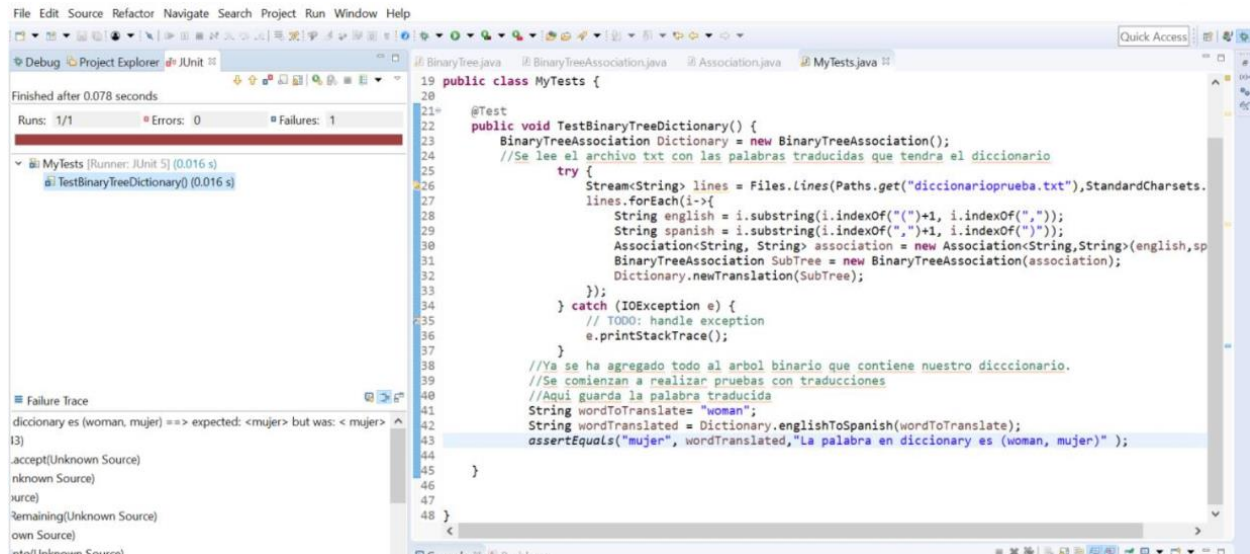


PRUEBAS JUNIT

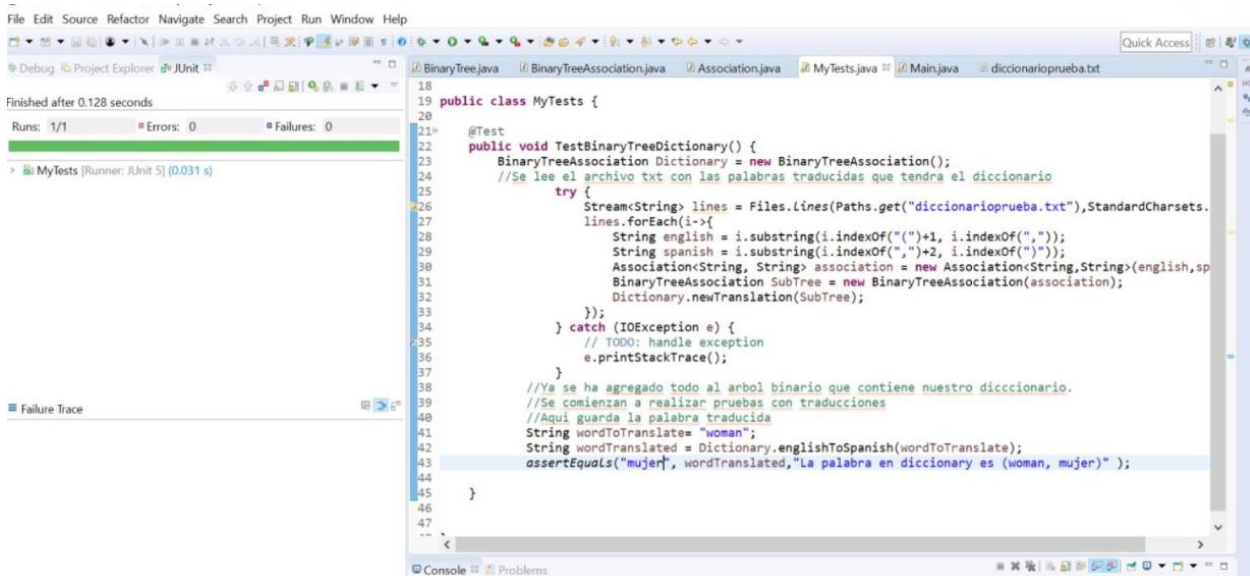


```
19 public class MyTests {
20
21     @Test
22     public void TestBinaryTreeDictionary() {
23         BinaryTreeAssociation Dictionary = new BinaryTreeAssociation();
24         //Se lee el archivo txt con las palabras traducidas que tendra el diccionario
25         try {
26             Stream<String> lines = Files.lines(Paths.get("diccionarioprueba.txt"), StandardCharsets.
27                 lines.forEach(i->{
28                     String english = i.substring(i.indexOf("(")+1, i.indexOf(")"));
29                     String spanish = i.substring(i.indexOf(",")+1, i.indexOf(")"));
30                     Association<String, String> association = new Association<String, String>(english, sp
31                         BinaryTreeAssociation SubTree = new BinaryTreeAssociation(association);
32                         Dictionary.newTranslation(SubTree);
33                     });
34                 } catch (IOException e) {
35                     // TODO: handle exception
36                     e.printStackTrace();
37                 }
38             //Ya se ha agregado todo al arbol binario que contiene nuestro diccionario.
39             //Se comienzan a realizar pruebas con traducciones
40             //Aqui guarda la palabra traducida
41             String wordToTranslate= "woman";
42             String wordTranslated = Dictionary.englishToSpanish(wordToTranslate);
43             assertEquals("mujer", wordTranslated, "La palabra en dictionary es (woman, mujer)");
44         }
45     }
46 }
47
48 }
```

Failure Trace

```
dictionary es (woman, mujer) ==> expected: <mujer> but was: <mujer >
[3]
.accept(Unknown Source)
.nknown Source)
.surce)
.Remaining(Unknown Source)
.own Source)
.intel.IntelOpen Source)
```

En esta primera prueba la palabra traducida estaba guardándose con un espacio de mas entonces el programa tiraba error.

