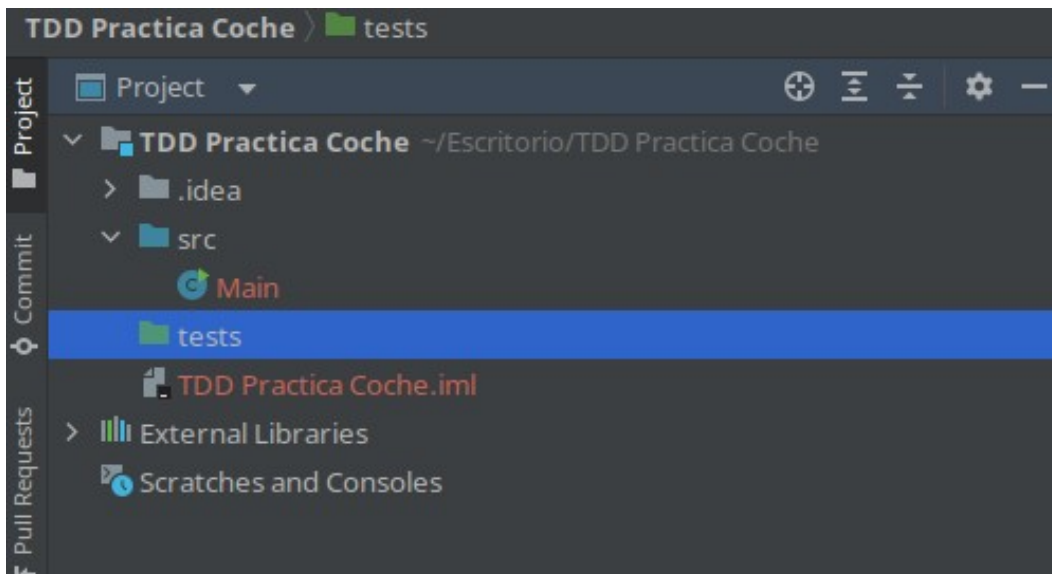
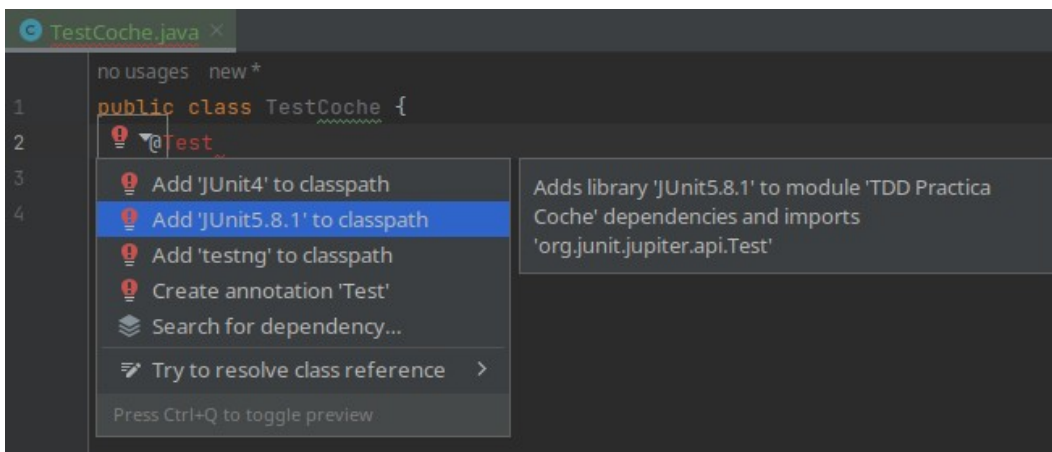


Memoria Mi primer TDD

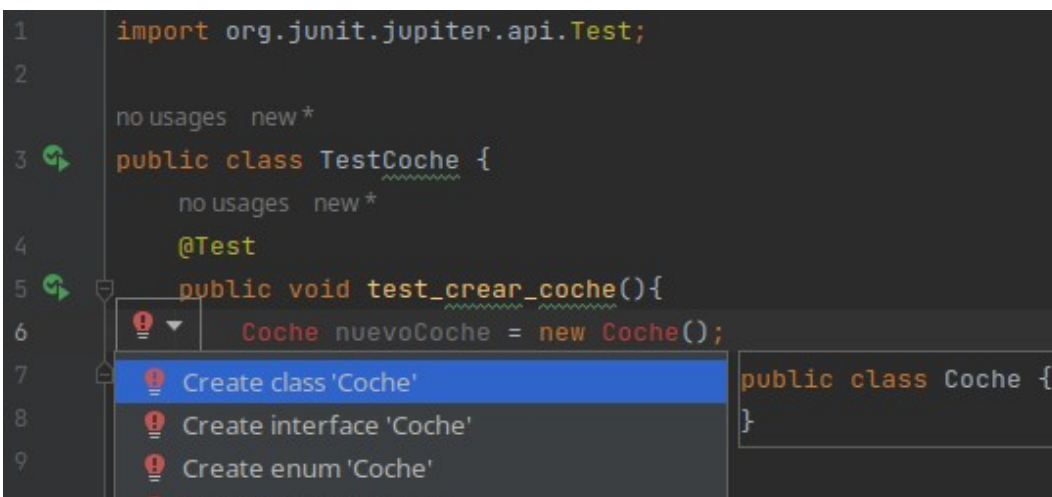
Creamos el proyecto nuevo TDD Practica Coche, con el directorio tests, y éste lo marcamos como un directorio de texto:



Creamos la java class TestCoche e indicamos que es @test, y tenemos que añadir J Unit 5:



Instanciamos la clase coche aunque aún no exista en un método y la creamos:



Le damos a Run y vemos que hemos pasado el primer test:

```
import org.junit.jupiter.api.Test;

public class TestCoche {

    @Test
    public void test_crear_coche(){
        Coche nuevoCoche = new Coche();
    }
}
```

TestCoche x

Tests passed: 1 of 1 test - 17 ms

TestCoche 17 ms /home/ALU1W/.jdk/openjdk-19.0.2/bin/java ...

test_crear_coche() 17 ms

Process finished with exit code 0

Mejoramos nuestro método usando `Assertions.assertEquals` y creamos la variable `velocidad` en la class `Coche`, ya que la requiere dentro de `assert`:

```
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.Test;

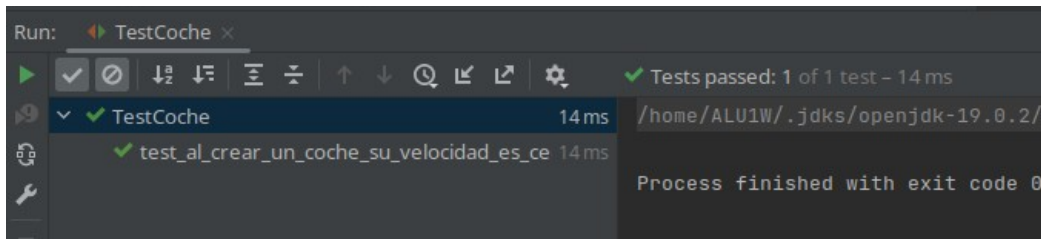
public class TestCoche {

    @Test
    public void test_al_crear_un_coche_su_velocidad_es_cero(){
        Coche nuevoCoche = new Coche();
        Assertions.assertEquals( expected: 0, nuevoCoche.velocidad);
    }
}
```

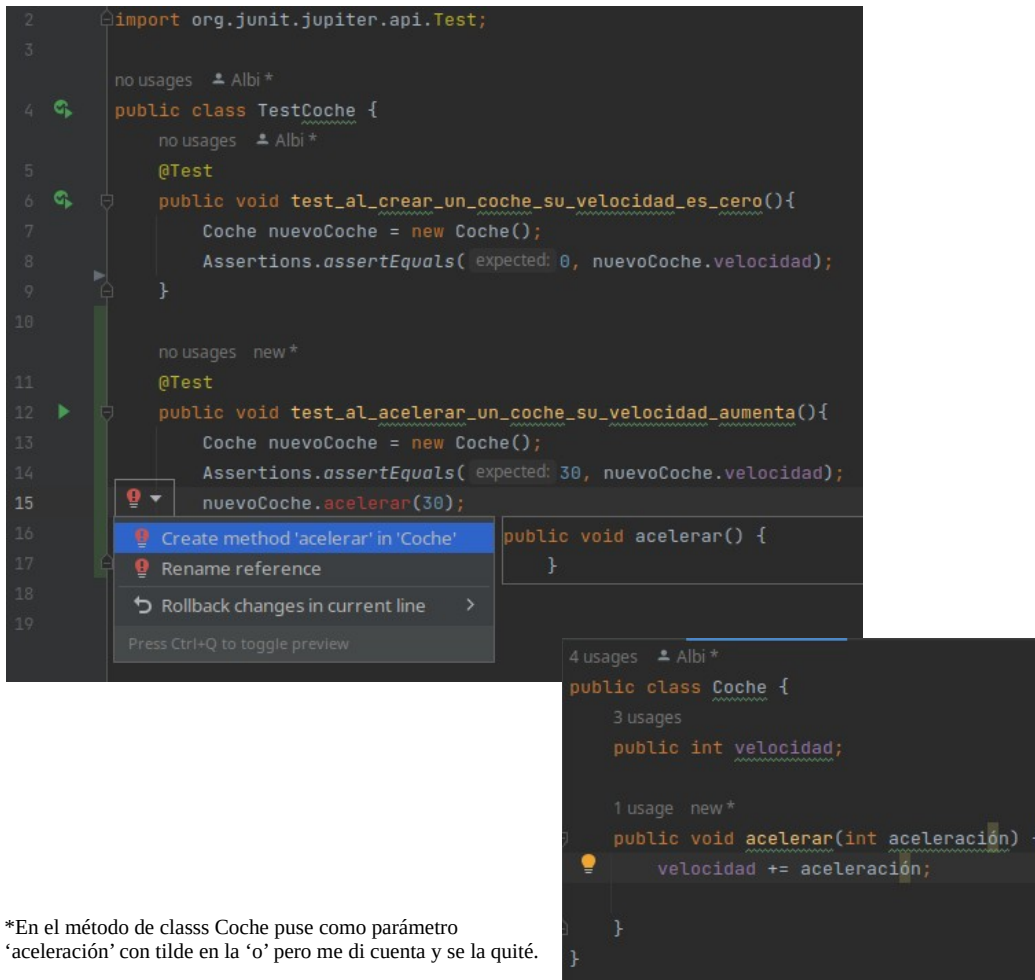
```
public class Coche {

    public int velocidad;
}
```

