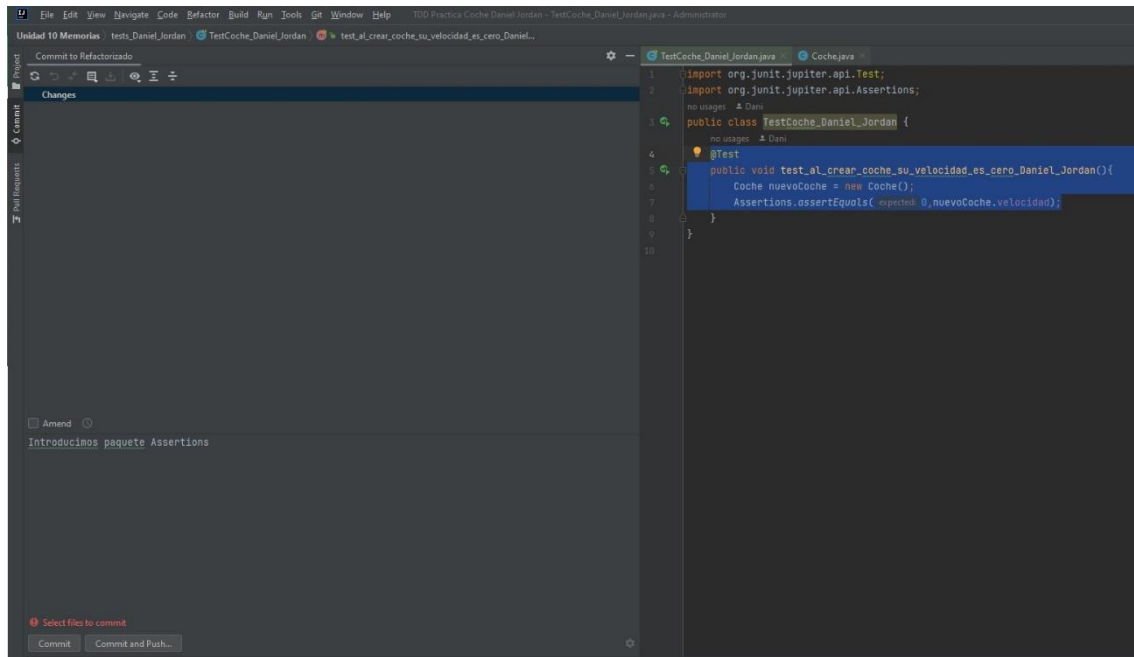
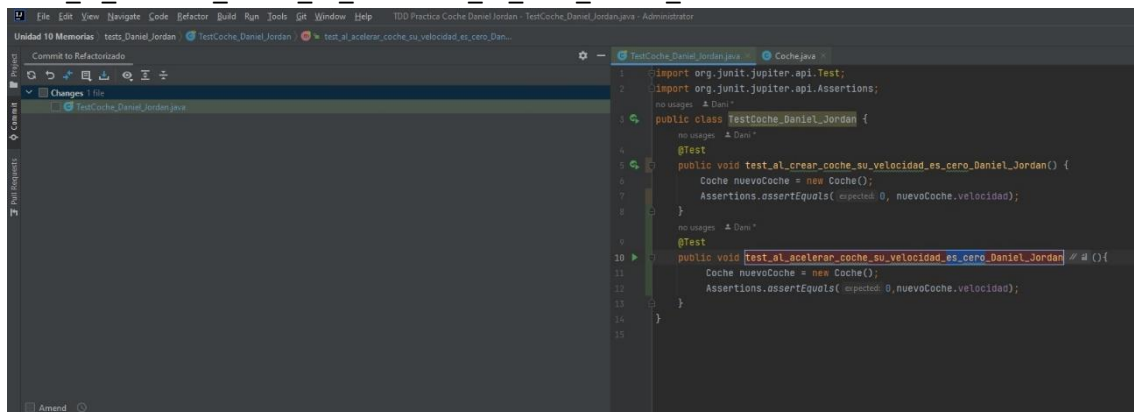


Entornos de Desarrollo

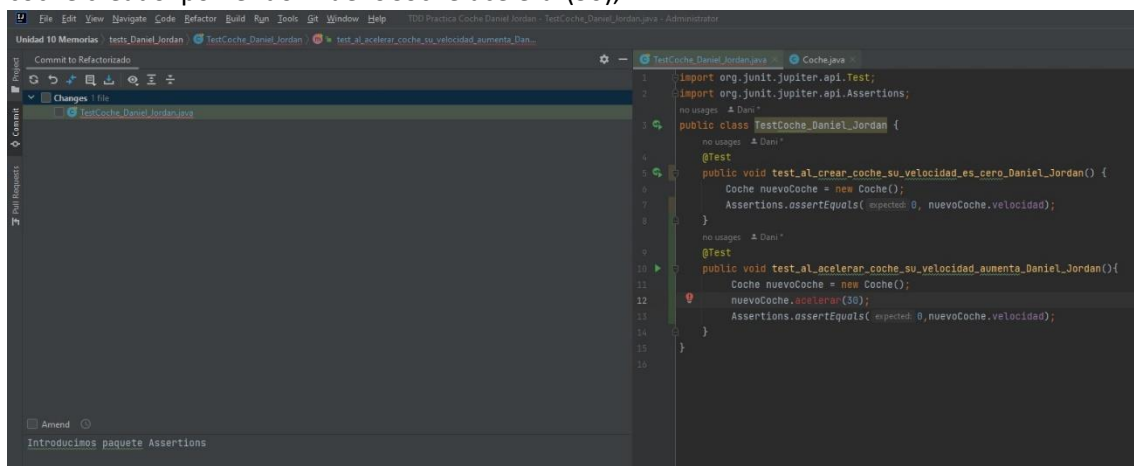
Test 2 TDD IntelliJ



Comenzamos copiando el método anterior y pegándolo abajo del mismo. Refactorizamos o modificamos el texto del método y lo modificamos para que ponga “test_al_acelerar_coche_su_velocidad_es_cero_Daniel_Jordan”

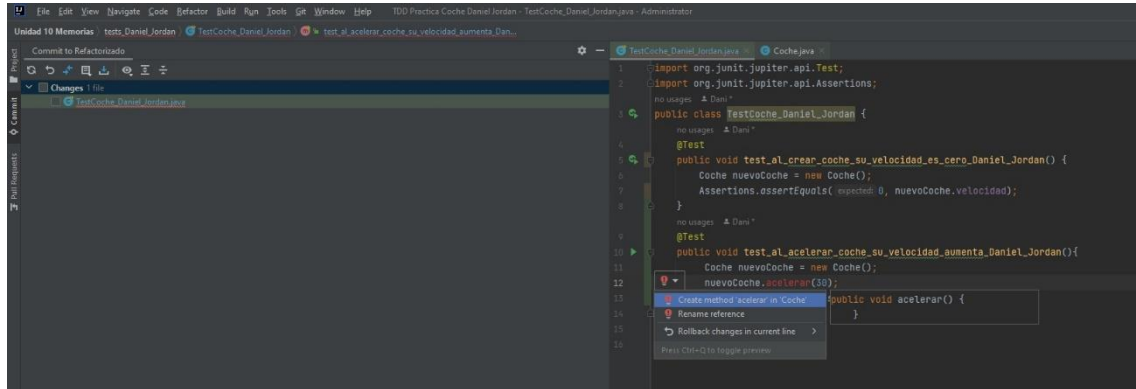


Una vez modificado, después de la creación del objeto coche le damos una aceleración al coche creado poniendo “nuevoCoche.acelerar(30);”

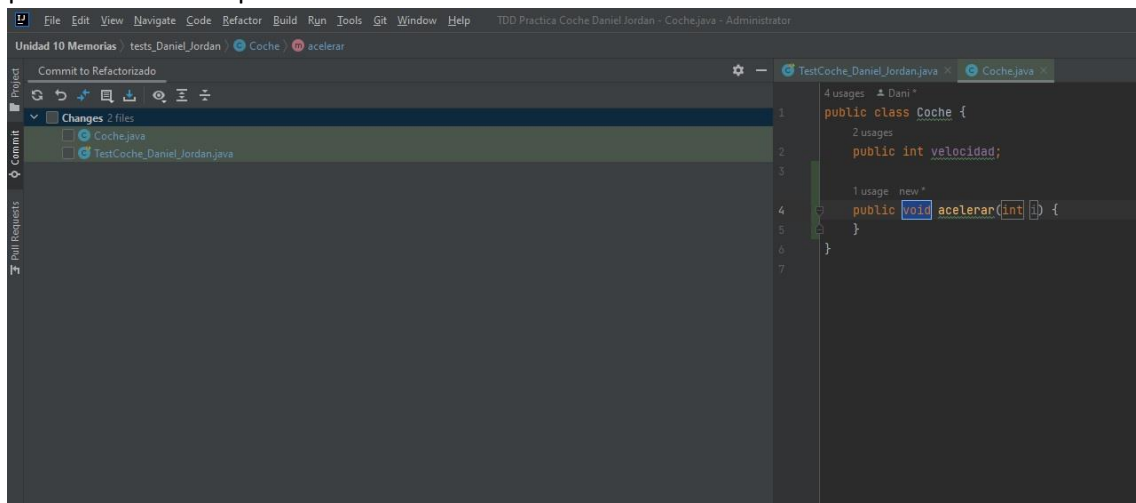


Ahora nos marcará en rojo el método acelerar de la clase coche ya que no lo tenemos creado, haciendo click en el método acelerar y luego haciendo click en la bombilla roja podemos crearlo automáticamente. Hay que poner en Assertions.assertEquals(30 una velocidad de 30 ...

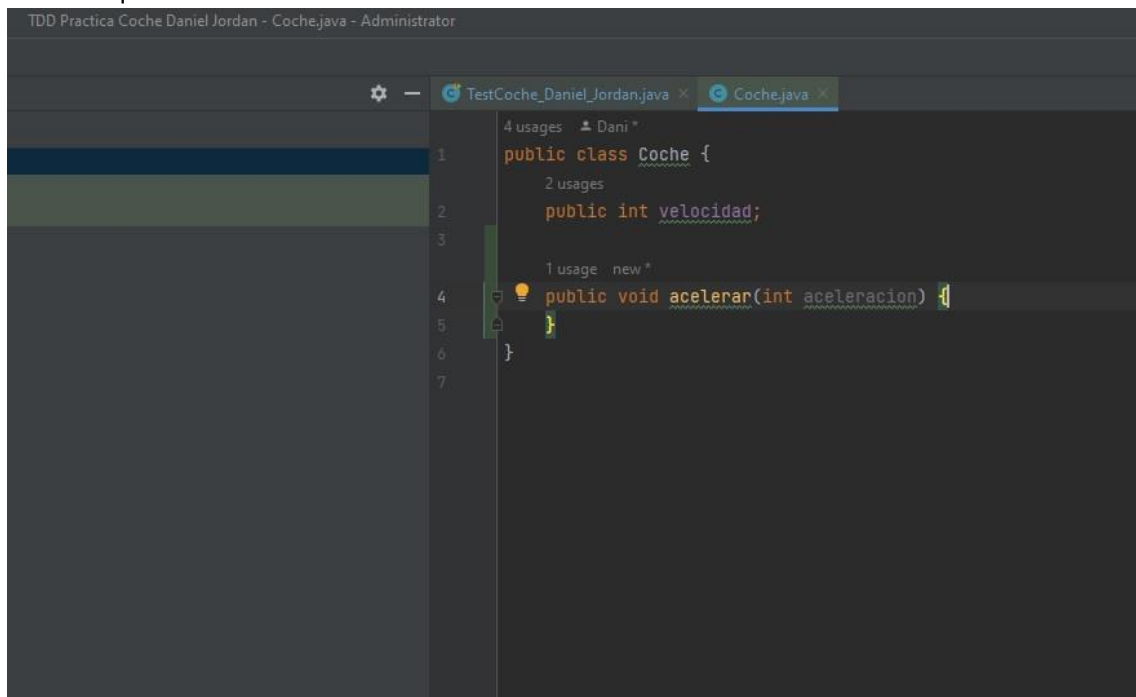
que es la velocidad que se le va a pasar para realizar el test es lo que se espera que nos de.