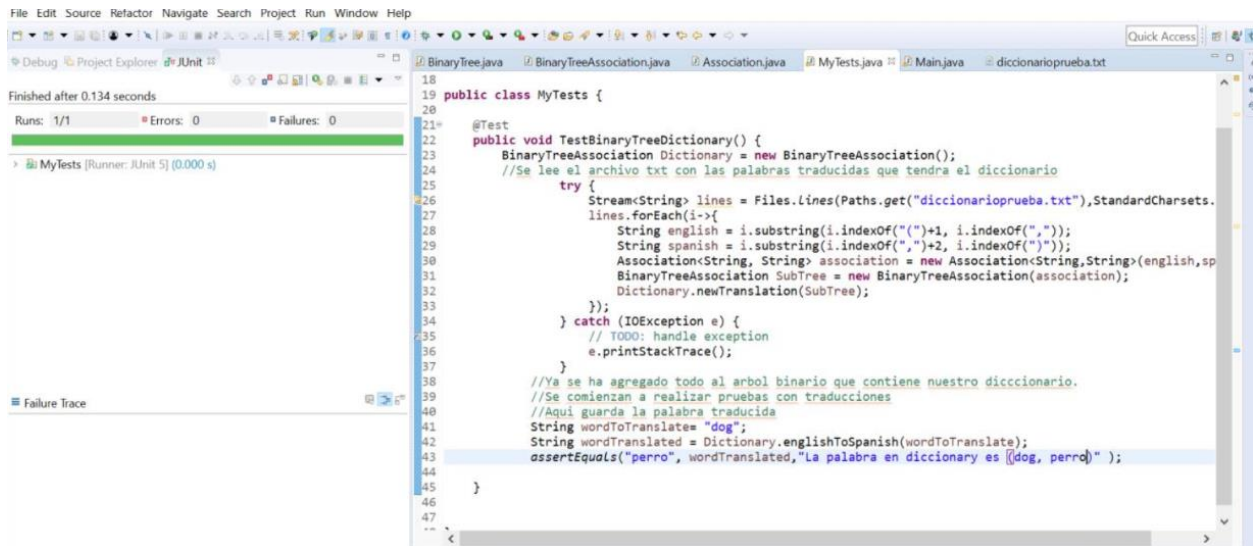


```
18 public class MyTests {
19
20     @Test
21     public void TestBinaryTreeDictionary() {
22         BinaryTreeAssociation Dictionary = new BinaryTreeAssociation();
23         //Se lee el archivo txt con las palabras traducidas que tendra el diccionario
24         try {
25             Stream<String> lines = Files.lines(Paths.get("diccionarioprueba.txt"), StandardCharsets.
26                 lines.forEach(i->{
27                     String english = i.substring(i.indexOf("(")+1, i.indexOf(")"));
28                     String spanish = i.substring(i.indexOf(",")+2, i.indexOf(")"));
29                     Association<String, String> association = new Association<String, String>(english, sp
30                         BinaryTreeAssociation SubTree = new BinaryTreeAssociation(association);
31                         Dictionary.newTranslation(SubTree);
32                     });
33                 } catch (IOException e) {
34                     // TODO: handle exception
35                     e.printStackTrace();
36                 }
37             //Ya se ha agregado todo al arbol binario que contiene nuestro diccionario.
38             //Se comienzan a realizar pruebas con traducciones
39             //Aqui guarda la palabra traducida
40             String wordToTranslate= "woman";
41             String wordTranslated = Dictionary.englishToSpanish(wordToTranslate);
42             assertEquals("mujer", wordTranslated, "La palabra en dictionary es (woman, mujer)");
43         }
44     }
45 }
46
47 }
```

Failure Trace

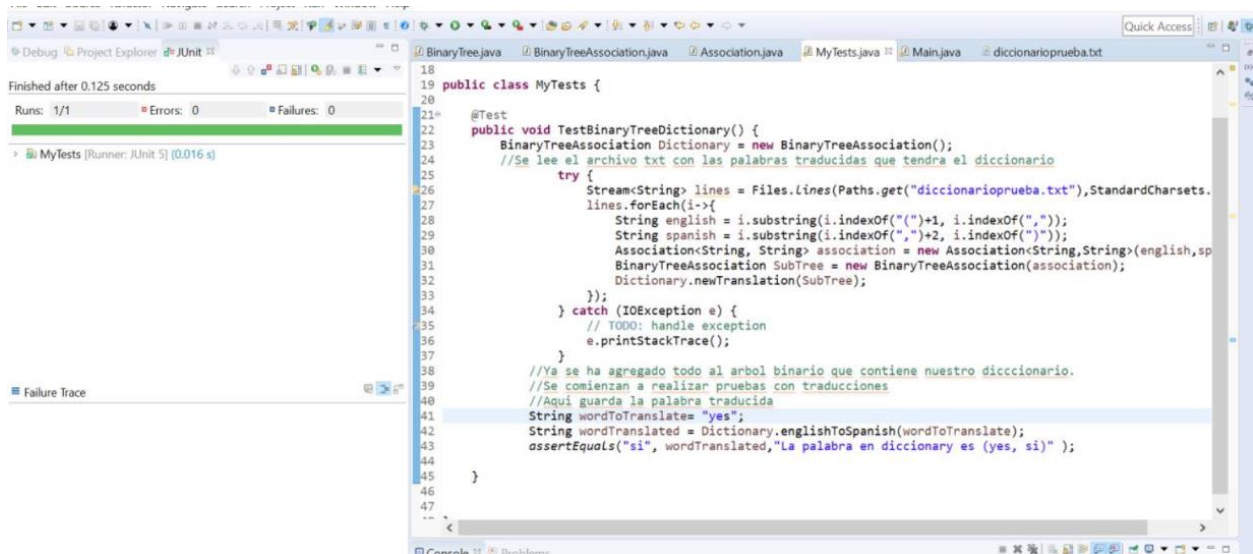
La palabra mujer ya es traducida correctamente.

Daniela Villamar
19086



```
18 public class MyTests {
19
20
21     @Test
22     public void TestBinaryTreeDictionary() {
23         BinaryTreeAssociation Dictionary = new BinaryTreeAssociation();
24         //Se lee el archivo txt con las palabras traducidas que tendra el diccionario
25         try {
26             Stream<String> lines = Files.lines(Paths.get("diccionarioprueba.txt"), StandardCharsets.
27                 lines.forEach(i->{
28                     String english = i.substring(i.indexOf("(")+1, i.indexOf(",");
29                     String spanish = i.substring(i.indexOf(",")+2, i.indexOf(")"));
30                     Association<String, String> association = new Association<String, String>(english, sp
31                     BinaryTreeAssociation SubTree = new BinaryTreeAssociation(association);
32                     Dictionary.newTranslation(SubTree);
33                 }));
34             } catch (IOException e) {
35                 // TODO: handle exception
36                 e.printStackTrace();
37             }
38             //Ya se ha agregado todo al arbol binario que contiene nuestro diccionario.
39             //Se comienzan a realizar pruebas con traducciones
40             //Aqui guarda la palabra traducida
41             String wordToTranslate = "dog";
42             String wordTranslated = Dictionary.englishToSpanish(wordToTranslate);
43             assertEquals("La palabra en dictionary es (dog, perro)");
44         }
45     }
46
47 }
```

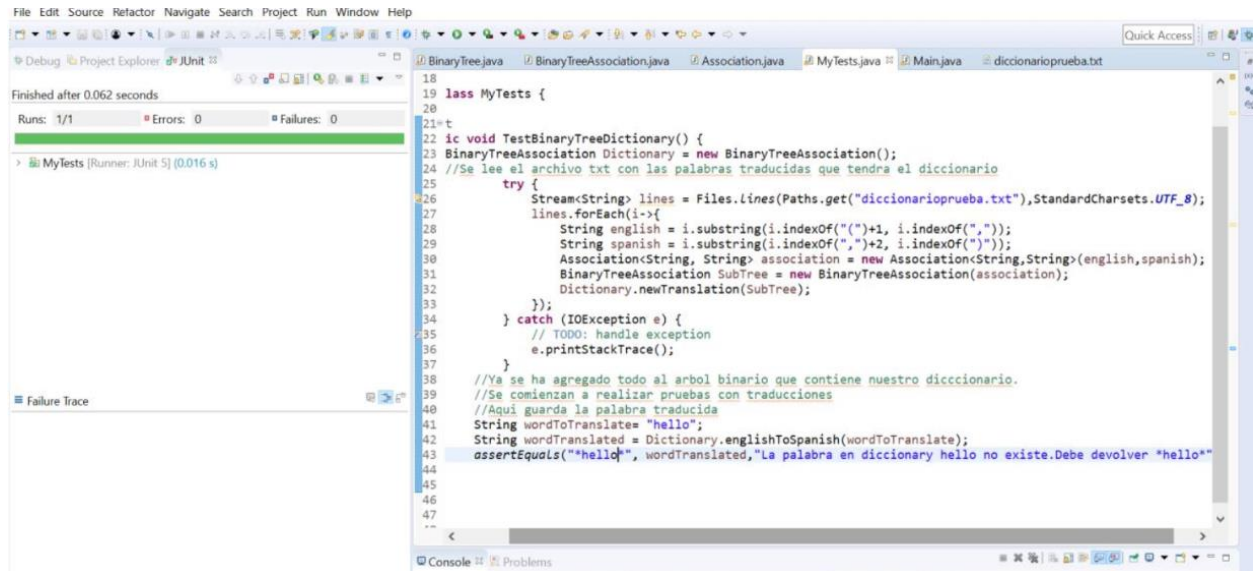
La palabra dog fue traducida correctamente.



```
18 public class MyTests {
19
20
21     @Test
22     public void TestBinaryTreeDictionary() {
23         BinaryTreeAssociation Dictionary = new BinaryTreeAssociation();
24         //Se lee el archivo txt con las palabras traducidas que tendra el diccionario
25         try {
26             Stream<String> lines = Files.lines(Paths.get("diccionarioprueba.txt"), StandardCharsets.
27                 lines.forEach(i->{
28                     String english = i.substring(i.indexOf("(")+1, i.indexOf(",");
29                     String spanish = i.substring(i.indexOf(",")+2, i.indexOf(")"));
30                     Association<String, String> association = new Association<String, String>(english, sp
31                     BinaryTreeAssociation SubTree = new BinaryTreeAssociation(association);
32                     Dictionary.newTranslation(SubTree);
33                 }));
34             } catch (IOException e) {
35                 // TODO: handle exception
36                 e.printStackTrace();
37             }
38             //Ya se ha agregado todo al arbol binario que contiene nuestro diccionario.
39             //Se comienzan a realizar pruebas con traducciones
40             //Aqui guarda la palabra traducida
41             String wordToTranslate = "yes";
42             String wordTranslated = Dictionary.englishToSpanish(wordToTranslate);
43             assertEquals("si", wordTranslated, "La palabra en dictionary es (yes, si)");
44         }
45     }
46
47 }
```

La palabra yes fue traducida correctamente.

Daniela Villamar
19086

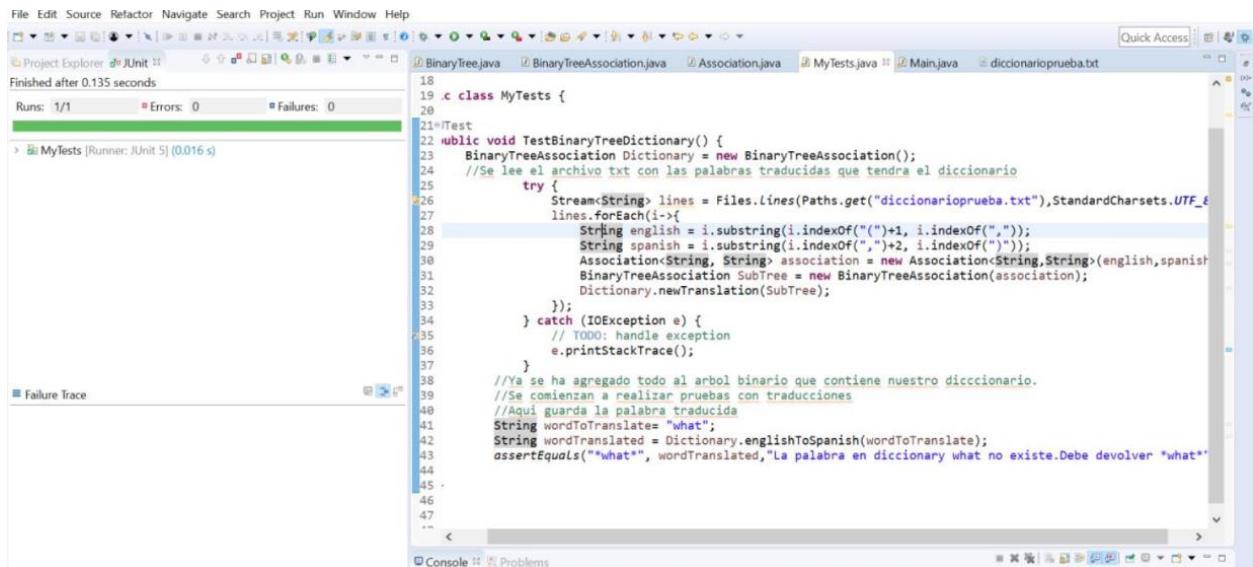


The screenshot shows an IDE with a Java project. The 'MyTests' class is being run, and the test 'TestBinaryTreeDictionary' is passing. The test code is as follows:

```
18 class MyTests {
19
20
21 @Test
22 void TestBinaryTreeDictionary() {
23     BinaryTreeAssociation Dictionary = new BinaryTreeAssociation();
24     //Se lee el archivo txt con las palabras traducidas que tendra el diccionario
25     try {
26         Stream<String> lines = Files.lines(Paths.get("diccionarioprueba.txt"), StandardCharsets.UTF_8);
27         lines.forEach(i->{
28             String english = i.substring(i.indexOf("(")+1, i.indexOf(","));
29             String spanish = i.substring(i.indexOf(",")+2, i.indexOf(")"));
30             Association<String, String> association = new Association<String, String>(english, spanish);
31             BinaryTreeAssociation SubTree = new BinaryTreeAssociation(association);
32             Dictionary.newTranslation(SubTree);
33         });
34     } catch (IOException e) {
35         // TODO: handle exception
36         e.printStackTrace();
37     }
38     //Ya se ha agregado todo al arbol binario que contiene nuestro diccionario.
39     //Se comienzan a realizar pruebas con traducciones
40     //Aqui guarda la palabra traducida
41     String wordToTranslate = "hello";
42     String wordTranslated = Dictionary.englishToSpanish(wordToTranslate);
43     assertEquals("hello", wordTranslated, "La palabra en dictionary hello no existe. Debe devolver 'hello'");
44 }
45
46
47 }
```

The test result shows 'Finished after 0.062 seconds' and 'Runs: 1/1', 'Errors: 0', 'Failures: 0'.

Como la palabra hello no esta en el diccionario la devuelve entre *



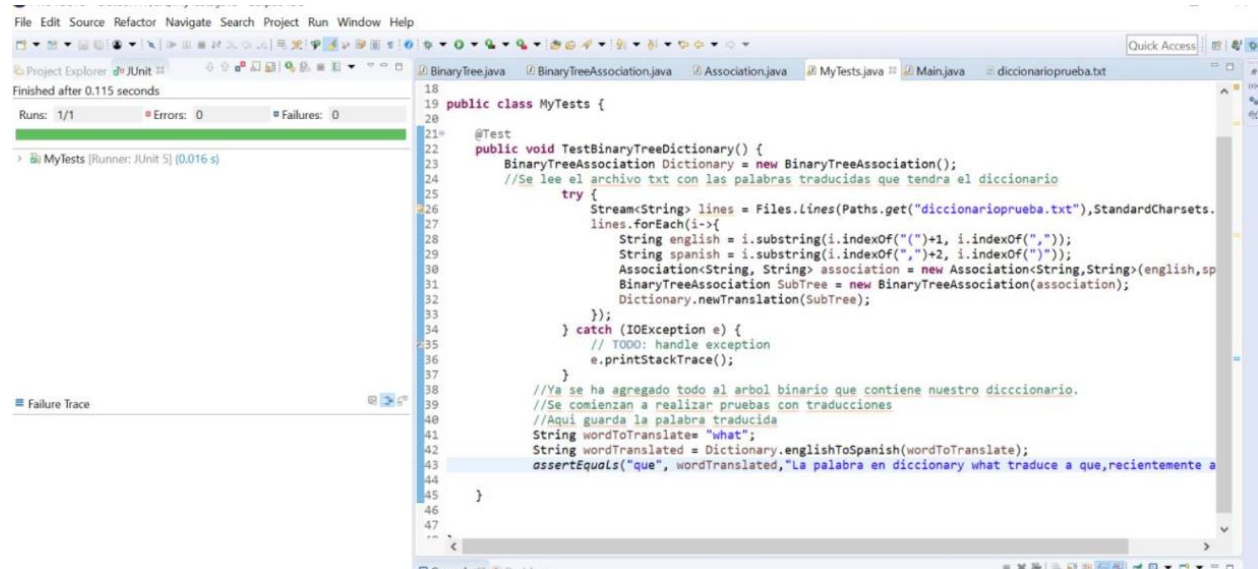
The screenshot shows the same IDE with the 'MyTests' class being run, but the test 'TestBinaryTreeDictionary' is failing. The test code is as follows:

```
18 class MyTests {
19
20
21 @Test
22 void TestBinaryTreeDictionary() {
23     BinaryTreeAssociation Dictionary = new BinaryTreeAssociation();
24     //Se lee el archivo txt con las palabras traducidas que tendra el diccionario
25     try {
26         Stream<String> lines = Files.lines(Paths.get("diccionarioprueba.txt"), StandardCharsets.UTF_8);
27         lines.forEach(i->{
28             String english = i.substring(i.indexOf("(")+1, i.indexOf(","));
29             String spanish = i.substring(i.indexOf(",")+2, i.indexOf(")"));
30             Association<String, String> association = new Association<String, String>(english, spanish);
31             BinaryTreeAssociation SubTree = new BinaryTreeAssociation(association);
32             Dictionary.newTranslation(SubTree);
33         });
34     } catch (IOException e) {
35         // TODO: handle exception
36         e.printStackTrace();
37     }
38     //Ya se ha agregado todo al arbol binario que contiene nuestro diccionario.
39     //Se comienzan a realizar pruebas con traducciones
40     //Aqui guarda la palabra traducida
41     String wordToTranslate = "what";
42     String wordTranslated = Dictionary.englishToSpanish(wordToTranslate);
43     assertEquals("what", wordTranslated, "La palabra en dictionary what no existe. Debe devolver 'what'");
44 }
45
46
47 }
```

The test result shows 'Finished after 0.135 seconds' and 'Runs: 1/1', 'Errors: 0', 'Failures: 0'.

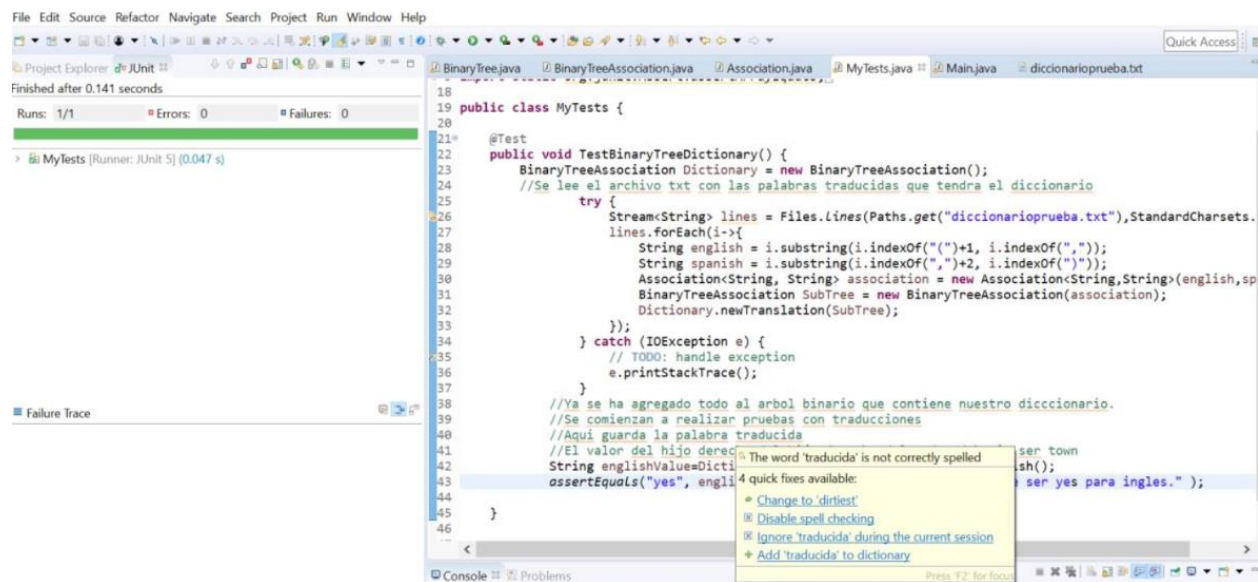
La palabra what no esta en el diccionario por eso la devuelve entre *

Daniela Villamar
19086



```
18
19 public class MyTests {
20
21     @Test
22     public void TestBinaryTreeDictionary() {
23         BinaryTreeAssociation Dictionary = new BinaryTreeAssociation();
24         //Se lee el archivo txt con las palabras traducidas que tendra el diccionario
25         try {
26             Stream<String> lines = Files.lines(Paths.get("diccionarioprueba.txt"), StandardCharsets.
27             lines.forEach(i->{
28                 String english = i.substring(i.indexOf("(")+1, i.indexOf(","));
29                 String spanish = i.substring(i.indexOf(",")+2, i.indexOf(")"));
30                 Association<String, String> association = new Association<String, String>(english, sp
31                 BinaryTreeAssociation SubTree = new BinaryTreeAssociation(association);
32                 Dictionary.newTranslation(SubTree);
33             });
34         } catch (IOException e) {
35             // TODO: handle exception
36             e.printStackTrace();
37         }
38         //Ya se ha agregado todo al arbol binario que contiene nuestro diccionario.
39         //Se comienzan a realizar pruebas con traducciones
40         //Aqui guarda la palabra traducida
41         String wordToTranslate = "what";
42         String wordTranslated = Dictionary.englishToSpanish(wordToTranslate);
43         assertEquals("que", wordTranslated, "La palabra en diccionario what traduce a que, recientemente a
44     }
45 }
46
47
```

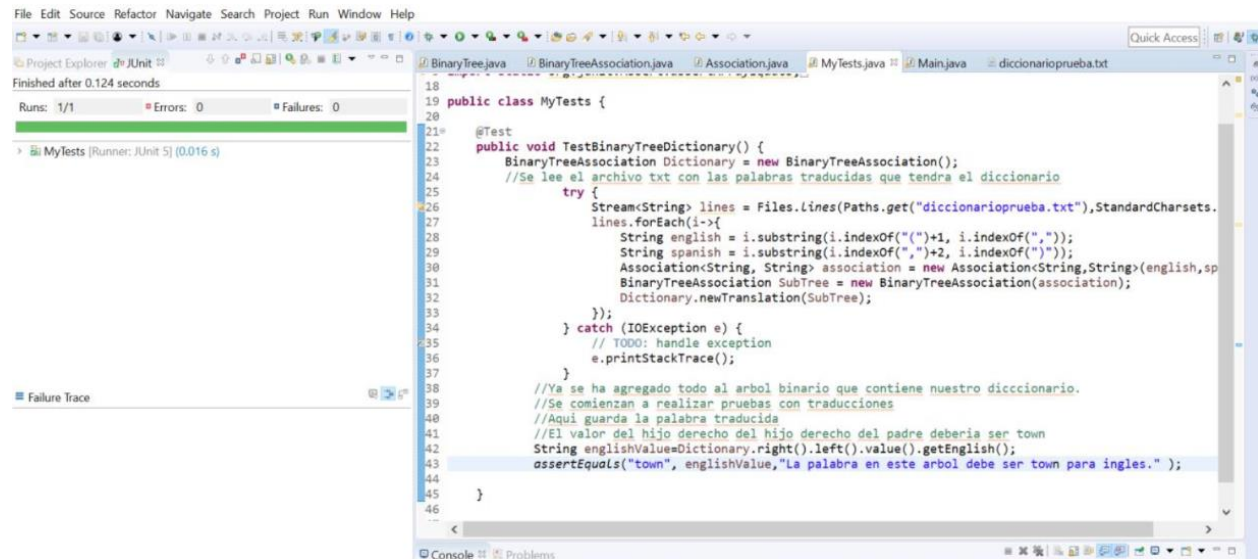
Al crear un diccionario prueba y agregarle (what, que) devuelve su traducción como: que.



```
18
19 public class MyTests {
20
21     @Test
22     public void TestBinaryTreeDictionary() {
23         BinaryTreeAssociation Dictionary = new BinaryTreeAssociation();
24         //Se lee el archivo txt con las palabras traducidas que tendra el diccionario
25         try {
26             Stream<String> lines = Files.lines(Paths.get("diccionarioprueba.txt"), StandardCharsets.
27             lines.forEach(i->{
28                 String english = i.substring(i.indexOf("(")+1, i.indexOf(","));
29                 String spanish = i.substring(i.indexOf(",")+2, i.indexOf(")"));
30                 Association<String, String> association = new Association<String, String>(english, sp
31                 BinaryTreeAssociation SubTree = new BinaryTreeAssociation(association);
32                 Dictionary.newTranslation(SubTree);
33             });
34         } catch (IOException e) {
35             // TODO: handle exception
36             e.printStackTrace();
37         }
38         //Ya se ha agregado todo al arbol binario que contiene nuestro diccionario.
39         //Se comienzan a realizar pruebas con traducciones
40         //Aqui guarda la palabra traducida
41         //El valor del hijo derec
42         String englishValue=Dicti
43         assertEquals("yes", englishValue, Dicti
44     }
45 }
46
```

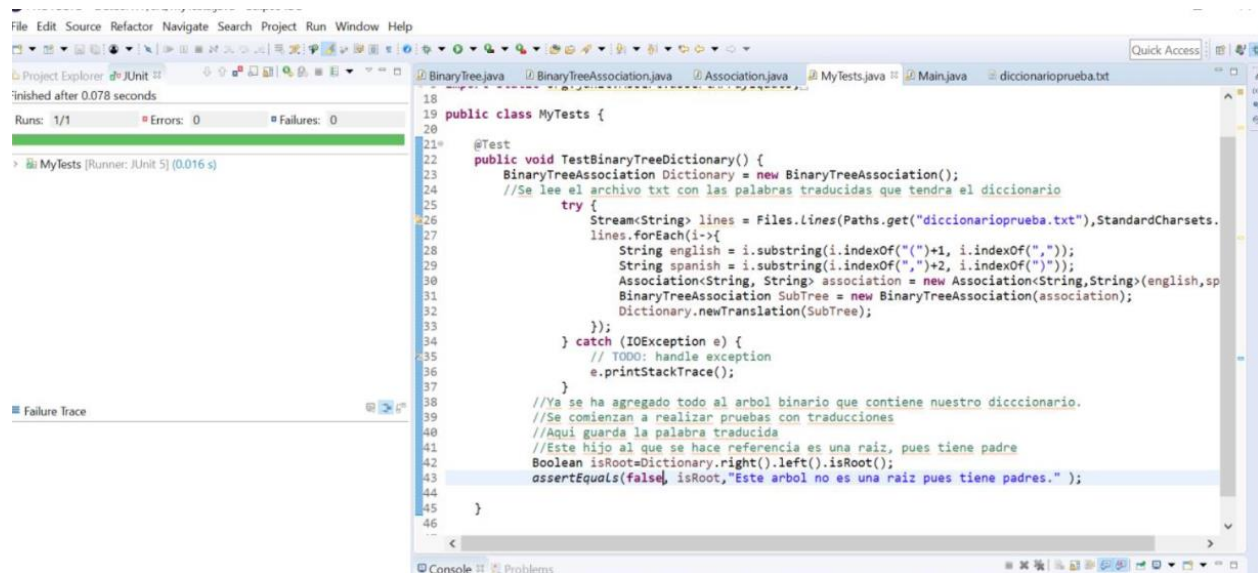
Los hijos guardan correctamente los datos en el orden solicitado.

Daniela Villamar
19086



```
18
19 public class MyTests {
20
21     @Test
22     public void TestBinaryTreeDictionary() {
23         BinaryTreeAssociation Dictionary = new BinaryTreeAssociation();
24         //Se lee el archivo txt con las palabras traducidas que tendra el diccionario
25         try {
26             Stream<String> lines = Files.lines(Paths.get("diccionarioprueba.txt"), StandardCharsets.
27                 lines.forEach(i->{
28                     String english = i.substring(i.indexOf("(")+1, i.indexOf(",");
29                     String spanish = i.substring(i.indexOf(",")+2, i.indexOf(")"));
30                     Association<String, String> association = new Association<String, String>(english, sp
31                     BinaryTreeAssociation SubTree = new BinaryTreeAssociation(association);
32                     Dictionary.newTranslation(SubTree);
33                 });
34             } catch (IOException e) {
35                 // TODO: handle exception
36                 e.printStackTrace();
37             }
38             //Ya se ha agregado todo al arbol binario que contiene nuestro diccionario.
39             //Se comienzan a realizar pruebas con traducciones
40             //Aqui guarda la palabra traducida
41             //El valor del hijo derecho del hijo derecho del padre deberia ser town
42             String englishValue=Dictionary.right().left().value().getEnglish();
43             assertEquals("town", englishValue, "La palabra en este arbol debe ser town para ingles." );
44
45         }
46     }
```

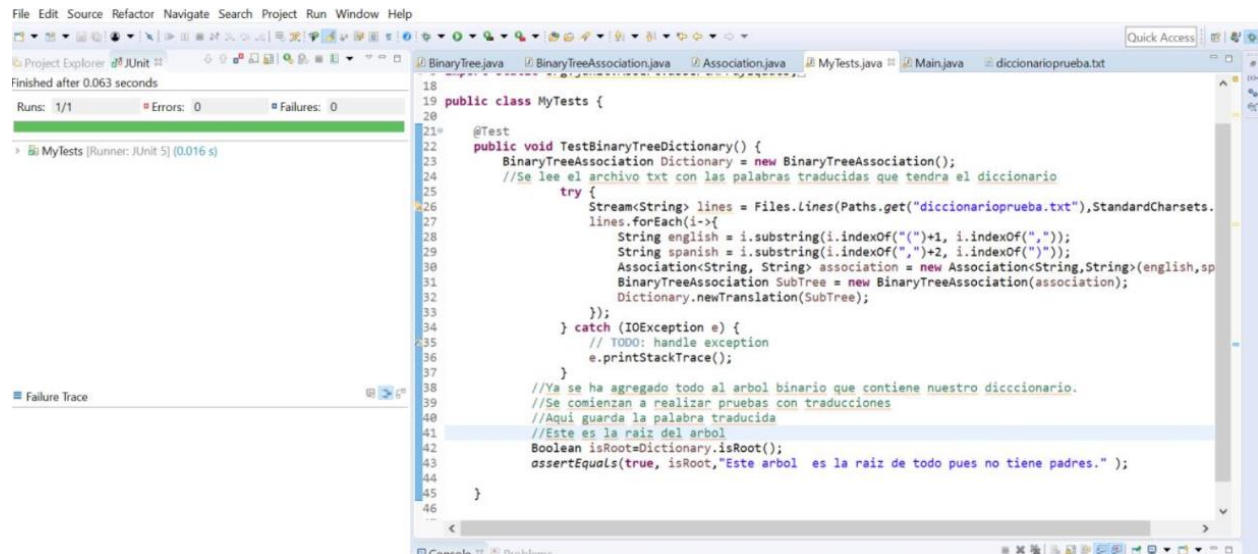
Los hijos guardan correctamente los datos en el orden solicitado.



```
18
19 public class MyTests {
20
21     @Test
22     public void TestBinaryTreeDictionary() {
23         BinaryTreeAssociation Dictionary = new BinaryTreeAssociation();
24         //Se lee el archivo txt con las palabras traducidas que tendra el diccionario
25         try {
26             Stream<String> lines = Files.lines(Paths.get("diccionarioprueba.txt"), StandardCharsets.
27                 lines.forEach(i->{
28                     String english = i.substring(i.indexOf("(")+1, i.indexOf(",");
29                     String spanish = i.substring(i.indexOf(",")+2, i.indexOf(")"));
30                     Association<String, String> association = new Association<String, String>(english, sp
31                     BinaryTreeAssociation SubTree = new BinaryTreeAssociation(association);
32                     Dictionary.newTranslation(SubTree);
33                 });
34             } catch (IOException e) {
35                 // TODO: handle exception
36                 e.printStackTrace();
37             }
38             //Ya se ha agregado todo al arbol binario que contiene nuestro diccionario.
39             //Se comienzan a realizar pruebas con traducciones
40             //Aqui guarda la palabra traducida
41             //Este hijo al que se hace referencia es una raiz, pues tiene padre
42             Boolean isRoot=Dictionary.right().left().isRoot();
43             assertEquals(false, isRoot, "Este arbol no es una raiz pues tiene padres." );
44
45         }
46     }
```

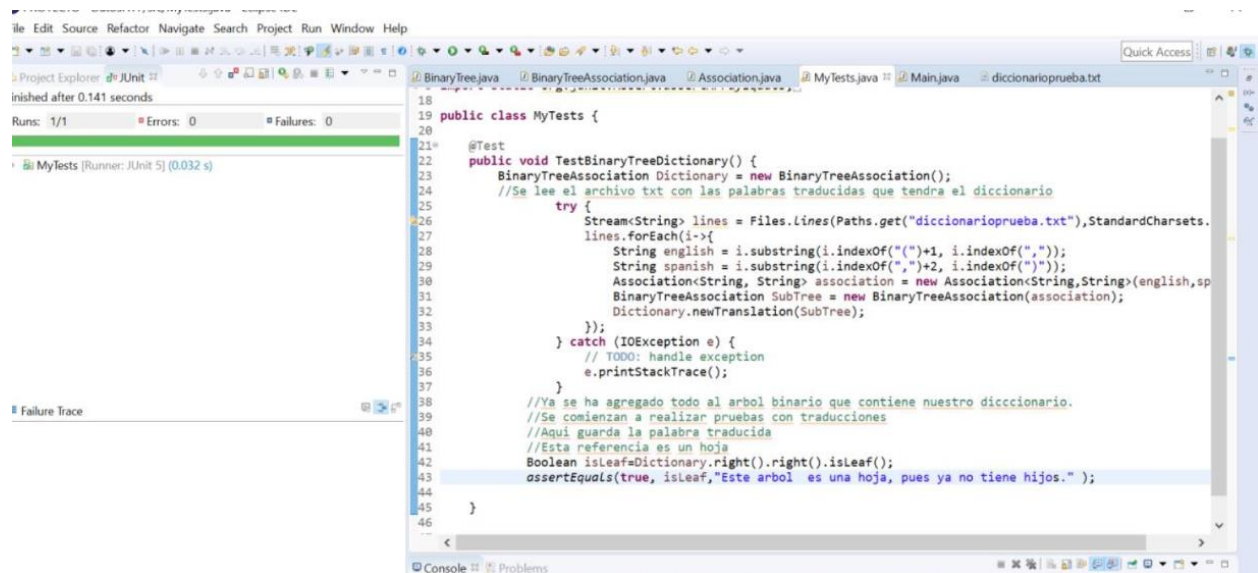
Verifica si un árbol es una raíz.

Daniela Villamar
19086



```
18
19 public class MyTests {
20
21     @Test
22     public void TestBinaryTreeDictionary() {
23         BinaryTreeAssociation Dictionary = new BinaryTreeAssociation();
24         //Se lee el archivo txt con las palabras traducidas que tendra el diccionario
25         try {
26             Stream<String> lines = Files.Lines(Paths.get("diccionarioprueba.txt"), StandardCharsets.
27                 lines.forEach(i->{
28                     String english = i.substring(i.indexOf("(")+1, i.indexOf(",");
29                     String spanish = i.substring(i.indexOf(",")+2, i.indexOf(")");
30                     Association<String, String> association = new Association<String, String>(english, sp
31                     BinaryTreeAssociation SubTree = new BinaryTreeAssociation(association);
32                     Dictionary.newTranslation(SubTree);
33                 });
34             } catch (IOException e) {
35                 // TODO: handle exception
36                 e.printStackTrace();
37             }
38             //Ya se ha agregado todo al arbol binario que contiene nuestro diccionario.
39             //Se comienzan a realizar pruebas con traducciones
40             //Aqui guarda la palabra traducida
41             //Este es la raíz del arbol
42             Boolean isRootsDictionary.isRoot();
43             assertEquals(true, isRoot, "Este arbol es la raíz de todo pues no tiene padres." );
44         }
45     }
46 }
```

Verifica que solo exista una raíz.



```
18
19 public class MyTests {
20
21     @Test
22     public void TestBinaryTreeDictionary() {
23         BinaryTreeAssociation Dictionary = new BinaryTreeAssociation();
24         //Se lee el archivo txt con las palabras traducidas que tendra el diccionario
25         try {
26             Stream<String> lines = Files.Lines(Paths.get("diccionarioprueba.txt"), StandardCharsets.
27                 lines.forEach(i->{
28                     String english = i.substring(i.indexOf("(")+1, i.indexOf(",");
29                     String spanish = i.substring(i.indexOf(",")+2, i.indexOf(")");
30                     Association<String, String> association = new Association<String, String>(english, sp
31                     BinaryTreeAssociation SubTree = new BinaryTreeAssociation(association);
32                     Dictionary.newTranslation(SubTree);
33                 });
34             } catch (IOException e) {
35                 // TODO: handle exception
36                 e.printStackTrace();
37             }
38             //Ya se ha agregado todo al arbol binario que contiene nuestro diccionario.
39             //Se comienzan a realizar pruebas con traducciones
40             //Aqui guarda la palabra traducida
41             //Esta referencia es un hoja
42             Boolean isLeaf=Dictionary.right().right().isLeaf();
43             assertEquals(true, isLeaf, "Este arbol es una hoja, pues ya no tiene hijos." );
44         }
45     }
46 }
```

Verifica cuando un árbol ya es una hoja.

Todos los métodos funcionan y el árbol busca e inserta correctamente.