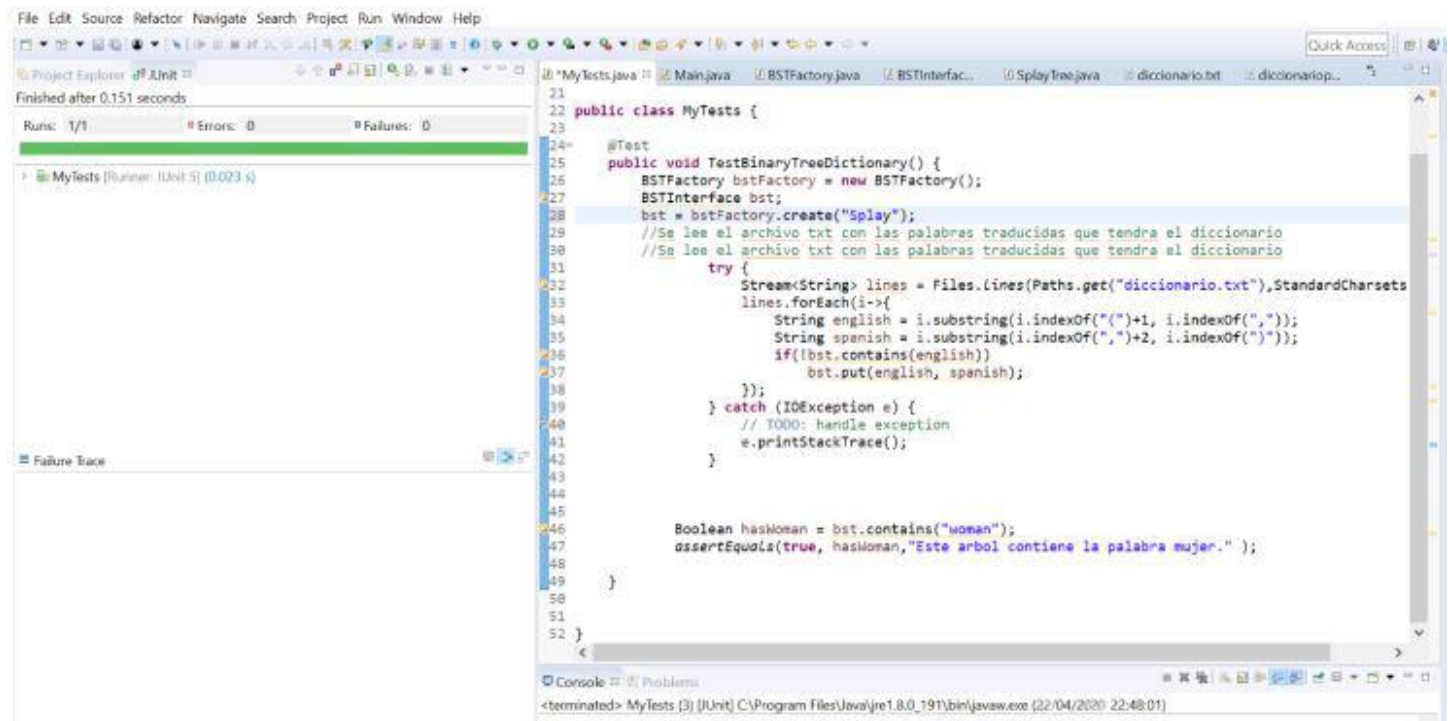


The screenshot shows an IDE with a Java project. The 'MyTests.java' file is open, showing a test method 'TestBinaryTreeDictionary()'. The test creates a 'BSTFactory' and a 'Dictionary' using 'RedBlack'. It reads a 'diccionario.txt' file and populates the dictionary. Then, it checks if the word 'tree' is in the dictionary and asserts that it is not, with the message 'La palabra no existe en el traductor'.

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22 public class MyTests {
23
24     @Test
25     public void TestBinaryTreeDictionary() {
26         BSTFactory bstFactory = new BSTFactory();
27         BSTInterface Dictionary;
28         Dictionary = bstFactory.create("RedBlack");
29         //Se lee el archivo txt con las palabras traducidas que tendra el diccionario
30         try {
31             Stream<String> lines = Files.lines(Paths.get("diccionario.txt"), StandardCharsets.UTF_8);
32             lines.forEach(i->{
33                 String english = i.substring(i.indexOf("(")+1, i.indexOf(","));
34                 String spanish = i.substring(i.indexOf(",")+2, i.indexOf(")"));
35                 if(!Dictionary.contains(english))
36                     Dictionary.put(english, spanish);
37             });
38         } catch (IOException e) {
39             // TODO: handle exception
40             e.printStackTrace();
41         }
42         //Ya se ha implementado el arbol binario
43         boolean containsTree = Dictionary.contains("tree");
44         assertEquals(false, containsTree, "La palabra no existe en el traductor");
45     }
46 }
47
48
49
```

Console: <terminated> MyTests (3) [JUnit] C:\Program Files\Java\jre1.8.0_191\bin\java.exe (22/04/2020 22:40:20)

Si existe la palabra, la inserta



The screenshot shows the same IDE with the 'MyTests.java' file. The test method 'TestBinaryTreeDictionary()' is modified to create a 'BSTFactory' and a 'BST' using 'Splay'. It reads the 'diccionario.txt' file and populates the BST. Then, it checks if the word 'woman' is in the BST and asserts that it is, with the message 'Este arbol contiene la palabra mujer.'.

```
21
22 public class MyTests {
23
24     @Test
25     public void TestBinaryTreeDictionary() {
26         BSTFactory bstFactory = new BSTFactory();
27         BSTInterface bst;
28         bst = bstFactory.create("Splay");
29         //Se lee el archivo txt con las palabras traducidas que tendra el diccionario
30         //Se lee el archivo txt con las palabras traducidas que tendra el diccionario
31         try {
32             Stream<String> lines = Files.lines(Paths.get("diccionario.txt"), StandardCharsets
33             lines.forEach(i->{
34                 String english = i.substring(i.indexOf("(")+1, i.indexOf(","));
35                 String spanish = i.substring(i.indexOf(",")+2, i.indexOf(")"));
36                 if(!bst.contains(english))
37                     bst.put(english, spanish);
38             });
39         } catch (IOException e) {
40             // TODO: handle exception
41             e.printStackTrace();
42         }
43
44         Boolean hasWoman = bst.contains("woman");
45         assertEquals(true, hasWoman, "Este arbol contiene la palabra mujer. ");
46     }
47 }
48
49
50
51
52
```

Console: <terminated> MyTests (3) [JUnit] C:\Program Files\Java\jre1.8.0_191\bin\java.exe (22/04/2020 22:48:01)