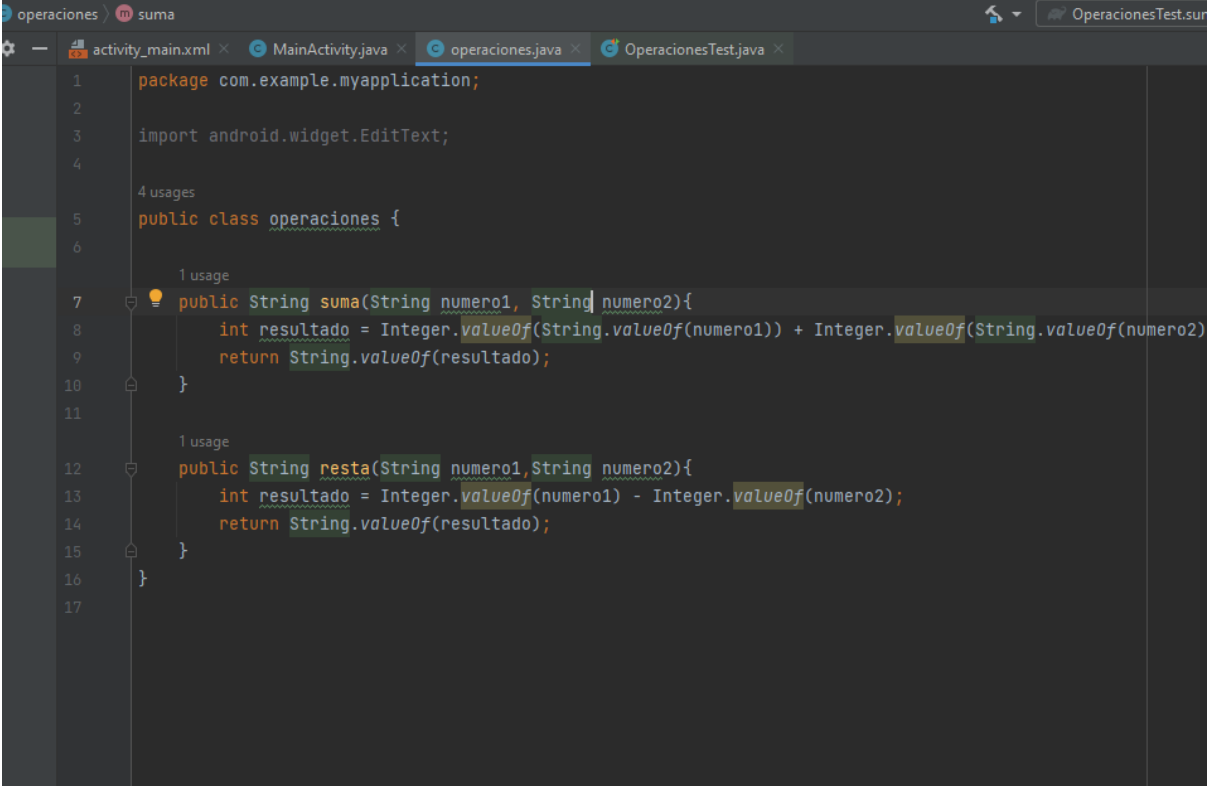


Tests Android Studio

1. Clase Operaciones con la funcionalidad de cada elemento

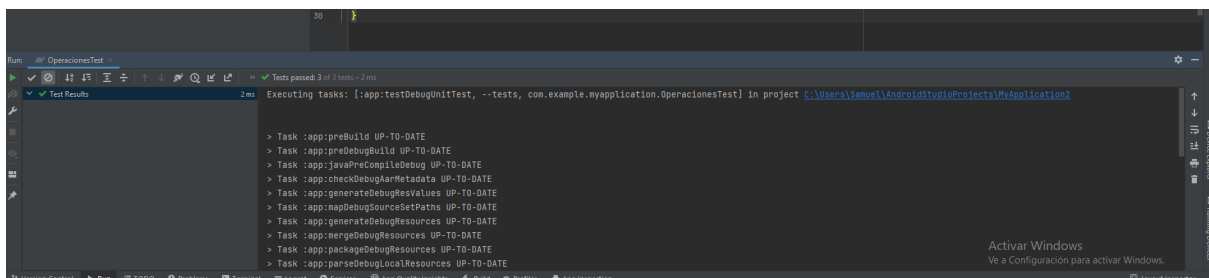


```
1 package com.example.myapplication;
2
3 import android.widget.EditText;
4
5 4 usages
6 public class operaciones {
7
8     1 usage
9     public String suma(String numero1, String numero2){
10         int resultado = Integer.valueOf(String.valueOf(numero1)) + Integer.valueOf(String.valueOf(numero2));
11         return String.valueOf(resultado);
12     }
13
14     1 usage
15     public String resta(String numero1, String numero2){
16         int resultado = Integer.valueOf(numero1) - Integer.valueOf(numero2);
17         return String.valueOf(resultado);
18     }
19 }
```

2. Se crea un test con Junit y se escribe el código para el funcionamiento con los datos a enviar y los datos esperados

```
activity_main.xml x MainActivity.java x operaciones.java x OperacionesTest.java x
1 package com.example.myapplication;
2
3 import ...
7
8 no usages
9 public class OperacionesTest {
10
11     4 usages
12     private operaciones nOper;
13
14     no usages
15     @Before
16     public void setUp() { nOper = new operaciones(); }
17
18     no usages
19     @Test
20     public void NotNull() { assertNotNull(nOper); }
21
22     no usages
23     @Test
24     public void suma() { assertEquals( expected: "7",nOper.suma("4","3")); }
25
26     no usages
27     @Test
28     public void resta() { assertEquals( expected: "1",nOper.resta("3","2")); }
29 }
30
```

3. Se corren los test para comprobar que el funcionamiento este correcto



En caso de que sea incorrecto

