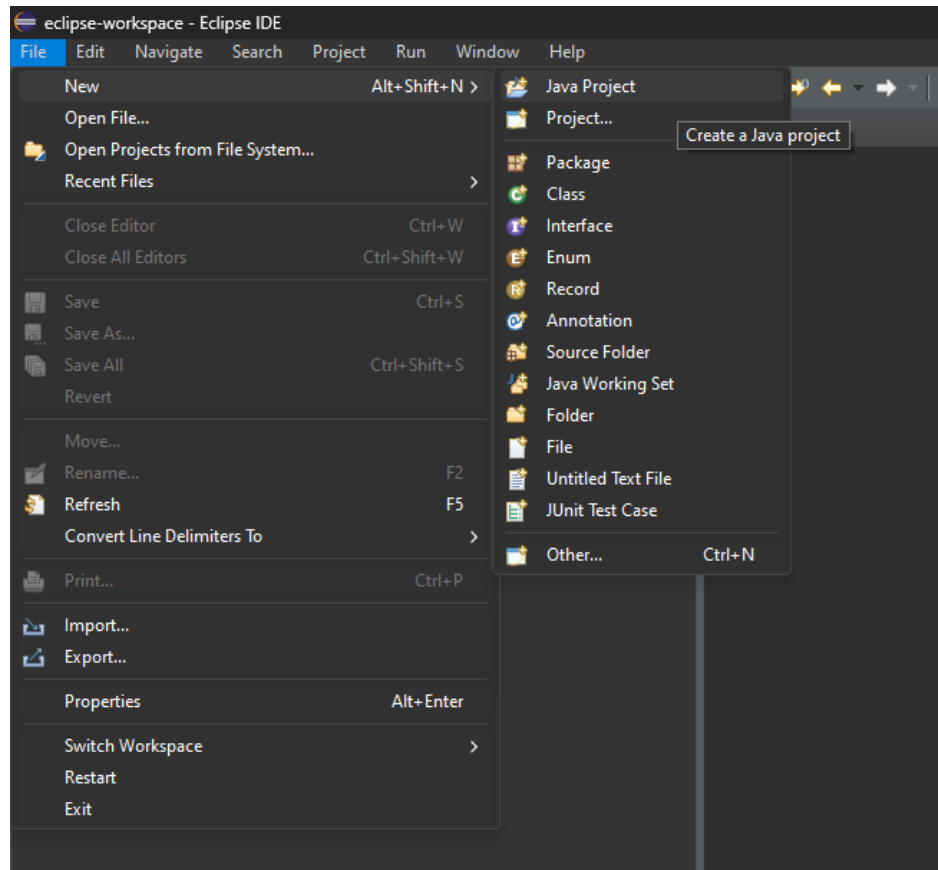


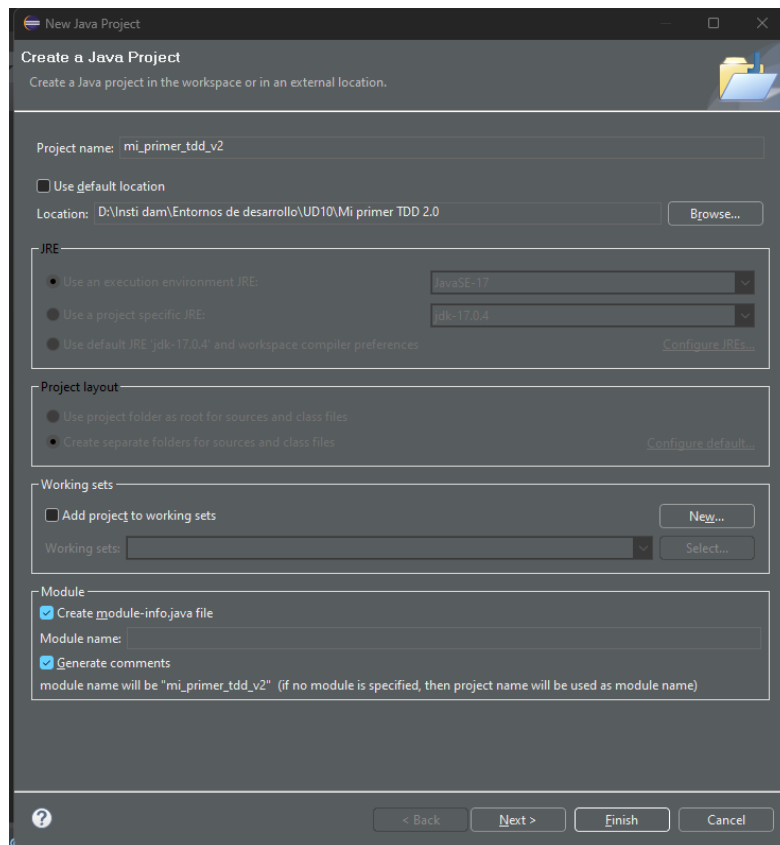
## Memoria de Mi primer TDD V2.0

Alejandro Vicente Carpena

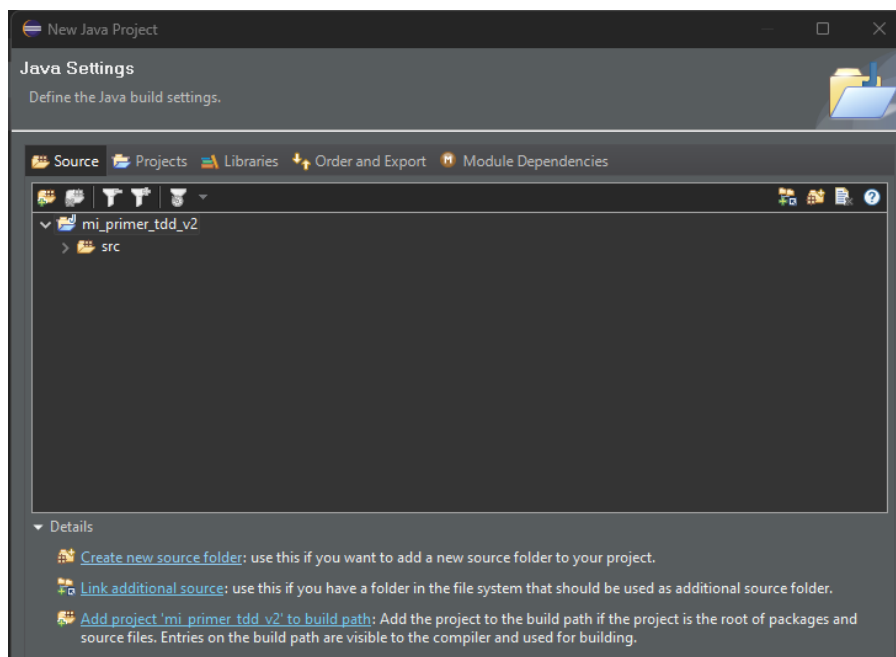
Creamos un proyecto de java en eclipse, dandole a file→new→java project



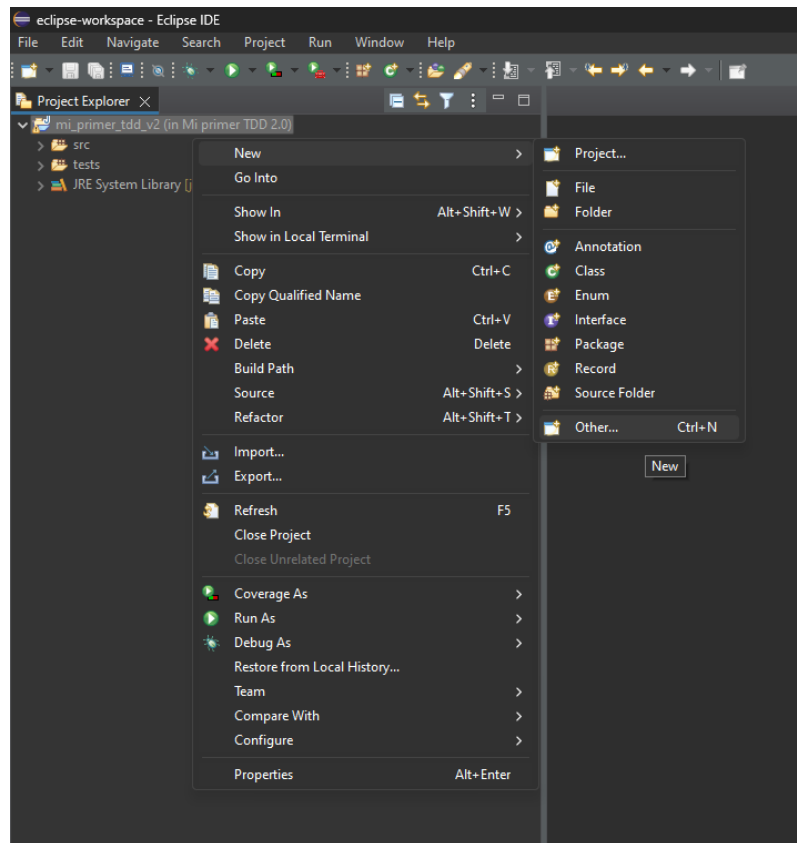
Seleccionamos la ruta en la que lo queremos crear, y le ponemos el nombre que queramos



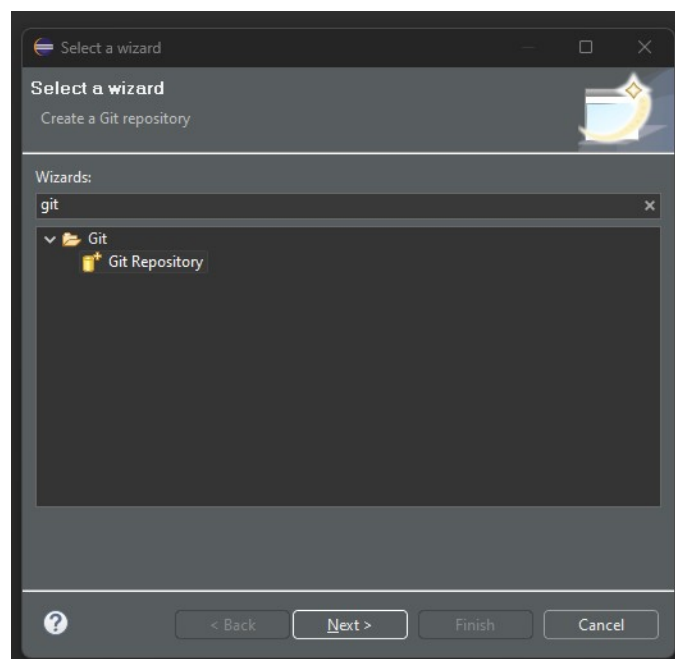
En este ajuste, podremos crear una carpeta dentro del proyecto, por lo que crearemos la carpeta tests aquí



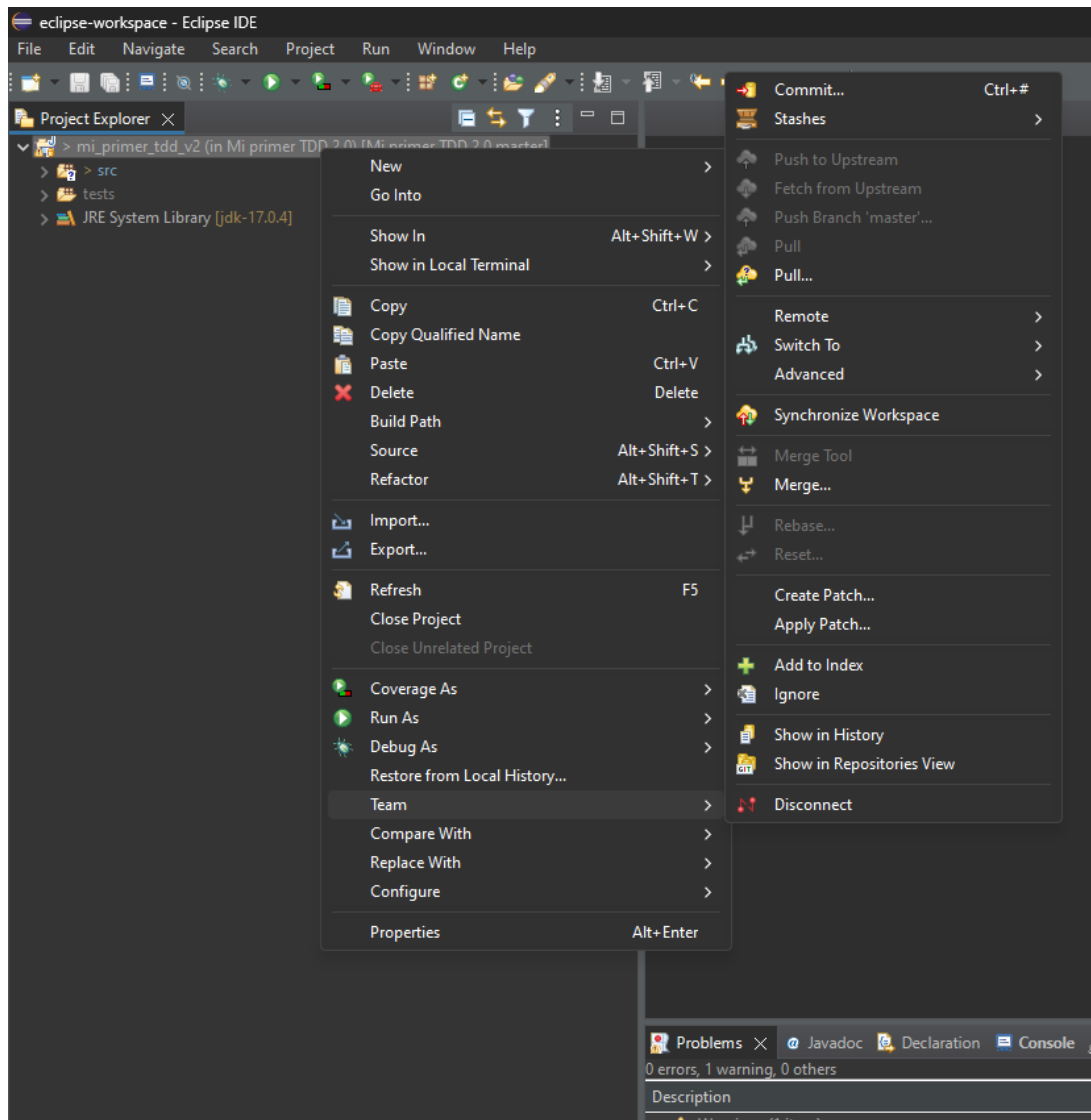
Para añadir a git el proyecto, lo podemos hacer dandole clic derecho al proyecto y en new→other



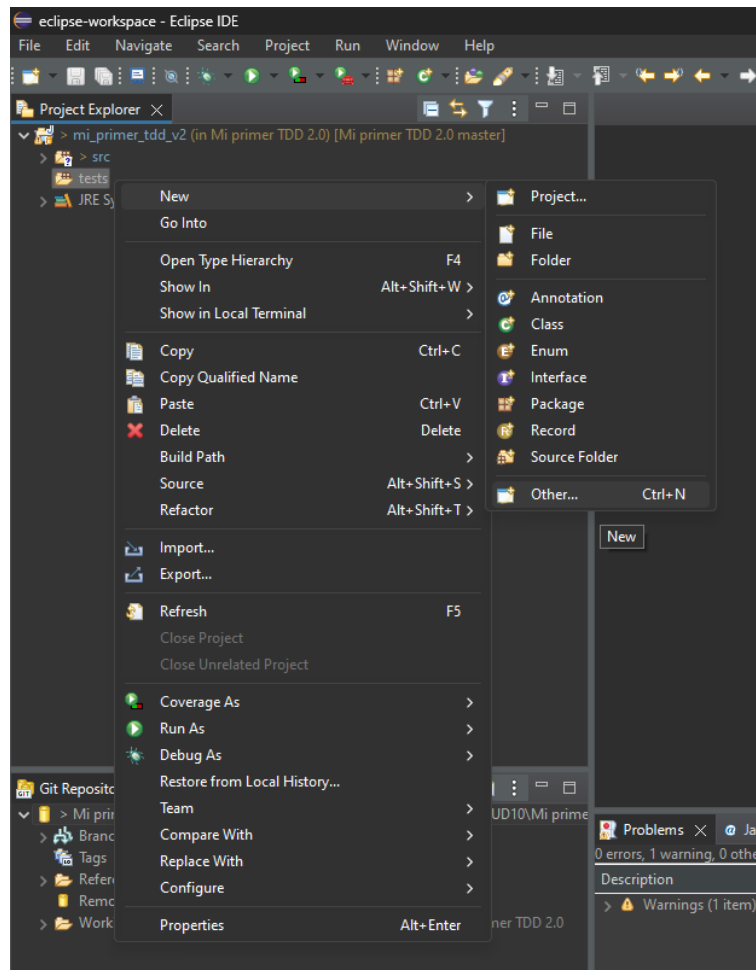
Y buscamos git, y nos saldra esta opcion



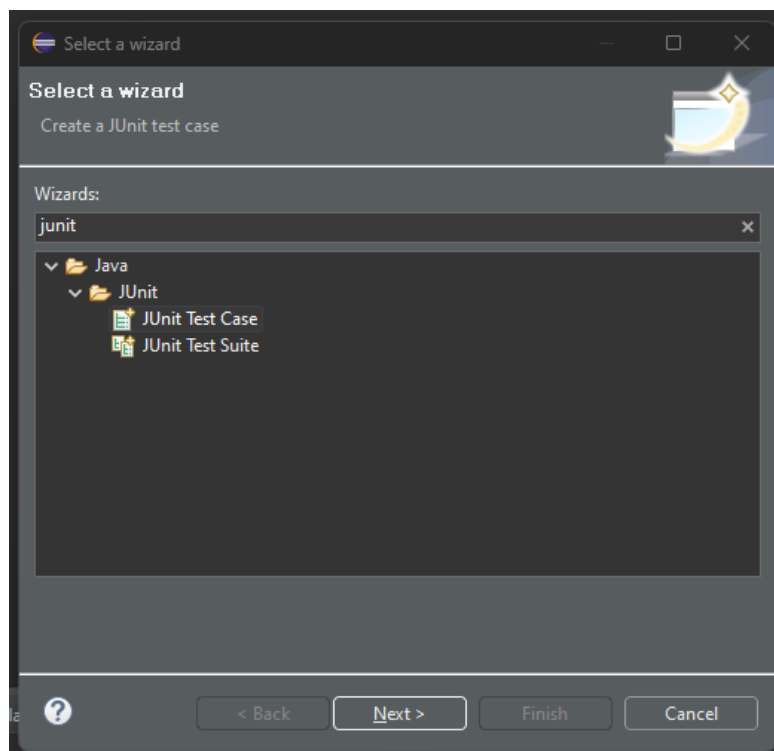
Luego dandole clic derecho al proyecto en el apartado Team, podemos realizar las opciones de git



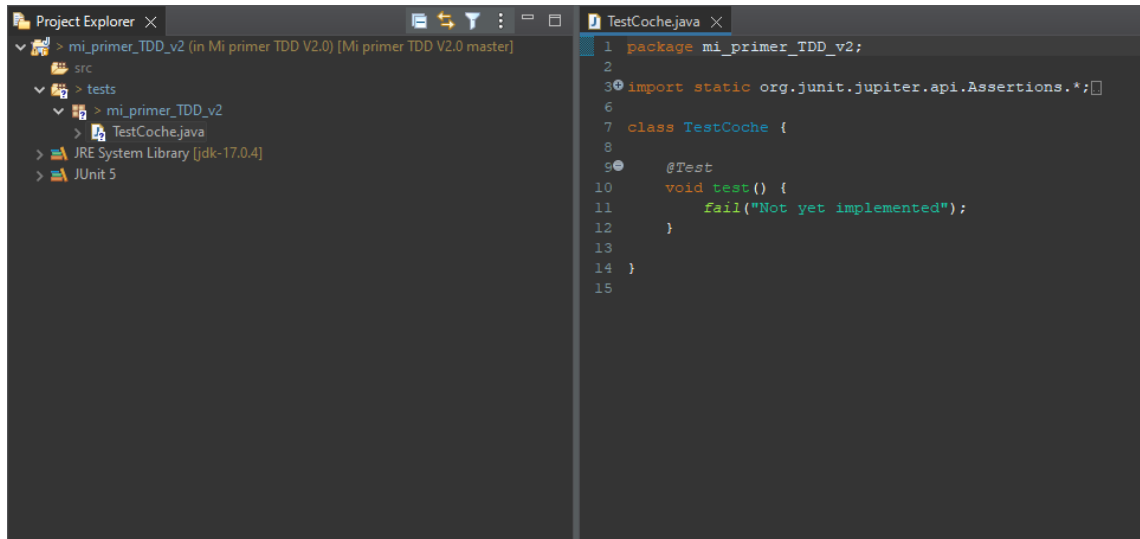
Para crear el primer test, le damos clic derecho a la carpeta tests, y en new→other



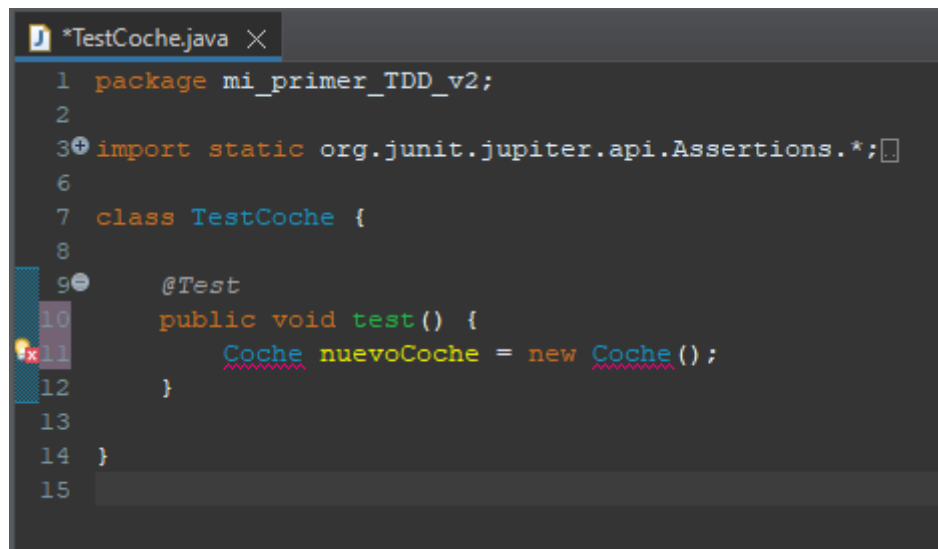
Buscamos Junit, y nos saldrán estas dos opciones, le damos a Test Case, y creamos el test



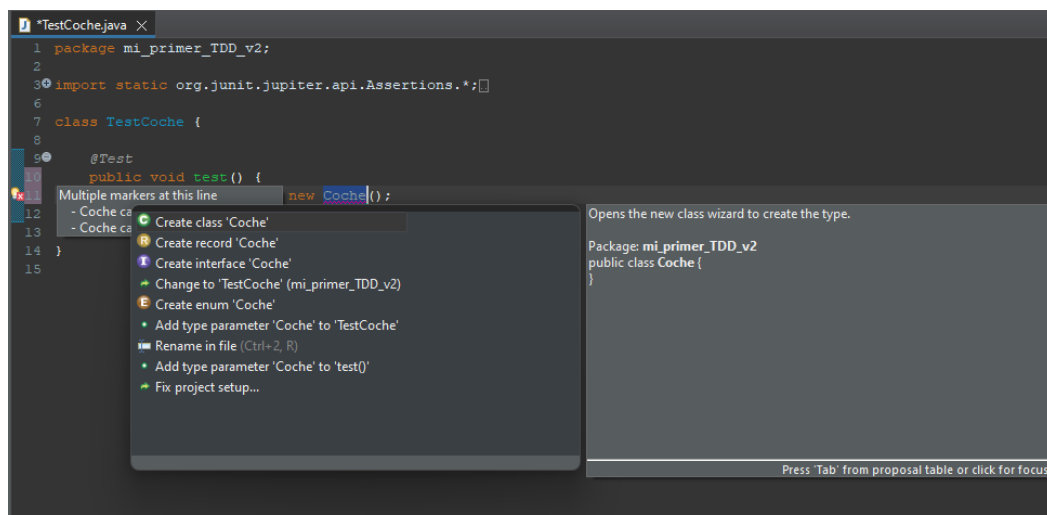
Se nos creara esta clase



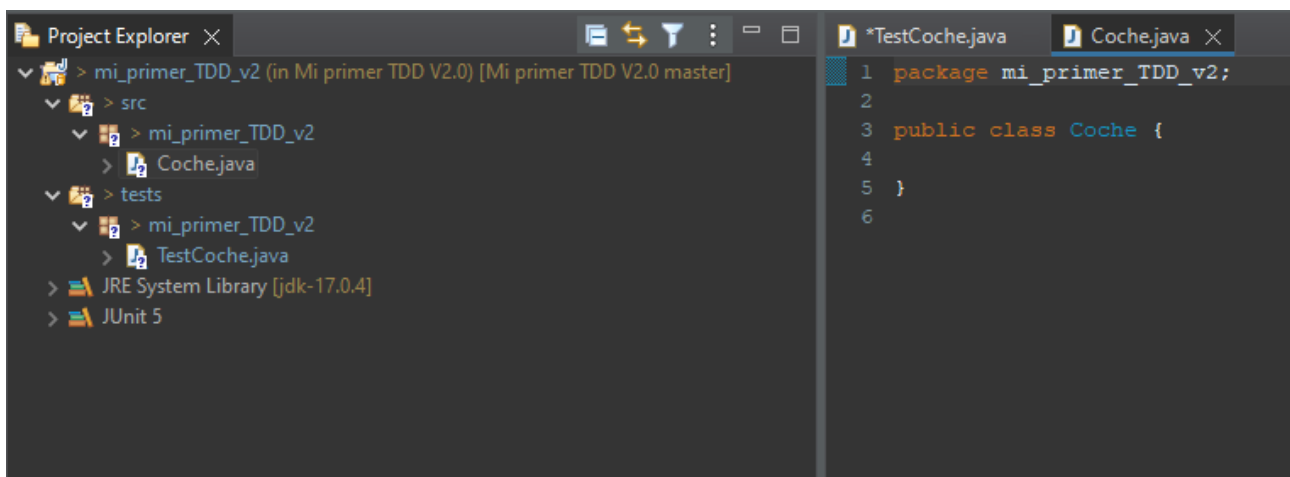
Lo cambiamos para añadir el test que queremos y vemos que no tenemos la clase coche creada



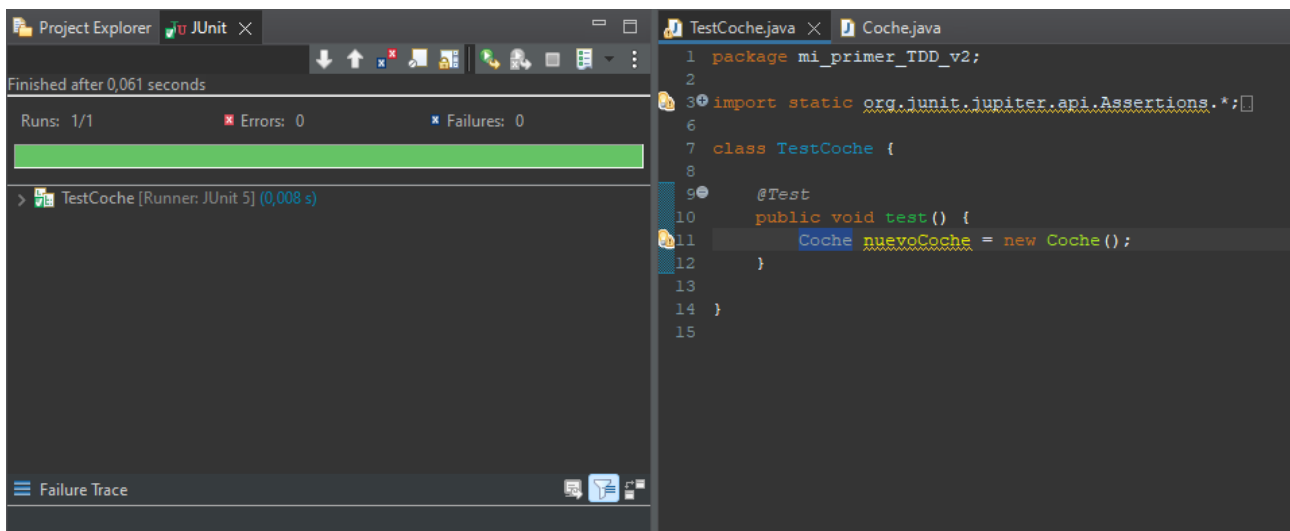
Si le damos a la bombilla de la izquierda, se nos abren estas opciones, una de ellas nos permite crear la clase Coche, y a la derecha tiene una previsualización de lo que se va a crear



Así se queda el proyecto al darle a la primera opción



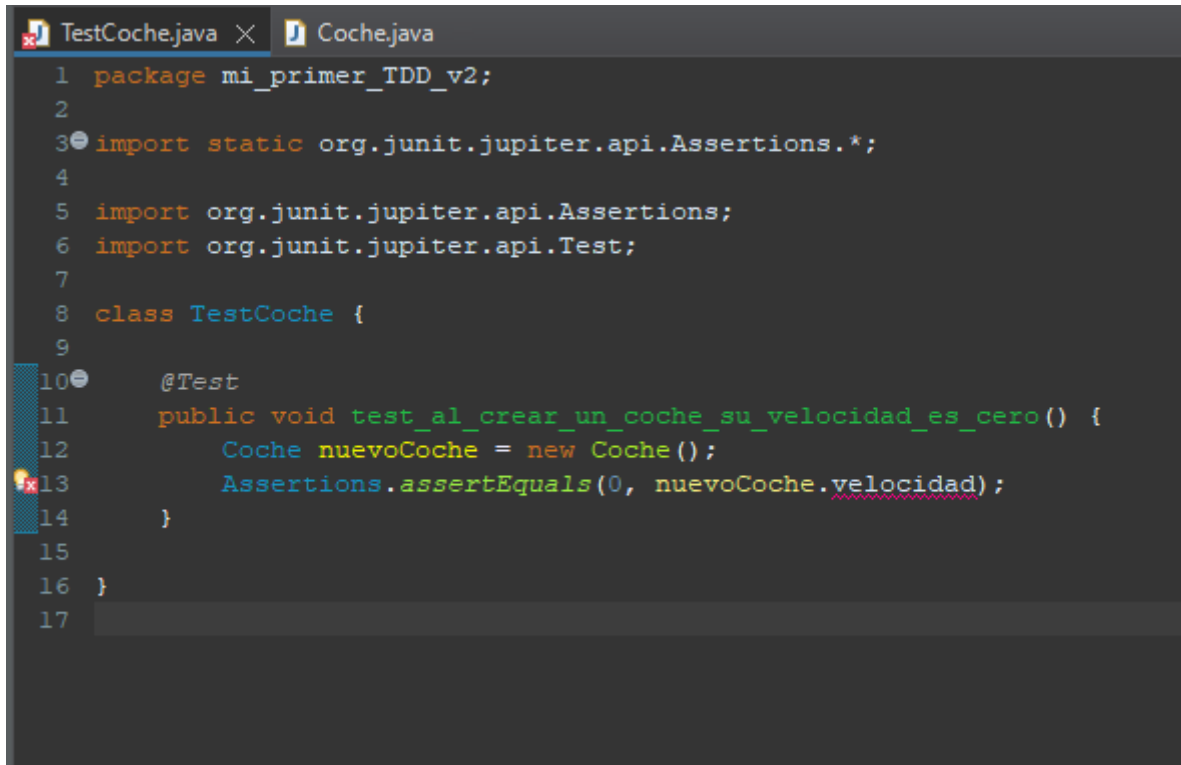
Corremos el test y lo pasa corretamente



Cambiamos el test para añadirle un poco de complejidad.

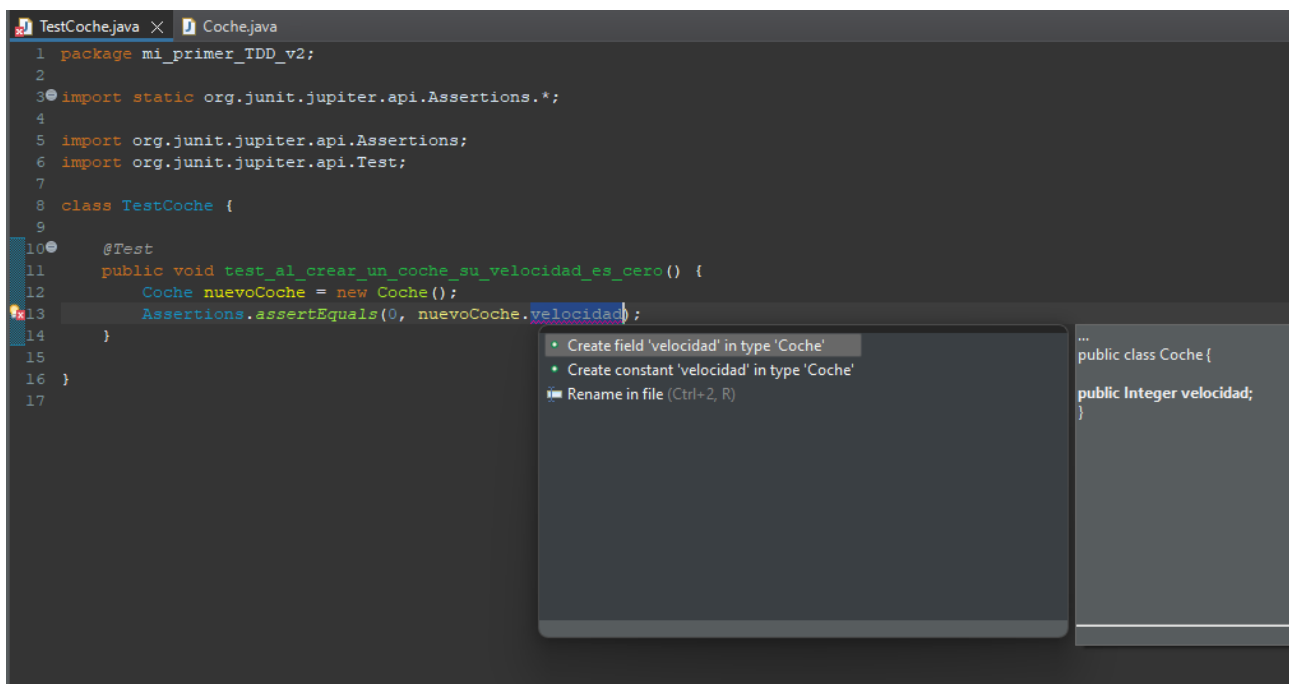
Con `Assertions.assertEquals(0, nuevoCoche.velocidad);` estamos comprobando que el parametro velocidad del objeto nuevoCoche sea 0

Velocidad sale en rojo porque no tenemos ese parametro creado en la clase Coche



```
1 package mi_primer_TDD_v2;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 import org.junit.jupiter.api.Assertions;
6 import org.junit.jupiter.api.Test;
7
8 class TestCoche {
9
10     @Test
11     public void test_al_crear_un_coche_su_velocidad_es_cero() {
12         Coche nuevoCoche = new Coche();
13         Assertions.assertEquals(0, nuevoCoche.velocidad);
14     }
15
16 }
17
```

Por lo que si le damos a la bombilla nos saldra una opcion para crearla. En este caso nos crea una variable Integer, pero no nos interesa, ya que por defecto la variable Integer es null, por lo que la cambiare por un int, que es 0 por defecto



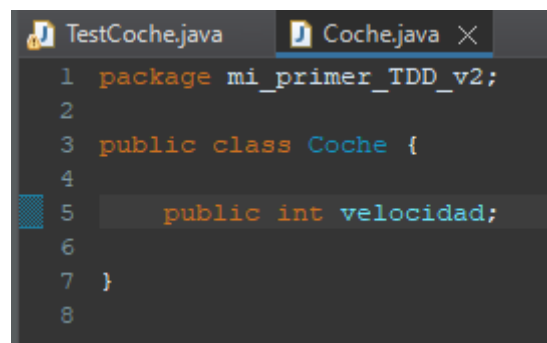
```
1 package mi_primer_TDD_v2;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 import org.junit.jupiter.api.Assertions;
6 import org.junit.jupiter.api.Test;
7
8 class TestCoche {
9
10     @Test
11     public void test_al_crear_un_coche_su_velocidad_es_cero() {
12         Coche nuevoCoche = new Coche();
13         Assertions.assertEquals(0, nuevoCoche.velocidad);
14     }
15
16 }
17
```

- Create field 'velocidad' in type 'Coche'
- Create constant 'velocidad' in type 'Coche'
- Rename in file (Ctrl+2, R)

```
public class Coche {
    public Integer velocidad;
}
```

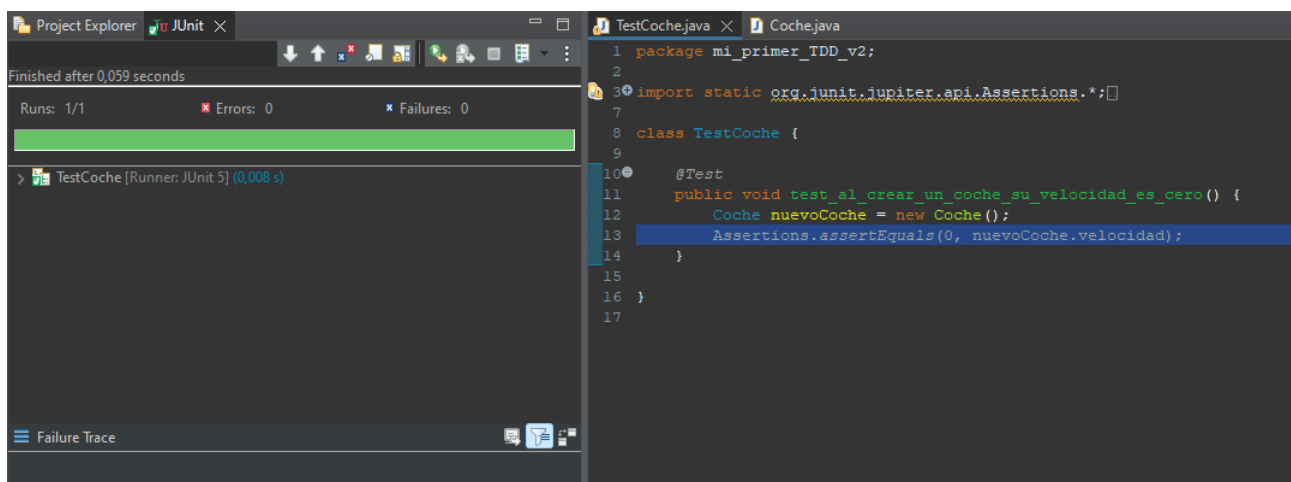


Asi queda la clase Coche



```
TestCoche.java  Coche.java X
1 package mi_primer_TDD_v2;
2
3 public class Coche {
4
5     public int velocidad;
6
7 }
8
```

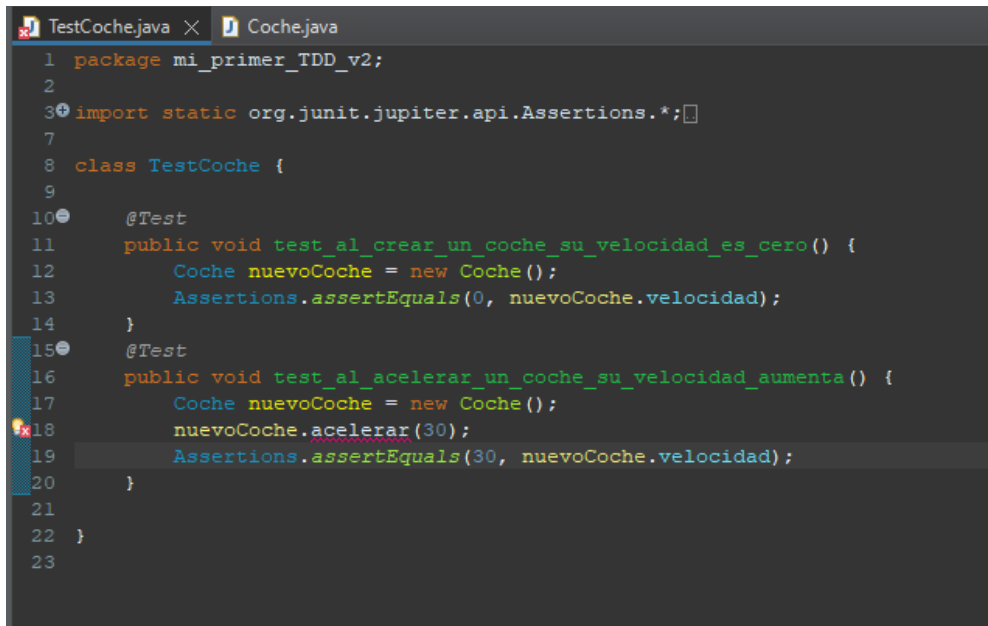
Corremos el test y no da fallos



```
Project Explorer  JUnit X
Finished after 0,059 seconds
Runs: 1/1      Errors: 0      Failures: 0
> TestCoche [Runner: JUnit 5] (0,008 s)
Failure Trace

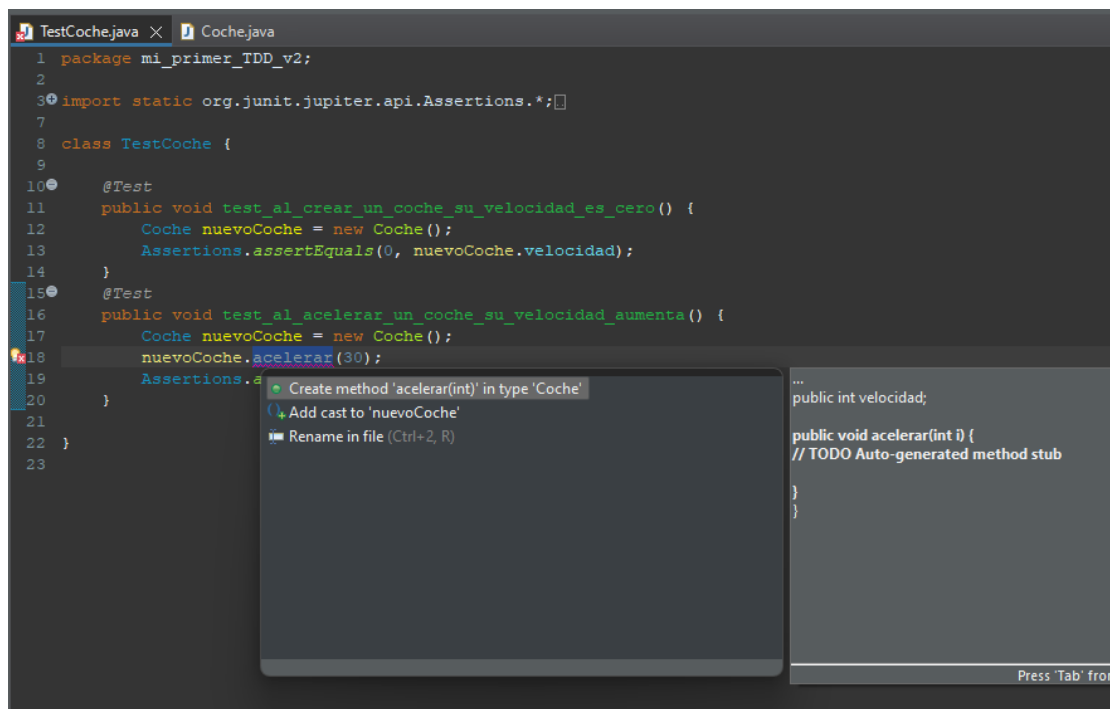
TestCoche.java  Coche.java
1 package mi_primer_TDD_v2;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 class TestCoche {
6
7     @Test
8     public void test_al_crear_un_coche_su_velocidad_es_cero() {
9         Coche nuevoCoche = new Coche();
10        Assertions.assertEquals(0, nuevoCoche.velocidad);
11    }
12 }
13
14
15
16
17
```

Añadimos un segundo test, que añade un nuevo metodo, que acelera el coche. Con el assertEquals, comprobamos que la velocidad final del coche, sea la suma de la velocidad inicial mas la aceleracion, que en este caso sera 30. El metodo no esta creado, por lo que tenemos que crearlo en la clase Coche



```
1 package mi_primer_TDD_v2;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5
6
7
8 class TestCoche {
9
10     @Test
11     public void test_al_crear_un_coche_su_velocidad_es_cero() {
12         Coche nuevoCoche = new Coche();
13         Assertions.assertEquals(0, nuevoCoche.velocidad);
14     }
15
16     @Test
17     public void test_al_acelerar_un_coche_su_velocidad_aumenta() {
18         Coche nuevoCoche = new Coche();
19         nuevoCoche.acelerar(30);
20         Assertions.assertEquals(30, nuevoCoche.velocidad);
21     }
22 }
23
```

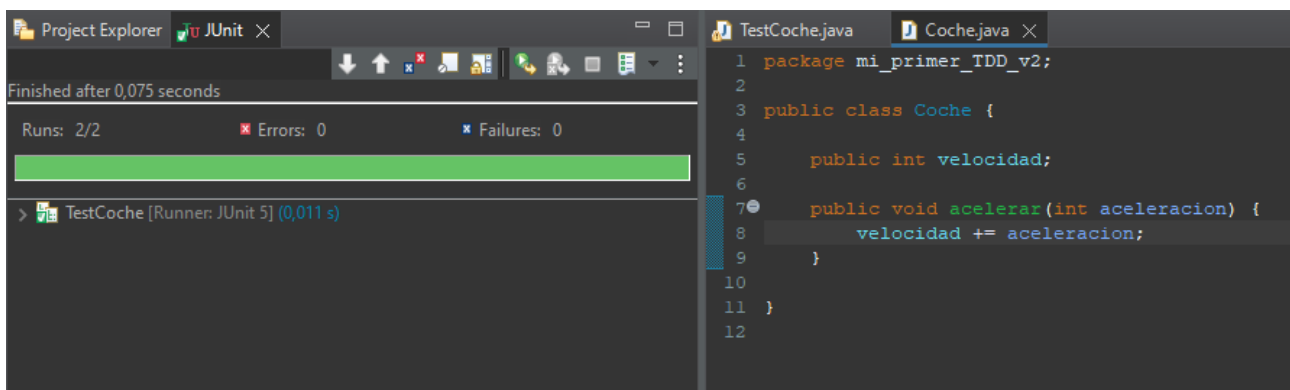
Dandole a la bombilla podemos darle a la opcion de crear metodo automaticamente



```
1 package mi_primer_TDD_v2;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5
6
7
8 class TestCoche {
9
10     @Test
11     public void test_al_crear_un_coche_su_velocidad_es_cero() {
12         Coche nuevoCoche = new Coche();
13         Assertions.assertEquals(0, nuevoCoche.velocidad);
14     }
15
16     @Test
17     public void test_al_acelerar_un_coche_su_velocidad_aumenta() {
18         Coche nuevoCoche = new Coche();
19         nuevoCoche.acelerar(30);
20         Assertions.a
21     }
22 }
23
```

```
... public int velocidad;
public void acelerar(int i) {
    // TODO Auto-generated method stub
}
}
```

Una vez creamos el metodo, le añadimos que sume la aceleracion a la velocidad y corremos el test, que no da fallos

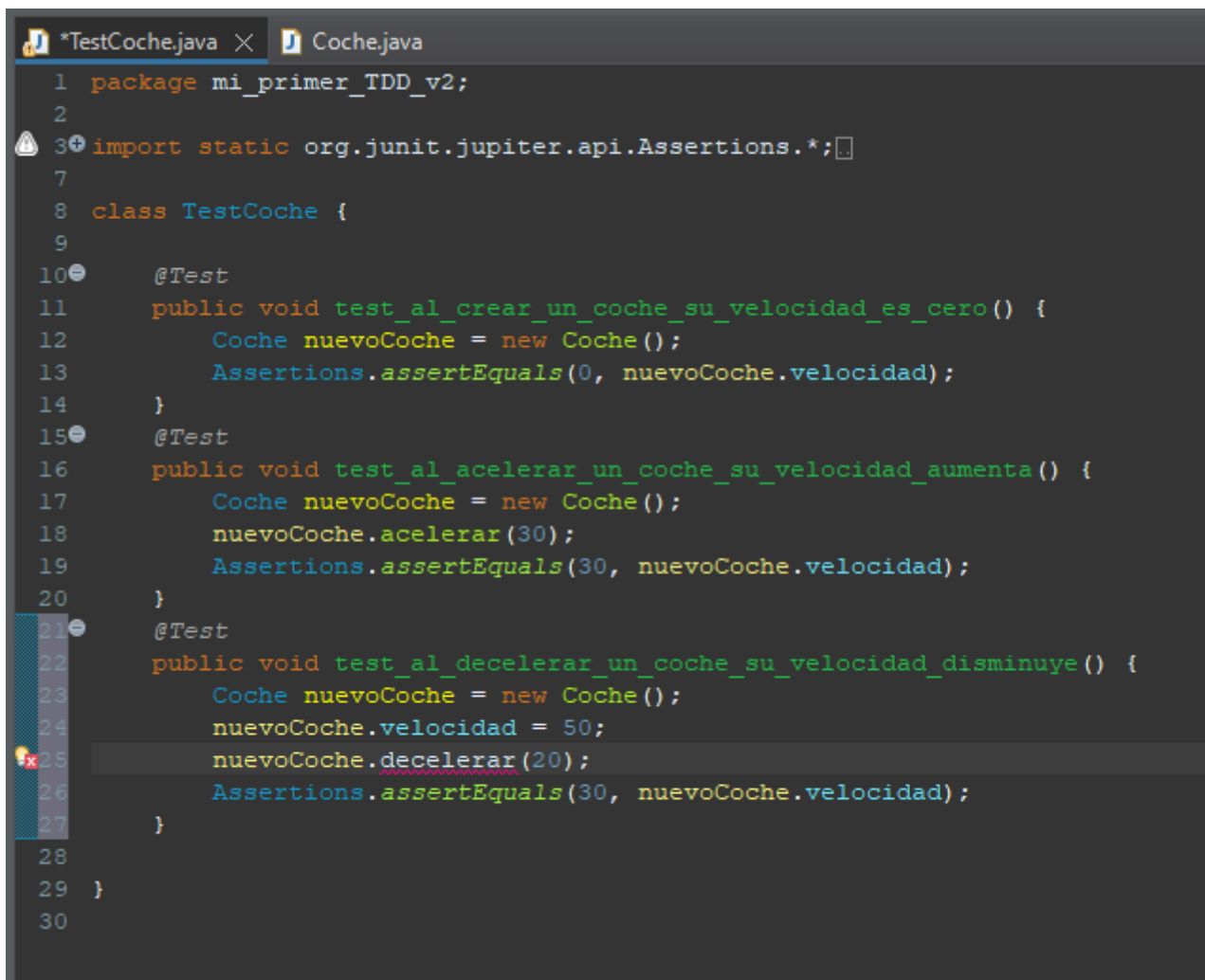


The screenshot shows an IDE with two tabs: 'TestCoche.java' and 'Coche.java'. The 'TestCoche.java' tab is active, displaying the results of a JUnit test run. The output indicates that the test was successful, with 2/2 runs, 0 errors, and 0 failures. The test took 0.011 seconds to complete. The 'Coche.java' tab shows the implementation of the 'acelerar' method, which increments the 'velocidad' attribute by the 'aceleracion' parameter.

```
1 package mi_primer_TDD_v2;
2
3 public class Coche {
4
5     public int velocidad;
6
7     public void acelerar(int aceleracion) {
8         velocidad += aceleracion;
9     }
10
11 }
12
```

Añadimos el tercer test, que disminuye la velocidad, el assertEquals debe dar la velocidad inicial menos la deceleracion.

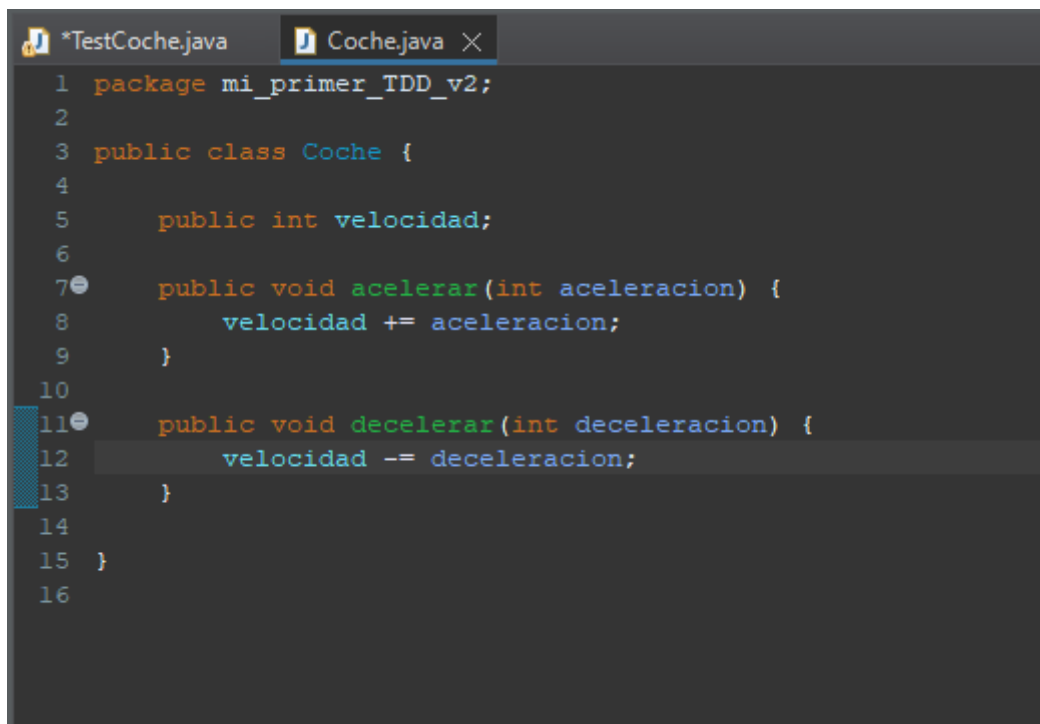
Para crear el metodo decelerar lo hacemos al igual que el de acelerar



The screenshot shows the same IDE with the 'TestCoche.java' tab active. The 'TestCoche' class now contains three test methods: 'test\_al\_crear\_un\_coche\_su\_velocidad\_es\_cero', 'test\_al\_acelerar\_un\_coche\_su\_velocidad\_aumenta', and 'test\_al\_decelerar\_un\_coche\_su\_velocidad\_disminuye'. The third test method is currently selected, and it calls the 'decelerar' method on a new 'Coche' object. The 'decelerar' method is not yet implemented, as indicated by the red squiggly line under the method call. The 'Coche.java' tab shows the implementation of the 'decelerar' method, which decrements the 'velocidad' attribute by the 'deceleracion' parameter.

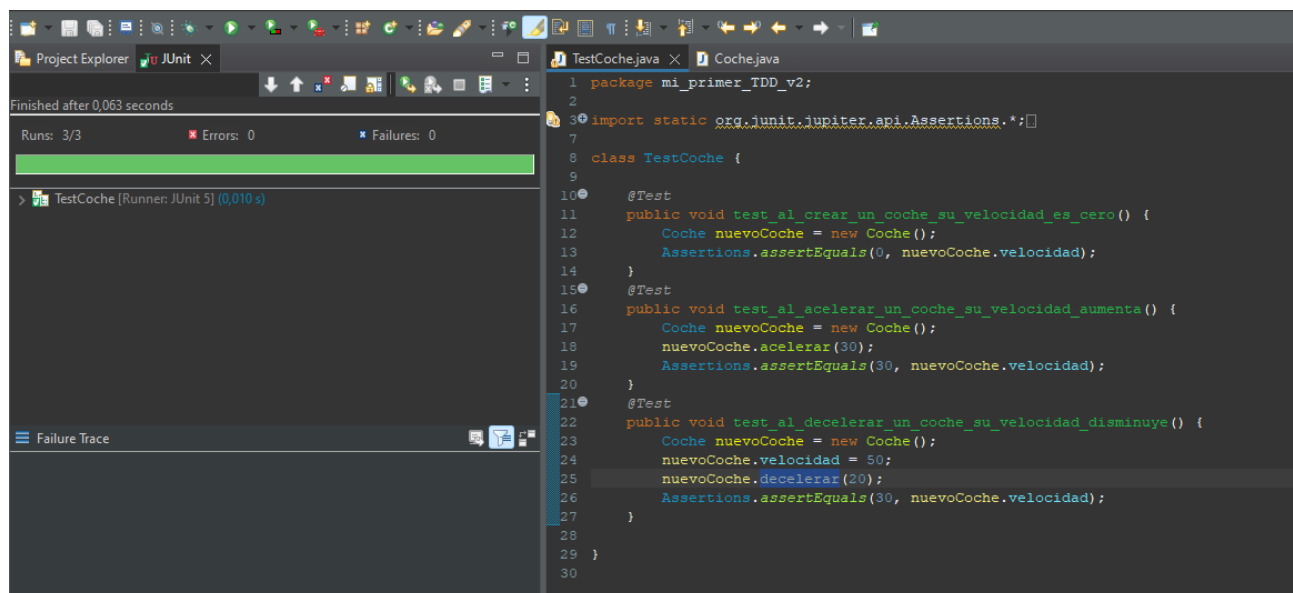
```
1 package mi_primer_TDD_v2;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 class TestCoche {
6
7     @Test
8     public void test_al_crear_un_coche_su_velocidad_es_cero() {
9         Coche nuevoCoche = new Coche();
10        Assertions.assertEquals(0, nuevoCoche.velocidad);
11    }
12
13    @Test
14    public void test_al_acelerar_un_coche_su_velocidad_aumenta() {
15        Coche nuevoCoche = new Coche();
16        nuevoCoche.acelerar(30);
17        Assertions.assertEquals(30, nuevoCoche.velocidad);
18    }
19
20    @Test
21    public void test_al_decelerar_un_coche_su_velocidad_disminuye() {
22        Coche nuevoCoche = new Coche();
23        nuevoCoche.velocidad = 50;
24        nuevoCoche.decelerar(20);
25        Assertions.assertEquals(30, nuevoCoche.velocidad);
26    }
27
28 }
29
30
```

Asi quedaria el metodo en la clase Coche, restandole la deceleracion a la velocidad



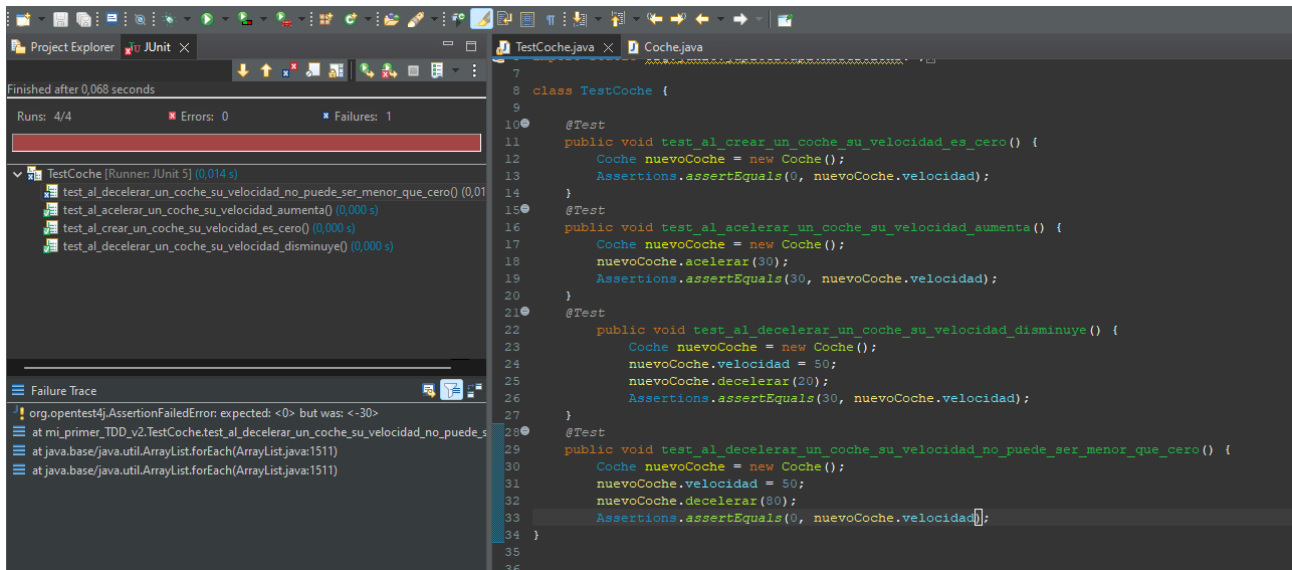
```
1 package mi_primer_TDD_v2;
2
3 public class Coche {
4
5     public int velocidad;
6
7     public void acelerar(int aceleracion) {
8         velocidad += aceleracion;
9     }
10
11     public void decelerar(int deceleracion) {
12         velocidad -= deceleracion;
13     }
14
15 }
16
```

Corremos el test sin fallos

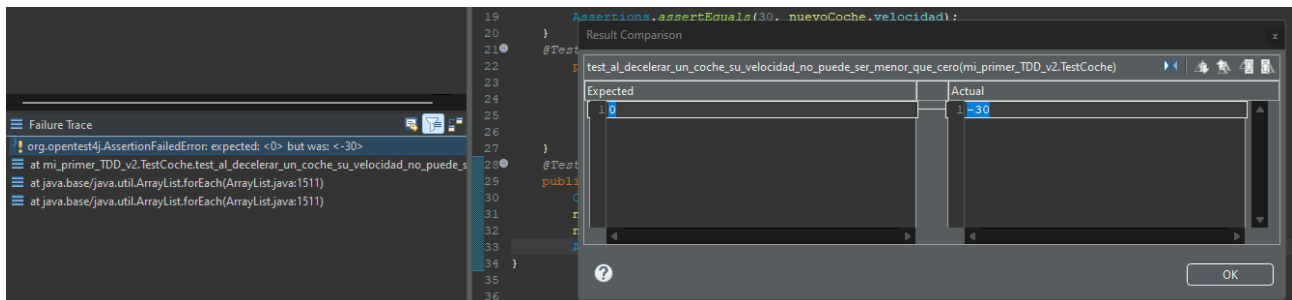


```
1 package mi_primer_TDD_v2;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 class TestCoche {
6
7     @Test
8     public void test_al_crear_un_coche_su_velocidad_es_cero() {
9         Coche nuevoCoche = new Coche();
10        Assertions.assertEquals(0, nuevoCoche.velocidad);
11    }
12
13     @Test
14     public void test_al_acelerar_un_coche_su_velocidad_aumenta() {
15        Coche nuevoCoche = new Coche();
16        nuevoCoche.acelerar(30);
17        Assertions.assertEquals(30, nuevoCoche.velocidad);
18    }
19
20     @Test
21     public void test_al_decelerar_un_coche_su_velocidad_disminuye() {
22        Coche nuevoCoche = new Coche();
23        nuevoCoche.velocidad = 50;
24        nuevoCoche.decelerar(20);
25        Assertions.assertEquals(30, nuevoCoche.velocidad);
26    }
27
28 }
29
30
```

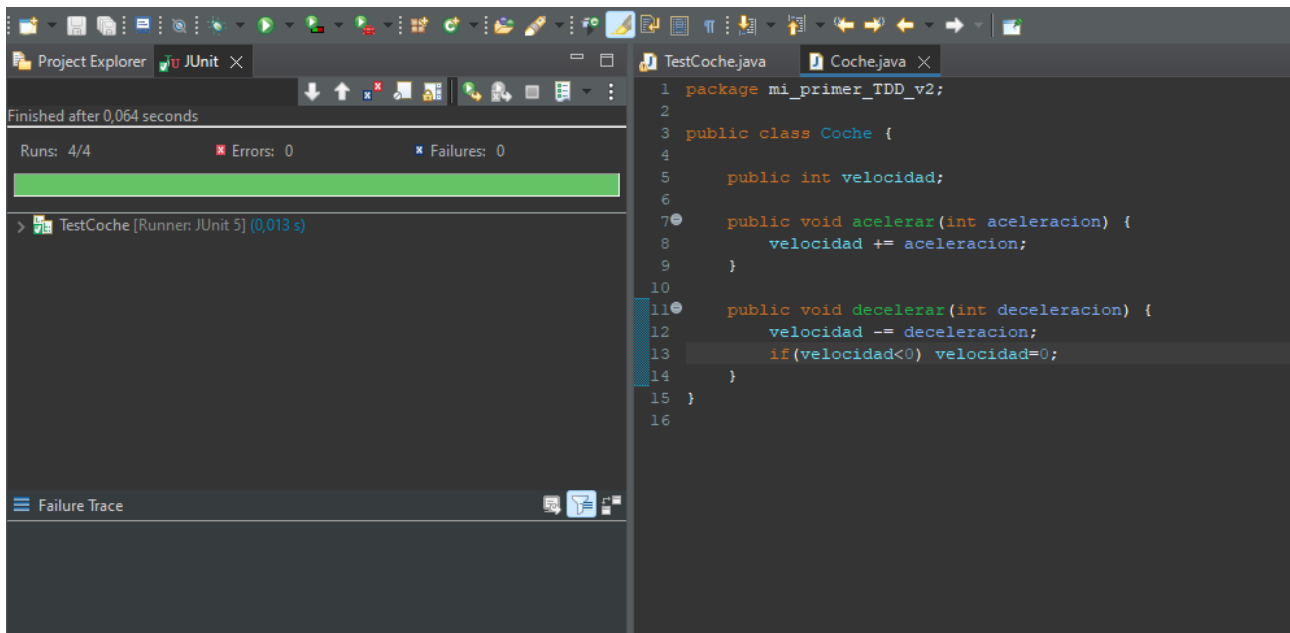
Añadimos el ultimo test, que comprueba que la velocidad del coche no sea menor que cero.  
Pero al correr el test, nos da error, porque la velocidad final es -30



Lo podemos comprobar dándole al test que falla y nos sale esta ventana



Para que no de error el fallo debemos añadir un if al metodo decelerar, para que si la velocidad sea menor que 0, le de el valor 0 a la variable



```
1 package mi_primer_TDD_v2;
2
3 public class Coche {
4
5     public int velocidad;
6
7     public void acelerar(int aceleracion) {
8         velocidad += aceleracion;
9     }
10
11     public void decelerar(int deceleracion) {
12         velocidad -= deceleracion;
13         if(velocidad<0) velocidad=0;
14     }
15 }
16
```

Project Explorer JUnit X

Finished after 0,064 seconds

Runs: 4/4 Errors: 0 Failures: 0

> TestCoche [Runner: JUnit 5] (0,013 s)

Failure Trace