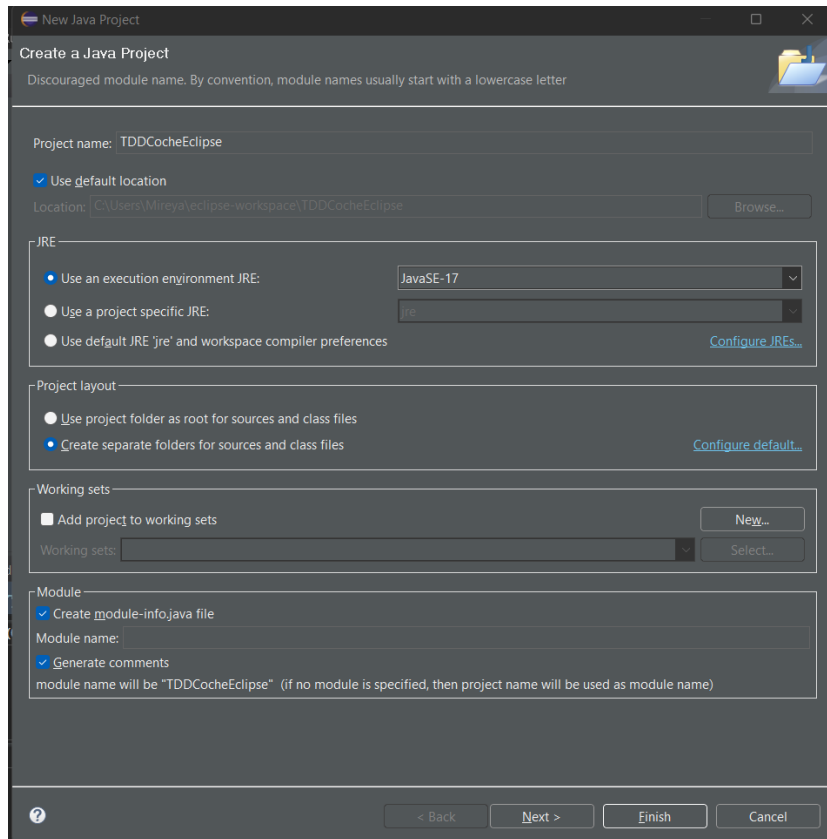
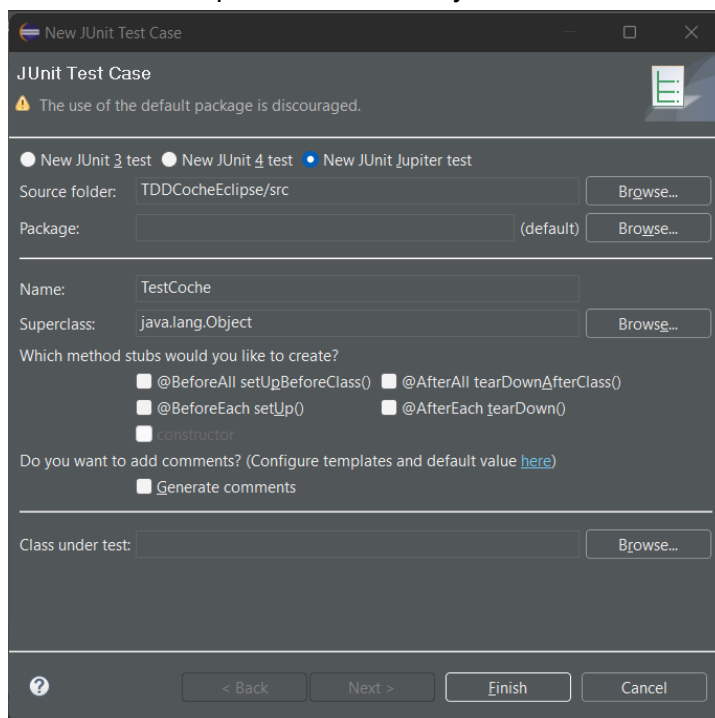
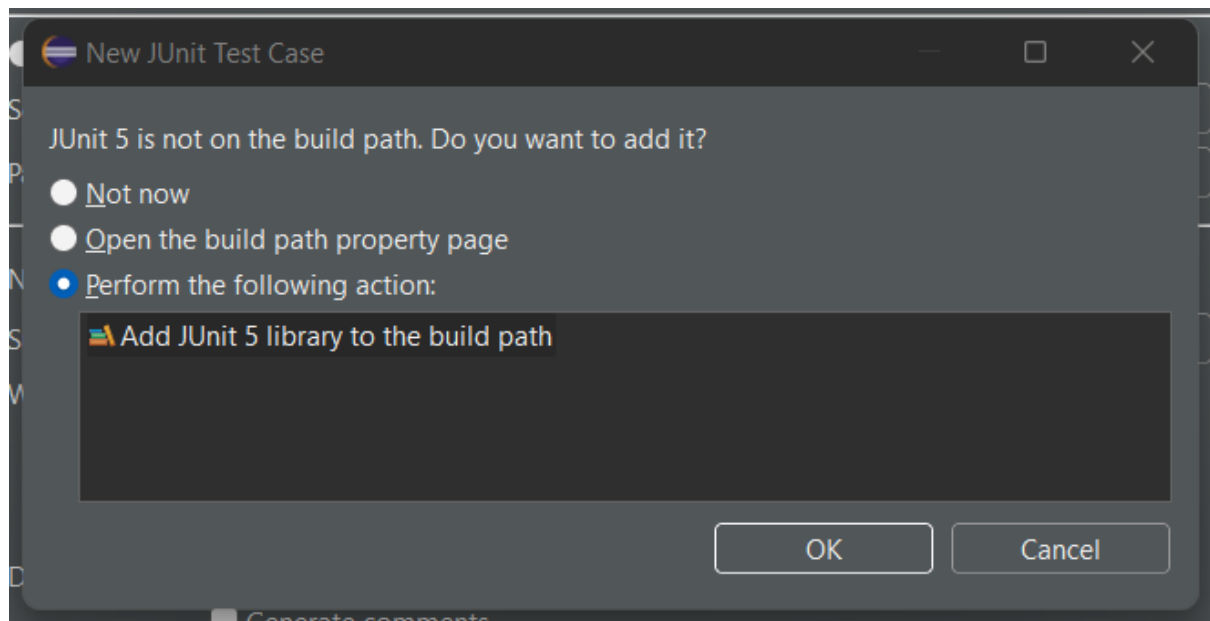


