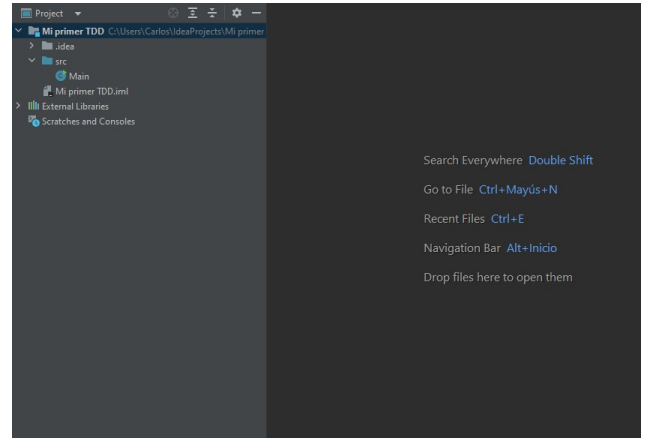
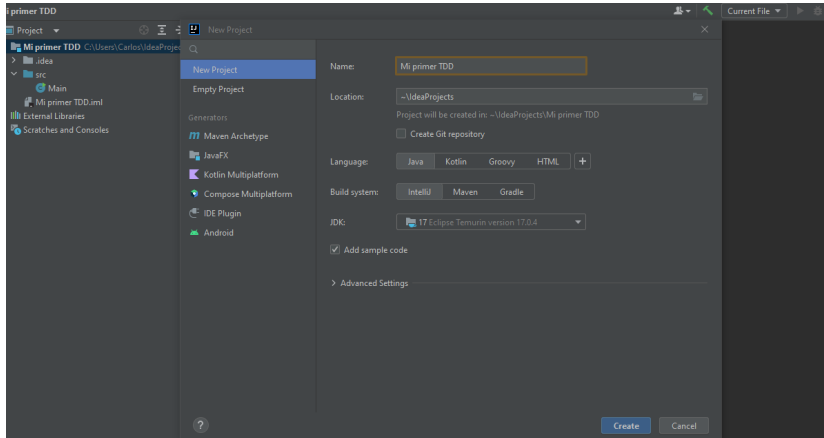
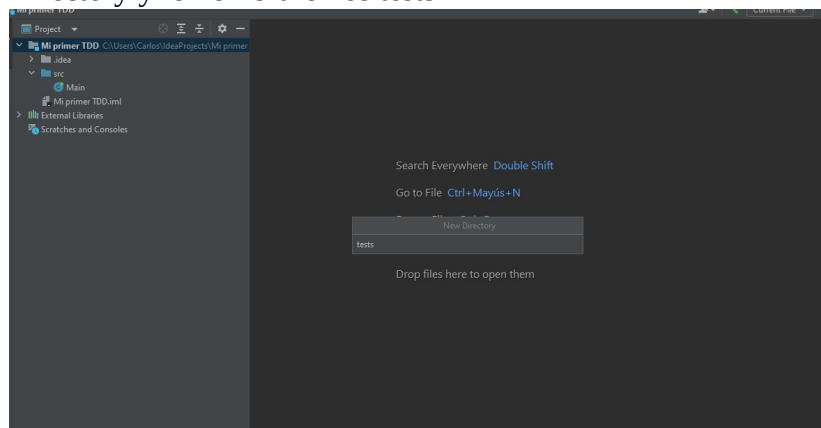


# Mi primer TDD

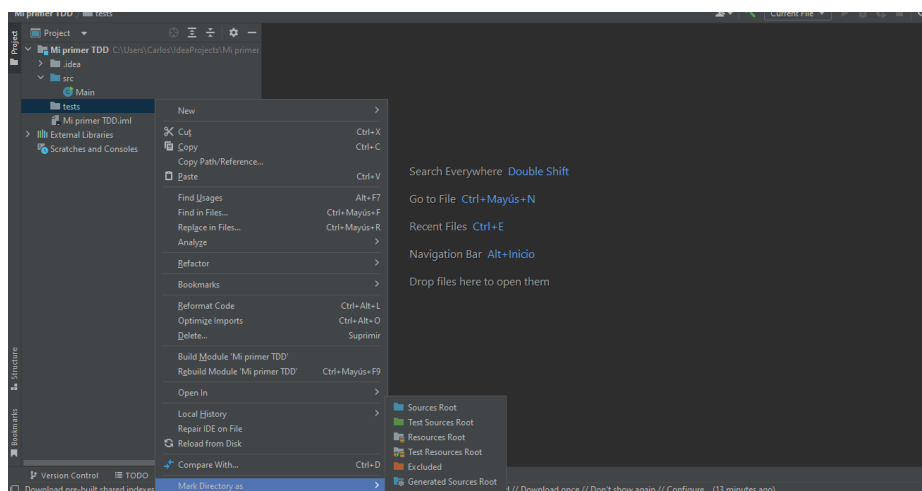
Creamos un nuevo proyecto en IntelliJ y lo llamamos “Mi primer TDD”, en mi caso no lo crearé ya que ya lo tengo hecho



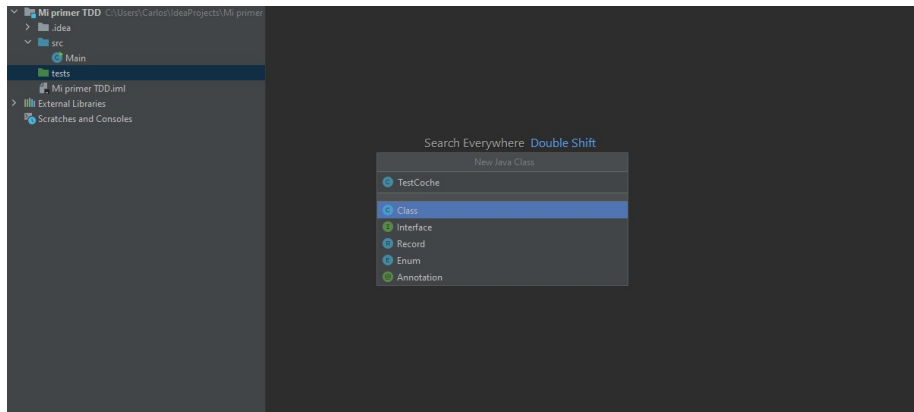
Seguido de esto crearemos un nuevo directorio, haremos click derecho encima de nuestro proyecto → new → Directory y lo nombraremos tests



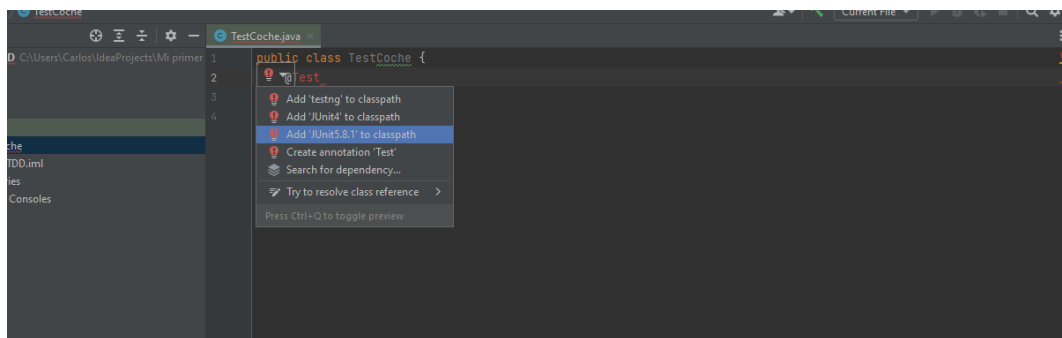
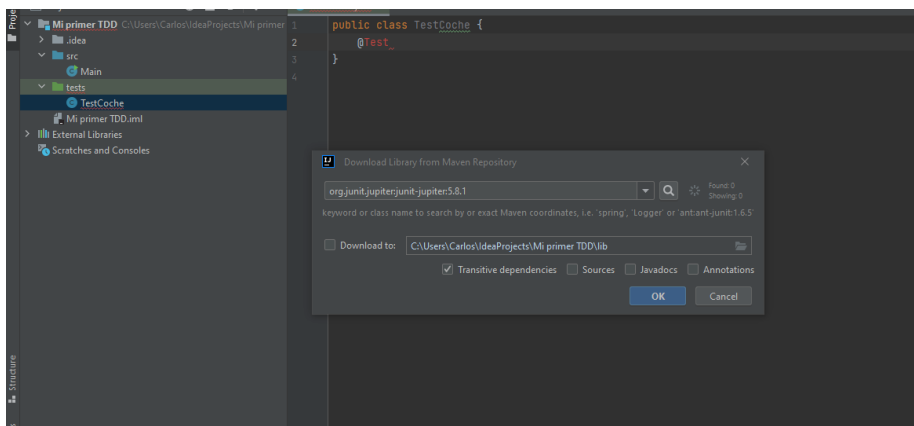
Seguido de esto marcaremos el directorio como un directorio de test, click drecho encima de este → mark directory as → test sources root



Creamos una nueva clase dentro de este llamada “TestCoche”



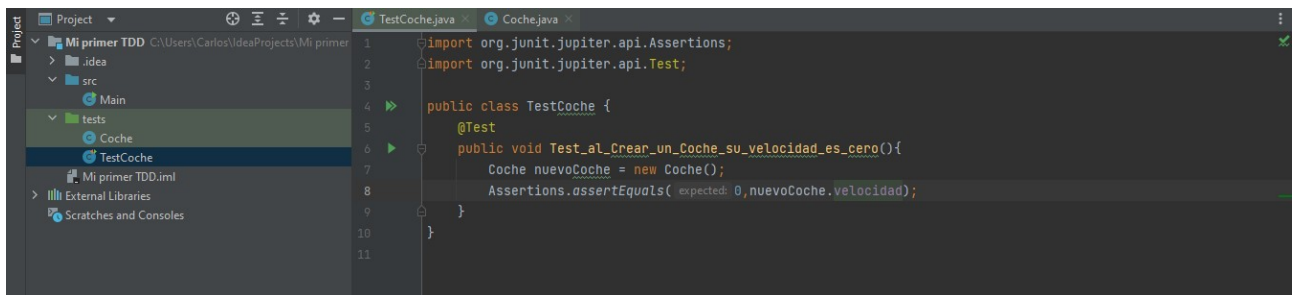
Lo primero que añadimos dentro de la clase será el test, seguido de esto no lo reconocerá y haremos que añada Junit 5



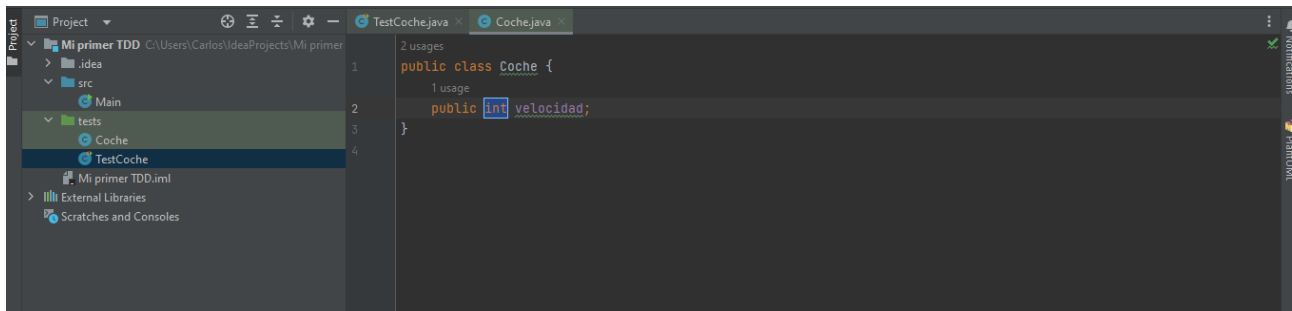
Creamos nuestro primer test, para ello crearemos un metodo y una clase llamada coche



Este test era muy simple así que lo mejoraremos, haremos que al crear su coche su velocidad es cero, y en la clase coche ya creada agregaremos el atributo int velocidad

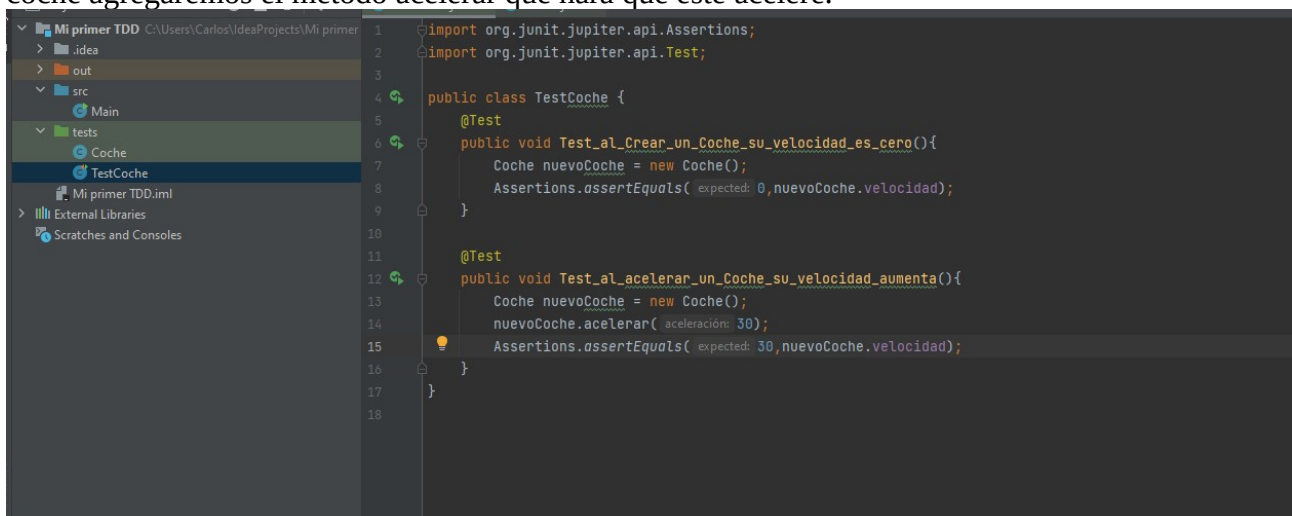


```
1 import org.junit.jupiter.api.Assertions;
2 import org.junit.jupiter.api.Test;
3
4 public class TestCoche {
5     @Test
6     public void Test_al_Crear_un_Coche_su_velocidad_es_cero(){
7         Coche nuevoCoche = new Coche();
8         Assertions.assertEquals( expected: 0,nuevoCoche.velocidad);
9     }
10
11 }
```

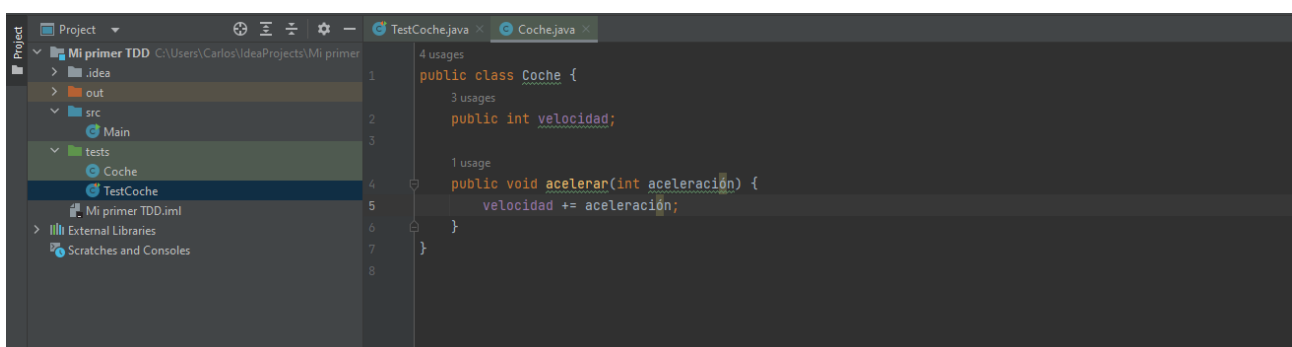


```
1 public class Coche {
2     public int velocidad;
3 }
4
```

Creamos un nuevo test que hace que al acelerar un coche aumente su velocidad y dentro de la clase coche agregaremos el método acelerar que hará que este acelere.

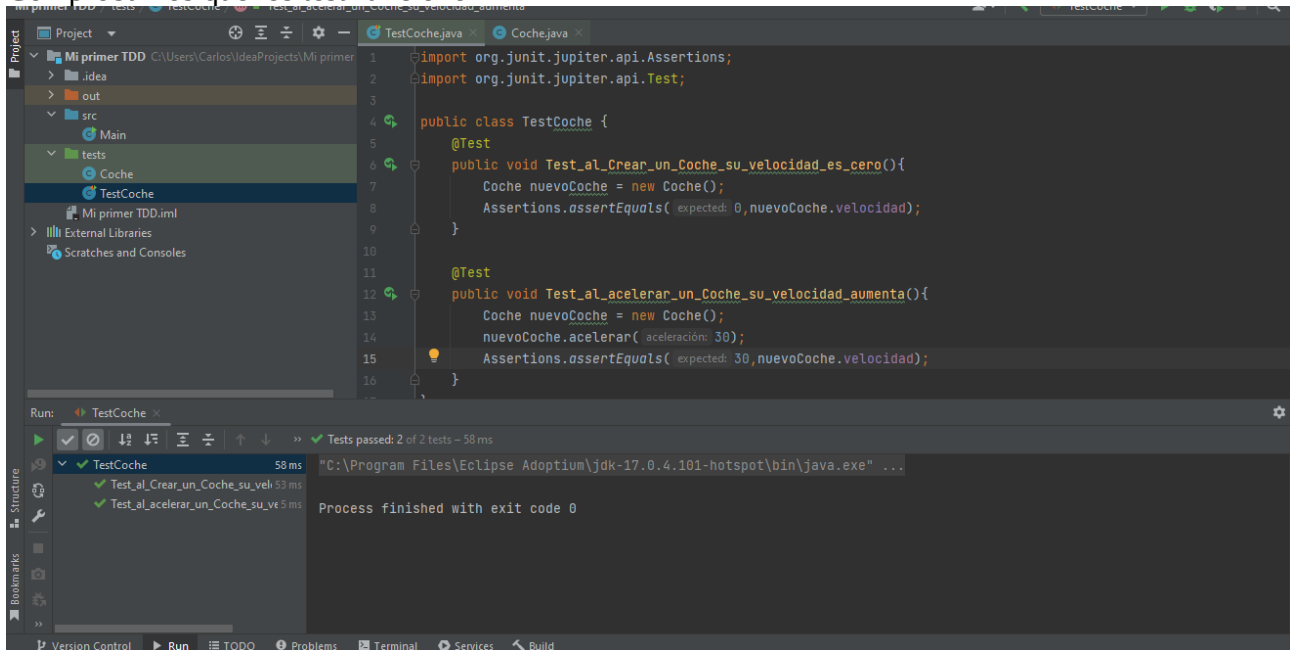


```
1 import org.junit.jupiter.api.Assertions;
2 import org.junit.jupiter.api.Test;
3
4 public class TestCoche {
5     @Test
6     public void Test_al_Crear_un_Coche_su_velocidad_es_cero(){
7         Coche nuevoCoche = new Coche();
8         Assertions.assertEquals( expected: 0,nuevoCoche.velocidad);
9     }
10
11     @Test
12     public void Test_al_acelerar_un_Coche_su_velocidad_aumenta(){
13         Coche nuevoCoche = new Coche();
14         nuevoCoche.acelerar( aceleración: 30);
15         Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
16     }
17
18 }
```

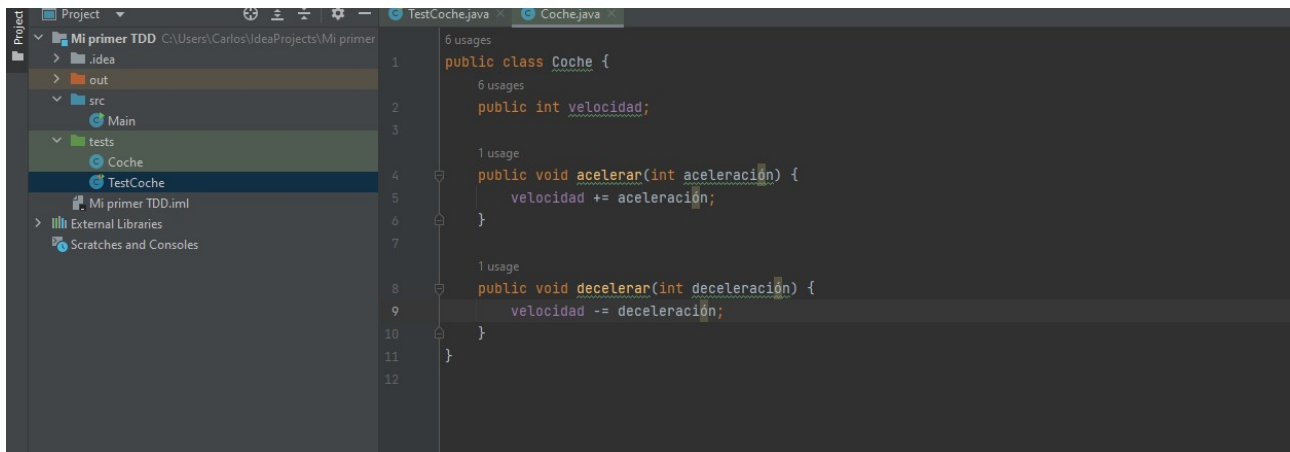
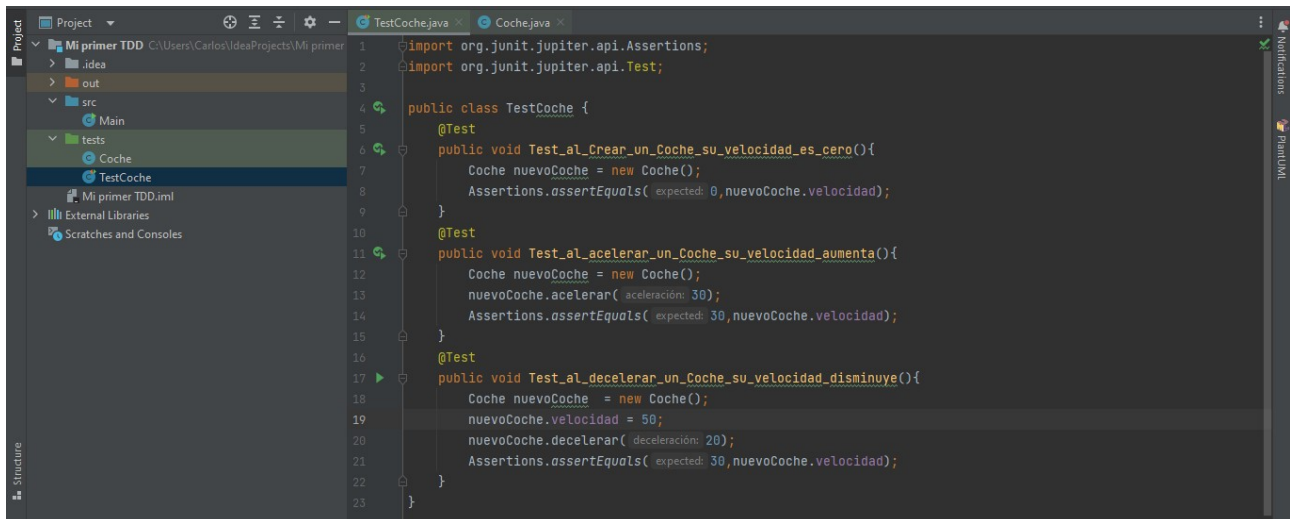


```
1 public class Coche {
2     public int velocidad;
3
4     public void acelerar(int aceleración) {
5         velocidad += aceleración;
6     }
7 }
8
```

