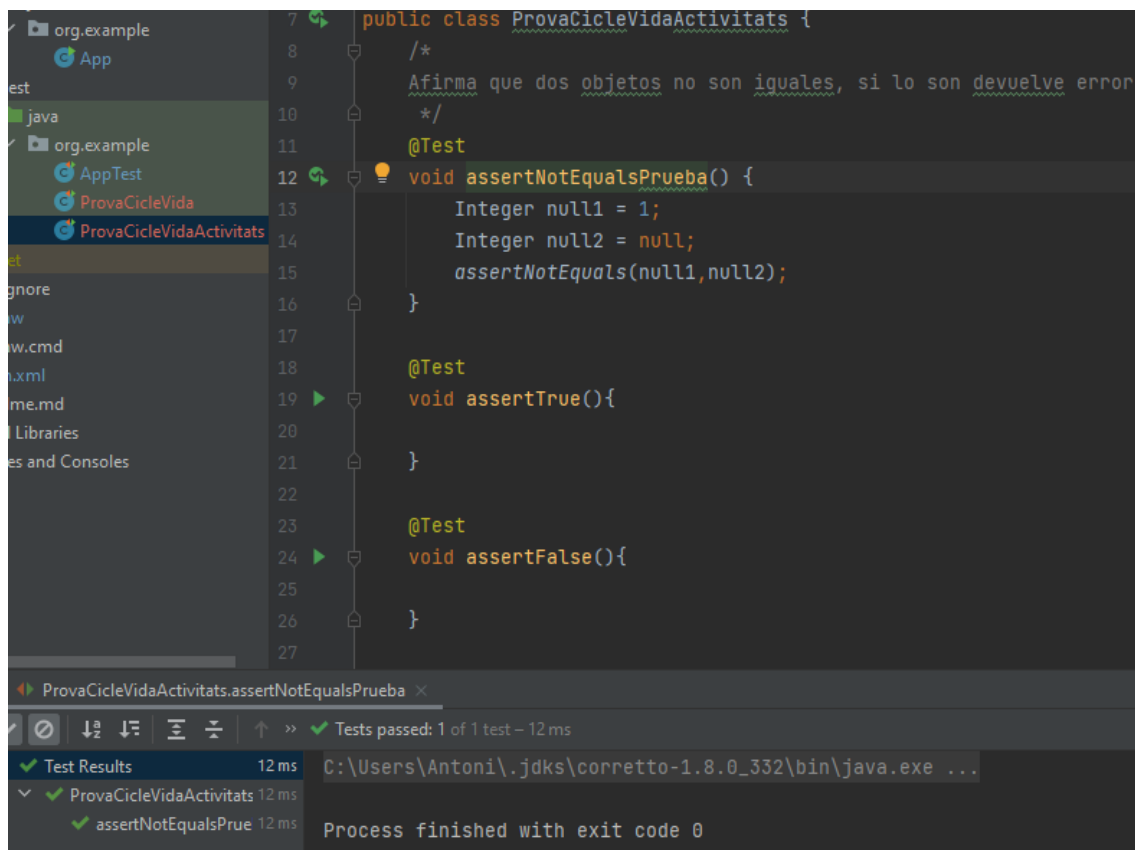
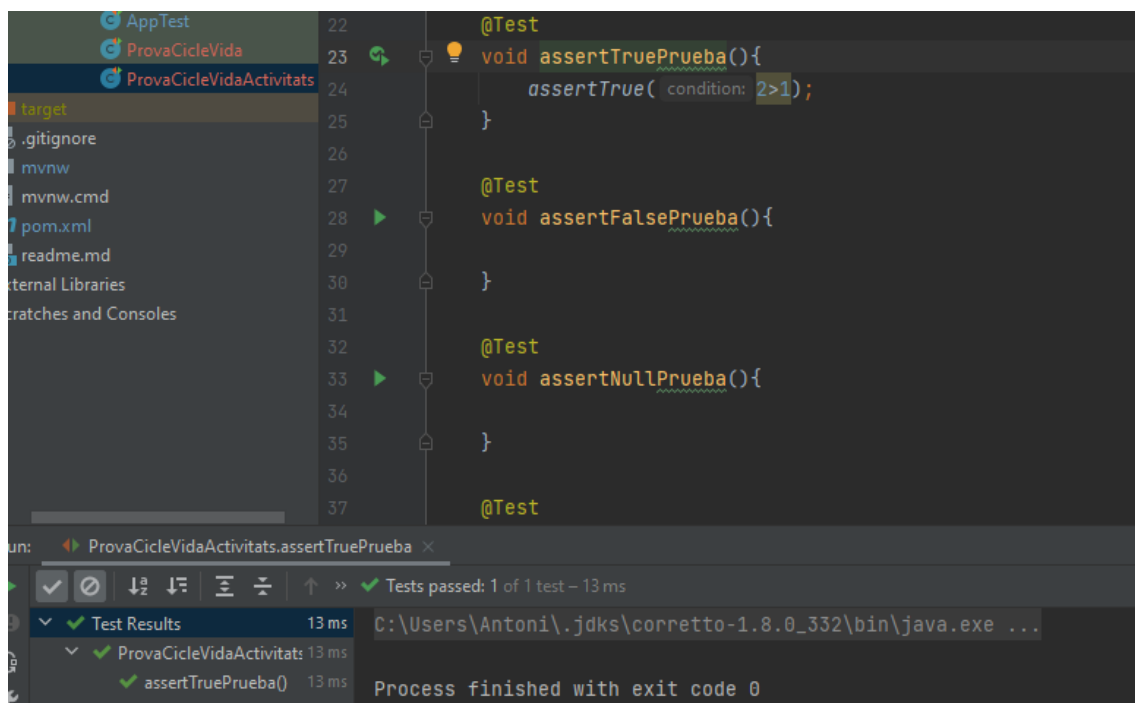