Para empezar creamos un nuevo proyecto en Eclipse



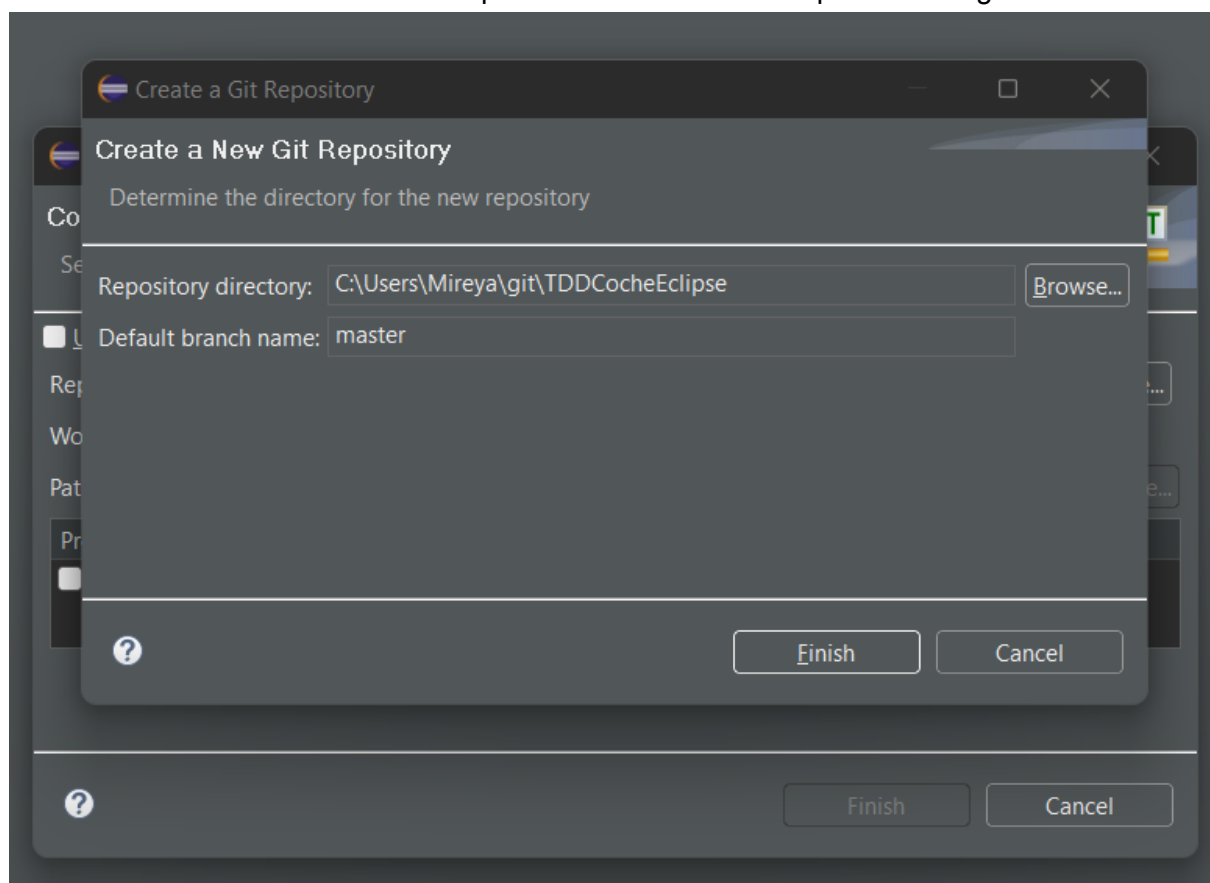
A continuación vamos a crear una clase Test de JUnit, haciendo click derecho en la carpeta src de nuestro proyecto, nos aparecerá la pestaña que está en la captura e a continuación, le ponemos nombre y le damos a finish



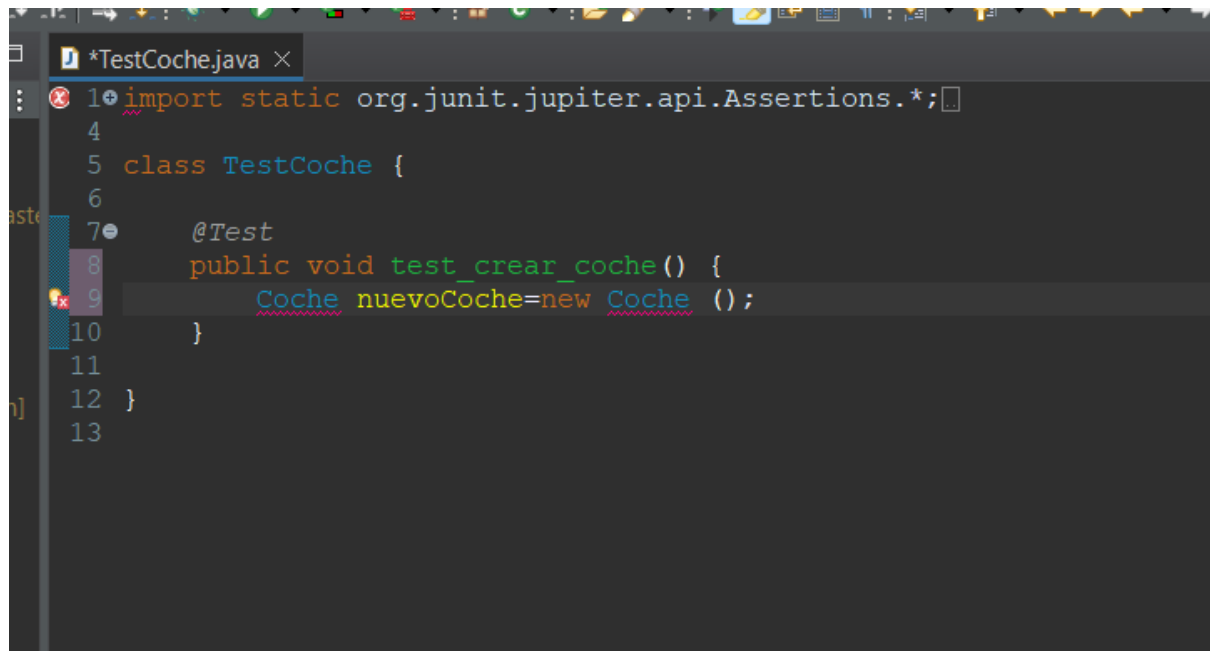


Pinchamos en ok

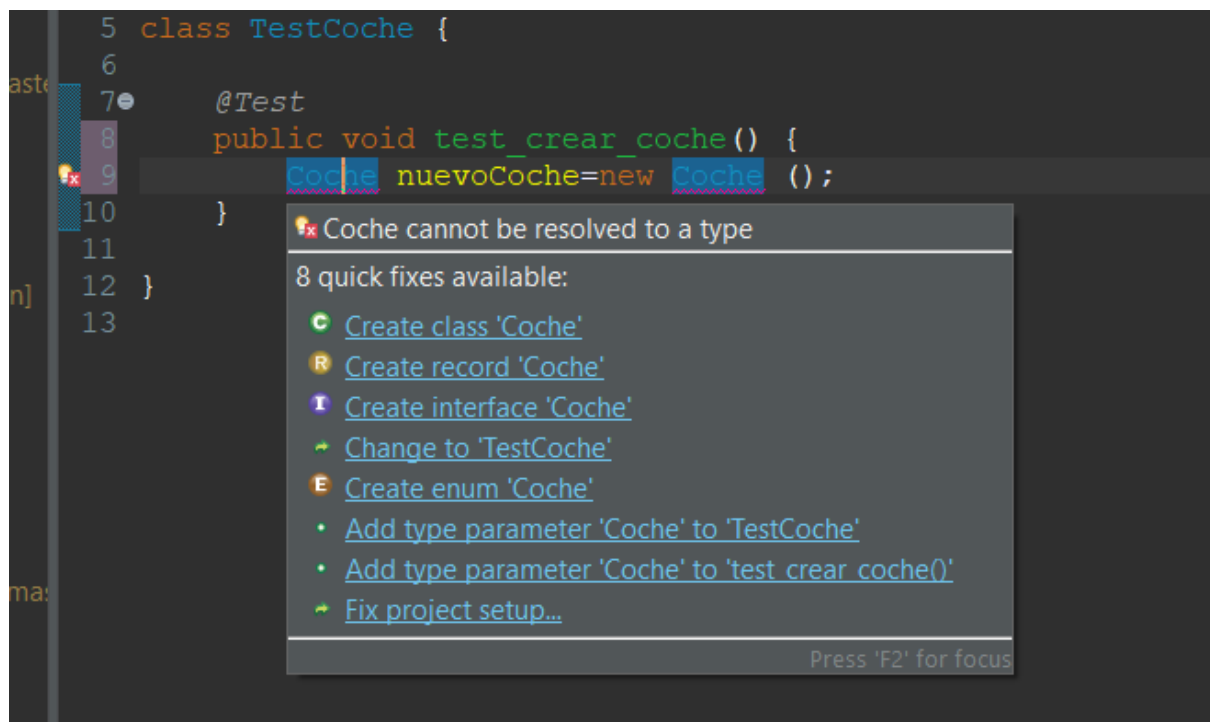
Para ir haciendo commits de todo el proceso iniciaremos un repositorio de git



Volvemos a la clase que habíamos creado de JUnit y escribimos el primer test donde pedimos crear un objeto Coche aunque no tenemos la clase creada, por eso aparecerá en rojo



```
1 import static org.junit.jupiter.api.Assertions.*;
4
5 class TestCoche {
6
7     @Test
8     public void test_crear_coche() {
9         Coche nuevoCoche=new Coche ();
10    }
11
12 }
13
```



```
5 class TestCoche {
6
7     @Test
8     public void test_crear_coche() {
9         Coche nuevoCoche=new Coche ();
10    }
11
12 }
13
```

Coche cannot be resolved to a type


8 quick fixes available:

- Create class 'Coche'
- Create record 'Coche'
- Create interface 'Coche'
- Change to 'TestCoche'
- Create enum 'Coche'
- Add type parameter 'Coche' to 'TestCoche'
- Add type parameter 'Coche' to 'test crear coche()'
- Fix project setup...

Press 'F2' for focus

New

## Java Class

 The use of the default package is discouraged.

Source folder:

Package:

☐ Enclosing type:

---

Name:

Modifiers:

☒ public ☐ package ☐ private ☐ protected

☐ abstract ☐ final ☐ static

☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass:

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

☒ Inherited abstract methods

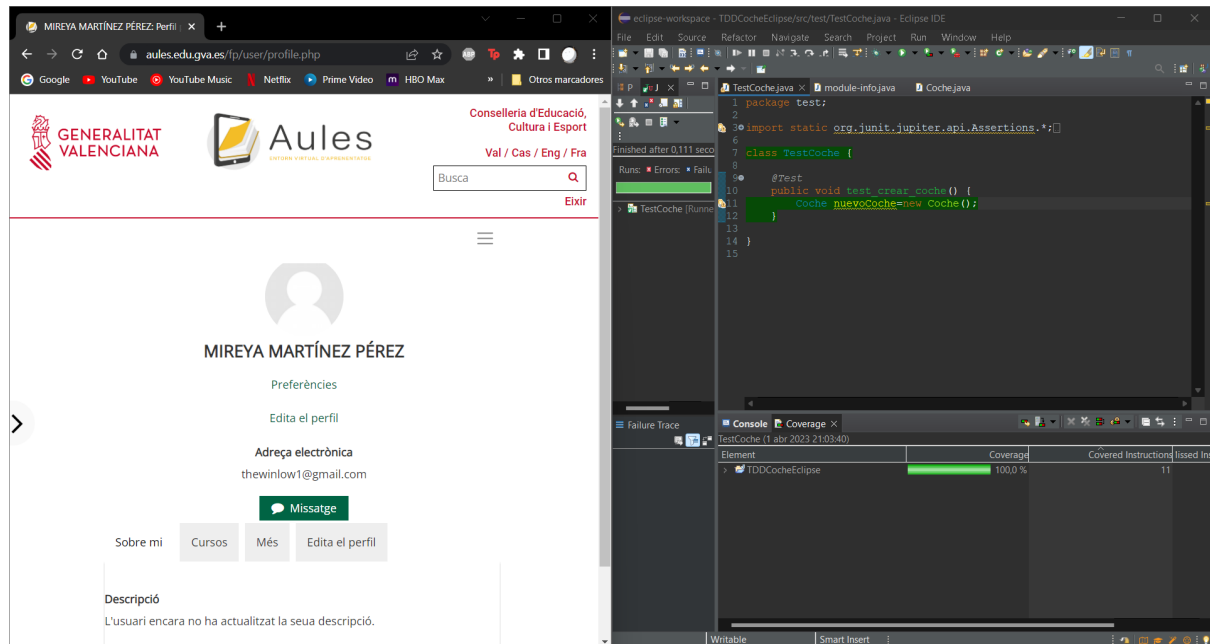
Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

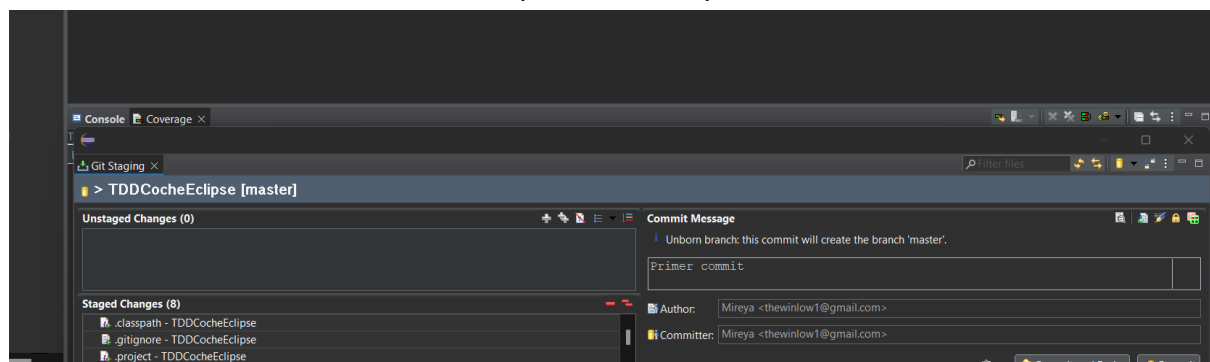
\*Testcoche.java Coche.java ×

```
1
2 public class Coche {
3
4 }
5
```

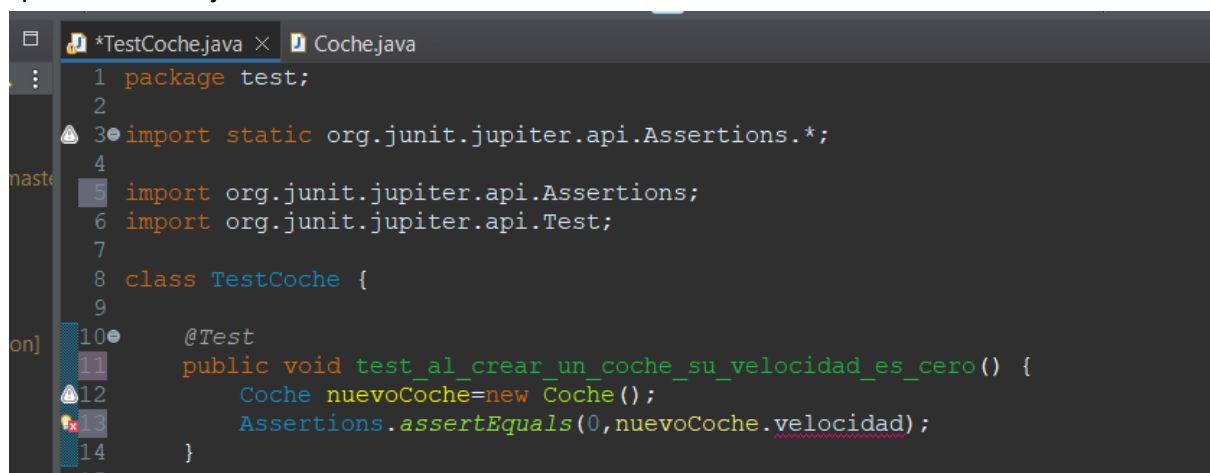
Hacemos un coverage y vemos que va perfectamente



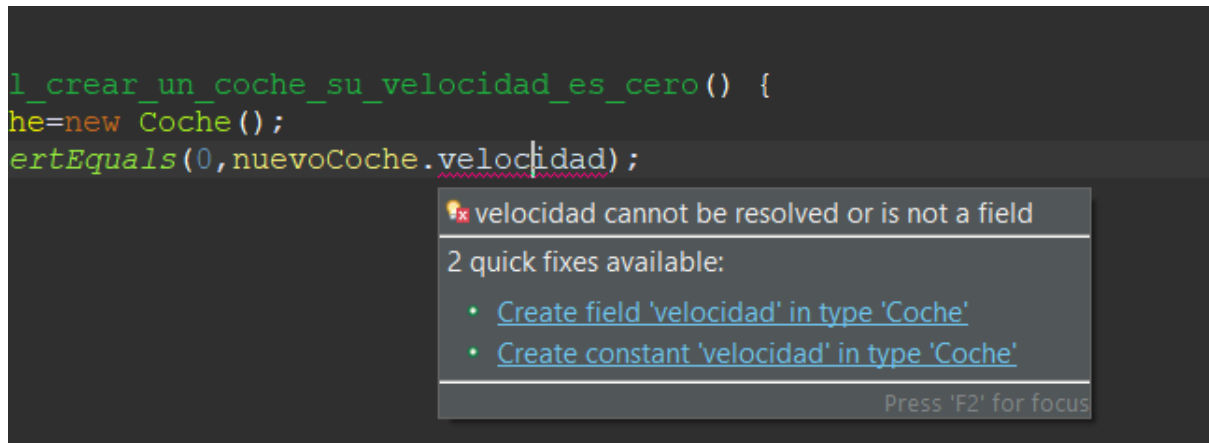
Mientras vamos haciendo los commits que creamos oportunos



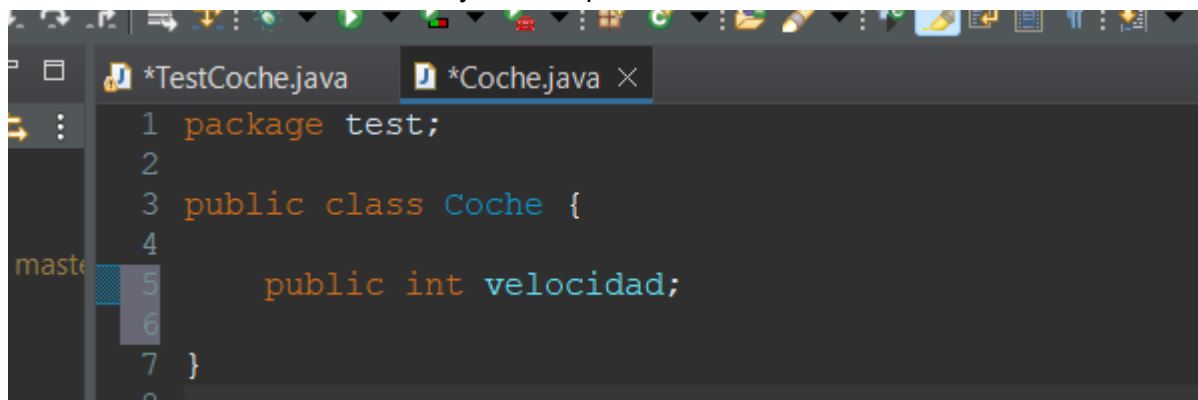
A continuación cambiamos el test y hacemos este en el que le indicamos que cuando se crea un coche su velocidad es 0, puesto que no tenemos la variable velocidad creada nos aparecerá en rojo



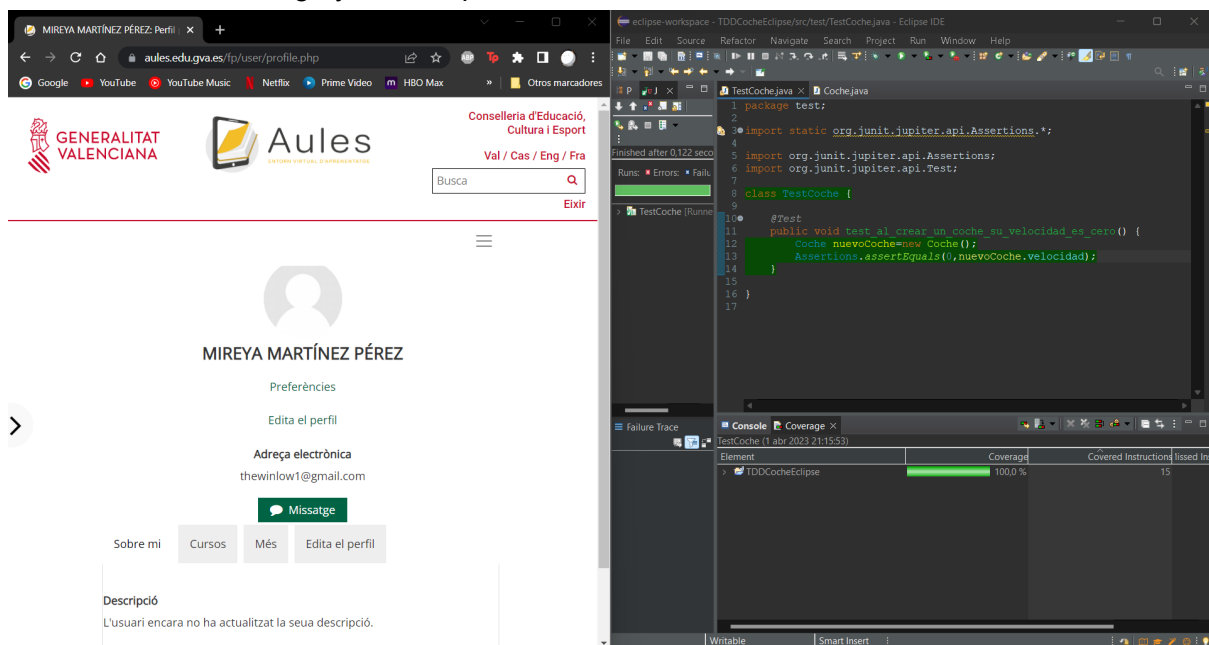
Nos colocamos encima de velocidad y pinchamos en la primera opción que nos aparece



Ahora nos vamos a la clase coche y vemos que se ha creado

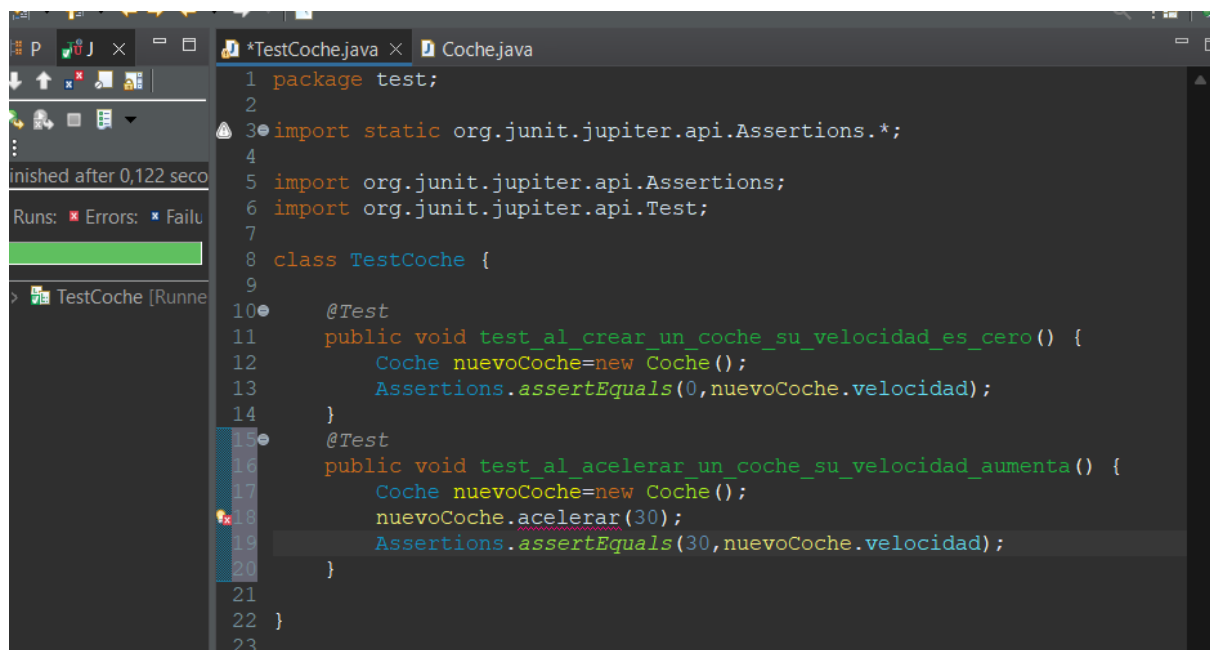


Realizamos un coverage y vemos que se ha hecho bien

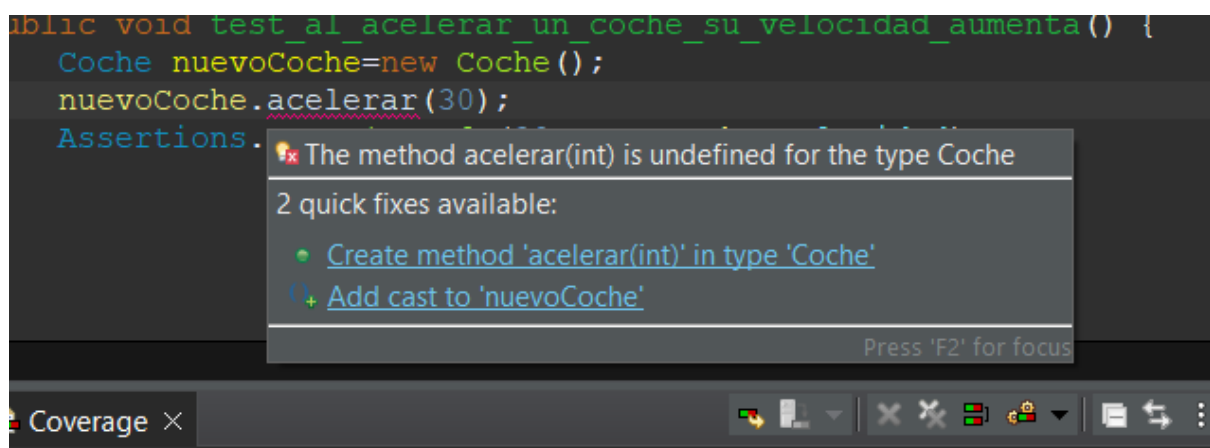


A continuación creamos otro test e intentamos usar el método acelerar que no existe aún en la clase coche, repetimos los pasos anteriores, nos ponemos encima de acelerar,

pinchamos la primera opción y cuando se cree el método ponemos lo que aumente la velocidad cuando se use ese método.



```
1 package test;
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     @Test
16     public void test_al_acelerar_un_coche_su_velocidad_aumenta() {
17         Coche nuevoCoche=new Coche();
18         nuevoCoche.acelerar(30);
19         Assertions.assertEquals(30,nuevoCoche.velocidad);
20     }
21
22 }
23
```



```
public void test_al_acelerar_un_coche_su_velocidad_aumenta() {
    Coche nuevoCoche=new Coche();
    nuevoCoche.acelerar(30);
    Assertions.
```

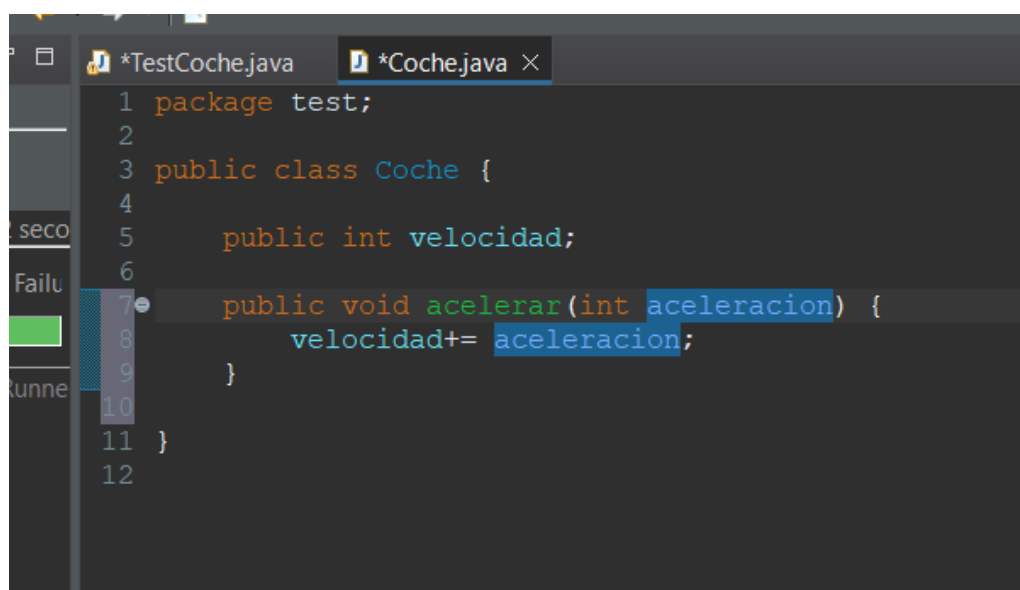
The method `acelerar(int)` is undefined for the type `Coche`

2 quick fixes available:

- Create method '`acelerar(int)`' in type '`Coche`'
- Add cast to '`nuevoCoche`'

Press 'F2' for focus

Coverage ×



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

Volvemos ha hacer un coverage

The screenshot shows a web browser on the left and an Eclipse IDE on the right. The browser displays the profile of MIREYA MARTÍNEZ PÉREZ on the Aules platform, with a search bar and navigation links. The IDE shows a Java file named TestCoche.java with two test methods. The bottom panel of the IDE displays a coverage report for the TestCoche class, showing 100.0% coverage and 34 covered instructions.

Element	Coverage	Covered Instructions	Issued Ins
TestCoche (1 abr 2023 21:23:06)	100.0 %	34	

Repetimos otra vez todo lo anterior pero con decelerar

```
@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);
}
```

en este caso pinchamos en la segunda opción

The screenshot shows the Eclipse IDE with a code completion suggestion for the `decelerar` method. The suggestion is highlighted, and the text "The method decelerar(int) is undefined for the type Coche" is displayed. Below the suggestion, three quick fixes are available: "Change to 'acelerar(...)'", "Create method 'decelerar(int)' in type 'Coche'", and "Add cast to 'nuevoCoche'".

The method `decelerar` is undefined for the type `Coche`. 3 quick fixes available:

- Change to 'acelerar(...)'
- Create method 'decelerar(int)' in type 'Coche'
- Add cast to 'nuevoCoche'



```

9      }
10
11     public void decelerar(int deceleracion) {
12         velocidad-= deceleracion;
13     }
14
15 }
16

```

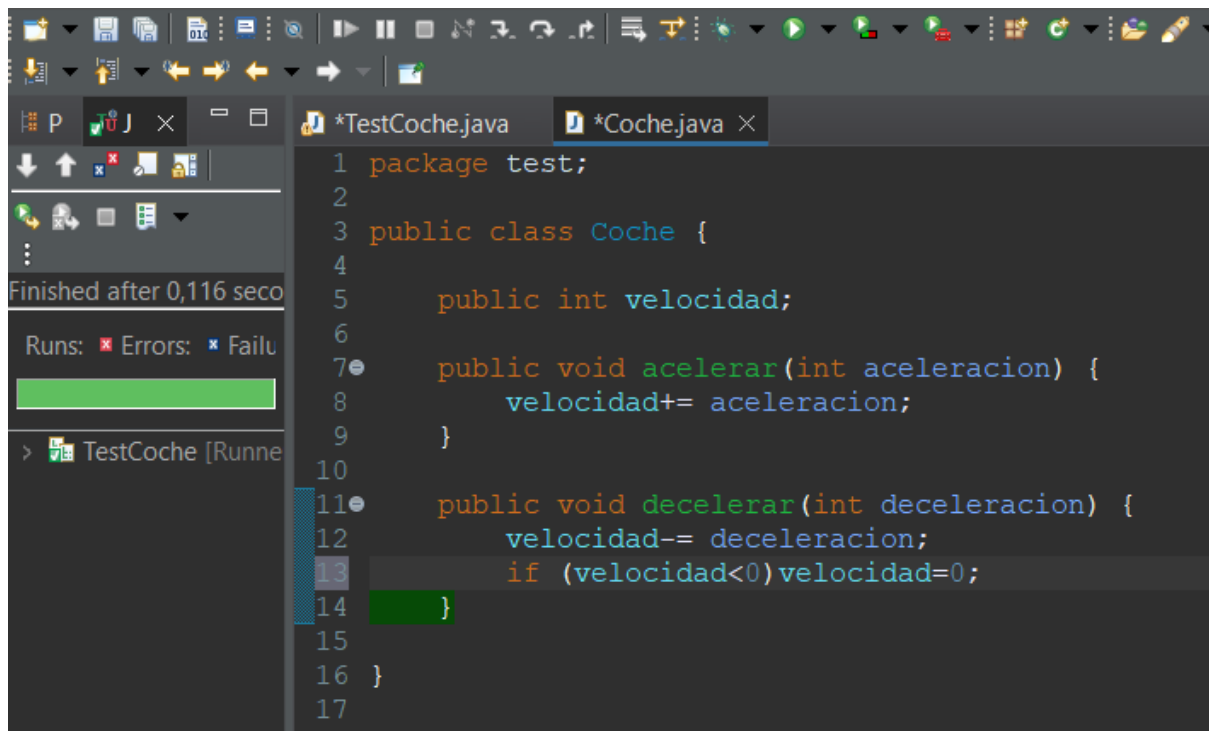
The screenshot shows two windows side-by-side. The left window is a web browser displaying the profile page of MIREYA MARTÍNEZ PÉREZ on the Aules platform. The right window is the Eclipse IDE, showing the source code of a Java project named 'TDDCocheEclipse'. The IDE displays the 'TestCoche.java' file, which contains several JUnit tests for the 'decelerar' method. The tests verify that the velocity decreases correctly and remains non-negative.

Por último método vamos a crear un método para indicar que pasaría si intentamos decelerar mas de la velocidad existente, puesto que la velocidad no puede ser negativa lo que haremos es crear un condicional el decelerar en el que le indicaremos que siempre que velocidad sea menor que 0 se quede en 0

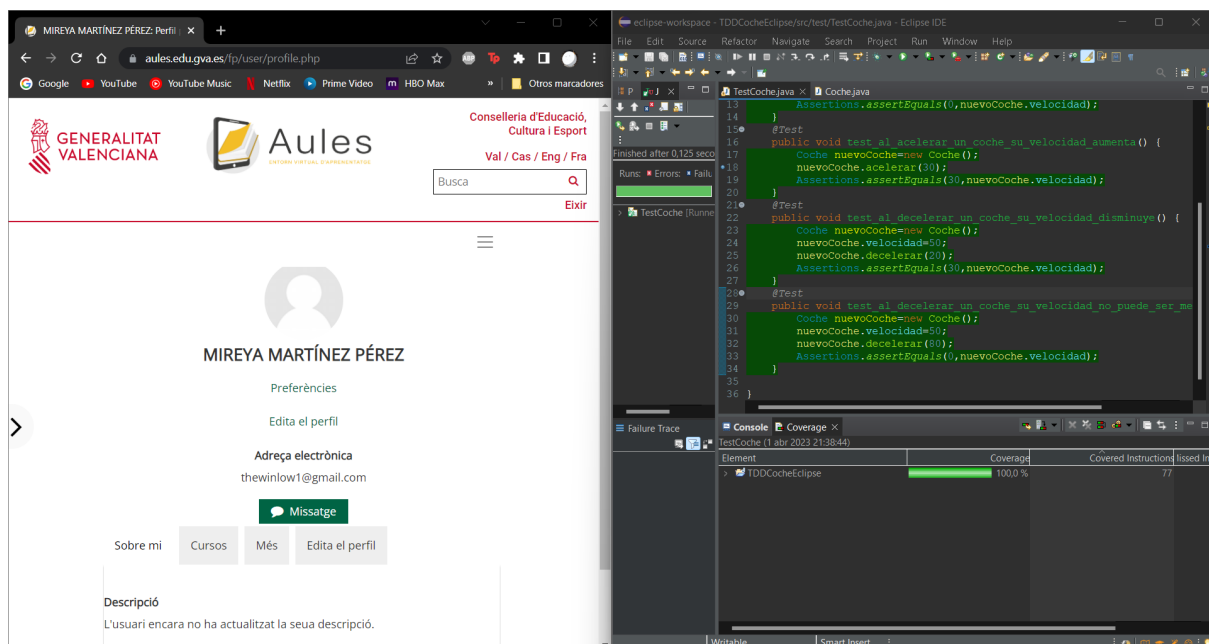
```

24     nuevoCoche.velocidad=50;
25     nuevoCoche.decelerar(20);
26     Assertions.assertEquals(30,nuevoCoche.velocidad);
27 }
28 public void test_al_decelerar_un_coche_su_velocidad_no_puede_ser_menor_que_cero() {
29     Coche nuevoCoche=new Coche();
30     nuevoCoche.velocidad=50;
31     nuevoCoche.decelerar(80);
32     Assertions.assertEquals(0,nuevoCoche.velocidad);
33 }
34
35 }

```



```
1 package test;
2
3 public class Car {
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 }
17
```



MIREYA MARTÍNEZ PÉREZ

Conselleria d'Educació, Cultura i Esport

Val / Cas / Eng / Fra

Busca

Eixir

MIREYA MARTÍNEZ PÉREZ

Preferències

Edita el perfil

Adreça electrònica

thewinlow1@gmail.com

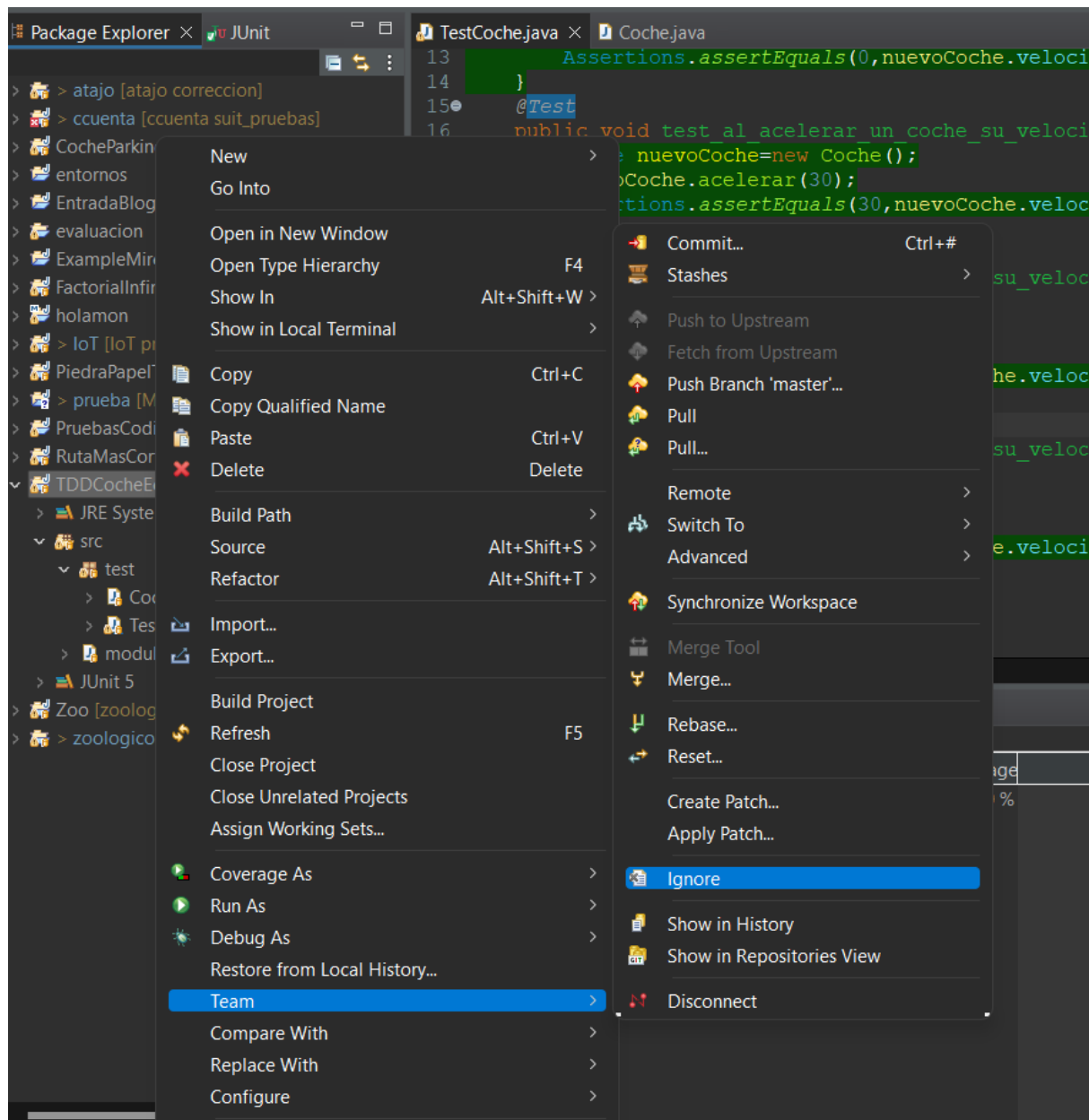
Missatge

Sobre mi Cursos Més Editar el perfil

Descripció

L'usuari encara no ha actualitzat la seua descripció.

Para terminar crearemos dos ramas en git



Una que se llamará Refactorizar, donde refactorizar vemos todos los nombres de los métodos y la otra donde meteremos esta memoria en pdf.

