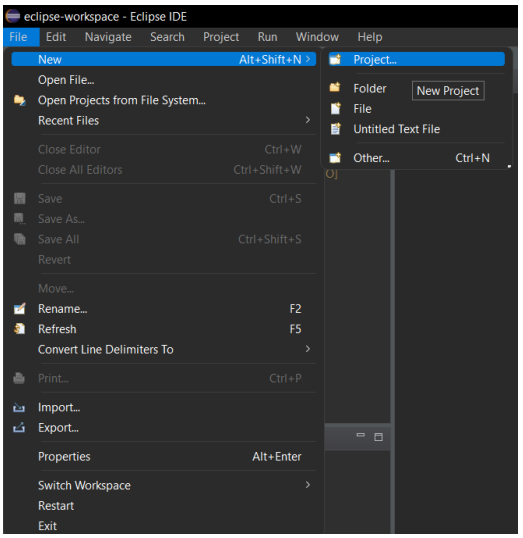
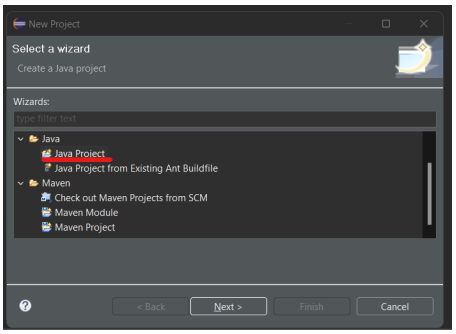
Mi Primer TDD en Eclipse

Fran Molina Adan

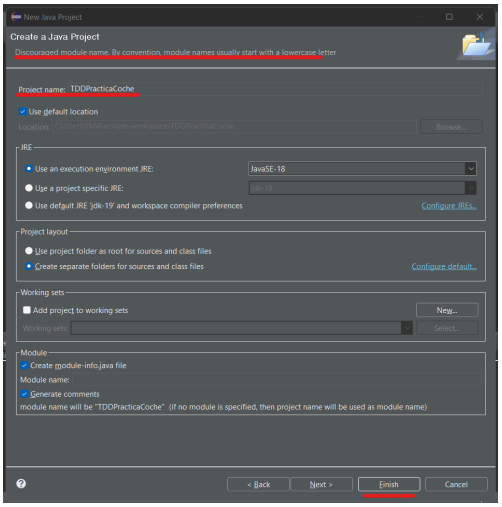
Lo primero que tenemos que hacer es crear nuestro proyecto.
Para esto damos en “File/New/Project…”



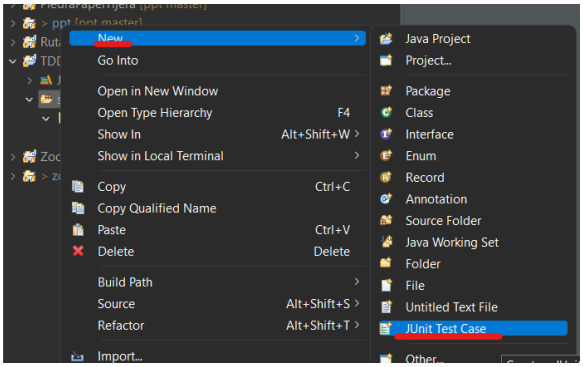
En la ventana emergente seleccionamos “Java project” y siguiente.



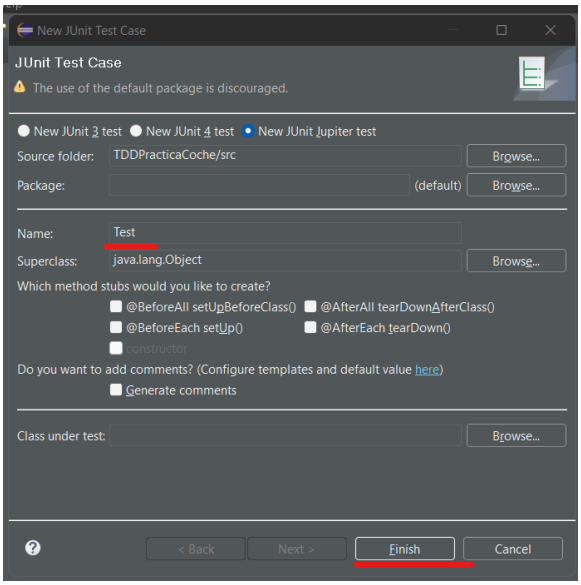
En la siguiente ventana insertamos el nombre de nuestro proyecto, es importante que este no contenga espacios y además es recomendable que empiece en minúsculas pero yo lo escribiré como en el proyecto anterior para identificarlo más fácilmente.



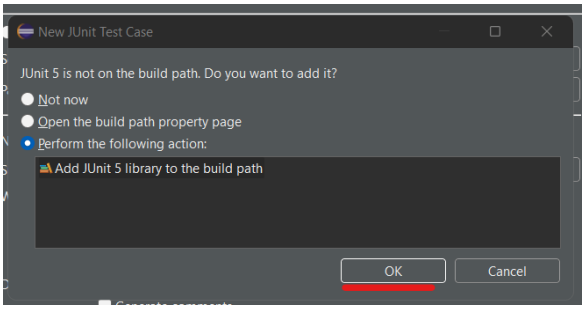
Ahora vamos a crear un archivo JUnit dentro de nuestra carpeta src para los test.
Para esto clic derecho sobre src, new y JUnit.



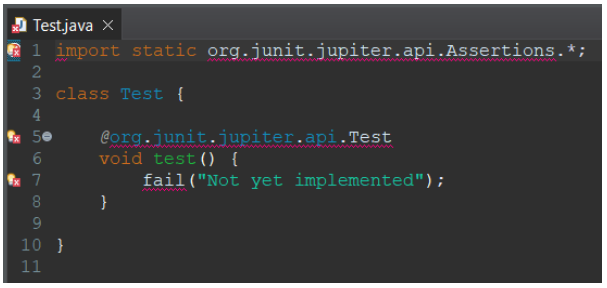
En la ventana emergente ponemos el nombre y finish.



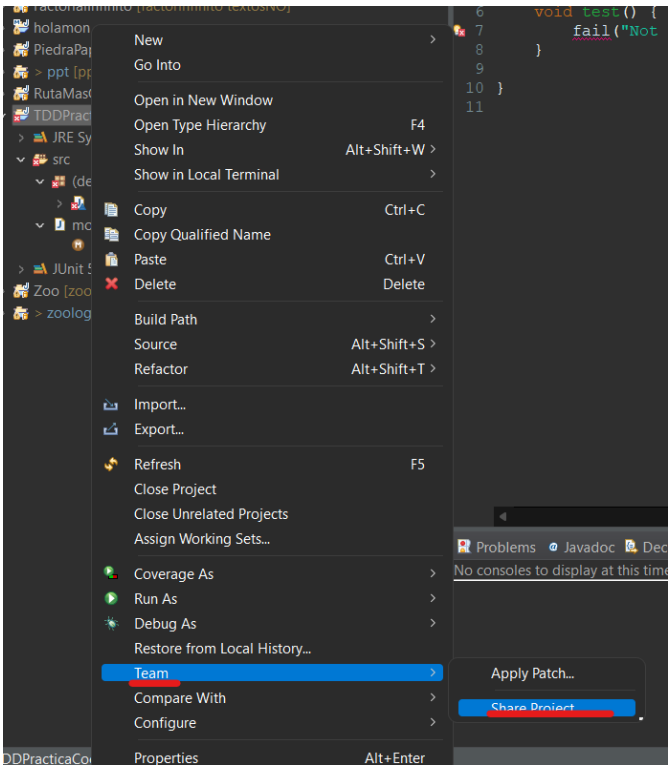
En la siguiente ventana damos en OK.



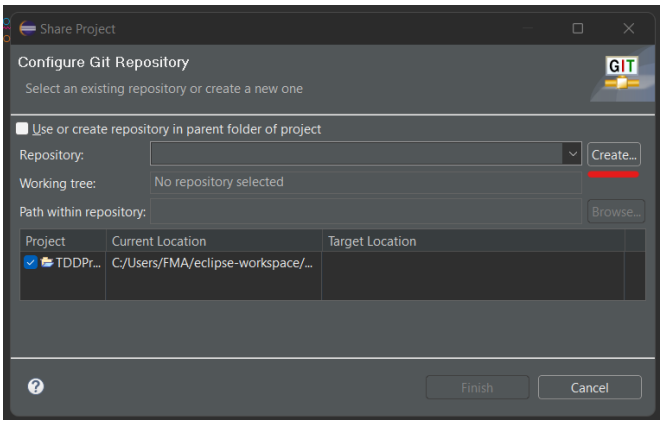
Debes tener algo así.



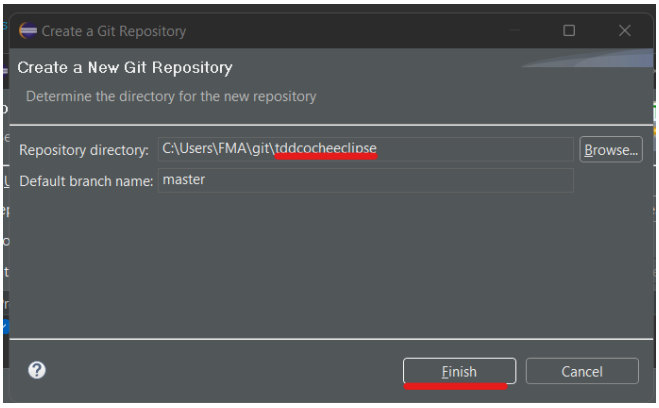
Vamos a compartir nuestro proyecto en git antes de empezar con los test.
Para esto clic derecho en el proyecto, teamy y share project.



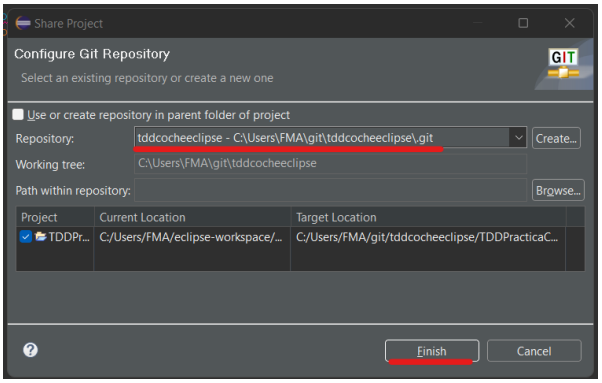
En la ventana emergente click en create.



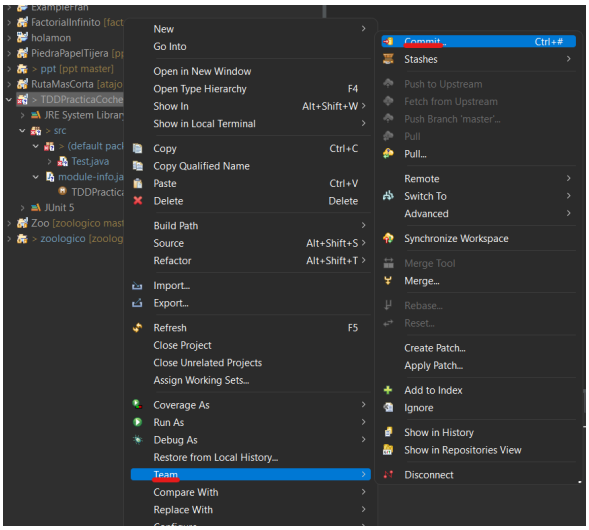
En la ventana emergente colocamos el nombre del proyecto y finish.



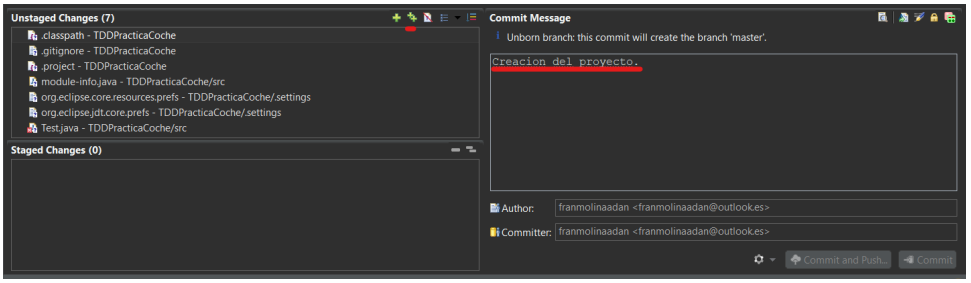
Se cerrara esta ultima y en la anterior se habran completado los campos del nombre, clic en finish.



Ahora clic derecho en el proyecto, team y commit.

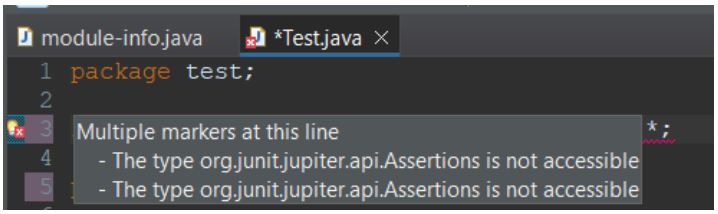


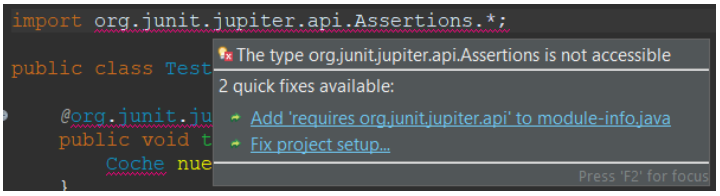
Ahora damos clic en los dos “+” verdes y añadimos el mensaje de commit.



Listo.

Ahora vamo a corregir errores al importar JUnit, para esto damos a la bombilla y escogemos el fix que nos propone eclipse.

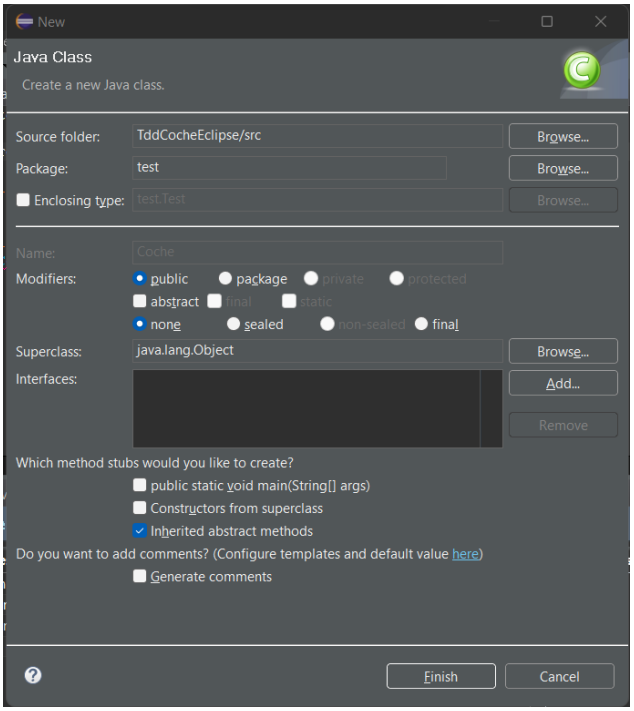




Ahora vamos a crear el siguiente @Test.

```java
package test;

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

public class Test {

    @org.junit.jupiter.api.Test
    public void test_al_crear_un_coche(){
        Coche nuevoCoche = new Coche();
    }

}
```

Nos colocamos encima de Coche y seleccionamos la corrección que nos sugiere.



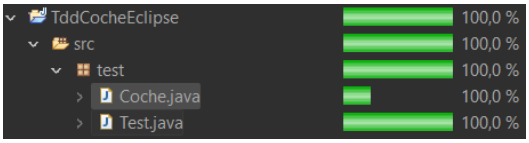
Ahora modificamos la clase test tal que asi.

```java
package test;

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

public class Test {

    @org.junit.jupiter.api.Test
    public void test_al_crear_un_coche_su_velocidad_es_cero(){
        Coche nuevoCoche = new Coche();
        Assertions.assertEquals(0, nuevoCoche.velocidad);
    }
```

Y escogemos la primera solución que nos ofrece velocidad.

```java
package test;

public class Coche {

    public Integer velocidad;

}
```

Vamos a probar a hacer coverage, dando clic derecho sobre el proyecto y coverage, debería pasarlo sin problemas.



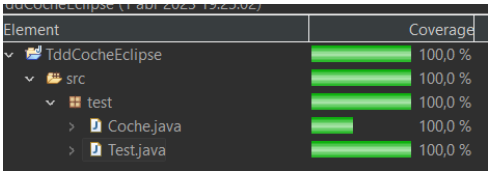
Ahora creamos otro test de la siguiente forma.

```
@org.junit.jupiter.api.Test
public void test_al_crear_un_coche_su_velocidad_es_cero(){
    Coche nuevoCoche = new Coche();
    Assertions.assertEquals(0, nuevoCoche.velocidad);
}
@org.junit.jupiter.api.Test
public void test_al_acelerar_un_coche_su_velocidad_aumenta(){
    Coche nuevoCoche = new Coche();
    nuevoCoche.acelerar(30);
    Assertions.assertEquals(30, nuevoCoche.velocidad);
}
```

Nos colocamos sobre acelerar y elegimos la primera corrección.
Nos crearon el siguiente método, lo modificamos tal que así.

```
package test;

public class Coche {

    public int velocidad;

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

}
```

Probamos otro coverage,todo debería ir bien.



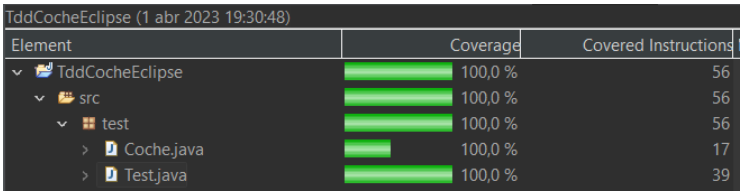
Creamos otro test y lo modificamos de esta forma.

```
@org.junit.jupiter.api.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);
}
```

Nos colocamos sobre decelerar y elegimos la corrección que crea el método decelerar.
Modificamos el método de esta manera.

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

}
```

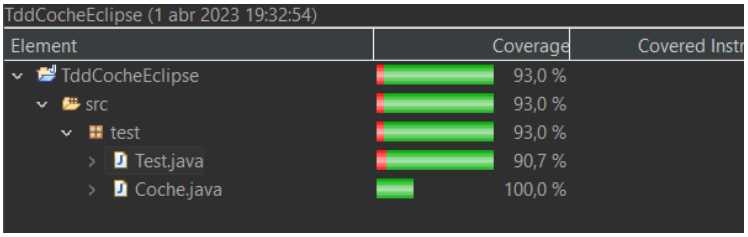
Si hacemos un coverage todo debería ir bien.



Vamos a crear otro test de la siguiente forma.

```java
@org.junit.jupiter.api.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(80);
    Assertions.assertEquals(0, nuevoCoche.velocidad);
}
```

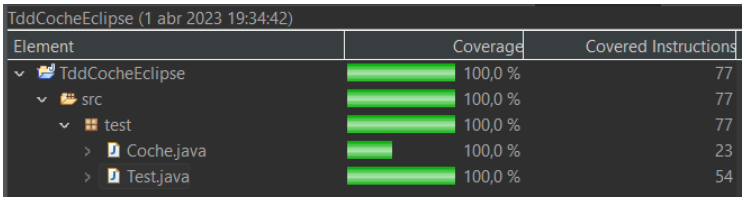
En este caso si hacemos un coverage tendremos un fallo.



Vamos a solucionarlo, para esto nos vamos al método decelerar y le añadimos la siguiente condición.

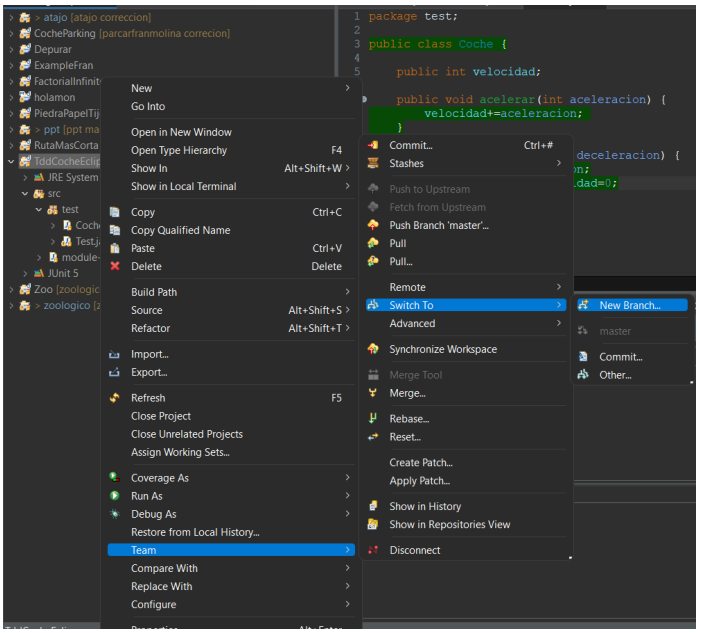
```java
public void decelerar(int deceleracion) {
    velocidad-=deceleracion;
    if(velocidad<0) velocidad=0;
}
```

Probamos de nuevo.

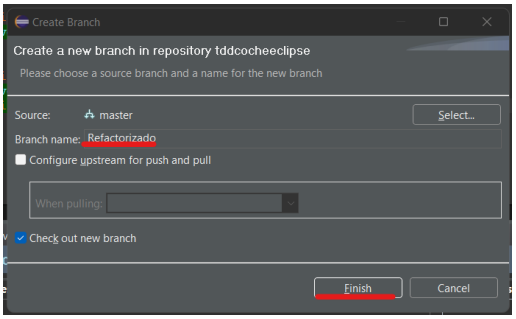


Ahora vamos a subir todo al repositorio.

Vamos a crear una nueva rama, para esto:



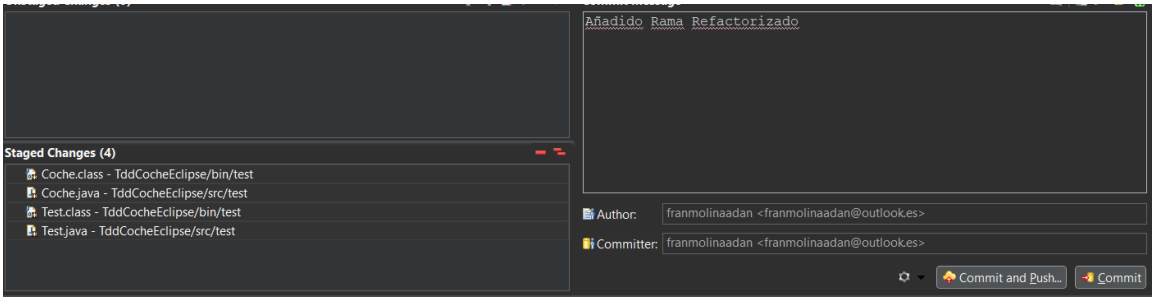
En la ventana emergente ponemos el nombre de la nueva rama.



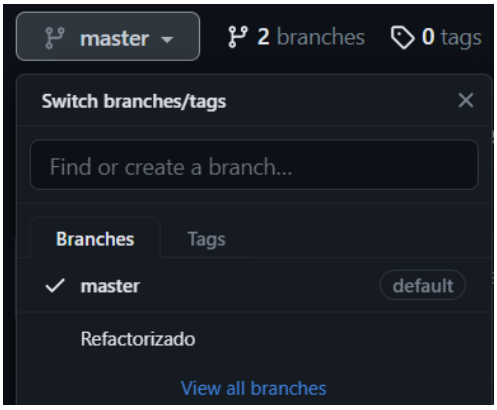
Ahora voy a cambiar el nombre de todos los métodos añadiendo mis iniciales.

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

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

```
@org.junit.jupiter.api.Test
public void test_al_crear_un_coche_su_velocidad_es_cero_FMA(){
    Coche nuevoCoche = new Coche();
    Assertions.assertEquals(0, nuevoCoche.velocidad);
}
@org.junit.jupiter.api.Test
public void test_al_acelerar_un_coche_su_velocidad_aumenta_FMA(){
    Coche nuevoCoche = new Coche();
    nuevoCoche.acelerarFMA(30);
    Assertions.assertEquals(30, nuevoCoche.velocidad);
}
@org.junit.jupiter.api.Test
public void test_al_decelerar_un_coche_su_velocidad_disminuye_FMA(){
    Coche nuevoCoche = new Coche();
    nuevoCoche.velocidad=50;
    nuevoCoche.decelerarFMA(20);
    Assertions.assertEquals(30, nuevoCoche.velocidad);
}
@org.junit.jupiter.api.Test
public void test_al_decelerar_un_coche_su_velocidad_no_puede_ser_menor_que_cero_FMA(){
    Coche nuevoCoche = new Coche();
    nuevoCoche.velocidad=50;
    nuevoCoche.decelerarFMA(80);
    Assertions.assertEquals(0, nuevoCoche.velocidad);
}
```


Añadido Rama Refactorizado
Staged Changes (4)
Coche.class - TddCocheEclipse/bin/test
Coche.java - TddCocheEclipse/src/test
Test.class - TddCocheEclipse/bin/test
Test.java - TddCocheEclipse/src/test
Author: franmolinaadan <franmolinaadan@outlook.es>
Committer: franmolinaadan <franmolinaadan@outlook.es>
Commit and Push...
Commit

Listo.


master
2 branches
0 tags
Switch branches/tags
Find or create a branch...
Branches
Tags
master
default
Refactorizado
View all branches