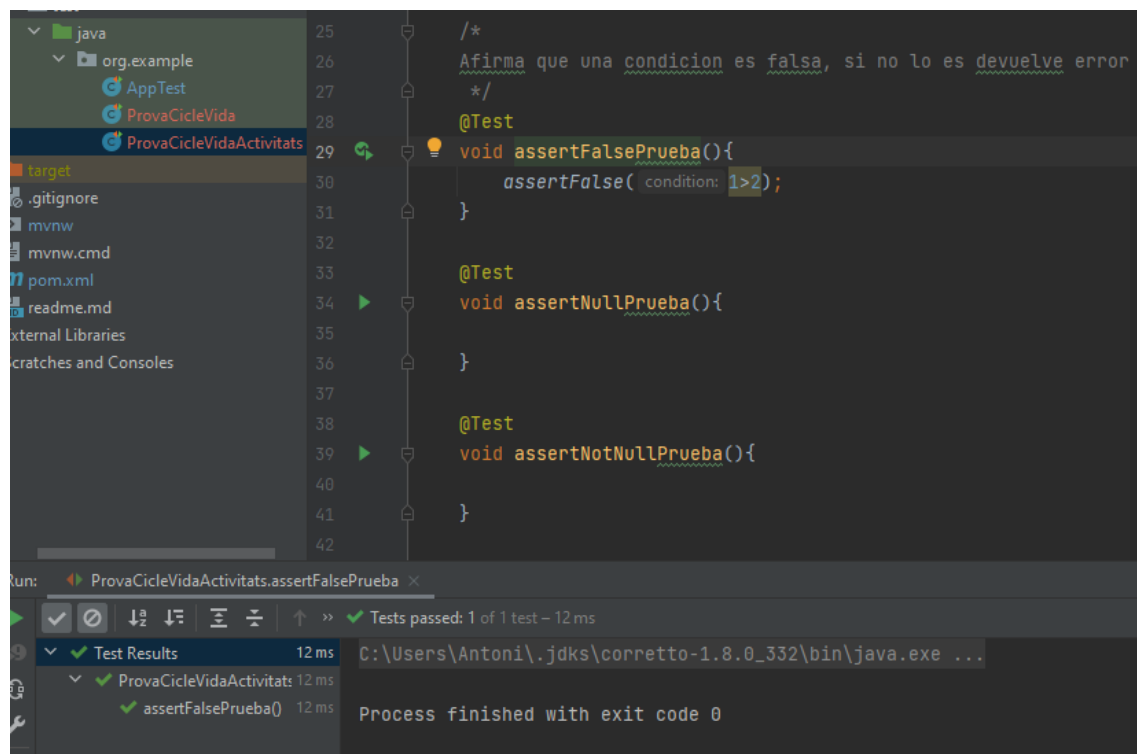
assertNotEquals



assertTrue



assertFalse

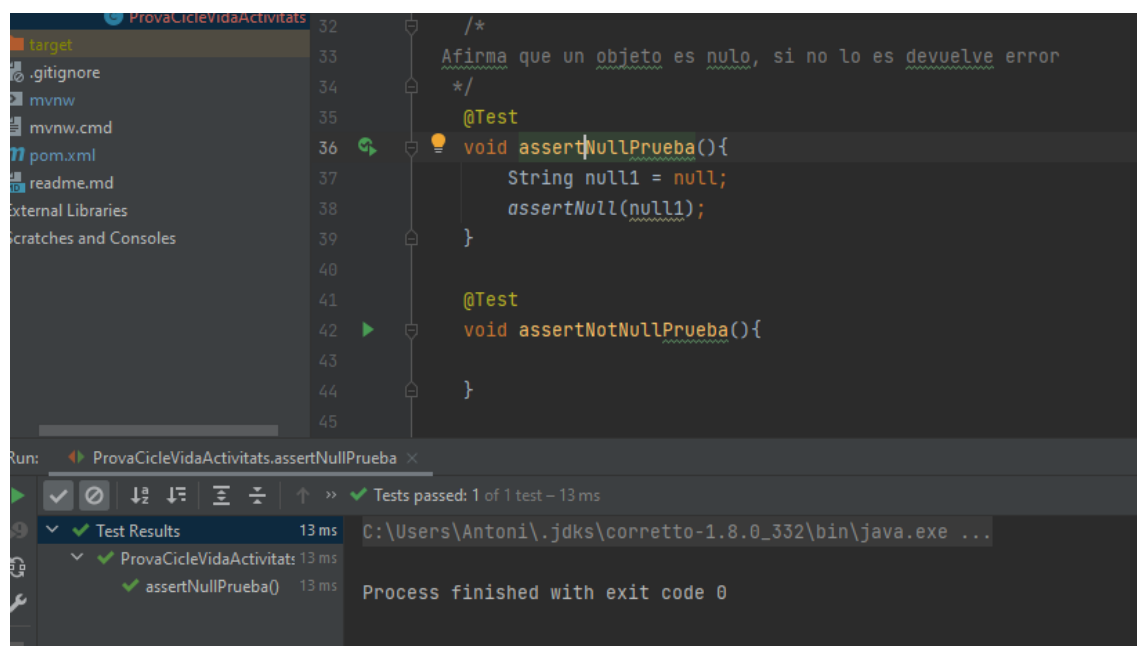


The screenshot shows an IDE with a project structure on the left and a code editor on the right. The project structure includes a 'java' package with 'org.example' containing 'AppTest', 'ProvaCicleVida', and 'ProvaCicleVidaActivitats'. The 'target' directory is also visible. The code editor shows the following Java code:

```
25  /*
26  Afirma que una condicion es falsa, si no lo es devuelve error
27  */
28  @Test
29  void assertFalsePrueba(){
30      assertFalse( condition: 1>2);
31  }
32
33  @Test
34  void assertNullPrueba(){
35  }
36
37
38  @Test
39  void assertNotNullPrueba(){
40  }
41
42
```

Below the code editor, the 'Run' tab shows the execution of 'ProvaCicleVidaActivitats.assertFalsePrueba'. The test passed in 12 ms. The console output shows the command 'C:\Users\Antoni\.jdk\corretto-1.8.0_332\bin\java.exe ...' and 'Process finished with exit code 0'.

