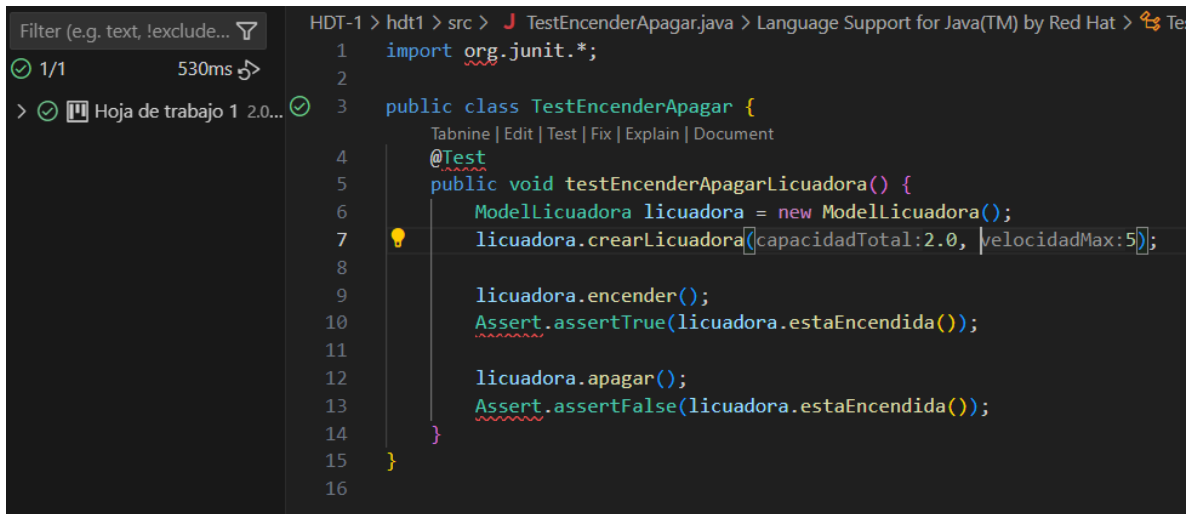


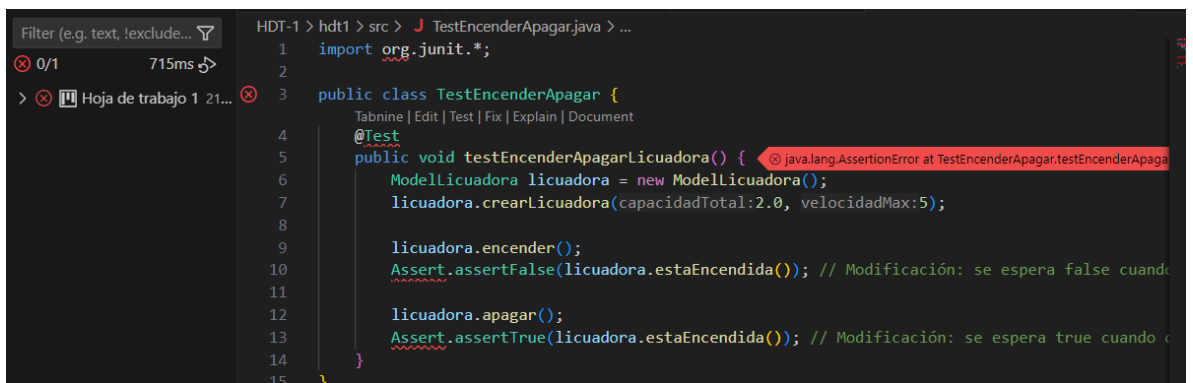
Encender y apagar



The screenshot shows an IDE window with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'HDT-1' with a sub-project 'hdt1' containing a source file 'TestEncenderApagar.java'. The code editor displays the following Java code:

```
1 import org.junit.*;
2
3 public class TestEncenderApagar {
4     @Test
5     public void testEncenderApagarLicuadora() {
6         ModelLicuadora licuadora = new ModelLicuadora();
7         licuadora.crearLicuadora(capacidadTotal:2.0, velocidadMax:5);
8
9         licuadora.encender();
10        Assert.assertTrue(licuadora.estaEncendida());
11
12        licuadora.apagar();
13        Assert.assertFalse(licuadora.estaEncendida());
14    }
15 }
```

The test run is successful, indicated by a green checkmark and a green arrow next to the test name '1/1' in the file explorer. The execution time is 530ms.

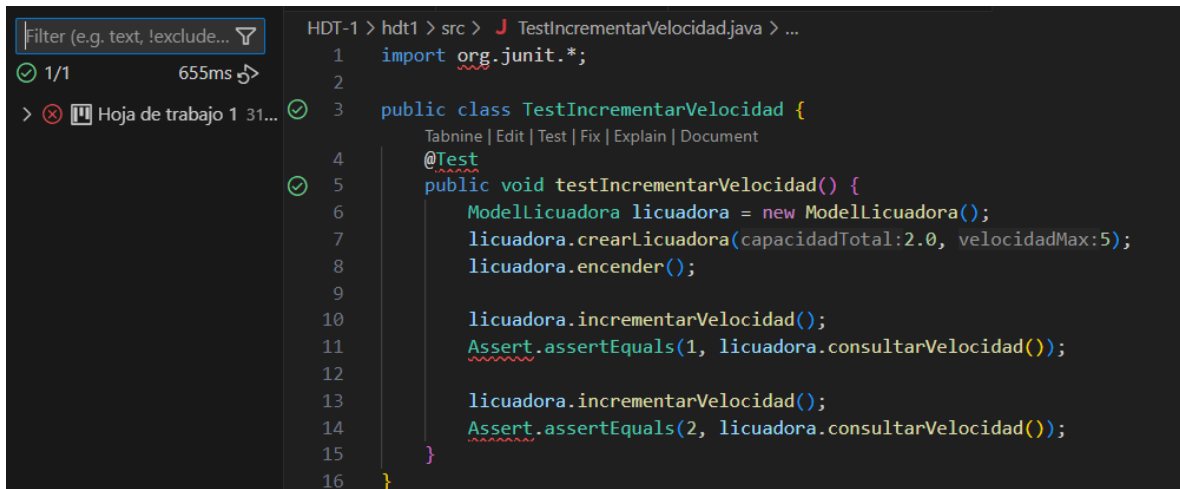


The screenshot shows the same IDE window as the previous one, but the test run has failed. The file explorer shows a red 'X' next to the test name '0/1' and a red arrow. The execution time is 715ms. The code editor displays the same Java code as before, but with a red error message in the background:

```
1 import org.junit.*;
2
3 public class TestEncenderApagar {
4     @Test
5     public void testEncenderApagarLicuadora() {
6         ModelLicuadora licuadora = new ModelLicuadora();
7         licuadora.crearLicuadora(capacidadTotal:2.0, velocidadMax:5);
8
9         licuadora.encender();
10        Assert.assertFalse(licuadora.estaEncendida()); // Modificación: se espera false cuando
11
12        licuadora.apagar();
13        Assert.assertTrue(licuadora.estaEncendida()); // Modificación: se espera true cuando
14    }
15 }
```

The error message is: `java.lang.AssertionError at TestEncenderApagar.testEncenderApagarLicuadora()`. The test is failing because the assertion on line 10 is incorrect. The comment indicates that the test expects false when the blender is turned on, but the code is using `Assert.assertFalse` instead of `Assert.assertTrue`.

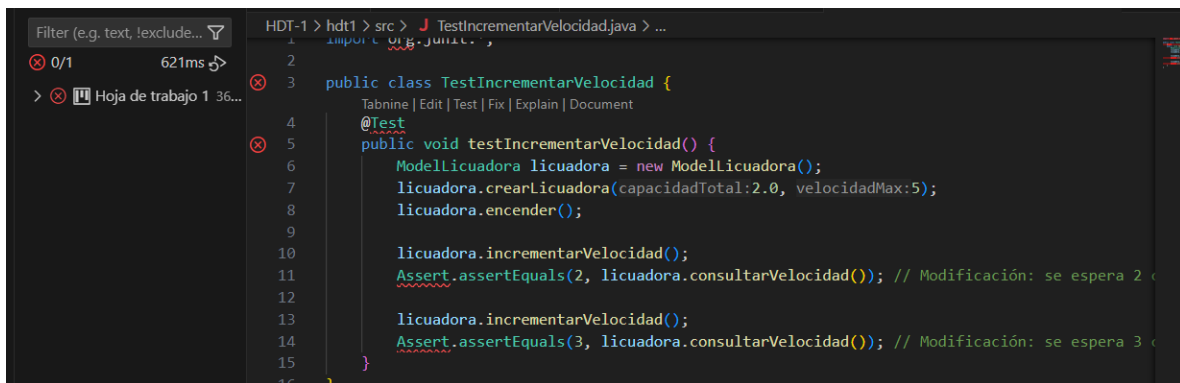
Incrementar Velocidad



The screenshot shows an IDE window with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'HDT-1' with a sub-project 'hdt1' containing a source file 'TestIncrementarVelocidad.java'. The code editor displays the following Java code:

```
1 import org.junit.*;
2
3 public class TestIncrementarVelocidad {
4     @Test
5     public void testIncrementarVelocidad() {
6         ModelLicuadora licuadora = new ModelLicuadora();
7         licuadora.crearLicuadora(capacidadTotal:2.0, velocidadMax:5);
8         licuadora.encender();
9
10        licuadora.incrementarVelocidad();
11        Assert.assertEquals(1, licuadora.consultarVelocidad());
12
13        licuadora.incrementarVelocidad();
14        Assert.assertEquals(2, licuadora.consultarVelocidad());
15    }
16 }
```

The test run results on the left show a green checkmark and '1/1' tests passed, with a duration of 655ms.

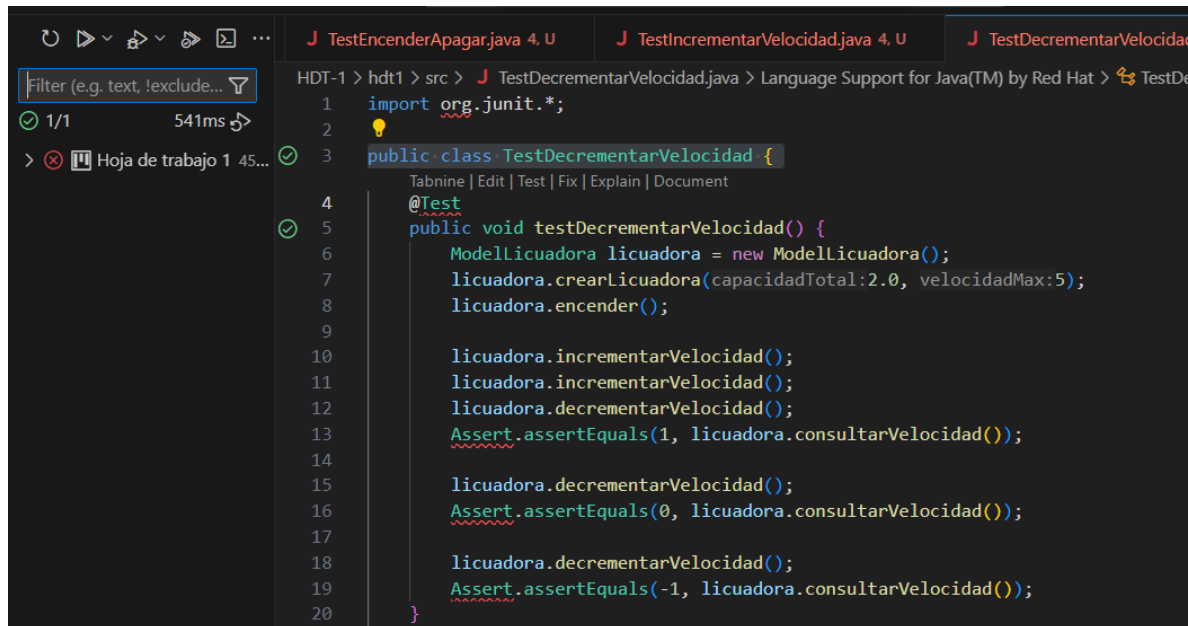


The screenshot shows the same IDE window as above, but the test run results on the left now show a red 'X' and '0/1' tests failed, with a duration of 621ms. The code editor displays the following Java code:

```
1 import org.junit.*;
2
3 public class TestIncrementarVelocidad {
4     @Test
5     public void testIncrementarVelocidad() {
6         ModelLicuadora licuadora = new ModelLicuadora();
7         licuadora.crearLicuadora(capacidadTotal:2.0, velocidadMax:5);
8         licuadora.encender();
9
10        licuadora.incrementarVelocidad();
11        Assert.assertEquals(2, licuadora.consultarVelocidad()); // Modificación: se espera 2
12
13        licuadora.incrementarVelocidad();
14        Assert.assertEquals(3, licuadora.consultarVelocidad()); // Modificación: se espera 3
15    }
16 }
```

The test run results on the left show a red 'X' and '0/1' tests failed, with a duration of 621ms.

Decrementar Velocidad



```
1 import org.junit.*;
2
3 public class TestDecrementarVelocidad {
4     @Test
5     public void testDecrementarVelocidad() {
6         ModelLicuadora licuadora = new ModelLicuadora();
7         licuadora.crearLicuadora(capacidadTotal:2.0, velocidadMax:5);
8         licuadora.encender();
9
10        licuadora.incrementarVelocidad();
11        licuadora.incrementarVelocidad();
12        licuadora.decrementarVelocidad();
13        Assert.assertEquals(1, licuadora.consultarVelocidad());
14
15        licuadora.decrementarVelocidad();
16        Assert.assertEquals(0, licuadora.consultarVelocidad());
17
18        licuadora.decrementarVelocidad();
19        Assert.assertEquals(-1, licuadora.consultarVelocidad());
20    }
```



```
1 import org.junit.*;
2
3 public class TestDecrementarVelocidad {
4     @Test
5     public void testDecrementarVelocidad() {
6         ModelLicuadora licuadora = new ModelLicuadora();
7         licuadora.crearLicuadora(capacidadTotal:2.0, velocidadMax:5);
8         licuadora.encender();
9
10        licuadora.incrementarVelocidad();
11        licuadora.incrementarVelocidad();
12        licuadora.decrementarVelocidad();
13        Assert.assertEquals(2, licuadora.consultarVelocidad()); // Modificación: se espera 2 c
14
15        licuadora.decrementarVelocidad();
16        Assert.assertEquals(1, licuadora.consultarVelocidad()); // Modificación: se espera 1 c
17    }
```