

Entornos de Desarrollo
Unidad 10 - Mi primer TDD

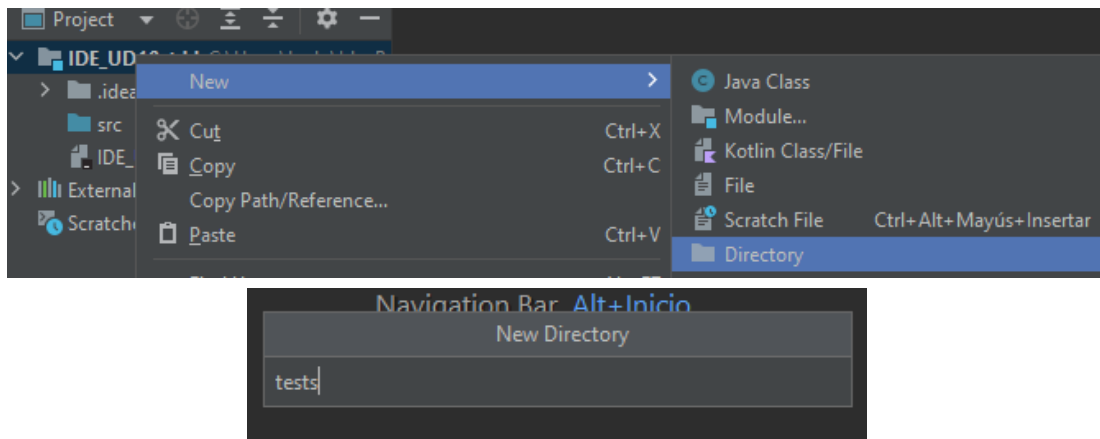
Carlos Blasco Espada
1º Q - DAM

Pasos previos:

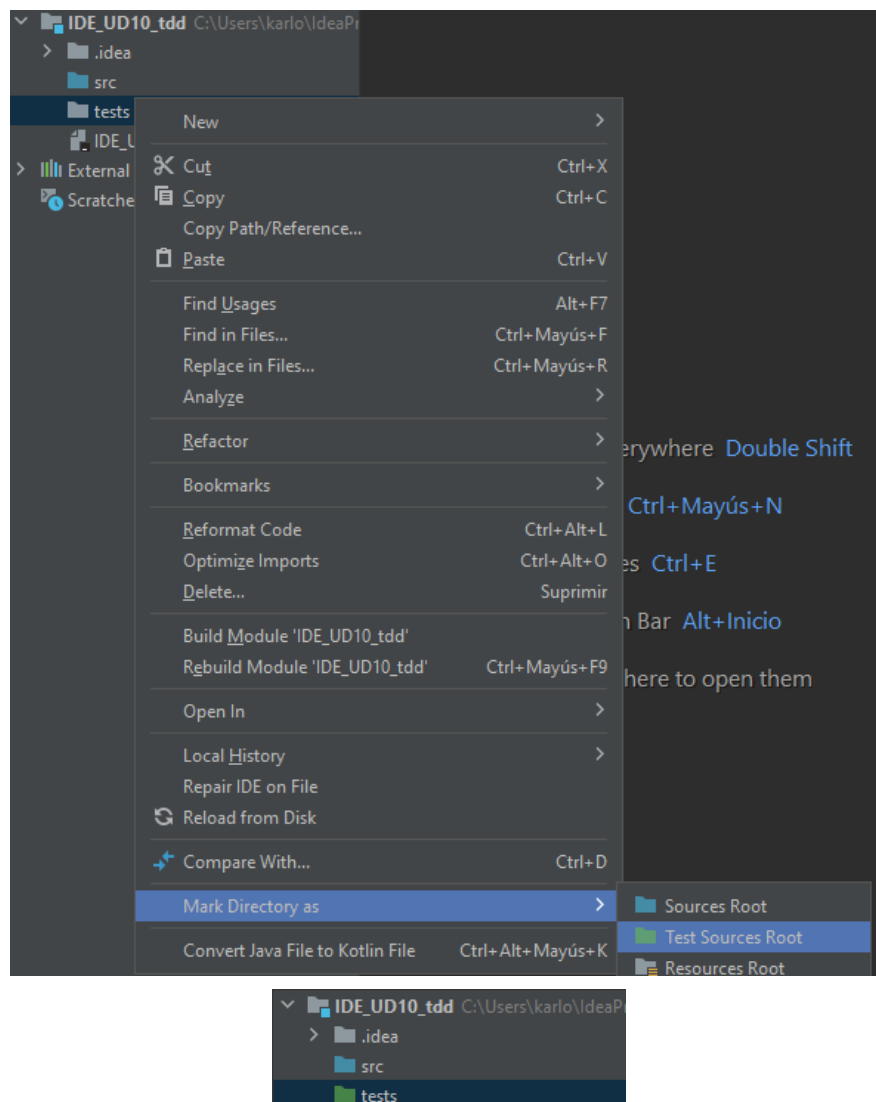
- Necesitamos el entorno de desarrollo IntelliJ IDEA.
- Tener el plugin JUnit 5 instalado.
- Crear el proyecto.

Pasos a seguir:

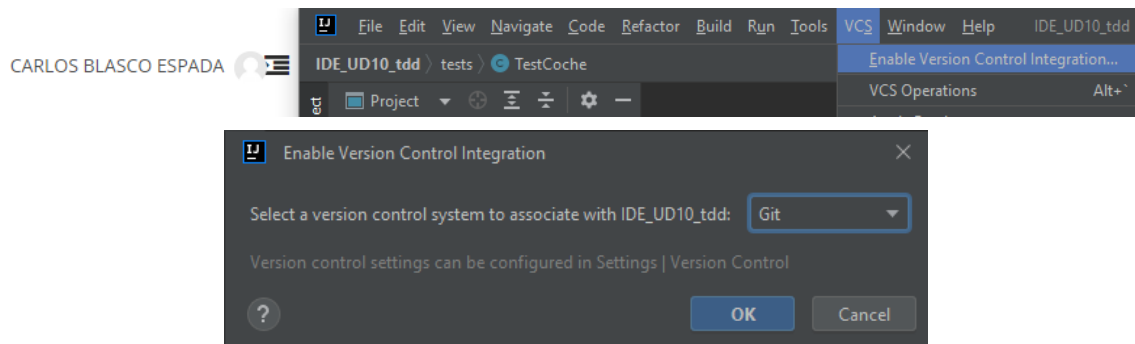
1. Creamos un directorio para los tests.



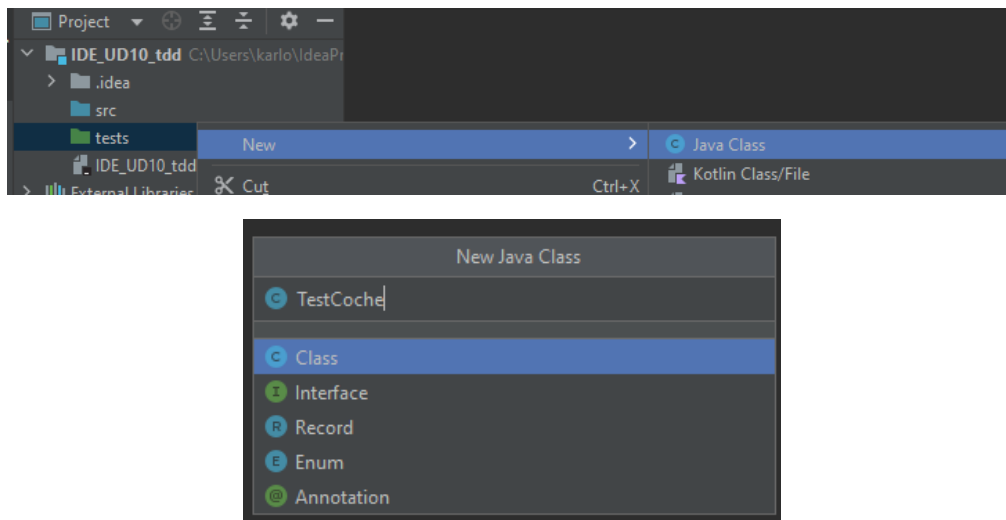
2. Lo marcamos como directorio de test.



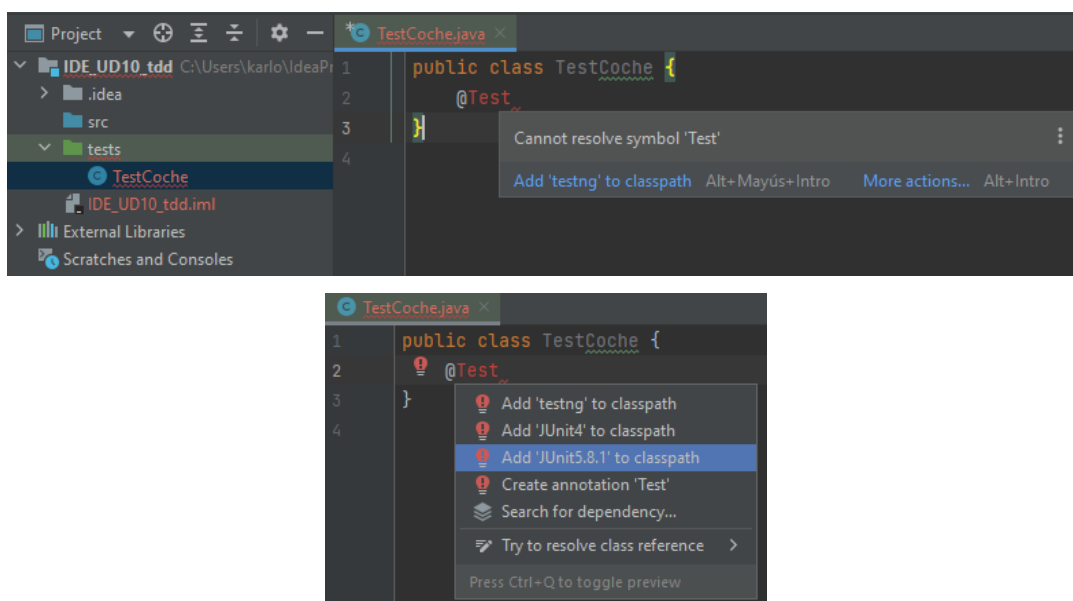
3. Habilitamos el control de versiones con Git.

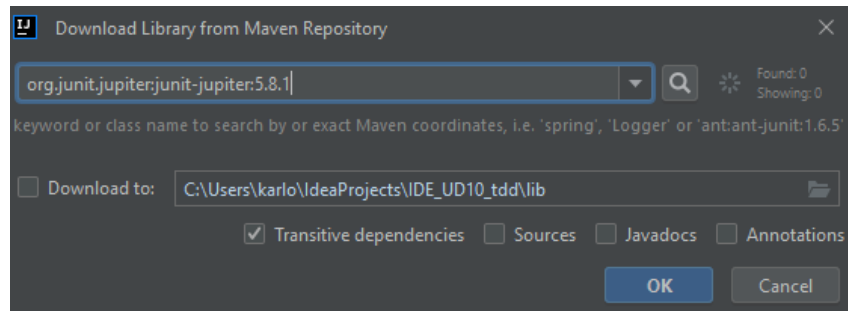


4. Creamos la clase Java TestCoche.



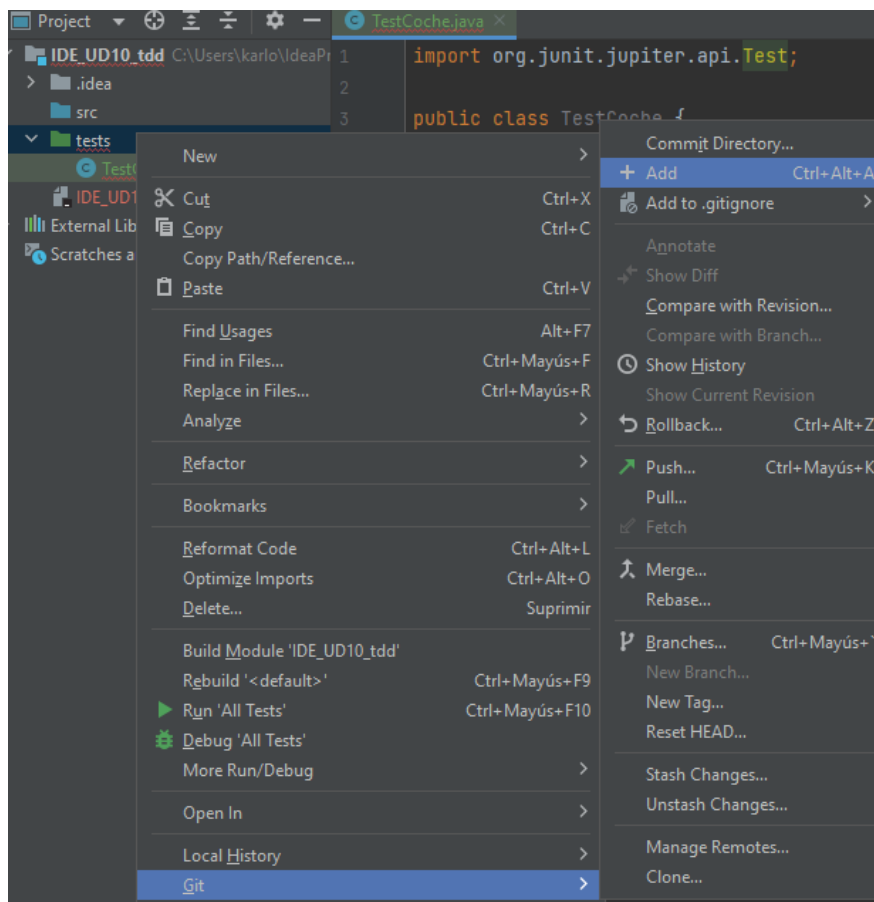
5. Creamos primer test. Escribimos @Test, posicionamos el cursor encima y damos Alt + Enter, añadimos JUnit 5 y se importará el paquete necesario.






```
TestCoche.java x
1 import org.junit.jupiter.api.Test;
2
3 public class TestCoche {
4     @Test
5 }
```

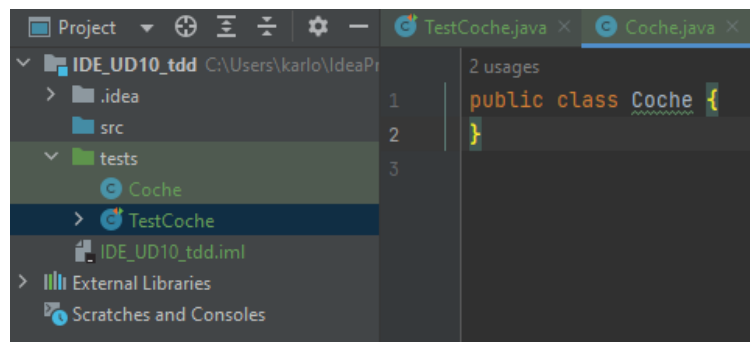
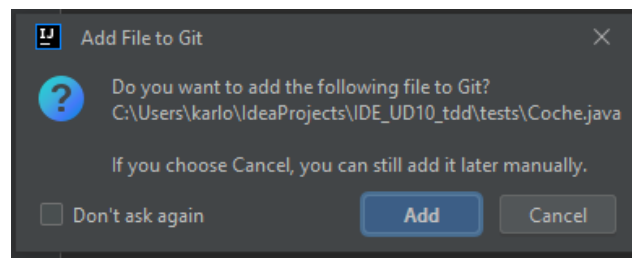
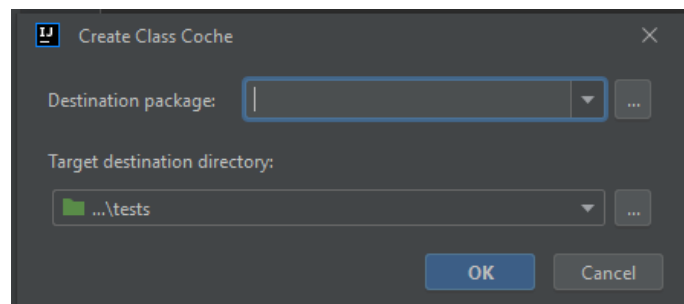
6. El nombre del archivo sale en rojo porque no lo hemos añadido a Git. Lo añadimos de esta forma:



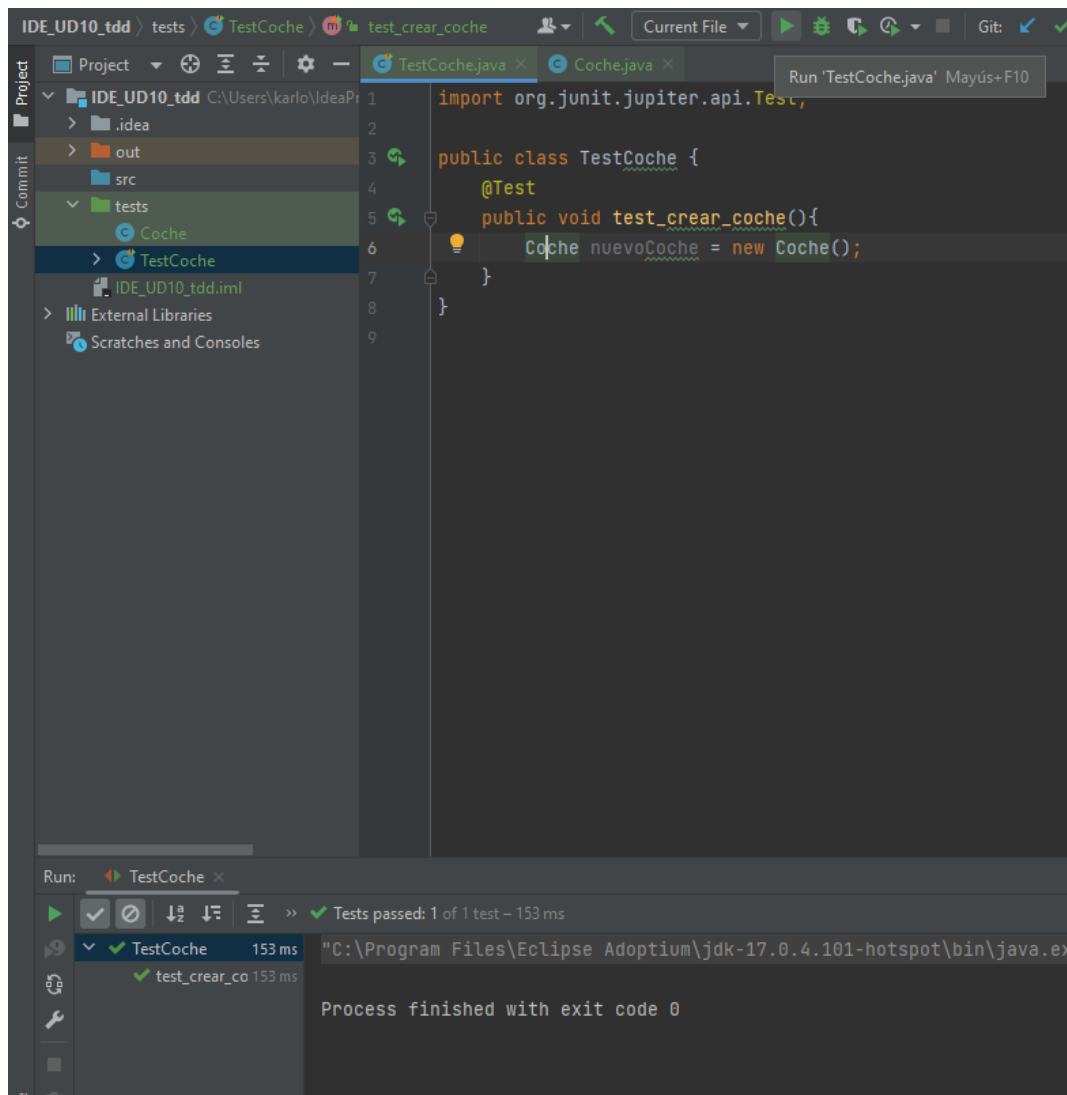
7. Creamos primer test. Posicionamos el cursor sobre "Coche" para crear la clase necesaria.



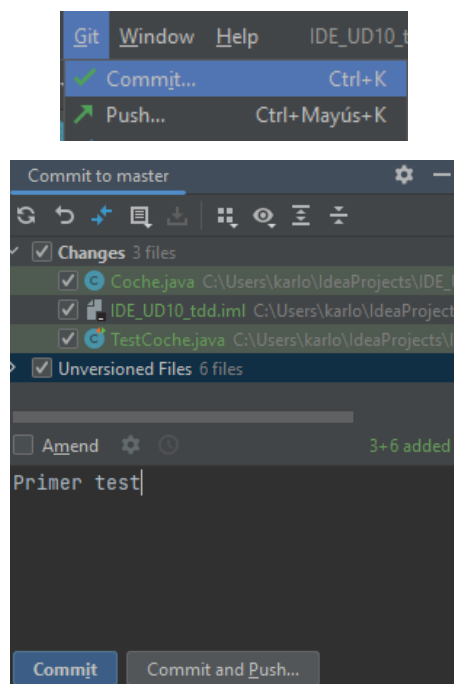
```
1 import org.junit.jupiter.api.Test;
2
3 public class TestCoche {
4     @Test
5     public void test_crear_coche(){
6         Coche nuevoCoche = new Coche();
7     }
8 }
9
```



8. Ejecutamos el primer test con el botón de ejecutar o Mayús + F10.



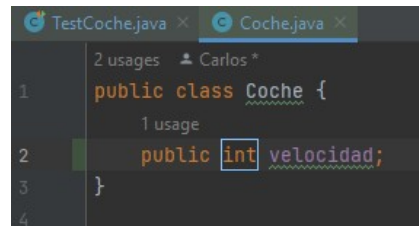
9. Realizamos un commit del estado actual. Seleccionamos todos los archivos.



10. Modificamos el test para hacerlo más complejo. Seleccionamos "velocidad" para crear el atributo en la clase Coche automáticamente.

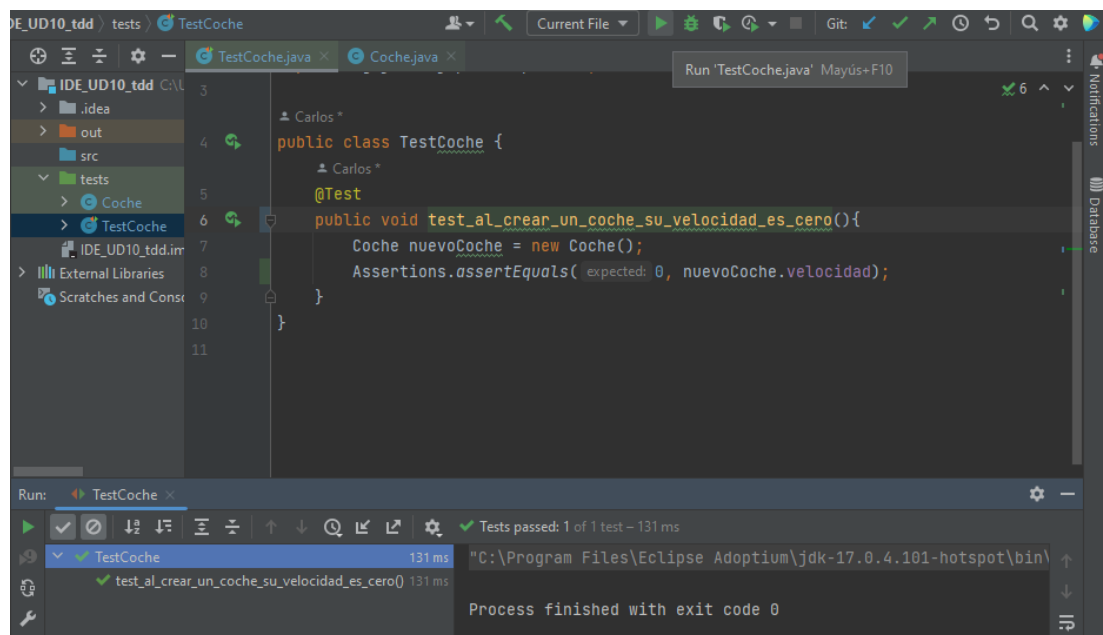


```
1 import org.junit.jupiter.api.Assertions;
2 import org.junit.jupiter.api.Test;
3
4 public class TestCoche {
5     @Test
6     public void test_al_crear_un_coche_su_velocidad_es_cero(){
7         Coche nuevoCoche = new Coche();
8         Assertions.assertEquals( expected: 0, nuevoCoche.velocidad);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



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

11. Ejecutamos el test y lo pasamos correctamente.



```
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
```

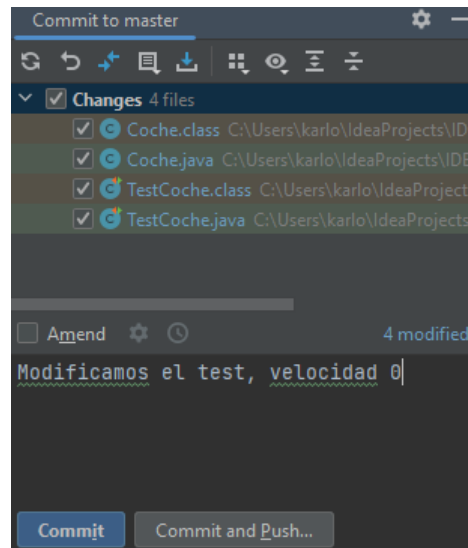
Run: TestCoche x

Tests passed: 1 of 1 test - 131 ms

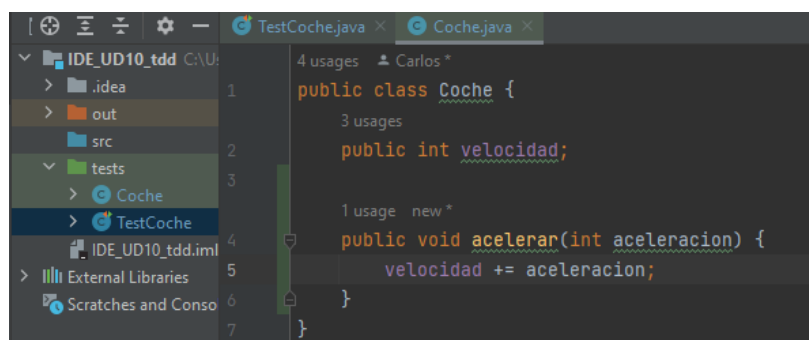
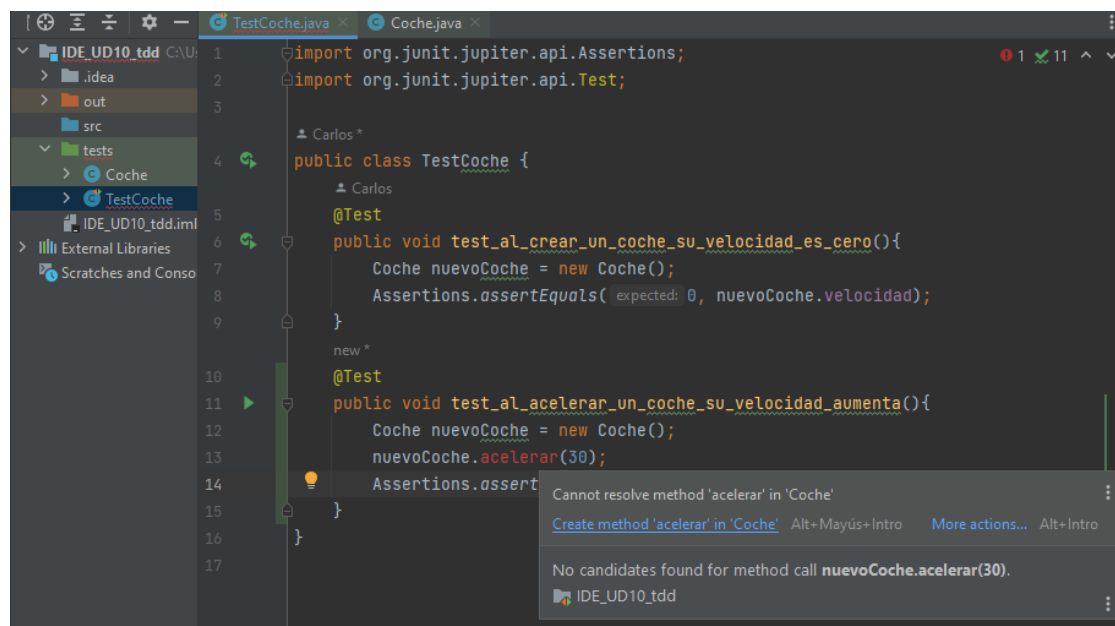
test_al_crear_un_coche_su_velocidad_es_cero() 131 ms

Process finished with exit code 0

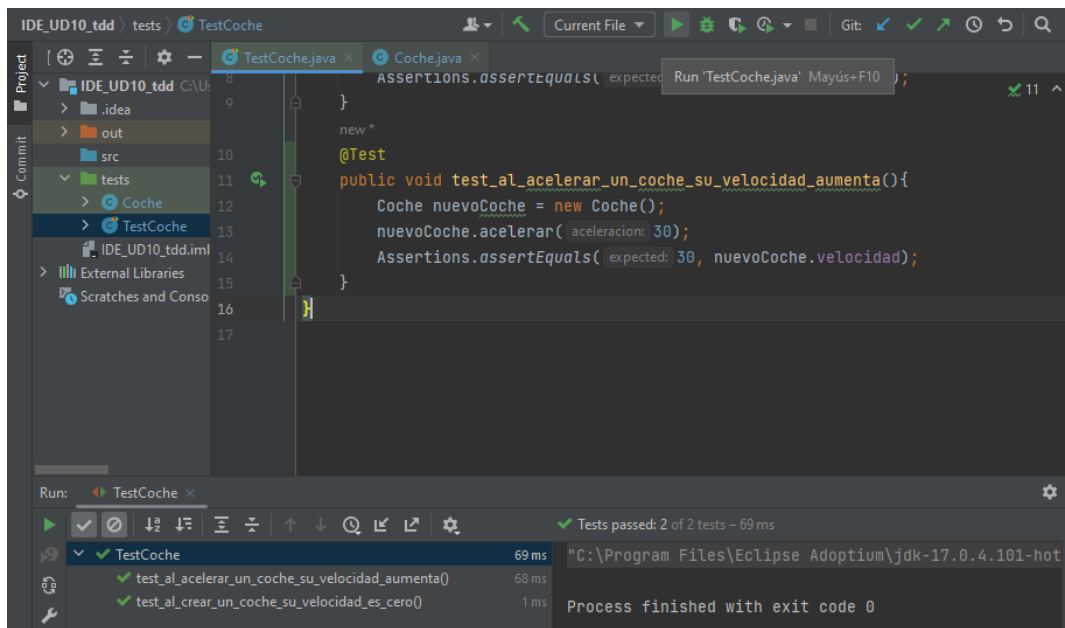
12. Realizamos un commit con los cambios.



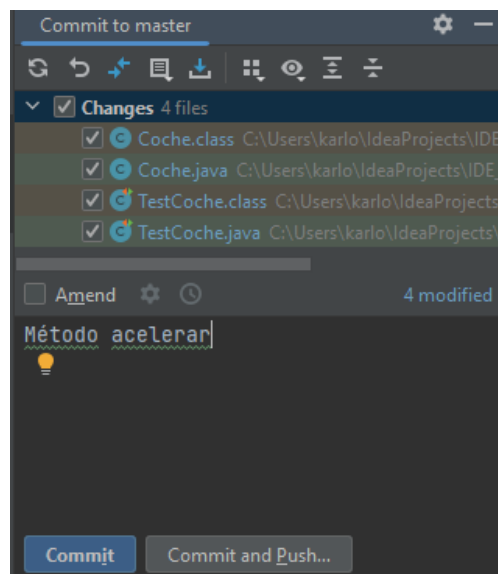
13. Creamos un nuevo método de prueba. Debemos crear el método "acelerar" en la clase Coche para resolver el error.



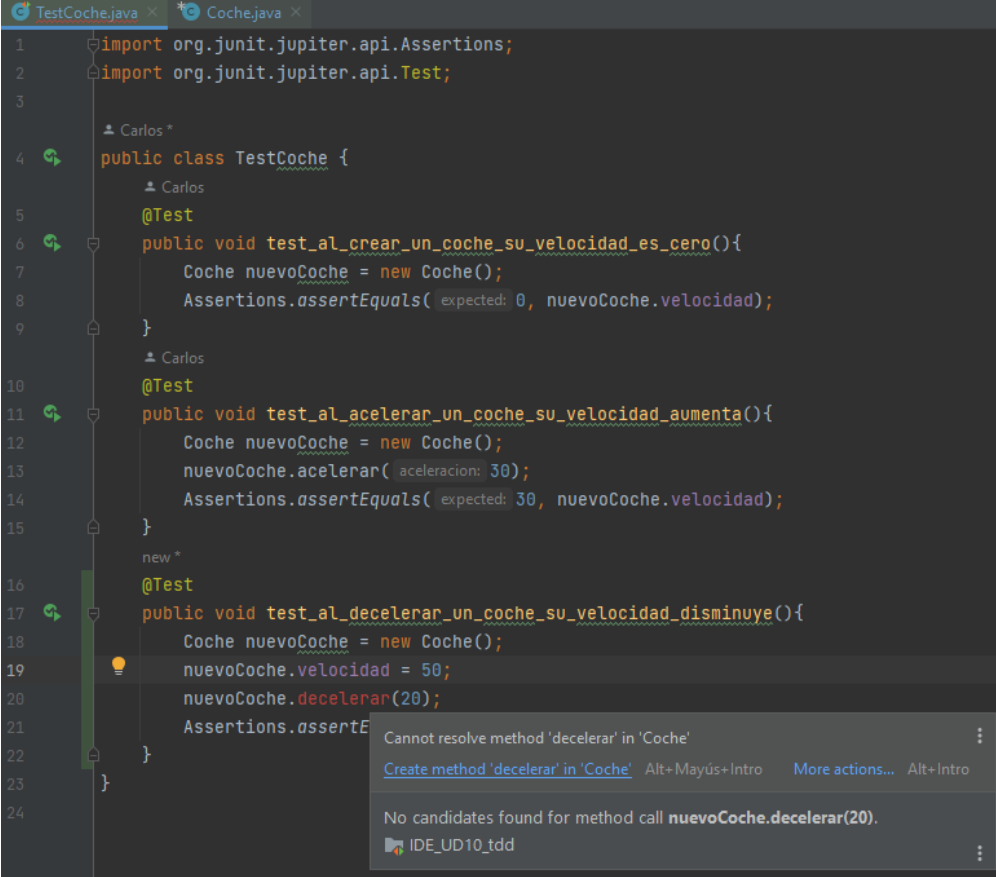
14. Realizamos un test con ambos métodos.



15. Hacemos un commit con los cambios.



16. Creamos el método decelerar. Al igual que en el método acelerar debemos crearlo en la clase Coche.




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

Cannot resolve method 'decelerar' in 'Coche'

[Create method 'decelerar' in 'Coche'](#) Alt+Mayús+Intro More actions... Alt+Intro

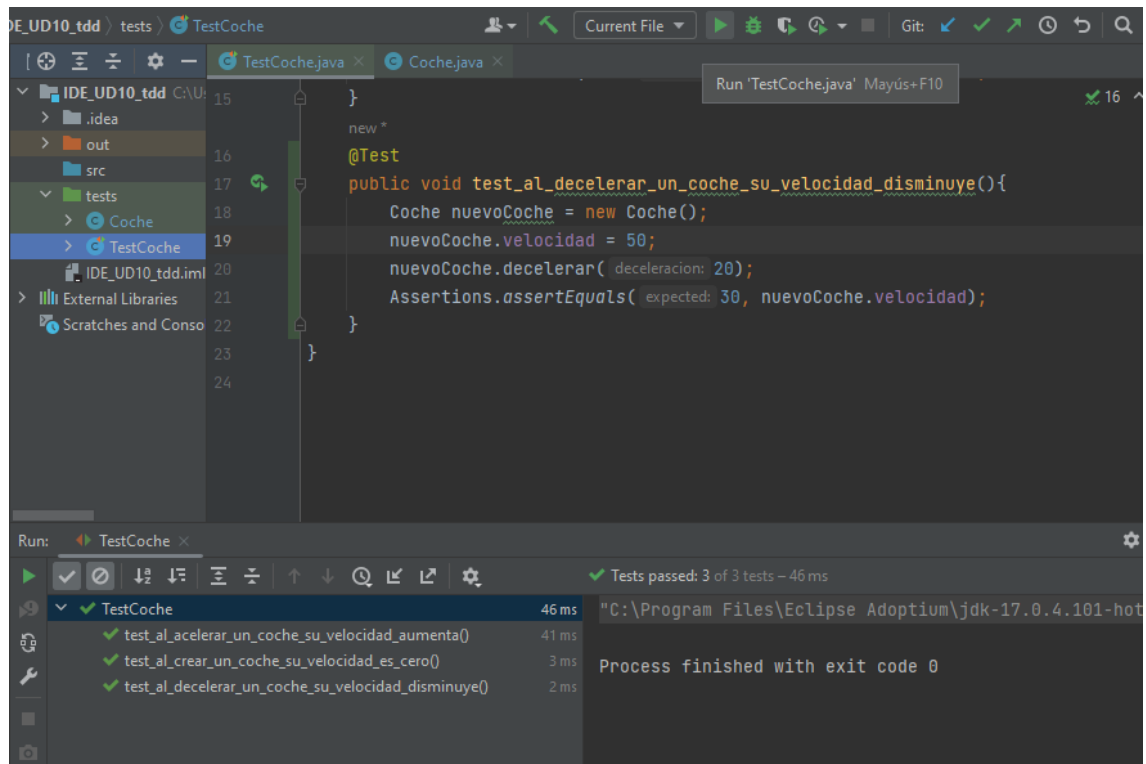
No candidates found for method call `nuevoCoche.decelerar(20)`.

IDE_UD10_tdd

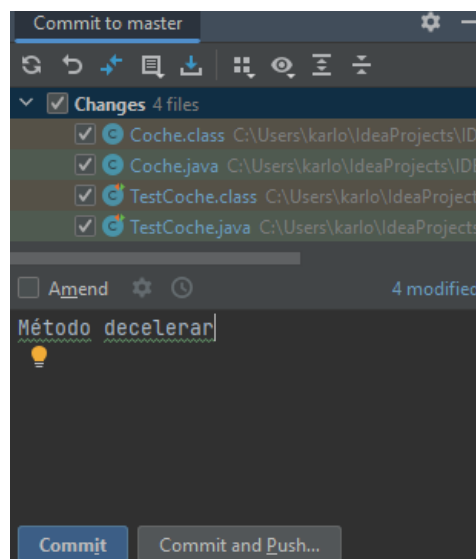


```
6 usages Carlos *
1 public class Coche {
2     6 usages
3     public int velocidad;
4
5     1 usage Carlos
6     public void acelerar(int aceleracion) {
7         velocidad += aceleracion;
8     }
9
10    1 usage new *
11    public void decelerar(int deceleracion) {
12        velocidad -= deceleracion;
13    }
14 }
```

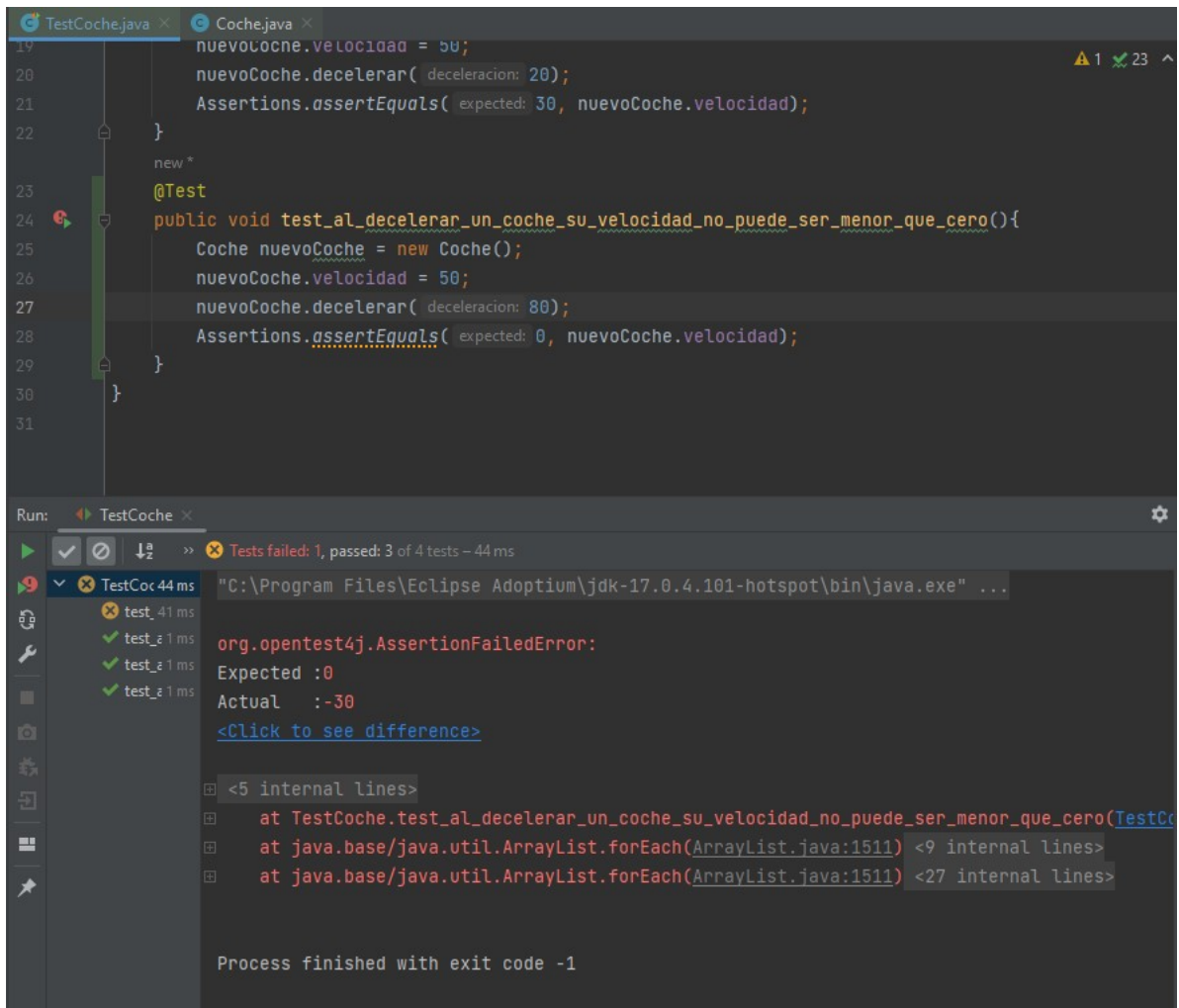
17. Pasamos el test con éxito.



18. Hacemos el commit.



19. Creamos otro método el cual dará error al ejecutar. Se esperaba 0 pero el valor resultante es -30.



The screenshot shows an IDE with two tabs: `TestCoche.java` and `Coche.java`. The `TestCoche.java` tab is active, showing a test method `test_al_decelerar_un_coche_su_velocidad_no_puede_ser_menor_que_cero()` that expects the velocity to be 0 after decelerating by 80. The `Coche.java` tab shows a `decelerar` method that subtracts the deceleration value from the current velocity. The Run window at the bottom shows that the test failed with the message: `org.opentest4j.AssertionFailedError: Expected :0 Actual :-30`. The stack trace indicates the failure occurred in the `test_al_decelerar_un_coche_su_velocidad_no_puede_ser_menor_que_cero` method.

```
19
20
21
22
23
24
25
26
27
28
29
30
31

nuevoCoche.velocidad = 50;
nuevoCoche.decelerar( deceleracion: 20);
Assertions.assertEquals( expected: 30, nuevoCoche.velocidad);
}

@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( deceleracion: 80);
    Assertions.assertEquals( expected: 0, nuevoCoche.velocidad);
}

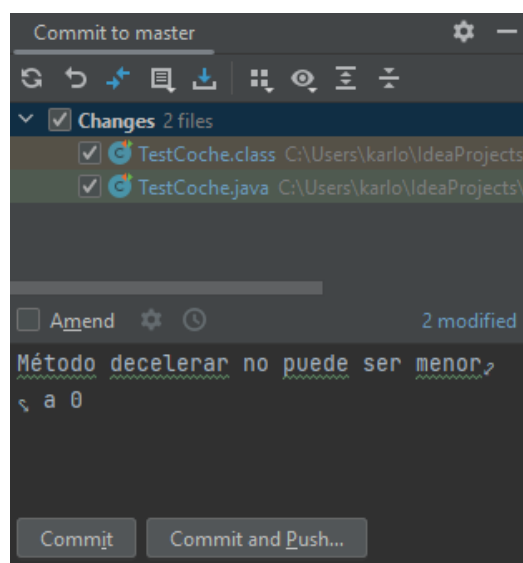
Run: TestCoche
Tests failed: 1, passed: 3 of 4 tests - 44 ms
TestCoc 44 ms
test_ 41 ms
test_ 1 ms
test_ 1 ms
test_ 1 ms

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(TestCo
    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 -1
```

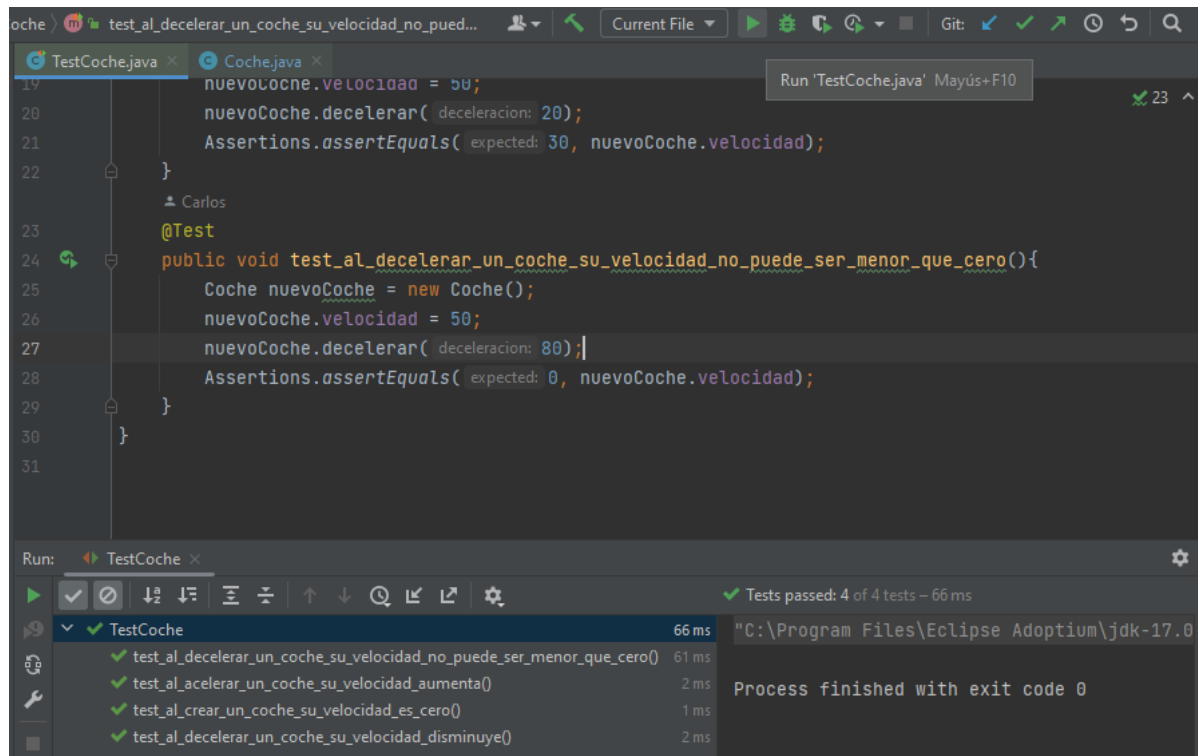
20. Commit del nuevo método.



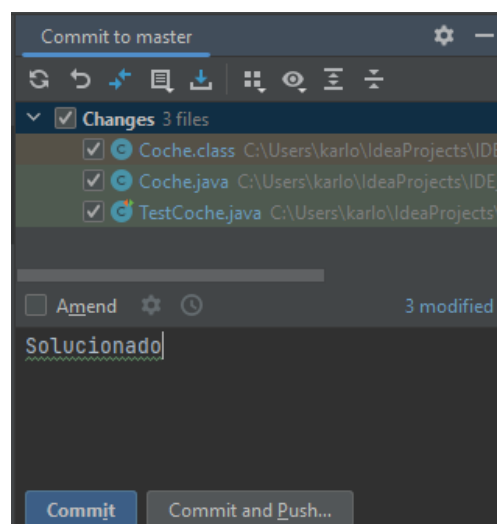
21. Corregimos en clase Coche para evitar errores.

```
2 usages Carlos *
8      public void decelerar(int deceleracion) {
9          velocidad -= deceleracion;
10         if (velocidad < 0) velocidad = 0;
11     }
12 }
13
```

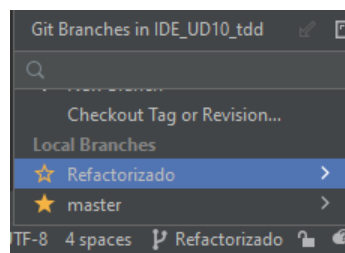
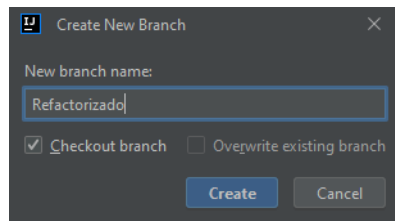
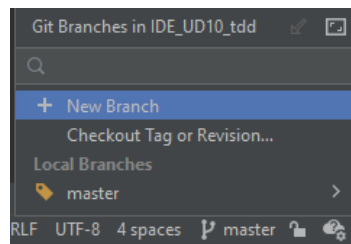
22. Test correcto.



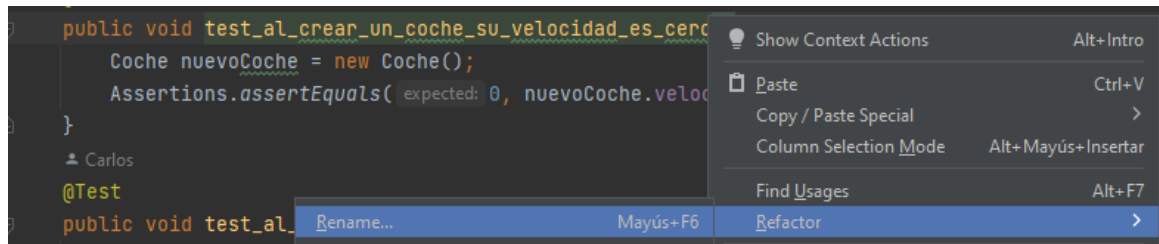
23. Nuevo commit.



24. Creamos la rama Refactorizado y nos situamos en ella.

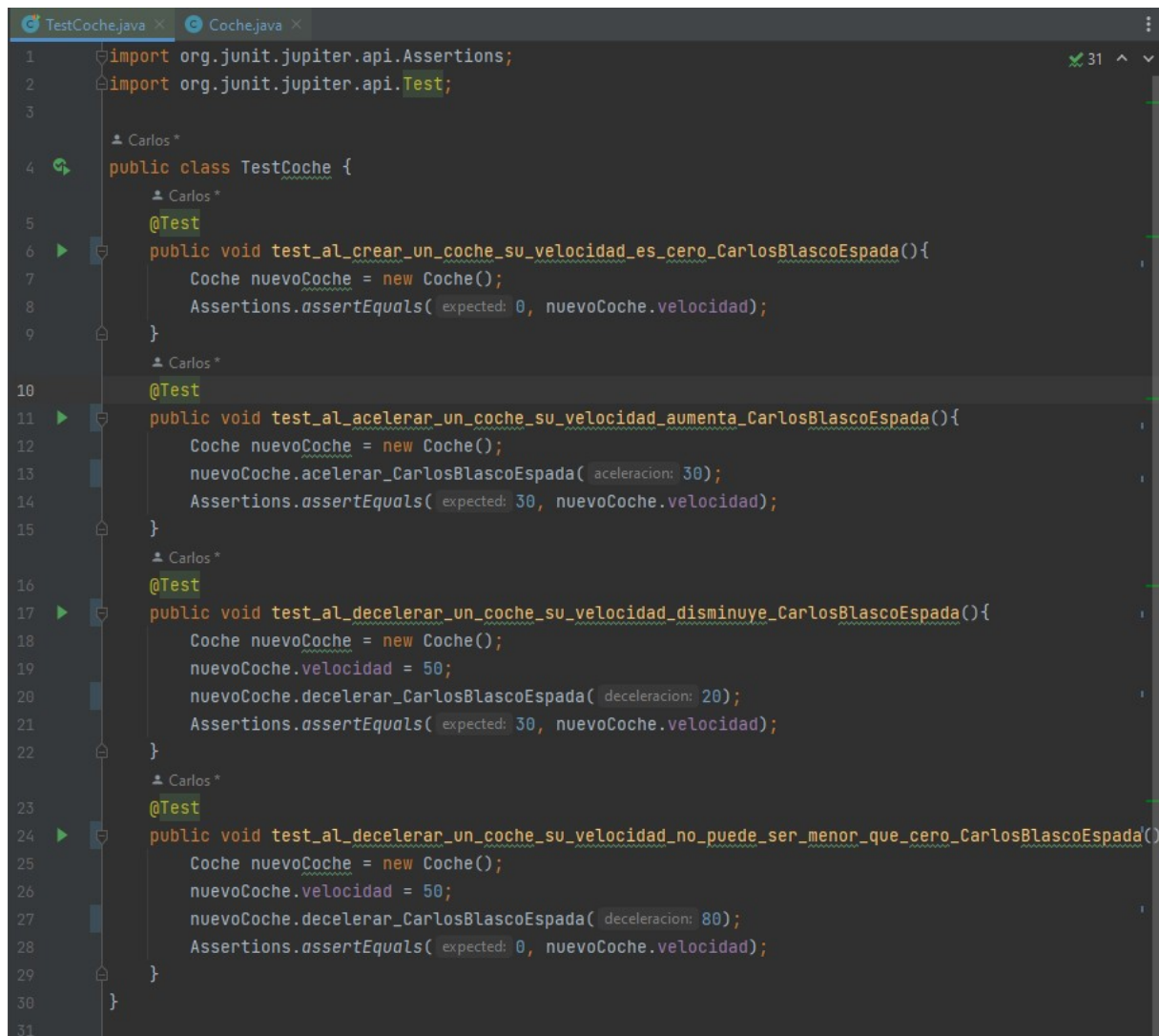


25. Refactorizamos de la siguiente manera:



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

@Test
public void test_al_
```



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

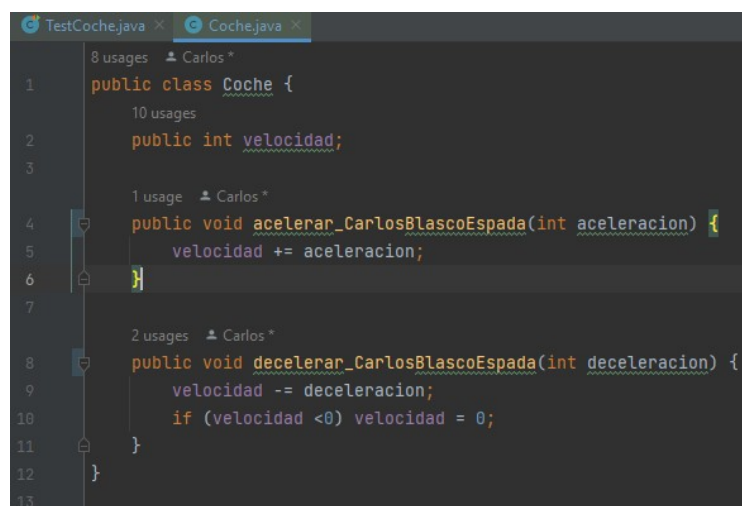
public class TestCoche {

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

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

    @Test
    public void test_al_decelerar_un_coche_su_velocidad_disminuye_CarlosBlascoEspada(){
        Coche nuevoCoche = new Coche();
        nuevoCoche.velocidad = 50;
        nuevoCoche.decelerar_CarlosBlascoEspada( deceleracion: 20);
        Assertions.assertEquals( expected: 30, nuevoCoche.velocidad);
    }

    @Test
    public void test_al_decelerar_un_coche_su_velocidad_no_puede_ser_menor_que_cero_CarlosBlascoEspada(){
        Coche nuevoCoche = new Coche();
        nuevoCoche.velocidad = 50;
        nuevoCoche.decelerar_CarlosBlascoEspada( deceleracion: 80);
        Assertions.assertEquals( expected: 0, nuevoCoche.velocidad);
    }
}
```

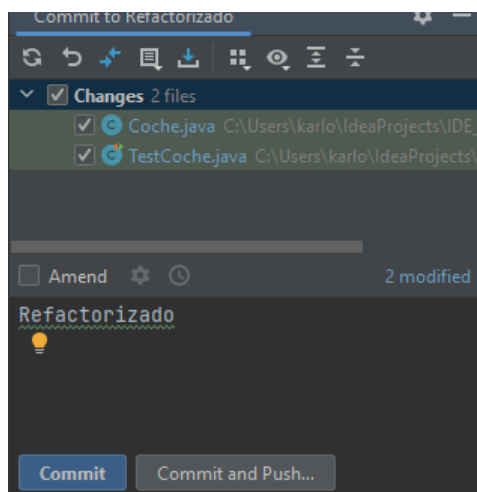


```
public class Coche {
    public int velocidad;

    public void acelerar_CarlosBlascoEspada(int aceleracion) {
        velocidad += aceleracion;
    }

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

26. Realizamos un commit con los cambios.



27. Siguiendo los mismos pasos creamos una nueva rama llamada Memoria, en la cual subiremos este pdf.