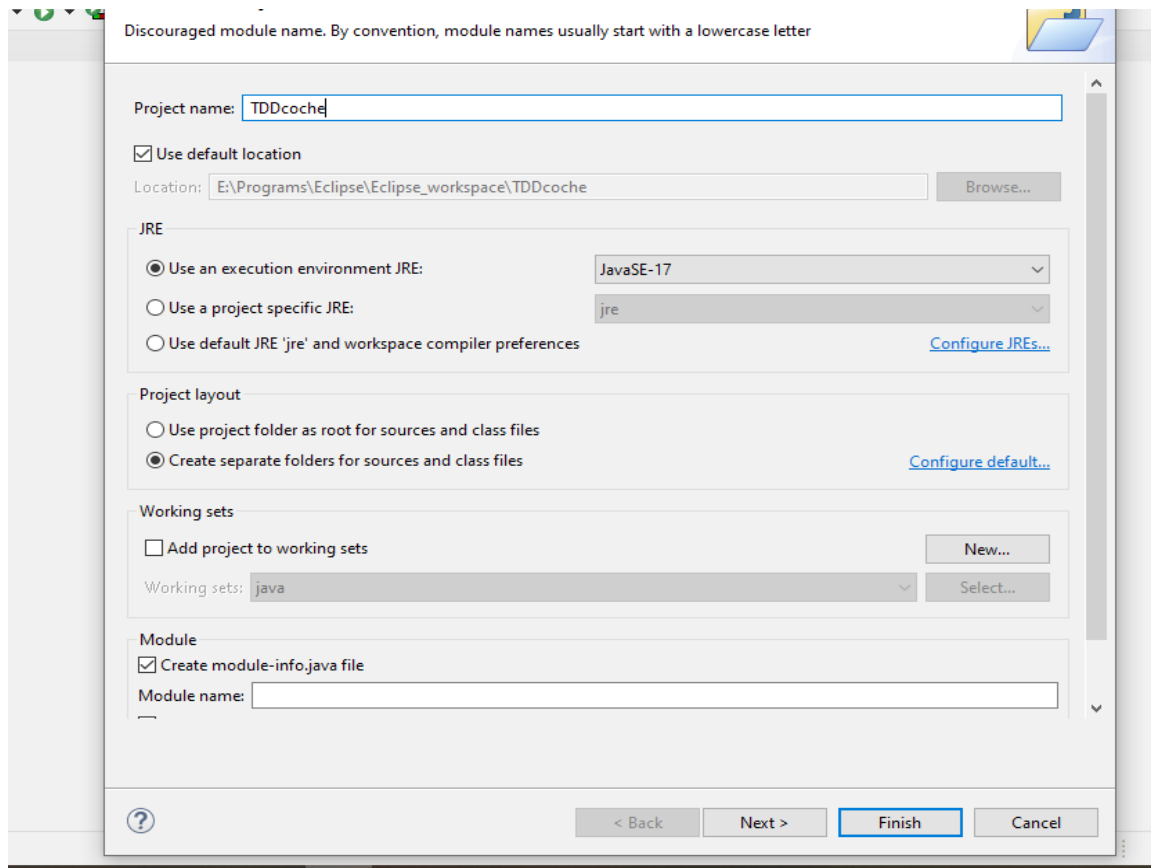
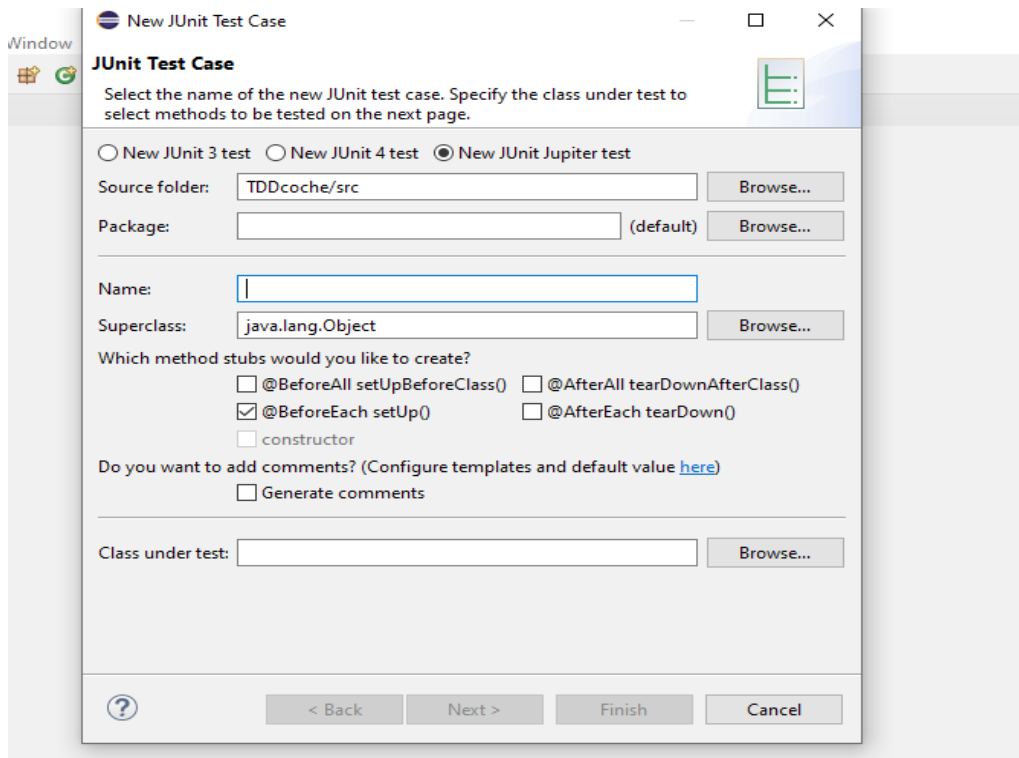


# MEMORIA TDD coche con Eclipse

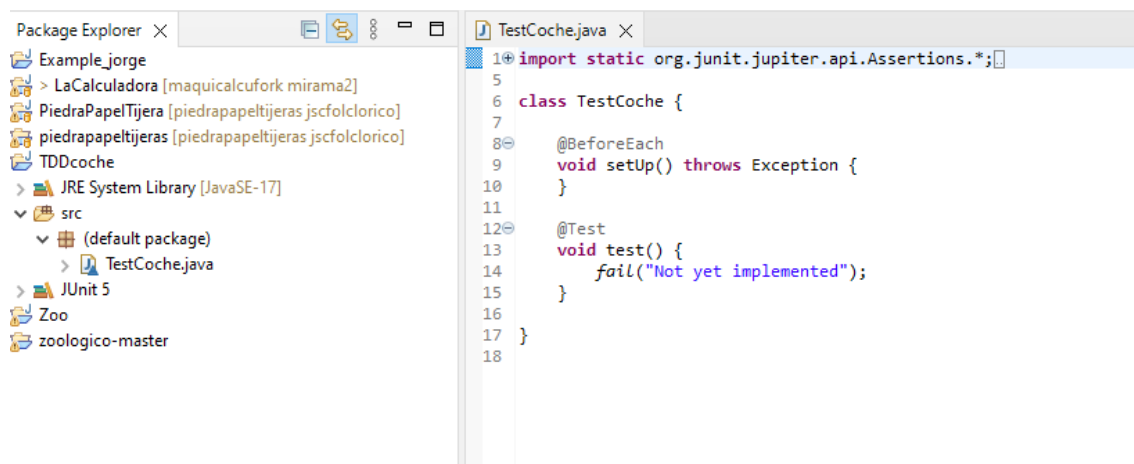
1. Creamos un nuevo proyecto al que llamaremos TDDcoche.



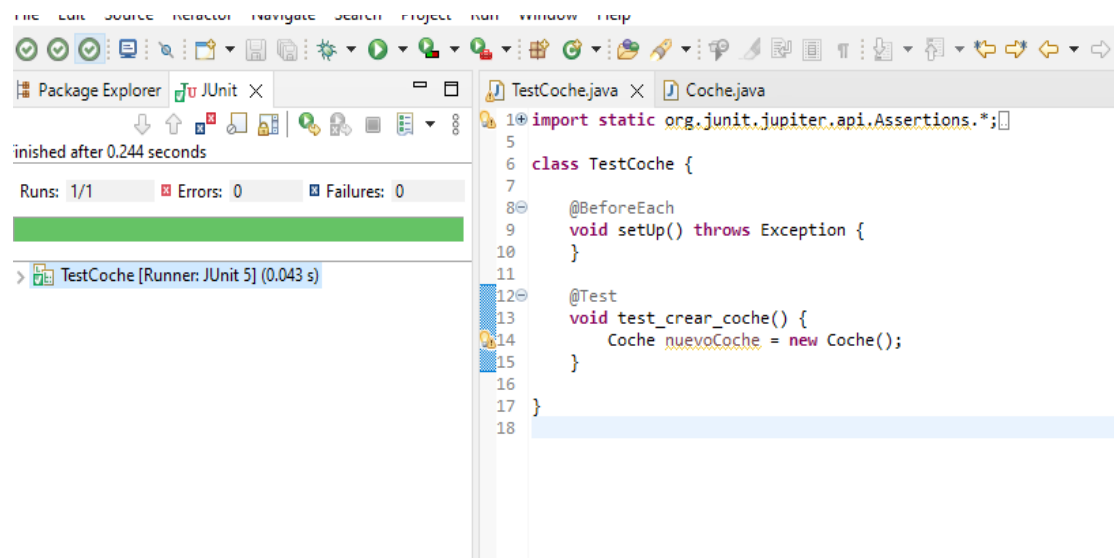
2. Creamos una clase java que la llamaremos TestCoche, para esto pincharemos boton derecho seleccionamos new y elegimos la opción de Junit test. Se abrirá una pantalla como la que se muestra a continuación.



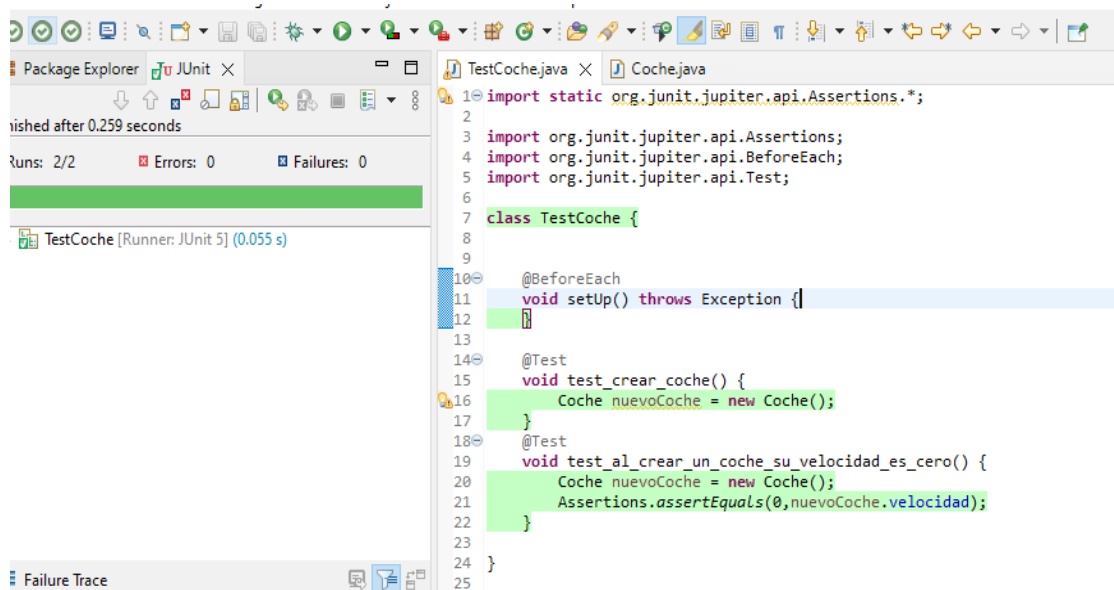
3. se nos creará una clase TestCoche como la que aparece en la imagen, donde se muestran ya las plantillas para implementar y hacer los test.



4. Haremos un test al que llamaremos test\_crear\_coche. Al implementar el método nos saldrá un error en coche y pinchando en él con boton derecho nos mostrará las opciones para solucionarlo. Seleccionaremos la opción de crear la clase Coche. Se creará la clase automáticamente. Seguidamente ejecutaremos el tes corriéndolo como Junit Test desde la flechita verde de la barra superior. En la imagen siguiente se puede ver que el resultado es correcto y no tiene errores.



5. Ahora vamos a probar que un coche cuando se crea tiene velocidad cero. Para ello haremos el test `test_al_crear_un_coche_su_velocidad_es_cero`. Como anteriormente, la velocidad dará un error, marcándola con el puntero nos mostrará opciones para solucionarlo, en este caso tendremos que crear la variable velocidad. En la siguiente imagen se puede ver que al ejecutar el test no se muestran errores.



Se puede ver que el test hemos introducido una línea de código que dice `Assertions`, pues bien con esta sentencia le introducimos el valor que esperamos nos de el test y la variable que queremos testear. En nuestro caso esperamos que de una velocidad de cero.

6. Ahora vamos a probar `test_al_acelerar_un_coche_su_velocidad_aumenta`. Para ello implementaremos el método `acelerar` en la clase `Coche`. Después le pasaremos los valores esperados el test y lo ejecutaremos.

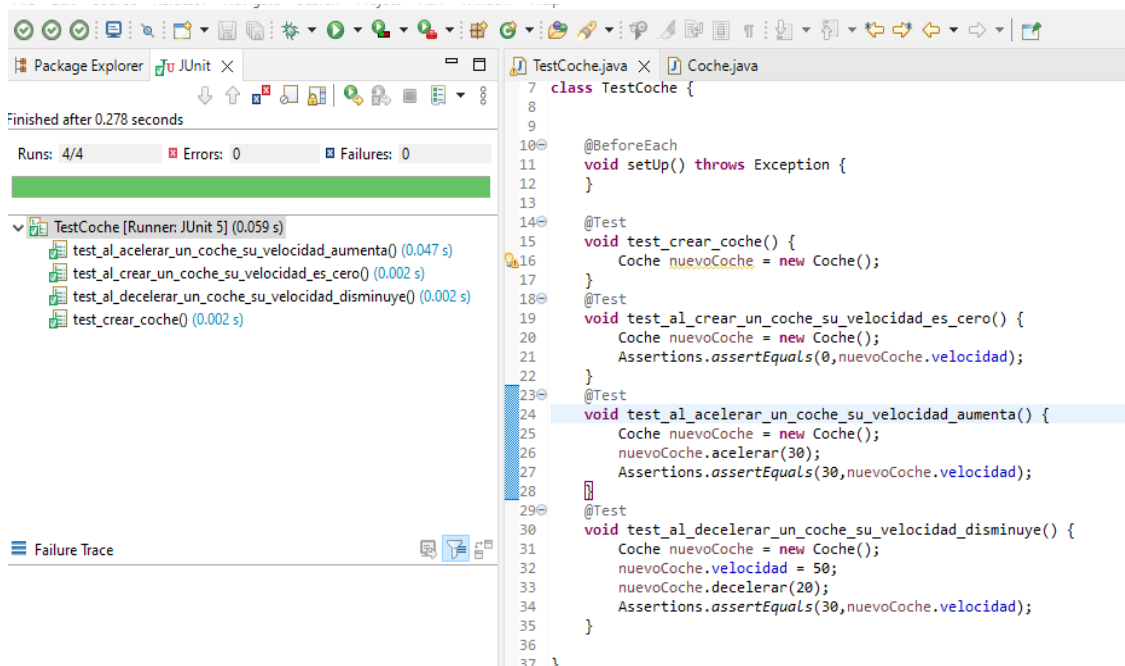
```
*TestCoche.java  *Coche.java X
1
2 public class Coche {
3
4     public Integer velocidad = 0;
5
6     public void acelerar(int aceleracion) {
7         velocidad += aceleracion;
8     }
9
10
11
12
13 }
14
```

```
Package Explorer  JUnit X
nished after 0.263 seconds
Runs: 3/3  Errors: 0  Failures: 0
TestCoche [Runner: JUnit 5] (0.050 s)
  test_al_acelerar_un_coche_su_velocidad_aumenta() (0.043 s)
  test_al_crear_un_coche_su_velocidad_es_cero() (0.002 s)
  test_crear_coche() (0.002 s)
Failure Trace

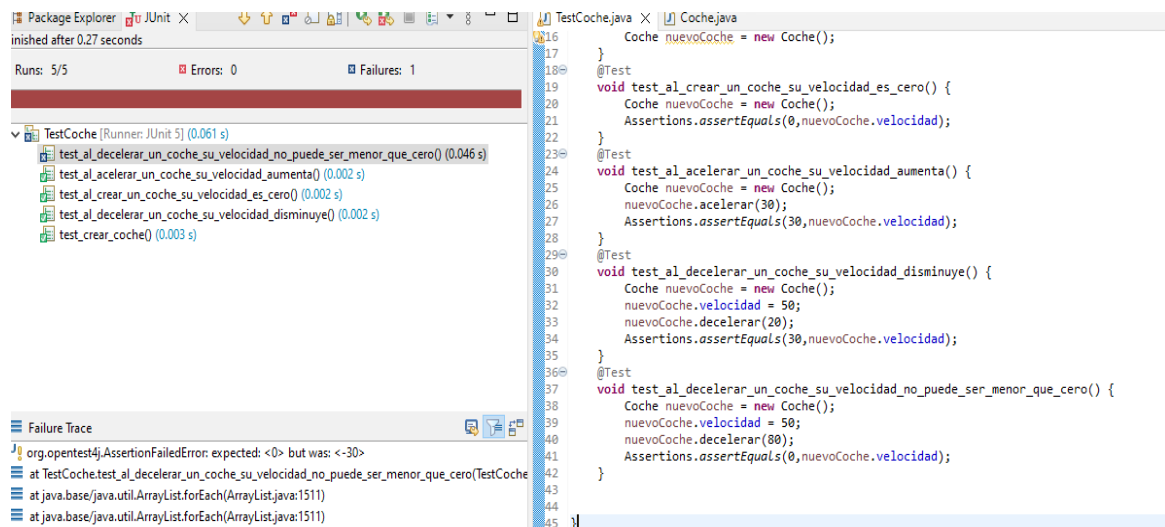
TestCoche.java  Coche.java
1 import static org.junit.jupiter.api.Assertions.*;
2
3 import org.junit.jupiter.api.Assertions;
4 import org.junit.jupiter.api.BeforeEach;
5 import org.junit.jupiter.api.Test;
6
7 class TestCoche {
8
9
10 @BeforeEach
11 void setUp() throws Exception {
12 }
13
14 @Test
15 void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

7. Haremos lo mismo pero para decelerar. A continuación se muestra como quedarían las clases al respecto.

```
*TestCoche.java  *Coche.java X
1
2 public class Coche {
3
4     public Integer velocidad = 0;
5
6     public void acelerar(int aceleracion) {
7         velocidad += aceleracion;
8     }
9
10
11     public void decelerar(int deceleracion) {
12         velocidad -= deceleracion;
13     }
14 }
15
16 }
17
```



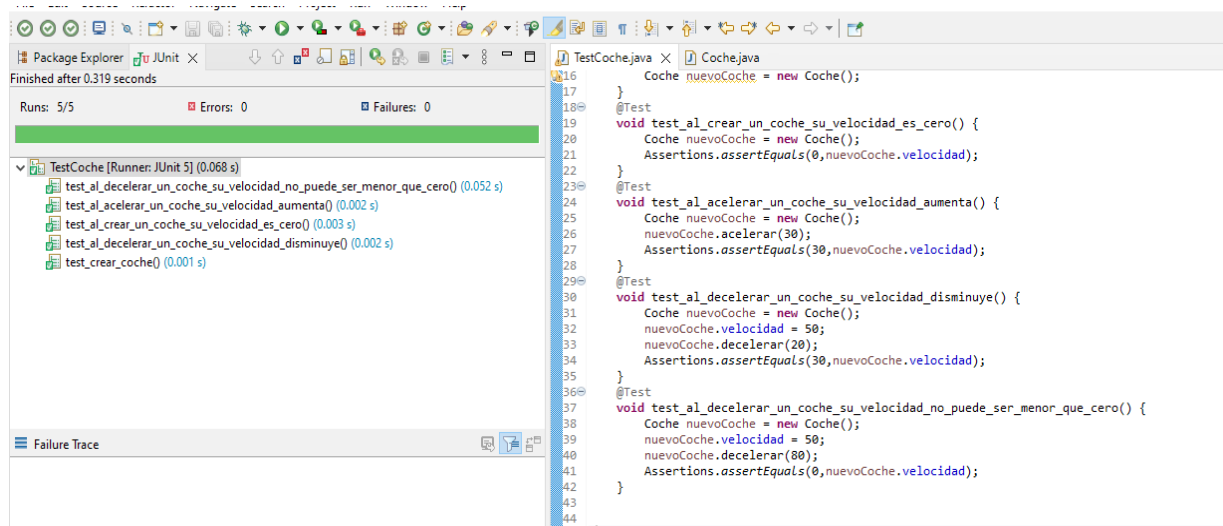
8. Ahora vamos a hacer que el test de decelerar falle, para ello pondremos los valores de que el coche tiene una velocidad de 50, su deceleracion es de 80 y esperamos un valor de velocidad de 0 al pararse el coche.



Vemos que nos da un error. Para solucionarlo vamos a modificar el método decelerar de la clase Coche, quedaría como se muestra en la imagen de abajo.

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

9. Una vez corregido esto, es decir se si la velocidad es menor que cero hacemos que la velocidad sea 0 en lugar de un valor negativo. Ejecutando el test podemos ver que ya no nos da el error, como se aprecia en la siguiente imagen.



**Jorge Selva Cerdá**