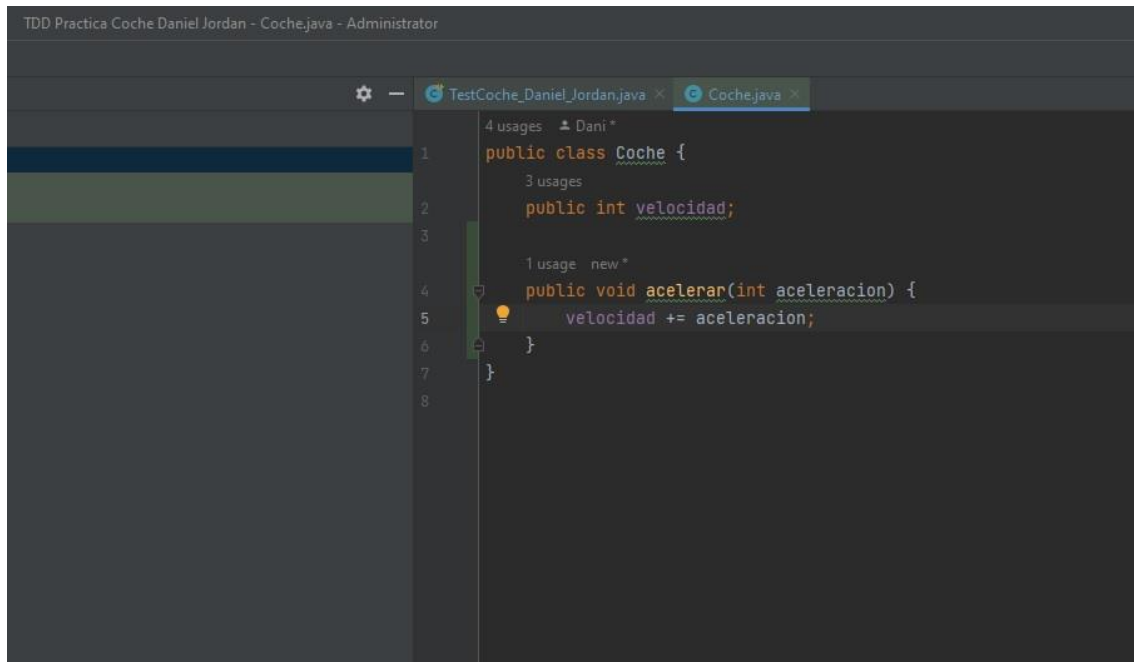
En la clase coche ya se nos crea el método acelerar, lo que tenemos que cambiar es la variable que se le pasa al método y la renombramos pinchando y arrastrando encima y reescribiendo para sustituir la “i” por “aceleración”



Y se nos queda así.



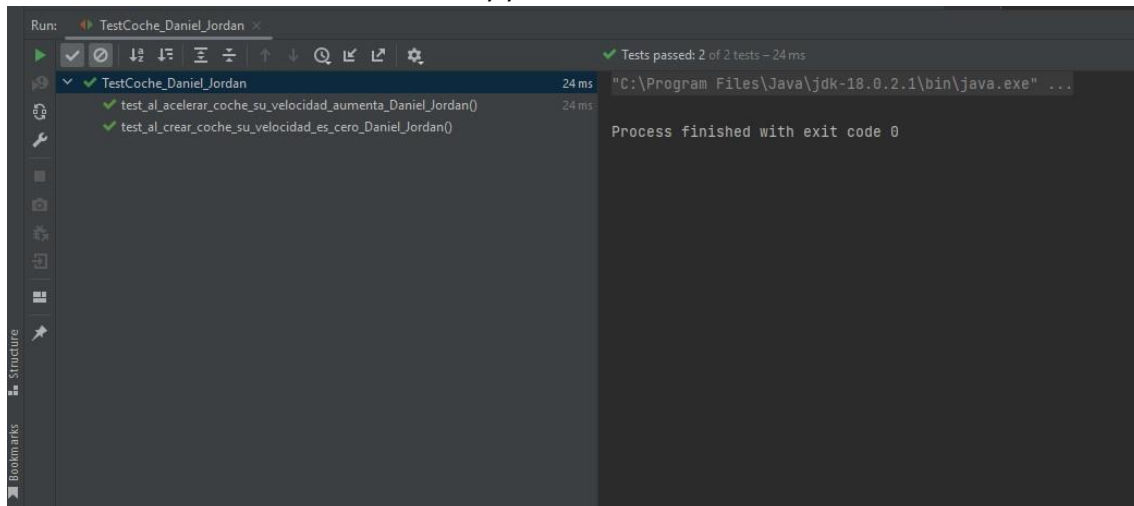
Dentro del método vamos a definirlo diciendo que la velocidad del coche aumenta con la aceleración que se le pasa del siguiente modo: “velocidad += aceleración”



The screenshot shows an IDE window titled "TDD Practica Coche Daniel Jordan - Coche.java - Administrator". It contains two tabs: "TestCoche_Daniel_Jordan.java" and "Coche.java". The "Coche.java" tab is active, displaying the following Java code:

```
1 public class Coche {  
2     public int velocidad;  
3  
4     public void acelerar(int aceleracion) {  
5         velocidad += aceleracion;  
6     }  
7 }  
8
```

Ahora le damos al símbolo verde del Play y realizamos el test satisfactoriamente



The screenshot shows the IDE's "Run" window. It displays the test results for "TestCoche_Daniel_Jordan". The tests passed are:

- test_al_acelerar_coche_su_velocidad_aumenta_Daniel_Jordan() (24 ms)
- test_al_crear_coche_su_velocidad_es_cero_Daniel_Jordan() (24 ms)

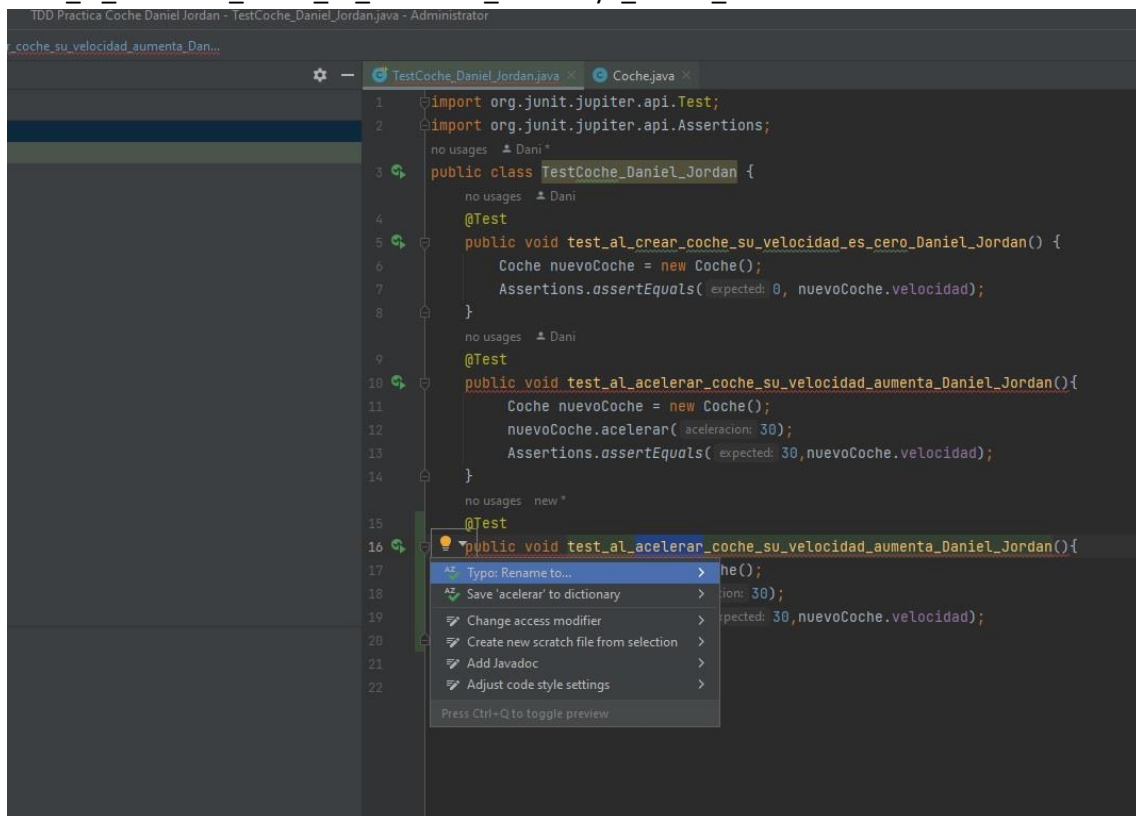
The output shows "Tests passed: 2 of 2 tests - 24 ms" and "Process finished with exit code 0".

Copiamos el test anterior y lo pegamos debajo de este para realizar el test de deceleración.

```
5  @Test
6  public void test_al_crear_coche_su_velocidad_es_cero_Daniel_Jordan() {
7      Coche nuevoCoche = new Coche();
8      Assertions.assertEquals( expected: 0, nuevoCoche.velocidad);
9  }
10 no usages Dani
11 @Test
12 public void test_al_acelerar_coche_su_velocidad_aumenta_Daniel_Jordan(){
13     Coche nuevoCoche = new Coche();
14     nuevoCoche.acelerar( aceleracion: 30);
15     Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
16 }
17 no usages new *
18 @Test
19 public void test_al_acelerar_coche_su_velocidad_aumenta_Daniel_Jordan(){
20     Coche nuevoCoche = new Coche();
21     nuevoCoche.acelerar( aceleracion: 30);
22     Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
23 }
24 }
```

Seleccionamos la palabra del test a modificar y con la bombilla amarilla de la izquierda tenemos la posibilidad de renombrar el texto marcado. Con lo que lo modificaremos para que quede de la siguiente manera:

“test_al_decelerar_coche_su_velocidad_disminuye_Daniel_Jordan.



```
1  import org.junit.jupiter.api.Test;
2  import org.junit.jupiter.api.Assertions;
3  public class TestCoche_Daniel_Jordan {
4      @Test
5      public void test_al_crear_coche_su_velocidad_es_cero_Daniel_Jordan() {
6          Coche nuevoCoche = new Coche();
7          Assertions.assertEquals( expected: 0, nuevoCoche.velocidad);
8      }
9      @Test
10     public void test_al_acelerar_coche_su_velocidad_aumenta_Daniel_Jordan(){
11         Coche nuevoCoche = new Coche();
12         nuevoCoche.acelerar( aceleracion: 30);
13         Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
14     }
15     @Test
16     public void test_al_acelerar_coche_su_velocidad_aumenta_Daniel_Jordan(){
17         Coche nuevoCoche = new Coche();
18         nuevoCoche.acelerar( aceleracion: 30);
19         Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
20     }
21     @Test
22     public void test_al_decelerar_coche_su_velocidad_disminuye_Daniel_Jordan(){
23         Coche nuevoCoche = new Coche();
24         nuevoCoche.decelerar( deceleracion: 30);
25         Assertions.assertEquals( expected: 0,nuevoCoche.velocidad);
26     }
27 }
```

Tal cual la siguiente imagen

```
no usages Dani
@Test
public void test_al_acelerar_coche_su_velocidad_aumenta_Daniel_Jordan(){
    Coche nuevoCoche = new Coche();
    nuevoCoche.acelerar( aceleracion: 30);
    Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
}

no usages new *
@Test
public void test_al_decelerar_coche_su_velocidad_disminuye_Daniel_Jordan // # () {
    Coche nuevoCoche = new Coche();
    nuevoCoche.acelerar( aceleracion: 30);
    Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
}
}
```

Ahora sustituimos el método acelerar aplicado al coche nuevo por el método decelerar y le pasamos un valor de (20) a la deceleración. Además tenemos que darle una velocidad actual al vehículo de (50) y al Assertion le indicamos 30 para que al realizar el test todo sea correcto.

```
no usages Dani
@Test
public void test_al_acelerar_coche_su_velocidad_aumenta_Daniel_Jordan(){
    Coche nuevoCoche = new Coche();
    nuevoCoche.acelerar( aceleracion: 30);
    Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
}

no usages new *
@Test
public void test_al_decelerar_coche_su_velocidad_disminuye_Daniel_Jordan(){
    Coche nuevoCoche = new Coche();
    nuevoCoche.decelerar(30);
    Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
}

no usages Dani
@Test
public void test_al_acelerar_coche_su_velocidad_aumenta_Daniel_Jordan(){
    Coche nuevoCoche = new Coche();
    nuevoCoche.acelerar( aceleracion: 30);
    Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
}

no usages new *
@Test
public void test_al_decelerar_coche_su_velocidad_disminuye_Daniel_Jordan(){
    Coche nuevoCoche = new Coche();
    nuevoCoche.velocidad = 50;
    nuevoCoche.decelerar(20);
    Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
}
}
```

Ahora seleccionamos la palabra marcada en rojo del método decelerar y con la bombilla roja nos dirigimos a crear automáticamente el método en la clase coche.

```
9      @Test
10     public void test_al_acelerar_coche_su_velocidad_aumenta_Daniel_Jordan(){
11         Coche nuevoCoche = new Coche();
12         nuevoCoche.acelerar( 30);
13         Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
14     }
15     no usages new *
16     @Test
17     public void test_al_decelerar_coche_su_velocidad_disminuye_Daniel_Jordan(){
18         Coche nuevoCoche = new Coche();
19         nuevoCoche.velocidad = 50;
20         nuevoCoche.decelerar(20);
21         // Create method 'decelerar' in 'Coche'
22         // Rollback changes in current line
23         // Press Ctrl+Q to toggle preview
```

Cambiamos la variable que por defecto sale “i” por la variable deceleración.

```
TDD Practica Coche Daniel Jordan - Coche.java - Administrator
TestCoche_Daniel_Jordan.java x Coche.java x
6 usages Dani *
1 public class Coche {
2     5 usages
3     public int velocidad;
4     1 usage Dani
5     public void acelerar(int aceleracion) {
6         velocidad += aceleracion;
7     }
8     1 usage new *
9     public void decelerar(int deceleracion) {
10    }
11 }
```

Y modificamos el método para decir que, la velocidad, es la resta de la velocidad en ese momento menos la deceleración. Velocidad -= deceleración;


```
TDD Practica Coche Daniel Jordan - Coche.java - Administrator

TestCoche_Daniel_Jordan.java x Coche.java x

6 usages Dani
1 public class Coche {
2     6 usages
3     public int velocidad;
4
5     1 usage Dani
6     public void acelerar(int aceleracion) {
7         velocidad += aceleracion;
8     }
9
10    1 usage Dani
11    public void decelerar(int deceleracion) {
12        velocidad -= deceleracion;
13    }
14 }
```

Probamos el test con el botón del Play verde y comprobamos que es todo un éxito.

```
Run: TestCoche_Daniel_Jordan x

Tests passed: 3 of 3 tests - 25 ms
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...

Process finished with exit code 0

TestCoche_Daniel_Jordan 25 ms
  ✓ test_al_acelerar_coche_su_velocidad_aumenta_Daniel_Jordan() 25 ms
  ✓ test_al_crear_coche_su_velocidad_es_cero_Daniel_Jordan()
  ✓ test_al_decelerar_coche_su_velocidad_disminuye_Daniel_Jordan()
```

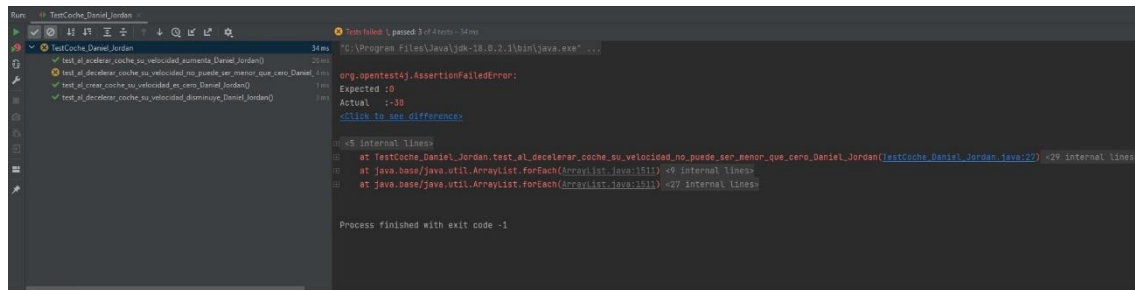
Ahora copiamos el test anterior y lo pegamos a continuación y lo renombramos tal y como hemos estado haciendo anteriormente.

```
15 @Test
16 public void test_al_decelerar_coche_su_velocidad_disminuye_Daniel_Jordan(){
17     Coche nuevoCoche = new Coche();
18     nuevoCoche.velocidad = 50;
19     nuevoCoche.decelerar( deceleracion: 20);
20     Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
21 }
22
23 @Test
24 public void test_al_decelerar_coche_su_velocidad_no_puede_ser_menor_que_cero_Daniel_Jordan(){
25     Coche nuevoCoche = new Coche();
26     nuevoCoche.velocidad = 50;
27     nuevoCoche.decelerar( deceleracion: 20);
28     Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
29 }
30 }
```

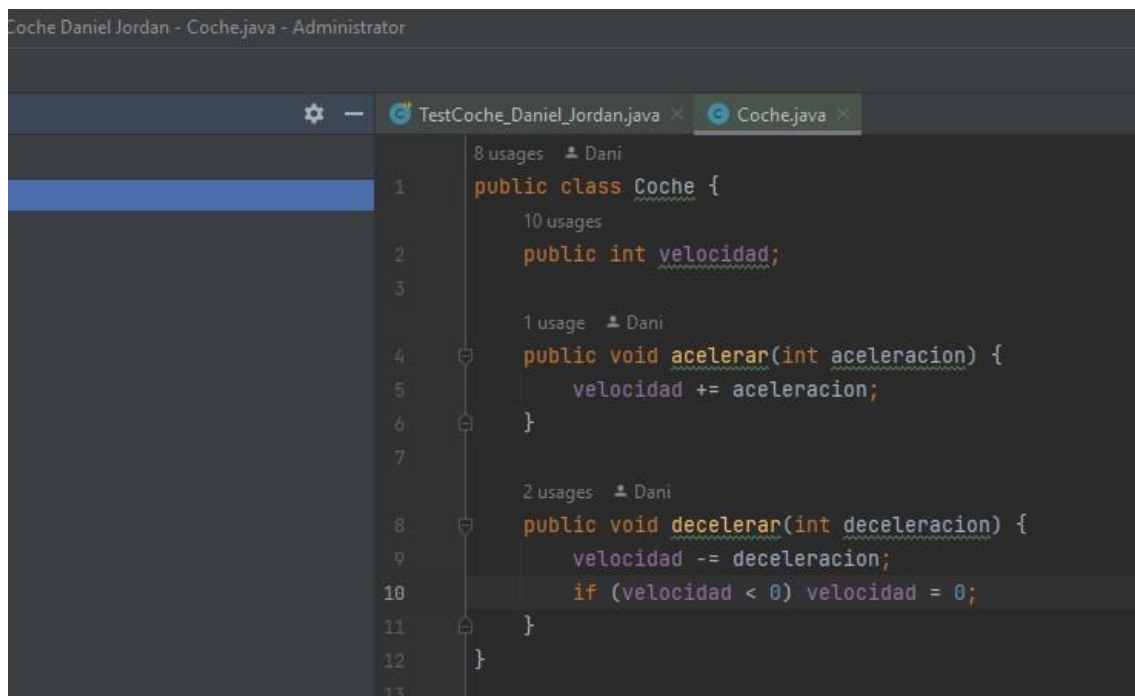
Le modificamos la deceleración y le pasamos un valor de 80, en este caso el vehículo al ir a 50km/h y decelerarlo en 80km/h vamos a recibir un valor negativo pero aún así vamos a indicarle al Assertion que esperamos una velocidad de 0.


```
15  @Test
16  public void test_al_decelerar_coche_su_velocidad_disminuye_Daniel_Jordan(){
17      Coche nuevoCoche = new Coche();
18      nuevoCoche.velocidad = 50;
19      nuevoCoche.decelerar( deceleracion: 20);
20      Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
21  }
22
23  @Test
24  public void test_al_decelerar_coche_su_velocidad_no_puede_ser_menor_que_cero_Daniel_Jordan(){
25      Coche nuevoCoche = new Coche();
26      nuevoCoche.velocidad = 50;
27      nuevoCoche.decelerar( deceleracion: 80);
28      Assertions.assertEquals( expected: 0,nuevoCoche.velocidad);
29  }
30  }
```

Recibimos un error al ejecutar el test ya que como comentaba antes la velocidad la va a dejar en -30 cuando se esperaba un resultado de 0. Para solucionar esto vamos a la clase Coche y en el método deceleracion incluimos un IF para indicar que la velocidad no puede ser inferior de 0 si se diese ese caso velocidad seria = 0.



```
Test failed, passed 3 of 4 tests - 34ms
D:\Program Files\Java\jdk-18.0.2\bin\java.exe ...
org.opentest4j.AssertionFailedError:
Expected :0
Actual   :-30
Click to see difference
at TestCoche.Daniel_Jordan.test_al_decelerar_coche_su_velocidad_no_puede_ser_menor_que_cero_Daniel_Jordan(TestCoche.Daniel_Jordan.java:27) <29 internal lines>
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511) <9 internal lines>
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511) <29 internal lines>
Process finished with exit code -1
```



```
8 usages  Dani
1  public class Coche {
2      10 usages
3      public int velocidad;
4
5      1 usage  Dani
6      public void acelerar(int aceleracion) {
7          velocidad += aceleracion;
8      }
9
10     2 usages  Dani
11     public void decelerar(int deceleracion) {
12         velocidad -= deceleracion;
13         if (velocidad < 0) velocidad = 0;
14     }
15 }
```

Y ahora si ejecutamos el ultimo test comprobamos que es tambien todo un éxito ya que ahora la velocidad se corresponde con nuestras expectativas ya que se queda a "0".