## Comprobamos que los test funcionen



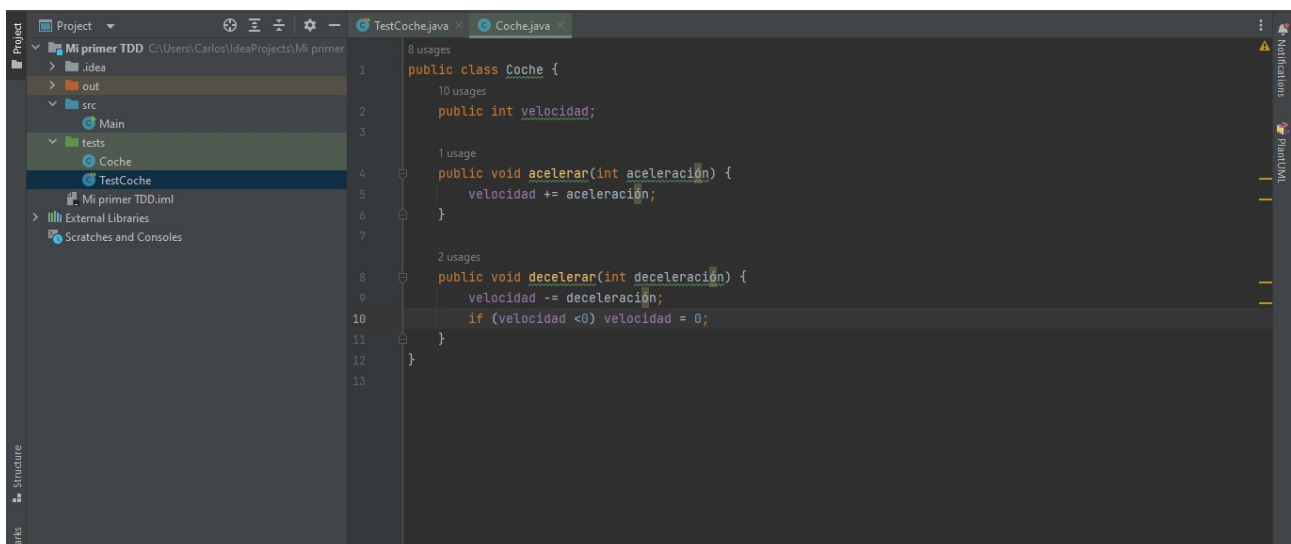
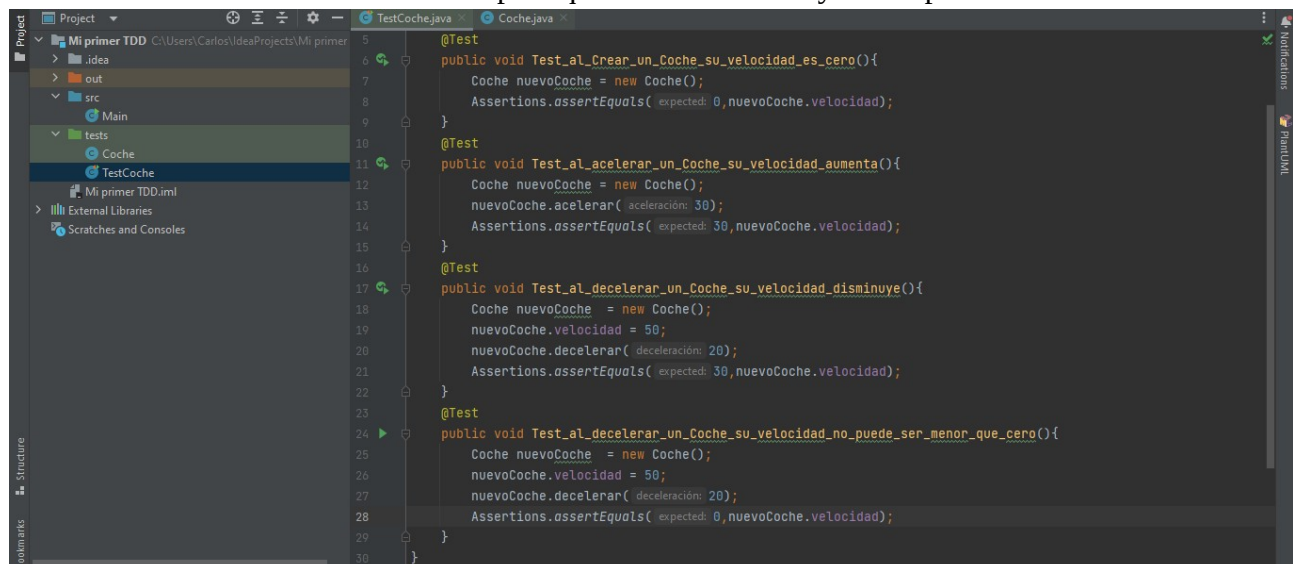
Creamos un nuevo test que al decelerar disminuirá la velocidad y dentro de la clase coche agregaremos el metodo decelerar que hará que este disminuya su velocidad



## Comprobamos que el test funcione



Hacemos un ultimo test que comprobará que al decelerar no salga de cero, para ello modificaremos el atributo decelerar de la clase coche para que cuando disminuya no supere el cero



## Comprobamos que los test funcionen correctamente

GENERALITAT VALENCIANA

Aules

1CFSK + W Entorns de ...

Conselleria d'Educació, Cultura i Esport

Buscar Val / Cas / Eng / Fra Salir

Área personal Eventos Este curso CARLOS ARROYO GALAN

Ocultar bloques Vista estándar

Área personal > Mis cursos > 1CFSK + W Entorns de desenrotllament 22/23 1664293976 > UD 9 > Mi primer TDD

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help Mi primer TDD - TestCoche.java

Project Mi primer TDD C:\Users\Carlos\IdeaProjects\Mi primer TDD

src Main tests

TestCoche.java

```
@Test
public void Test_al_decelerar_un_Coche_su_velocidad_no_puede_ser_menor_que_cero(){
    Coche nuevoCoche = new Coche();
    nuevoCoche.velocidad = 50;
    nuevoCoche.decelerar( deceleración: 80);
    Assertions.assertFauals( expected: 0 nuevoCoche.velocidad);
}
```

Run: TestCoche

Tests passed: 4 of 4 tests - 60 ms

TestCoche 60 ms

- Test\_al\_Crear\_un\_Coche\_su\_ve 54 ms
- Test\_al\_acelerar\_un\_Coche\_su\_v 2 ms
- Test\_al\_decelerar\_un\_Coche\_su\_2 ms
- Test\_al\_decelerar\_un\_Coche-su\_2 ms

Process finished with exit code 0