De nuevo pasamos el test:

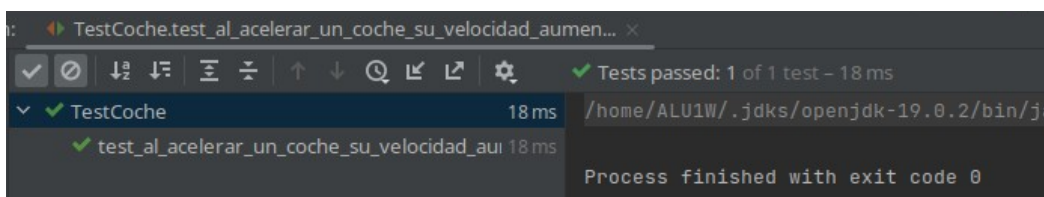


Creamos otro método que aumenta la velocidad cuando acelere el coche y en la class Coche creamos el método acelerar:



*En el método de class Coche puse como parámetro 'aceleración' con tilde en la 'o' pero me di cuenta y se la quité.

Volvemos a pasar el test:



Creamos un método que disminuye la velocidad justo debajo del que la aumenta y en class Coche creamos el método decelerar:

```
@Test
public void test_al_acelerar_un_coche_su_velocidad_aumenta(){
    Coche nuevoCoche = new Coche();
    nuevoCoche.acelerar( aceleracion: 30);
    Assertions.assertEquals( expected: 30, nuevoCoche.velocidad);
}

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

}

public class Coche {
    6 usages
    public int velocidad;

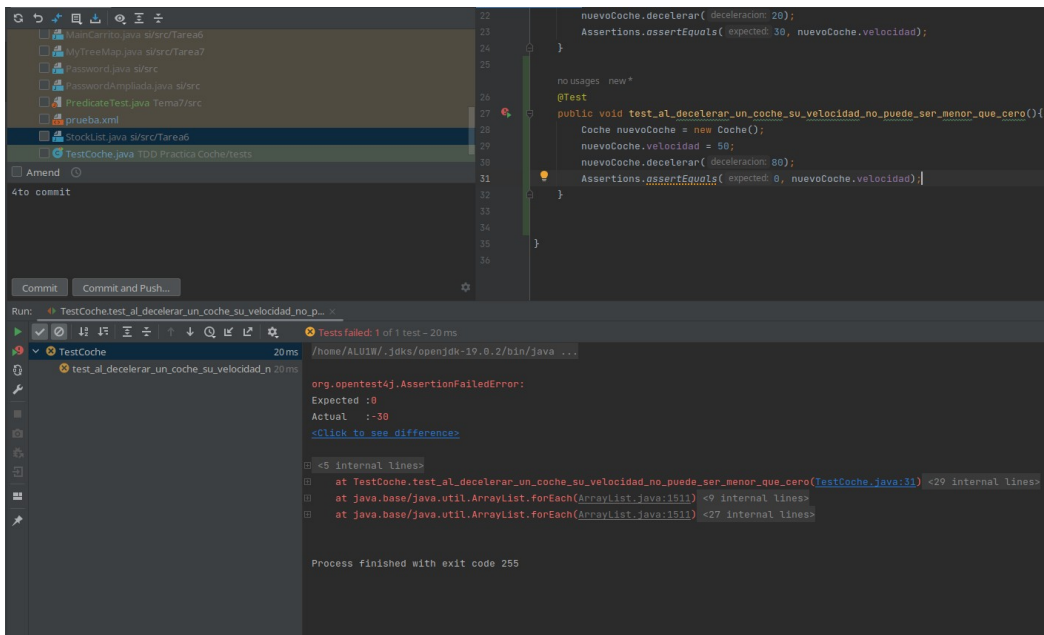
    1 usage new *
    public void acelerar(int aceleracion) {
        velocidad += aceleracion;
    }

    1 usage new *
    public void decelerar(int deceleracion) {
        velocidad -= deceleracion;
    }
}
```