assertNull

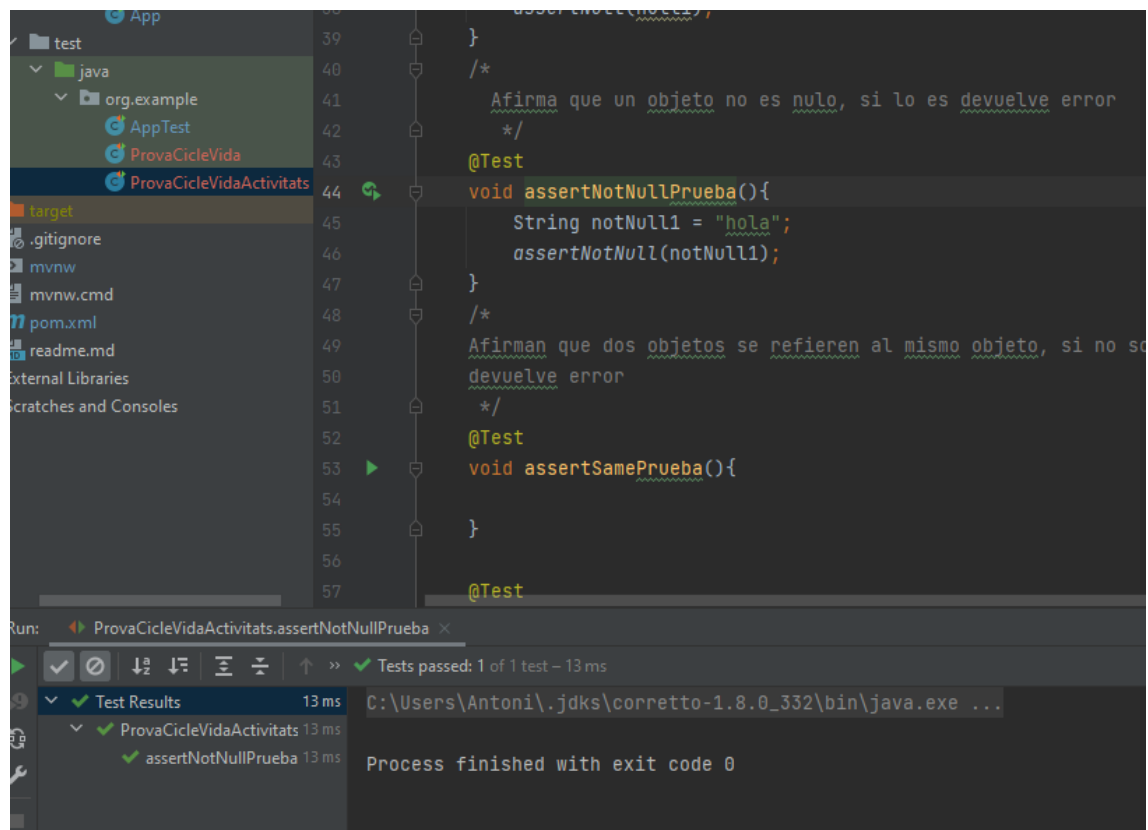


The screenshot shows an IDE with a project structure on the left and a code editor on the right. The project structure is the same as the previous screenshot. The code editor shows the following Java code:

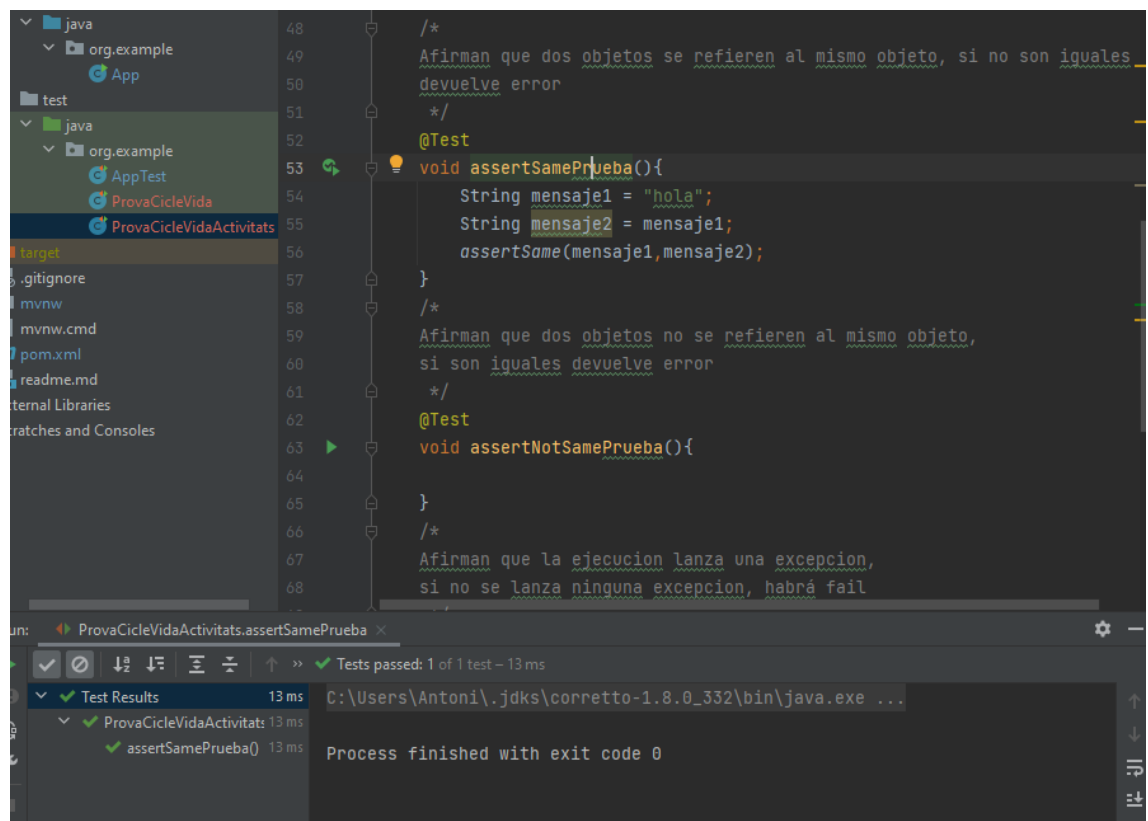
```
32  /*
33  Afirma que un objeto es nulo, si no lo es devuelve error
34  */
35  @Test
36  void assertNullPrueba(){
37      String null1 = null;
38      assertNull(null1);
39  }
40
41  @Test
42  void assertNotNullPrueba(){
43  }
44
45
```

Below the code editor, the 'Run' tab shows the execution of 'ProvaCicleVidaActivitats.assertNullPrueba'. The test passed in 13 ms. The console output shows the command 'C:\Users\Antoni\.jdk\corretto-1.8.0_332\bin\java.exe ...' and 'Process finished with exit code 0'.

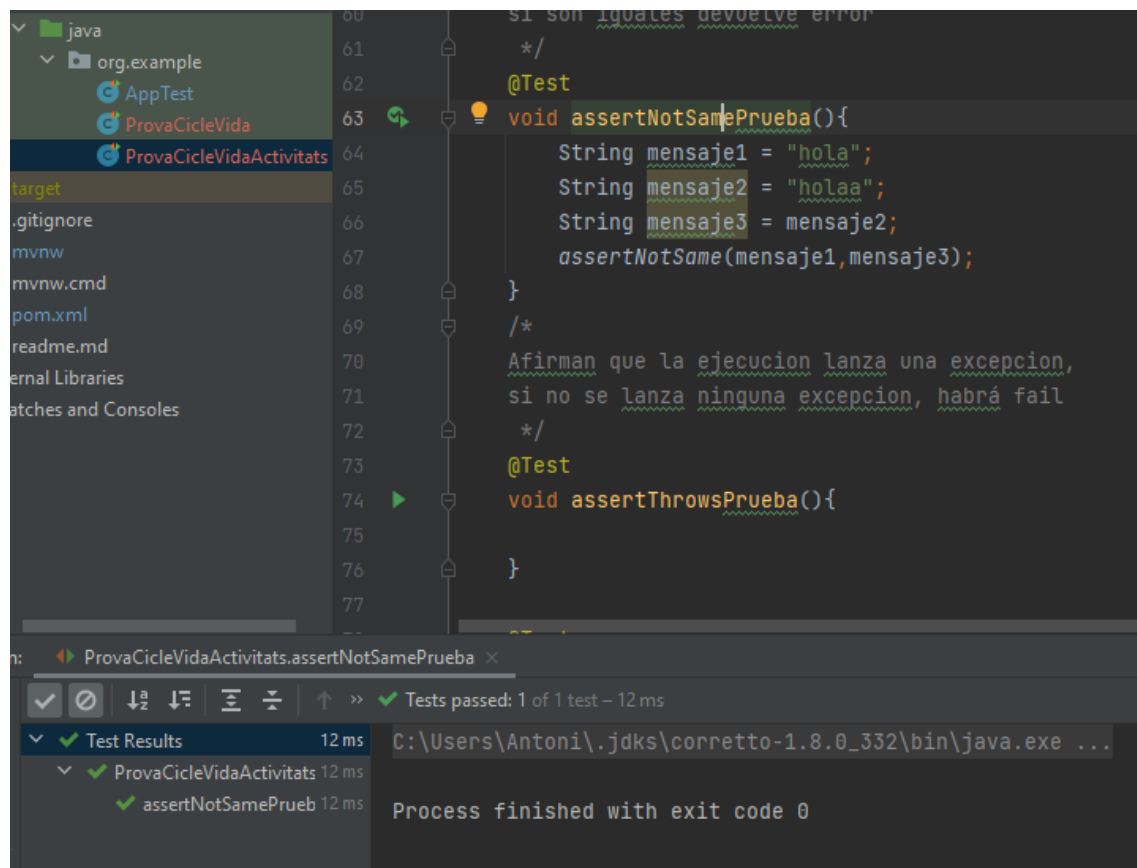
assertNotNull



assertSame



assertNotSame



```
60  /*
61  */
62  @Test
63  void assertNotSamePrueba(){
64      String mensaje1 = "hola";
65      String mensaje2 = "holaa";
66      String mensaje3 = mensaje2;
67      assertNotSame(mensaje1,mensaje3);
68  }
69  /*
70  Afirman que la ejecucion lanza una excepcion,
71  si no se lanza ninguna excepcion, habrá fail
72  */
73  @Test
74  void assertThrowsPrueba(){
75
76  }
77
```

Tests passed: 1 of 1 test – 12 ms

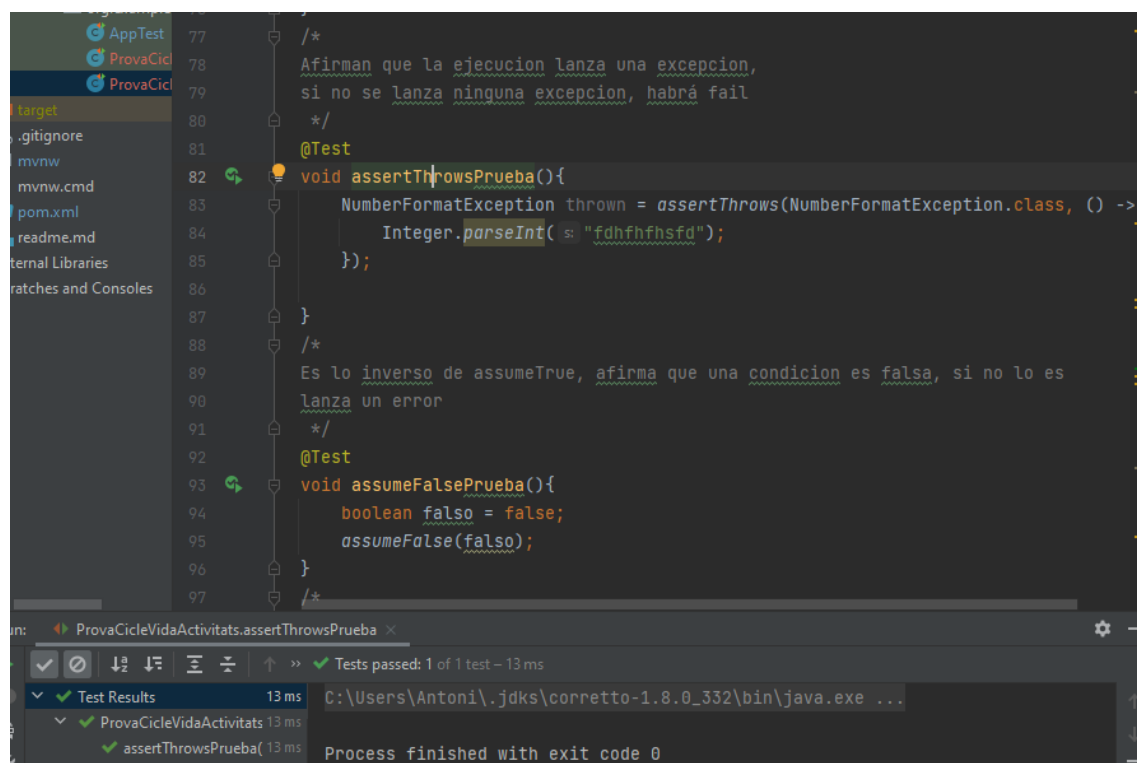
Test Results 12 ms

ProvaCicleVidaActivitats 12 ms

assertNotSamePrueb 12 ms

Process finished with exit code 0

assertThrows



```
77  /*
78  Afirman que la ejecucion lanza una excepcion,
79  si no se lanza ninguna excepcion, habrá fail
80  */
81  @Test
82  void assertThrowsPrueba(){
83      NumberFormatException thrown = assertThrows(NumberFormatException.class, () ->
84          Integer.parseInt("fdhfhfsfd");
85      });
86  }
87  /*
88  Es lo inverso de assumeTrue, afirma que una condicion es falsa, si no lo es
89  lanza un error
90  */
91  @Test
92  void assumeFalsePrueba(){
93      boolean falso = false;
94      assumeFalse(falso);
95  }
96  }
97  /*
```

Tests passed: 1 of 1 test – 13 ms

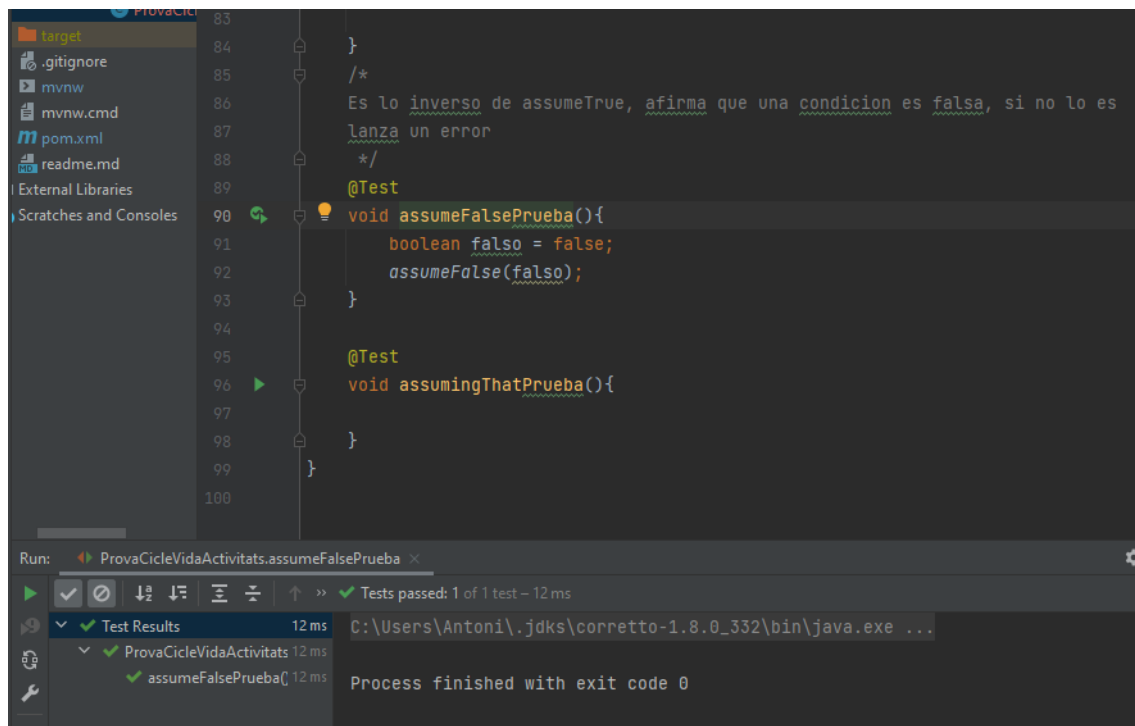
Test Results 13 ms

ProvaCicleVidaActivitats 13 ms

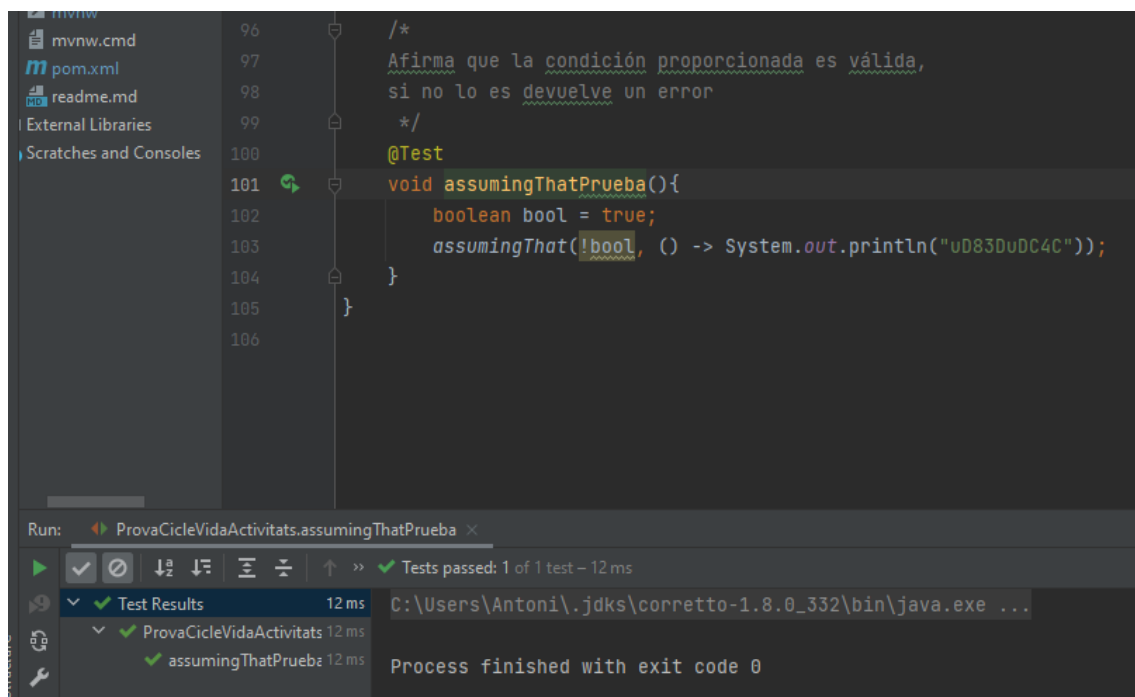
assertThrowsPrueba(13 ms

Process finished with exit code 0

assumeFalse



assumingThat



Todos:

