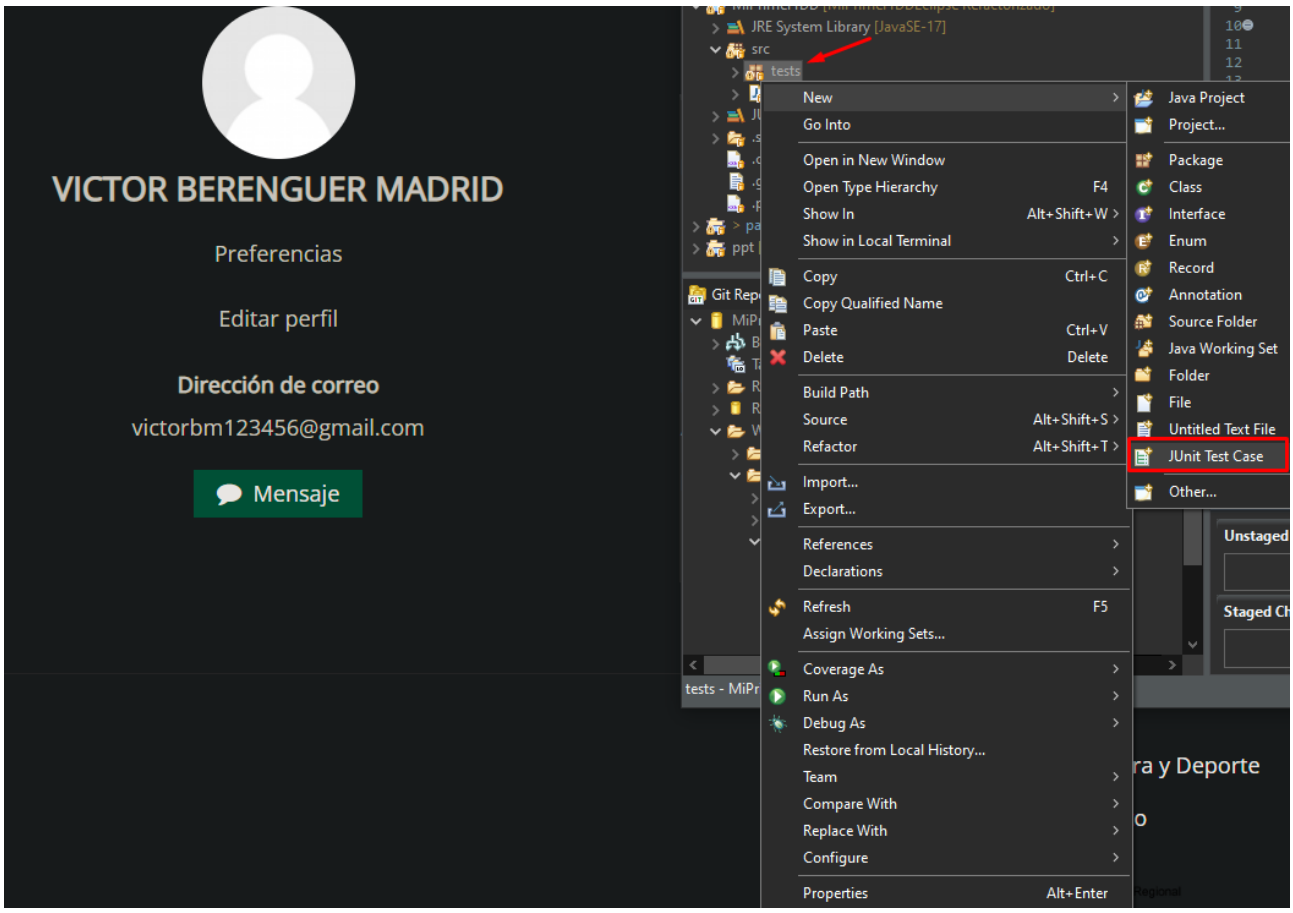
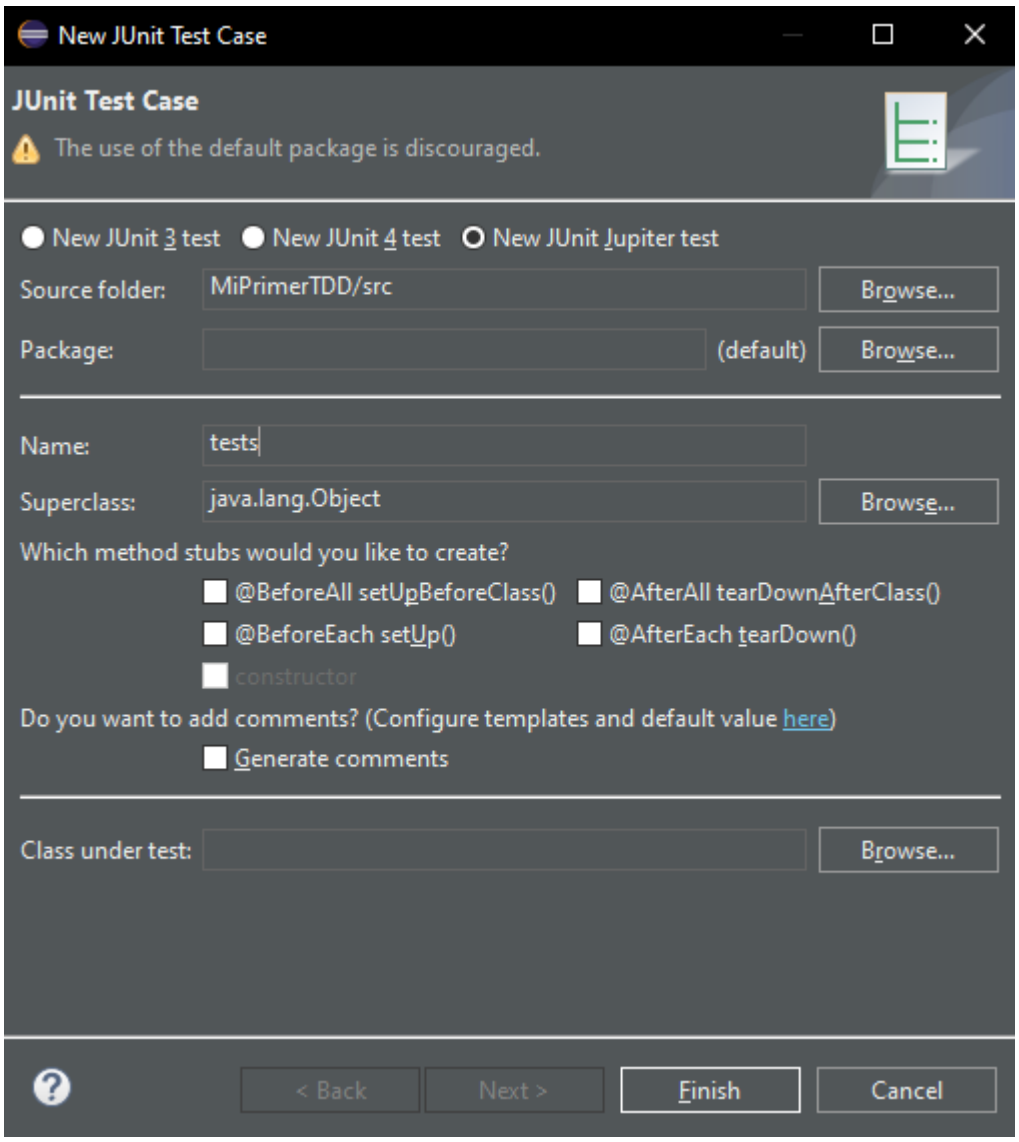
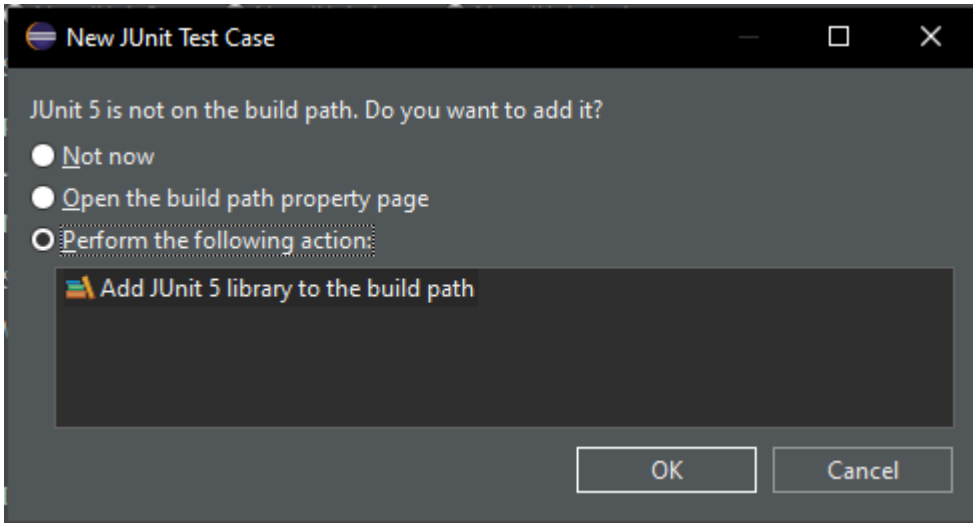


Comenzamos creando un nuevo proyecto de Java en Eclipse y, dentro de la carpeta src del proyecto, crearemos un nuevo paquete. Dentro de este paquete, crearemos un directorio de tests de JUnit con click derecho > New > JUnit Test Case.



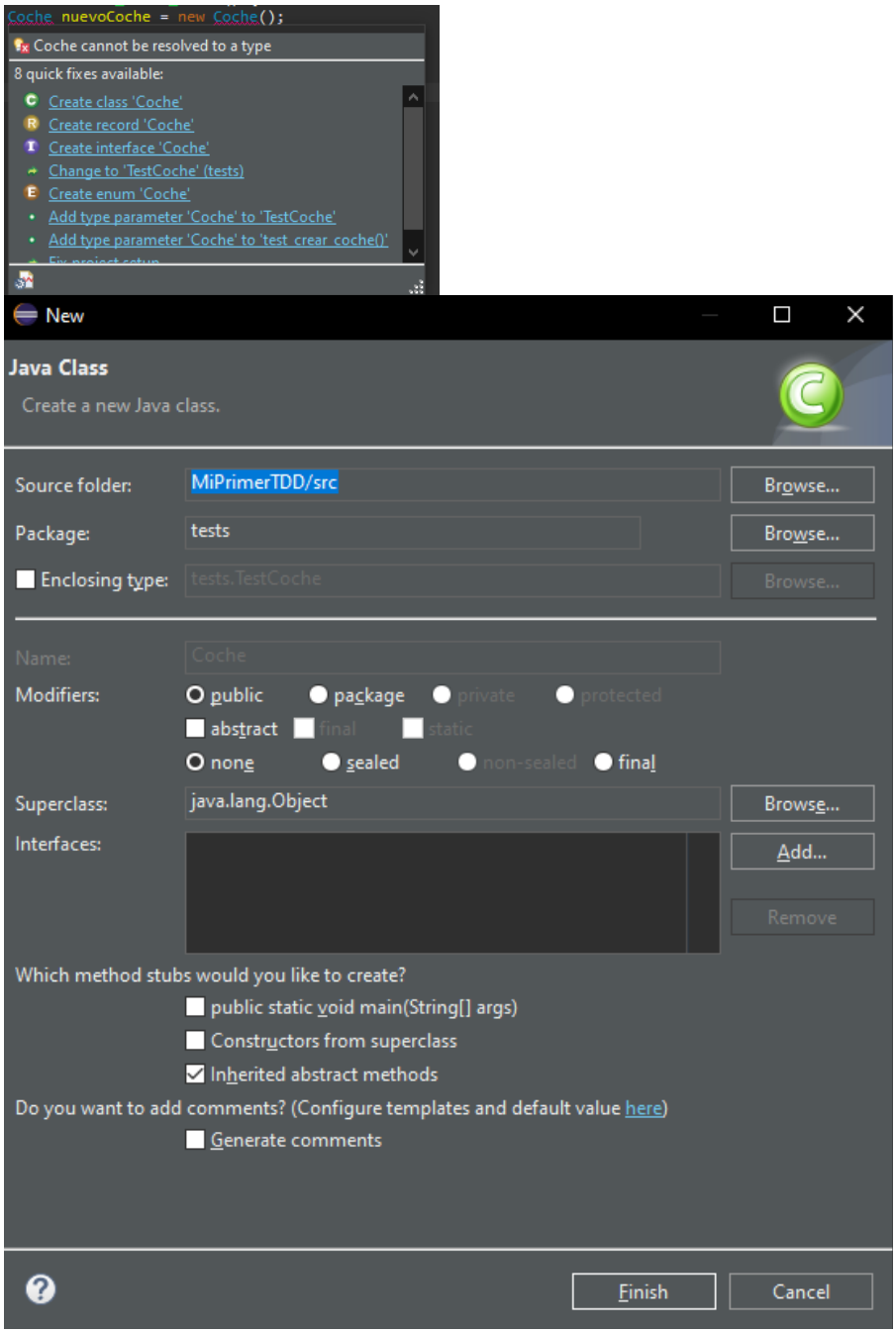
Aparecerá una ventana en la que indicaremos la versión de JUnit (Jupiter en este caso) y el nombre del directorio.



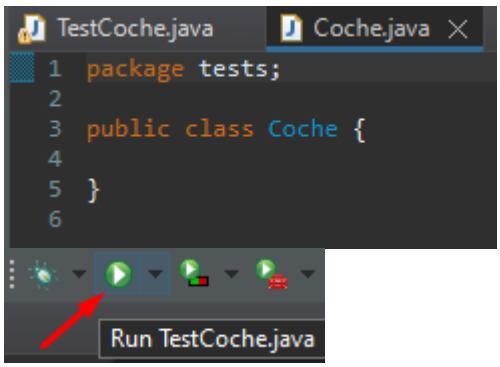
Indicaremos que queremos añadir la librería de JUnit 5 en la siguiente ventana.

```java
package tests;

import static org.junit.jupiter.api.Assertions.*;

class TestCoche {

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

}
```

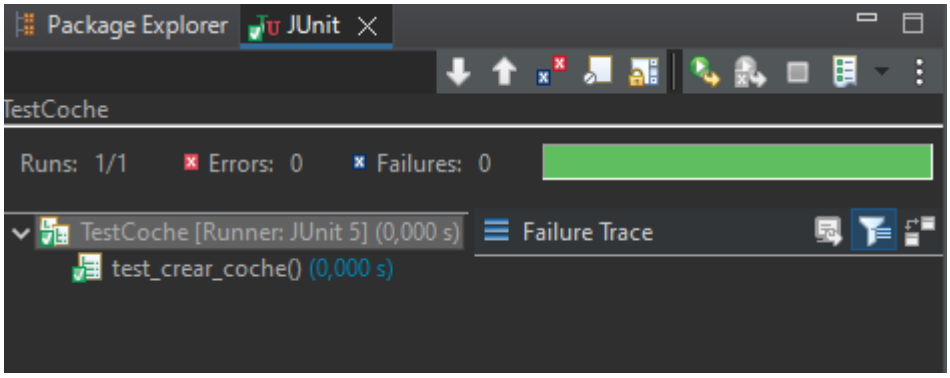
Siguiendo los mismos pasos que en la versión de TDD de IntelliJ, creamos un primer test para crear el coche.



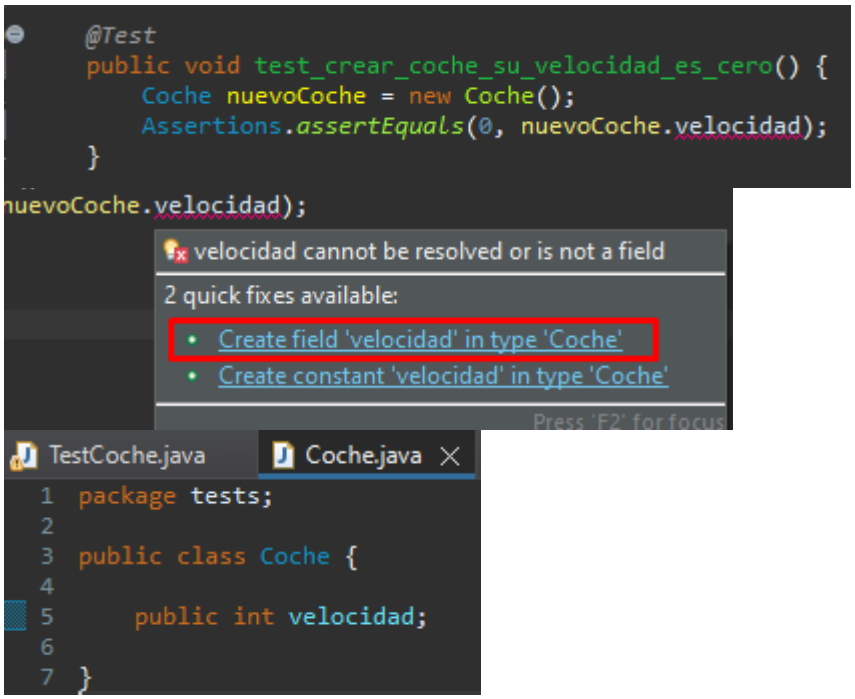
Aprovechamos y creamos la clase Coche al pasar el cusor por encima de la palabra subrayada en rojo y haciendo click en Create class ‘Coche’. Aparecerá un recuadro de creación de nueva clase en el que indicaremos el destino de la clase.



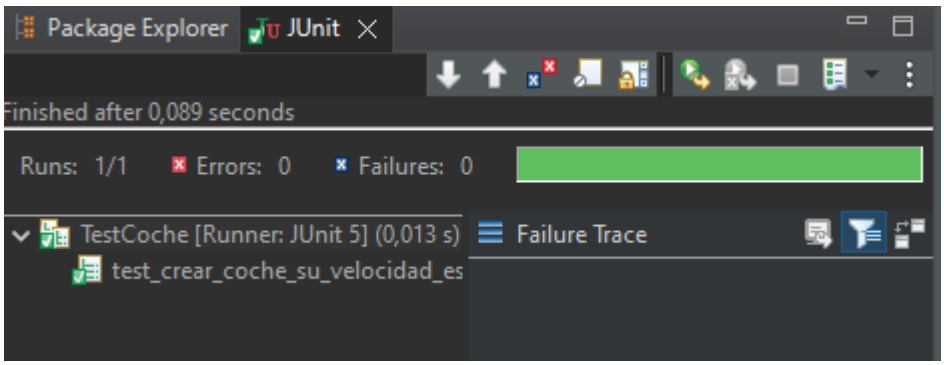
Una vez creada, podemos iniciar el primer test iniciando el programa.



Pasa el test sin problemas.



De la misma manera que en IntelliJ,creamos el método de la velocidad inicial, además de crear el atributo en la clase Coche.

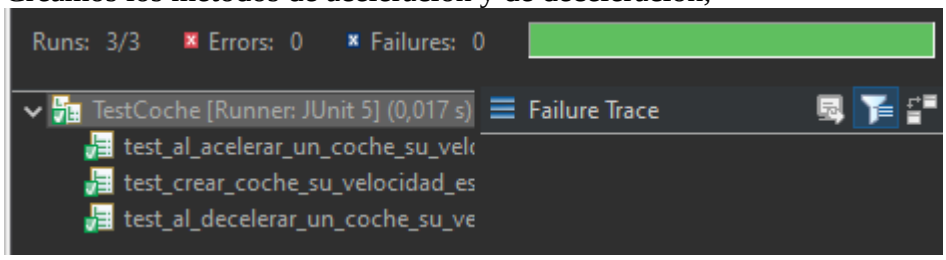


El test es correcto.

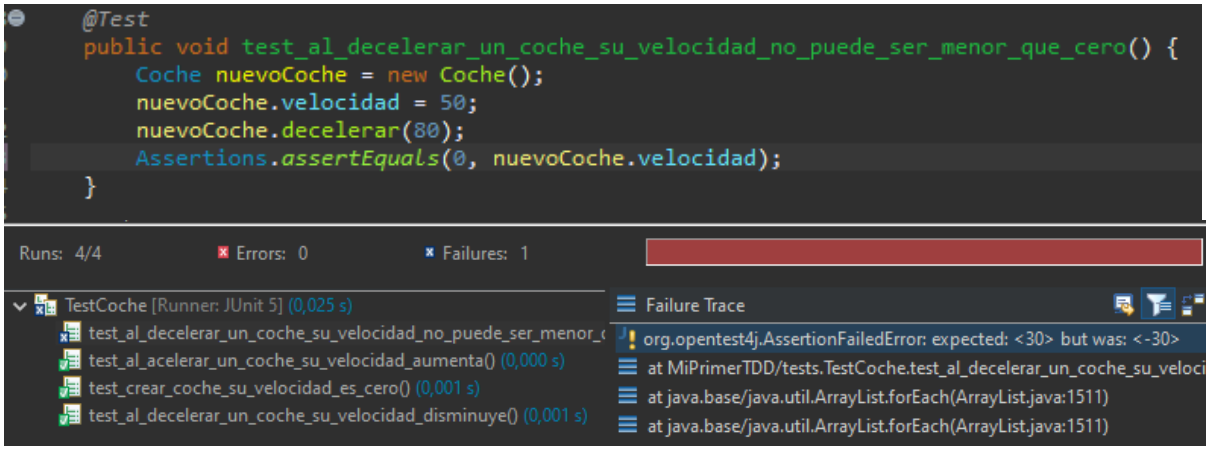
```java
    @Test
    public void test_al_acelerar_un_coche_su_velocidad_aumenta() {
        Coche nuevoCoche = new Coche();
        nuevoCoche.acelerar(30);
        Assertions.assertEquals(30, nuevoCoche.velocidad);
    }
    @Test
    public void test_al_decelerar_un_coche_su_velocidad_disminuye() {
        Coche nuevoCoche = new Coche();
        nuevoCoche.velocidad = 50;
        nuevoCoche.decelerar(20);
        Assertions.assertEquals(30, nuevoCoche.velocidad);
    }

}
```

```java
    public void acelerar(int aceleracion) {
        velocidad += aceleracion;

    }

    public void decelerar(int deceleracion) {
        velocidad -= deceleracion;
    }

}
```

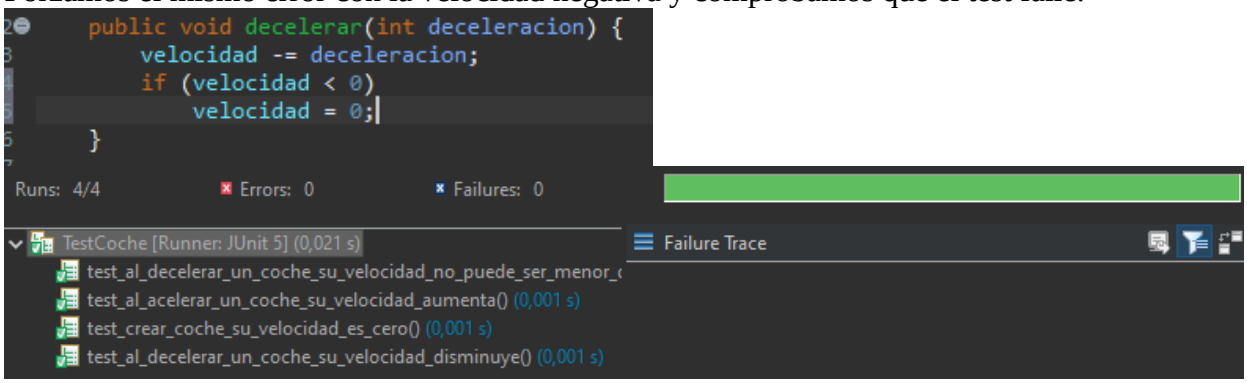
Creamos los métodos de aceleración y de deceleración,



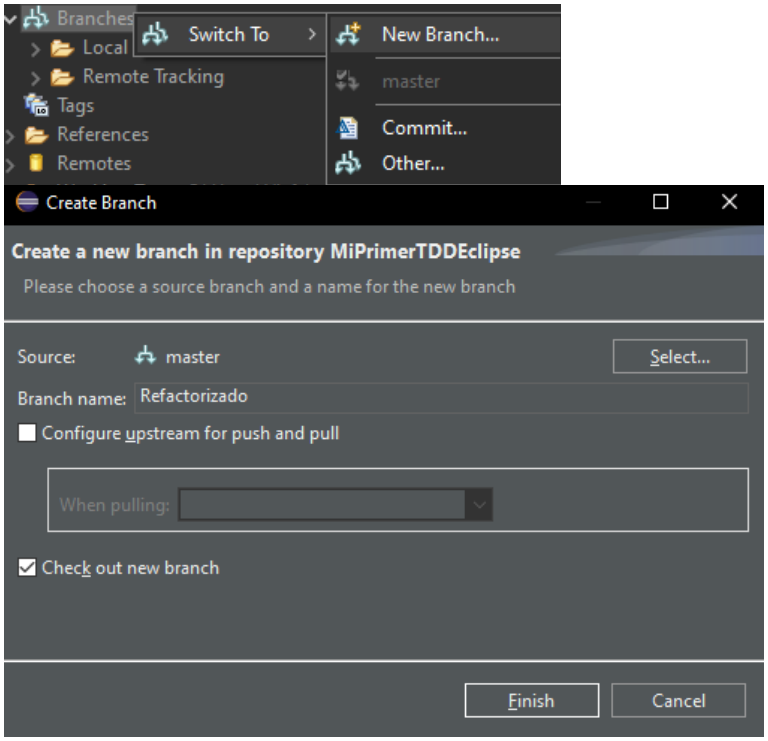
Comprobamos que el test sea correcto.



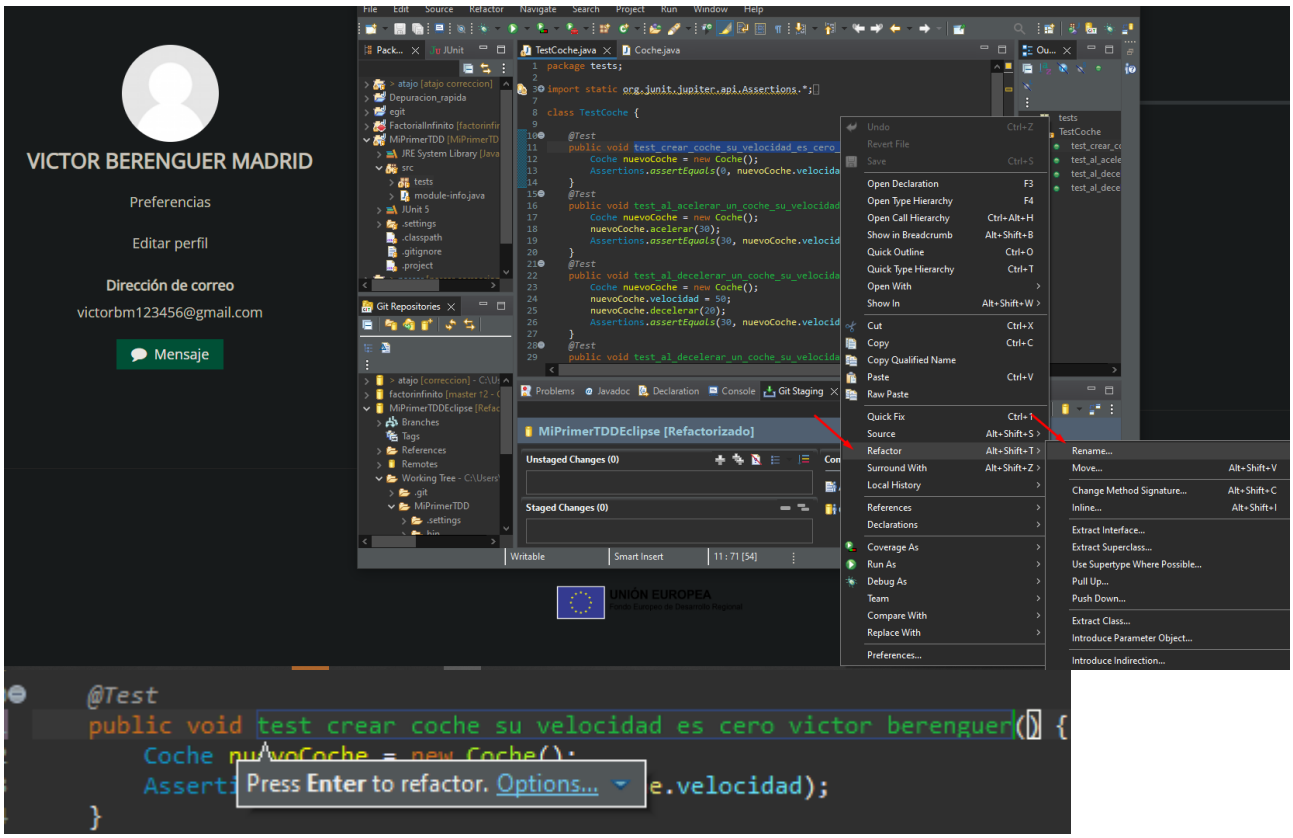
Forzamos el mismo error con la velocidad negativa y comprobamos que el test falle.



Por último, arreglamos el error con un condicional en el método decelerar y comprobamos que el test sea correcto.



Creamos la rama Refactorizado haciendo click derecho en el apartado de ramas de la interfaz de Git en Eclipse > Switch To > New Branch. Confirmamos en la siguiente ventana marcando la casilla de Check out new branch para cambiarnos a la rama recién creada.



Para refactorizar, hacemos click derecho sobre el método > Refactor > Rename.