Volvemos a pasar el test:

```
Run: TestCoche.test_al_decelerar_un_coche_su_velocidad_dismi... x
[Icons] [TestCoche] 14 ms /home/ALU1W/.jdk/openjdk-19.0.2/bin/java
[Icons] [test_al_decelerar_un_coche_su_velocidad_d 14 ms]
Process finished with exit code 0
```

Creamos otro método para que no haya velocidades negativas, osea por debajo del 0, y colocamos un valor para las variables, pero da error cuando lo testeamos:



```
22 nuevoCoche.decelerar( deceleracion: 20);
23 Assertions.assertEquals( expected: 30, nuevoCoche.velocidad);
24
25
26
27 no usages new *
28 @test
29 public void test_al_decelerar_un_coche_su_velocidad_no_puede_ser_menor_que_cero(){
30     Coche nuevoCoche = new Coche();
31     nuevoCoche.velocidad = 50;
32     nuevoCoche.decelerar( deceleracion: 80);
33     Assertions.assertEquals( expected: 0, nuevoCoche.velocidad);
34 }
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Run: TestCoche.test_al_decelerar_un_coche_su_velocidad_no_puede_ser_menor_que_cero

Tests failed: 1 of 1 test - 20 ms

TestCoche 20 ms /home/ALU1W/.jdk/openjdk-19.0.2/bin/java ...

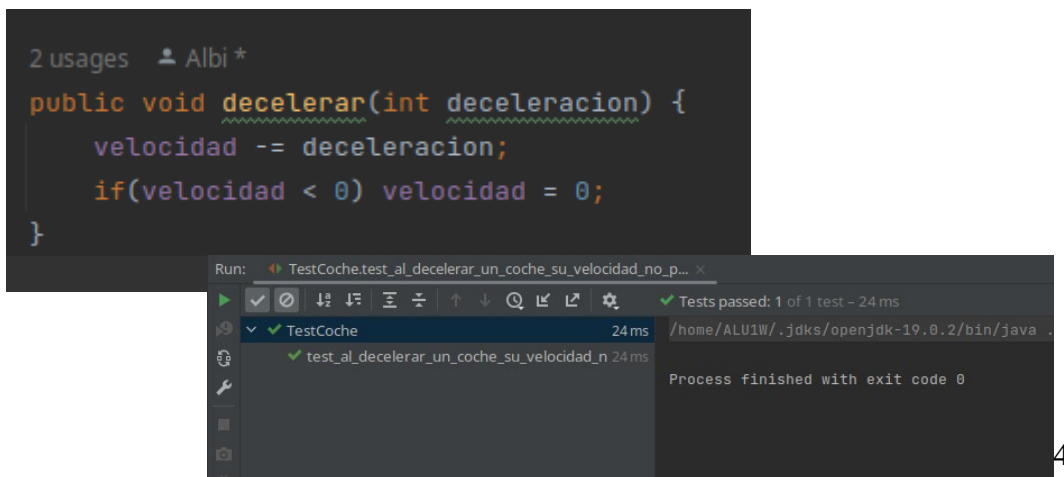
test_al_decelerar_un_coche_su_velocidad_n 20 ms

org.opentest4j.AssertionFailedError:
Expected :0
Actual :-30
[Click to see difference](#)

<5 internal lines>
at TestCoche.test_al_decelerar_un_coche_su_velocidad_no_puede_ser_menor_que_cero(TestCoche.java:31) <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) <27 internal lines>

Process finished with exit code 255

Cambiamos un par de cosas en el método decelerar de la class Coche, como asignar la velocidad en cero cuando se vuelve negativa para que no dé error y volvemos a testear:



```
2 usages Albi *
public void decelerar(int deceleracion) {
    velocidad -= deceleracion;
    if(velocidad < 0) velocidad = 0;
}

Run: TestCoche.test_al_decelerar_un_coche_su_velocidad_no_puede_ser_menor_que_cero
Tests passed: 1 of 1 test - 24 ms
TestCoche 24 ms /home/ALU1W/.jdk/openjdk-19.0.2/bin/java ...
test_al_decelerar_un_coche_su_velocidad_n 24 ms
Process finished with exit code 0
```

todo correcto (: