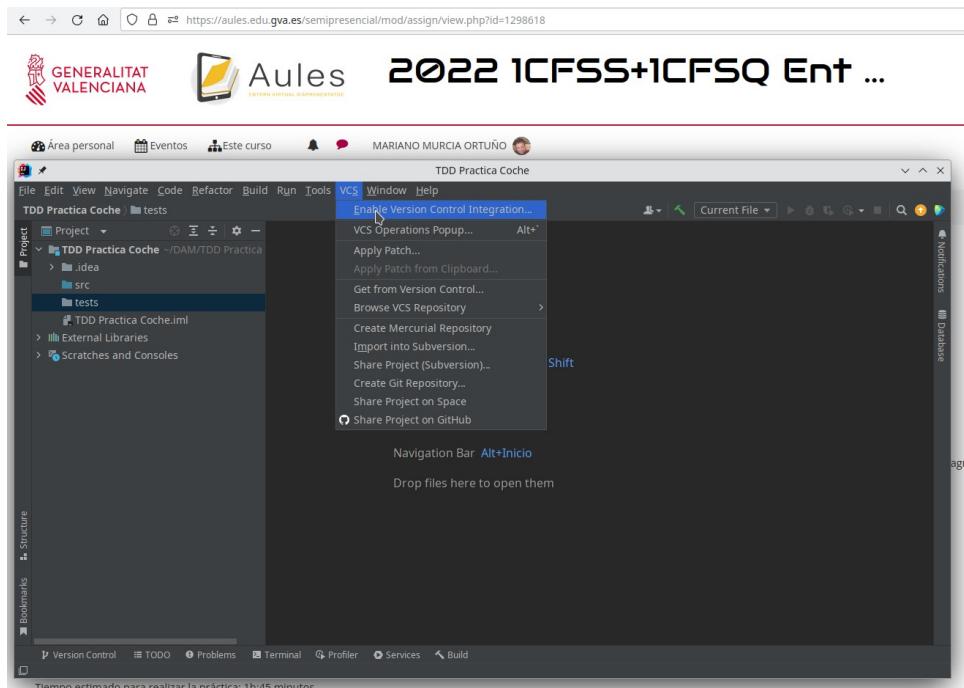
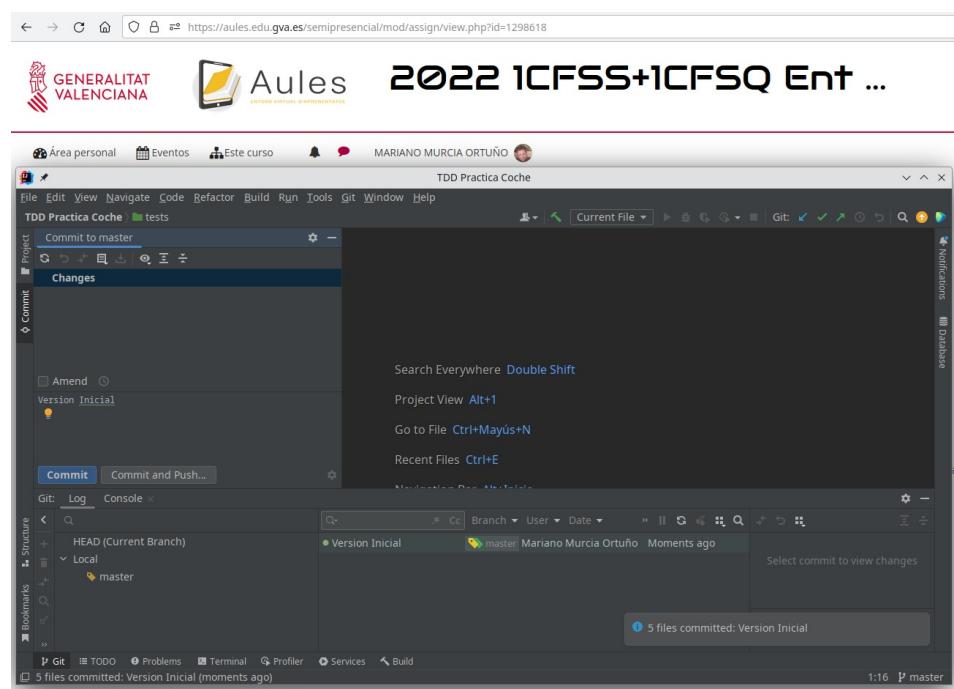


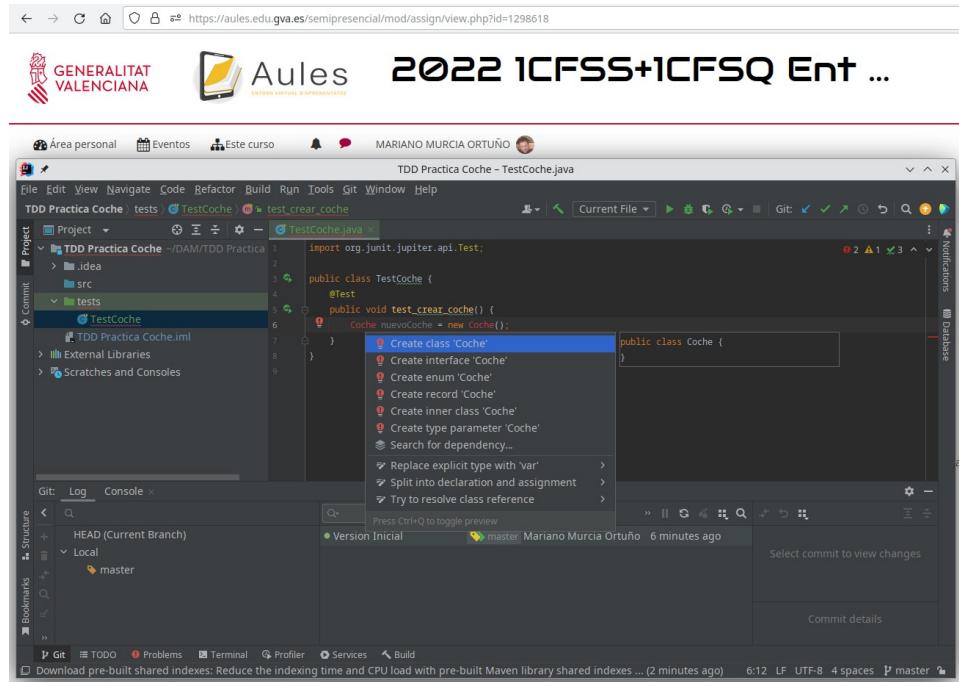
MI PRIMER TDD



Creamos nuestro proyecto “TDD Proyecto Coche” en IntelliJ IDEA. Creando una nueva carpeta “tests” y la marcamos como “Test Source Root”. A continuación iniciamos el control de versiones con “git”.



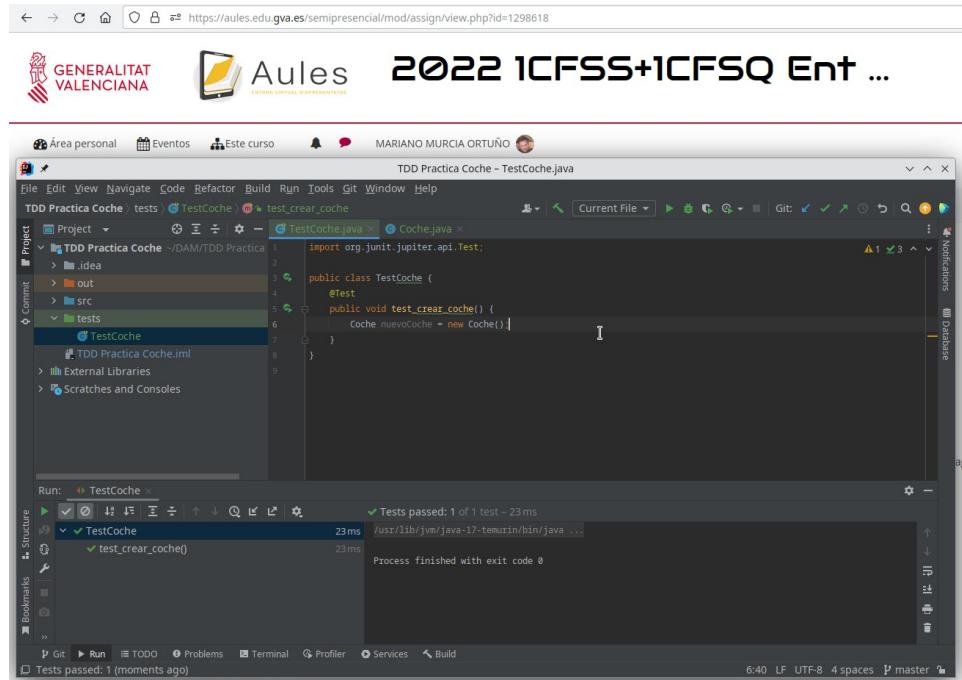
Realizamos nuestro primer commit “Version Inicial”.



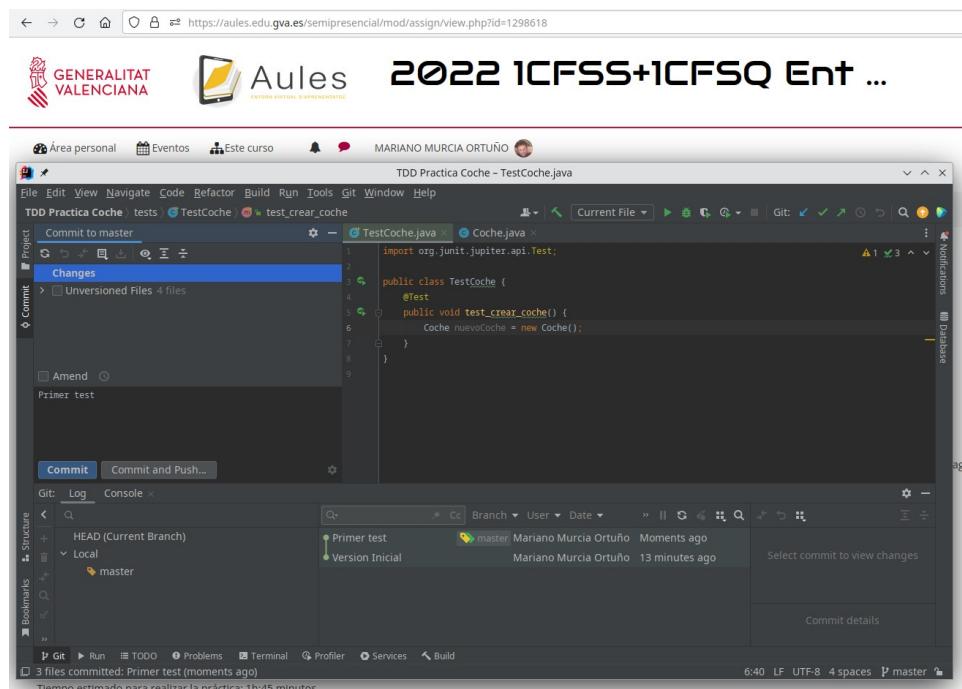
Creamos el primer test, usando la palabra clave `@Test`, y pulsando Alt+Intro, seleccionamos el Framework que queremos usar, en nuestro caso JUnit5.8.1.

Completamos el test creando un objeto de la clase Coche, el cual, al no existir, IntelliJ nos recomienda crearlo. Haremos Alt+Intro y seleccionaremos “Create class ‘Coche’”.

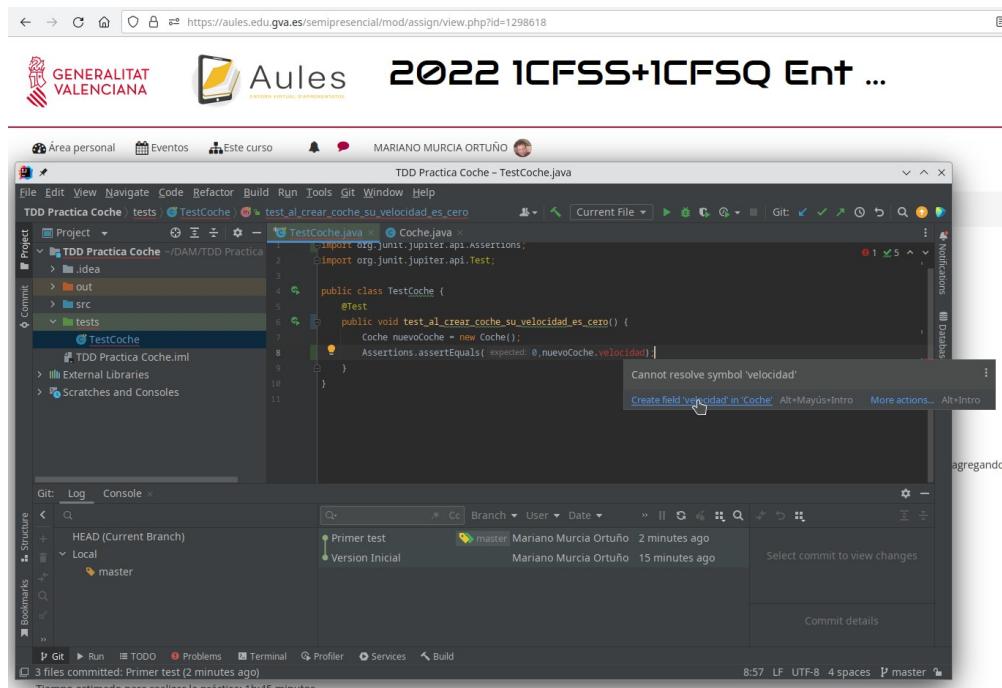
Seleccionaremos la carpeta correcta y pulsaremos OK.



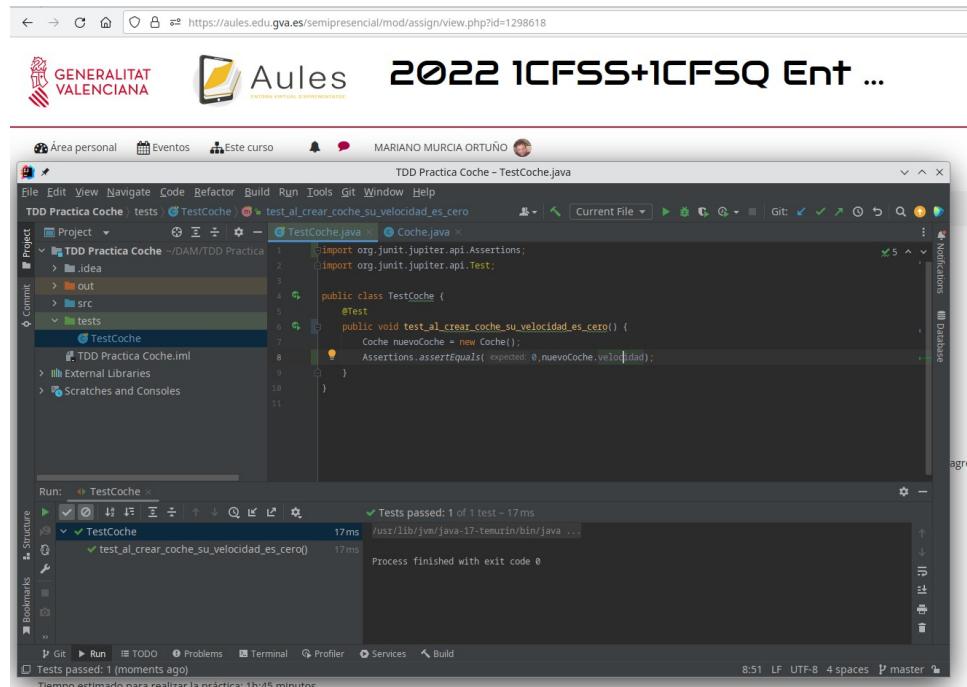
Ejecutamos, y comprobamos que tenemos el “check verde” que nos indica que el test a terminado satisfactoriamente.



Creamos un nuevo Commit, con los cambios realizados.



Modificamos el metodo, a “test_al_crear_coche_su_velocidad_es_cero()”. E introducimos una asercción para comprobar si el resultado esperado del atrituto del objeto coincide con el que nos devuelve. Como no existe tal atributo, el IDE nos sugiere crearlo.



Ejecutamos el nuevo test, y observamos que se ha completado correctamente, dado que la inicialización por defecto de un entero es 0, lo cual corresponde con el valor esperado.

```

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

public class TestCoche {
    @Test
    public void test_al_crear_coche_su_velocidad_es_cero() {
        Coche nuevoCoche = new Coche();
        Assertions.assertEquals(0, nuevoCoche.velocidad);
    }

    @Test
    public void test_al_acelerar_un_coche_su_velocidad_aumenta() {
        Coche nuevoCoche = new Coche();
        nuevoCoche.acelerar(30);
        Assertions.assert
    }
}

```

Y realizamos un nuevo Commit, ya que hemos realizado cambios que ha resultado satisfactorios.

```

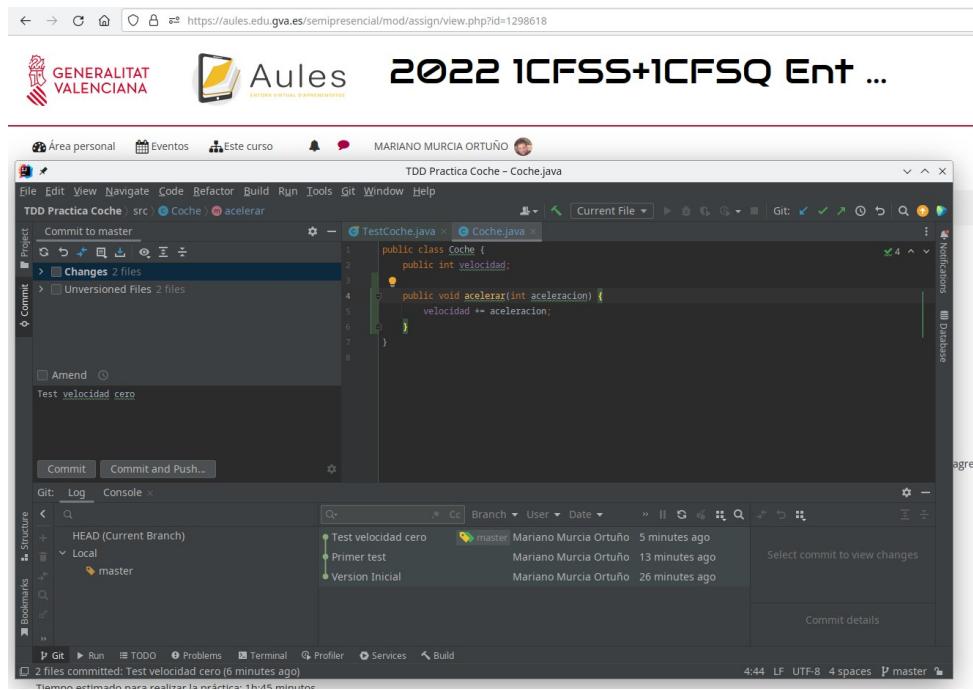
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.Test;

public class TestCoche {
    @Test
    public void test_al_crear_coche_su_velocidad_es_cero() {
        Coche nuevoCoche = new Coche();
        Assertions.assertEquals(0, nuevoCoche.velocidad);
    }

    @Test
    public void test_al_acelerar_un_coche_su_velocidad_aumenta() {
        Coche nuevoCoche = new Coche();
        nuevoCoche.acelerar(30);
        Assertions.assert
    }
}

```

Creamos un nuevo test "test_al_acelerar_un_coche_su_velocidad_aumenta()", en el que hacemos una llamada a un nuevo metodo, el cual al no existir, el IDE nos recomienda crearlo.



Lo modificamos para que se ajuste a nuestra necesidades.

The screenshot shows a Java code editor with the following code:

```

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_aumenta() {
        Coche nuevoCoche = new Coche();
        Assertions.assertEquals( expected: 0, nuevoCoche.velocidad);
    }

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

```

The code editor interface includes tabs for 'TestCoche.java' and 'Coche.java'. Below the code, the terminal output shows:

```

Tests passed: 2 of 2 tests - 18ms
/usr/lib/jvm/java-17-temurin/bin/java ...
  ✓ test_al_crear_un_coche_su_velocidad_aumenta 17ms
  ✓ test_al_acelerar_un_coche_su_velocidad_aumenta() 17ms
Process finished with exit code 0

```

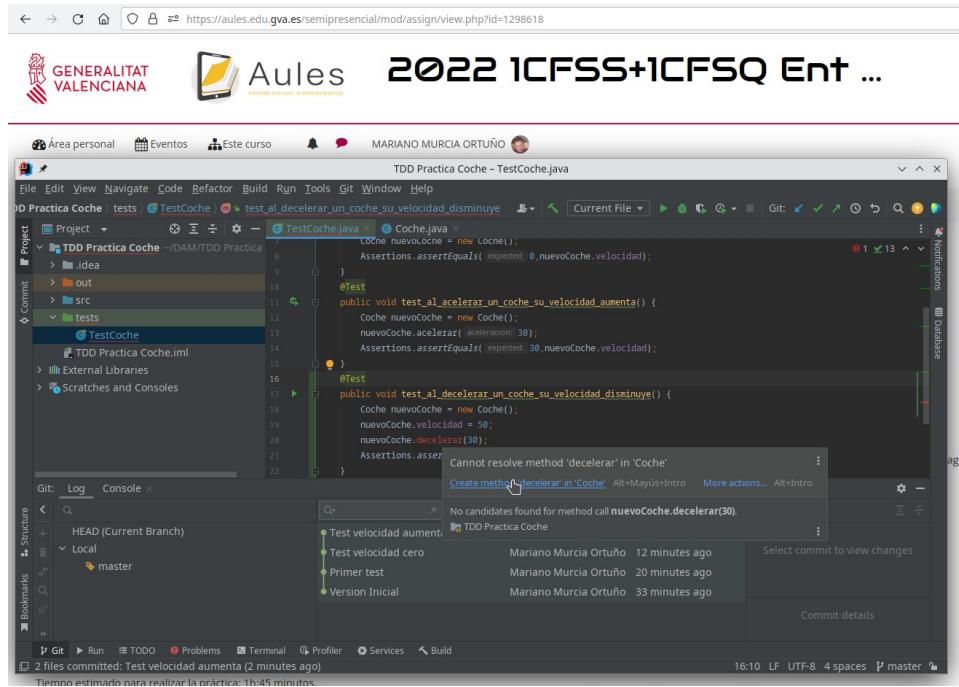
Modificamos la aserción, para que el valor esperado corresponda a nuestras nuevas expectativas y ejecutamos los tests. En ambos tenemos el check verde que nos indica que han finalizado con éxito.

The screenshot shows the Git log in the IDE. The commit history is as follows:

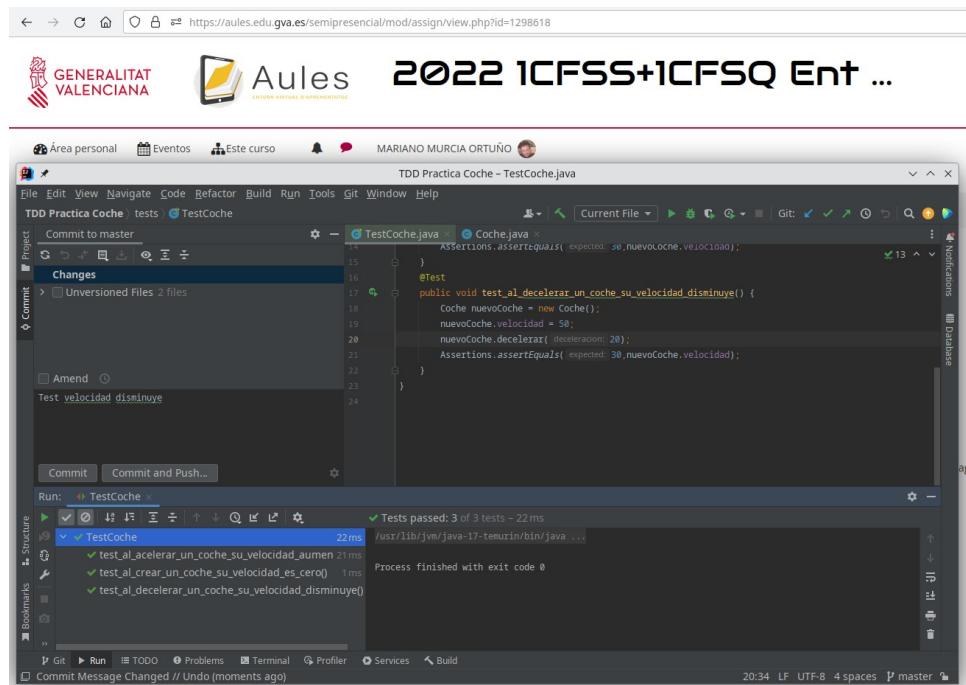
- HEAD (Current Branch)
- Local
 - master
 - Test velocidad aumenta Mariano Murcia Ortúñ Moments ago
 - Test velocidad cero Mariano Murcia Ortúñ 10 minutes ago
 - Primer test Mariano Murcia Ortúñ 19 minutes ago
 - Version Inicial Mariano Murcia Ortúñ 31 minutes ago

The commit message 'Test velocidad aumenta' is highlighted in green, indicating it was just committed.

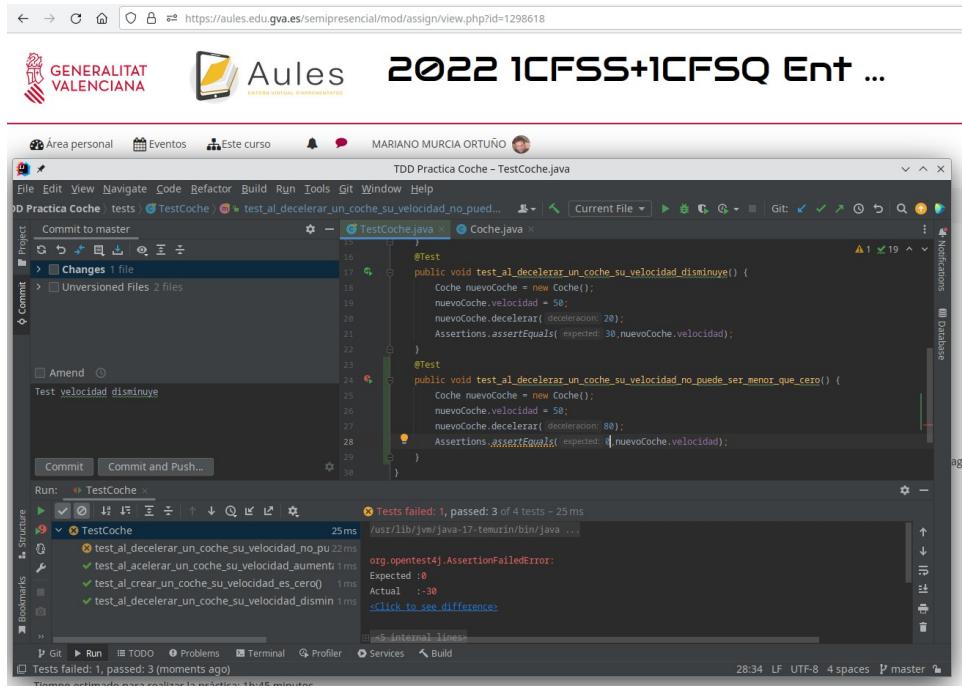
Realizamos un nuevo Commit con los nuevos cambios.



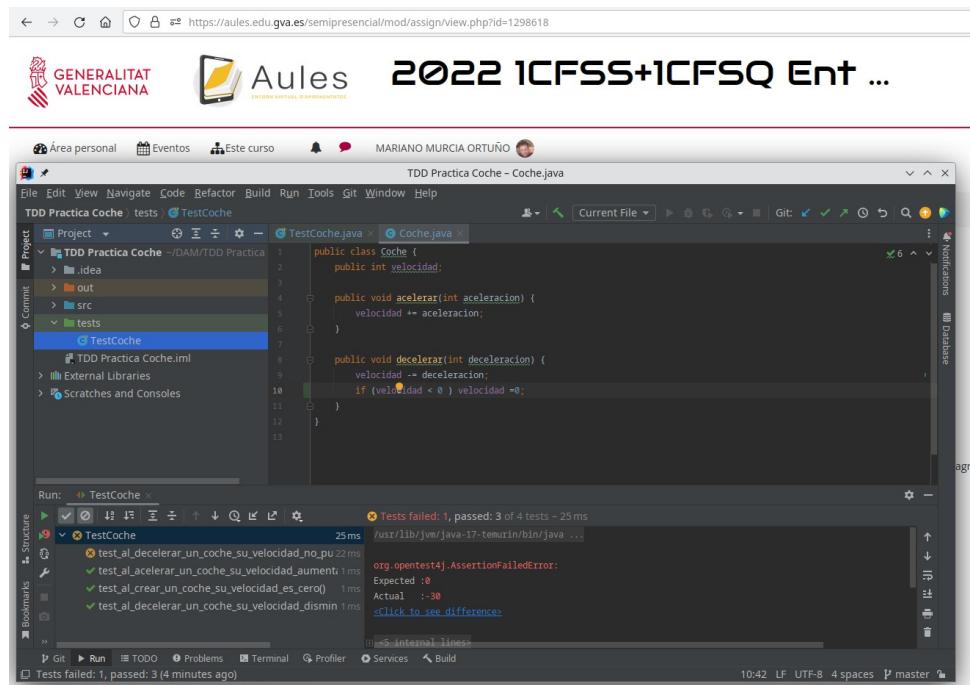
Creamos un nuevo método, en el que hacemos la llamada a otro método de la clase Coche que no existe. Siguiendo los mismos pasos que anteriormente, la creamos.



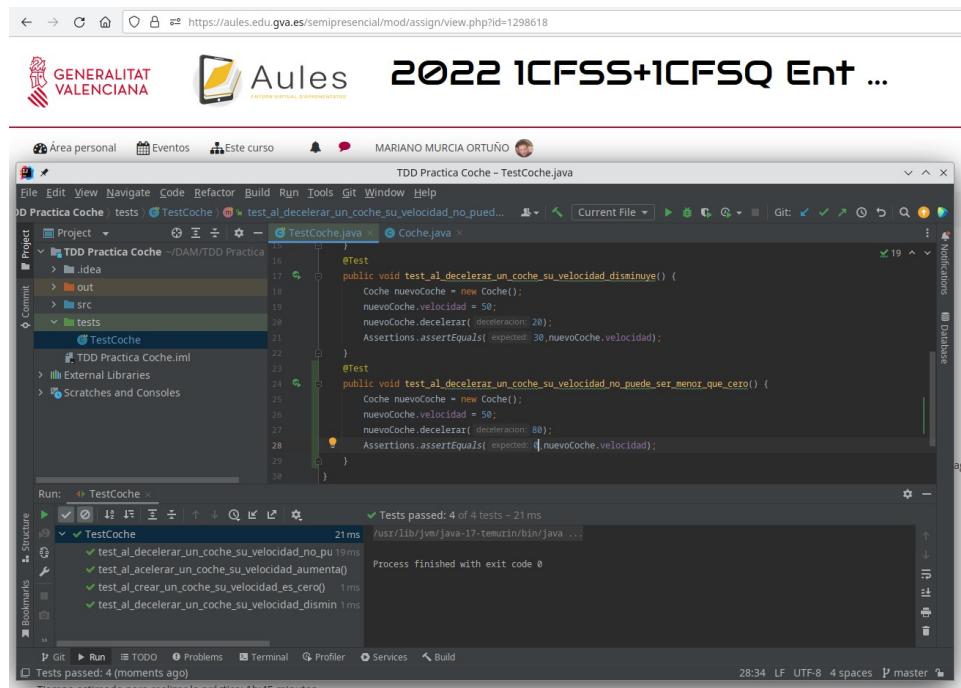
Los test acaban de forma correcta, por lo que realizamos un nuevo Commit, con los nuevos cambios.



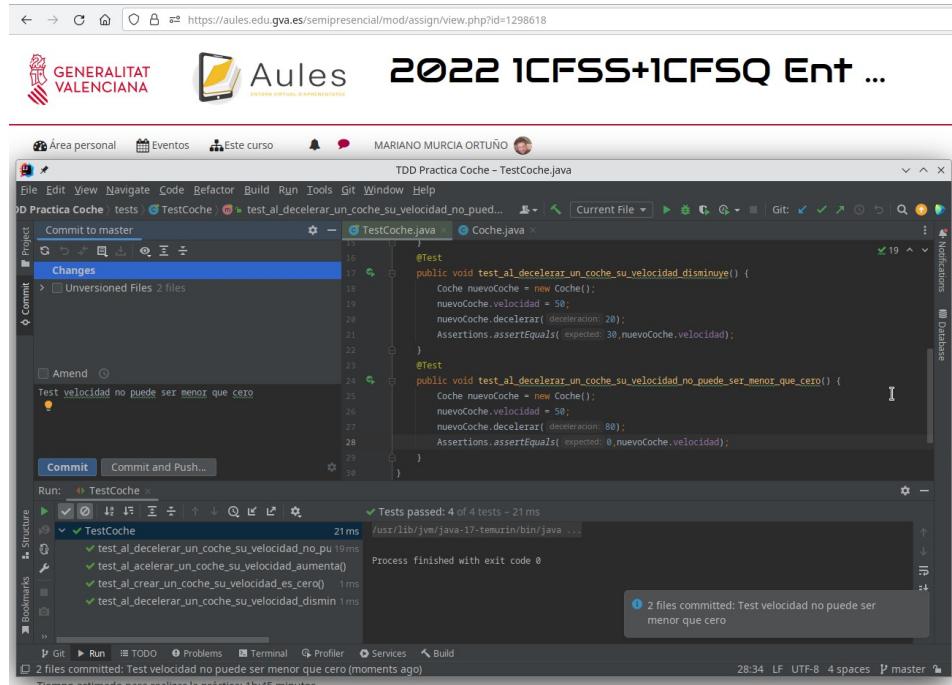
Creamos un nuevo test, para comprobar si la velocidad al decelerar no toma valores negativos. Al realizar la nueva prueba, el test falla.



Modificamos el método para adecuarlo.

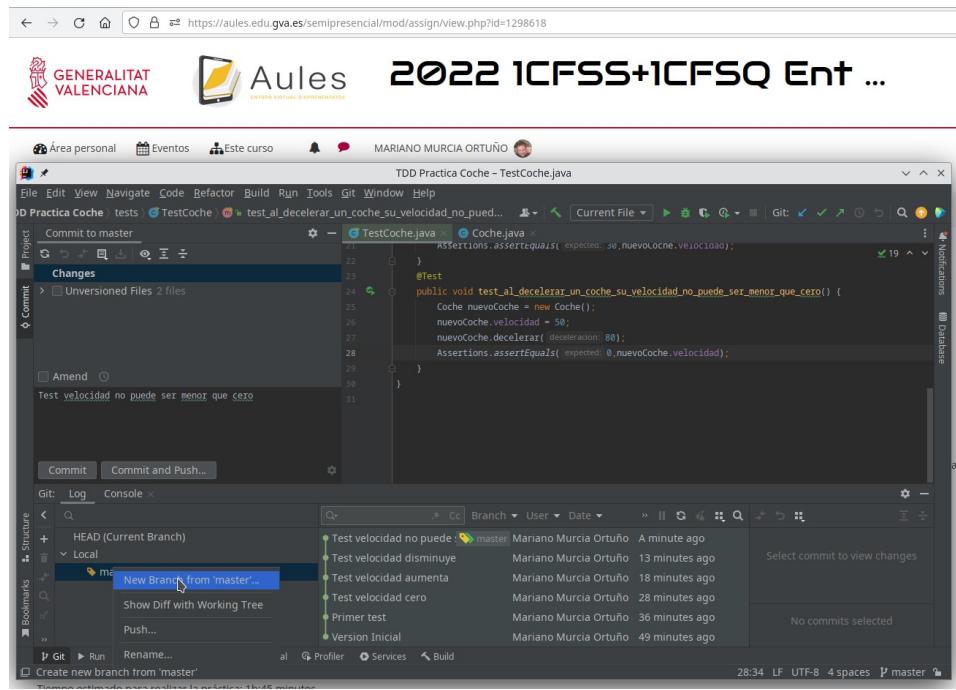


Y volvemos a ejecutar los tests, comprobando que ya funcionan correctamente.

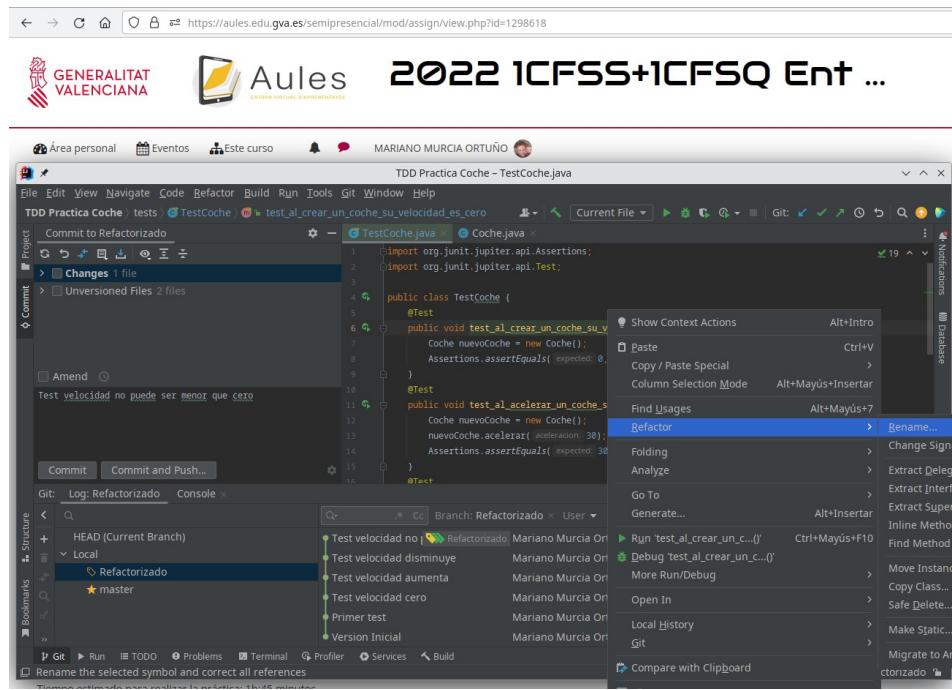


The screenshot shows the Aules software interface. At the top, there's a header with the logo of the Generalitat Valenciana, the word "Aules", and the year "2022 1CFSS+1CFSQ Ent ...". Below the header, the main window has a title bar "TDD Practica Coche - TestCoche.java". The left sidebar shows a project structure with "Changes" selected, showing a single amend: "Test velocidad no puede ser menor que cero". The main area displays Java code for a "Coche" class and its test cases in "TestCoche.java". The code includes annotations like @Test and assertions like assertEquals. The terminal at the bottom shows the output of the test run: "Tests passed: 4 of 4 tests - 21ms" and "Process finished with exit code 0". A tooltip indicates "2 files committed: Test velocidad no puede ser menor que cero".

Realizamos un nuevo commit con los nuevos cambios.



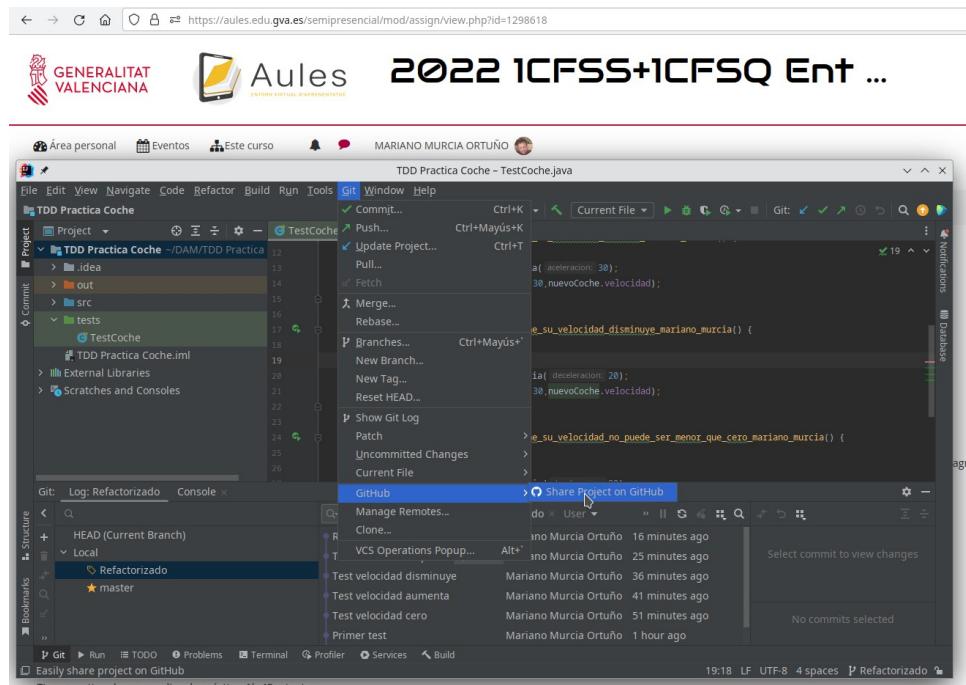
Creamos una nueva rama a la que llamaremos Refactorizado.



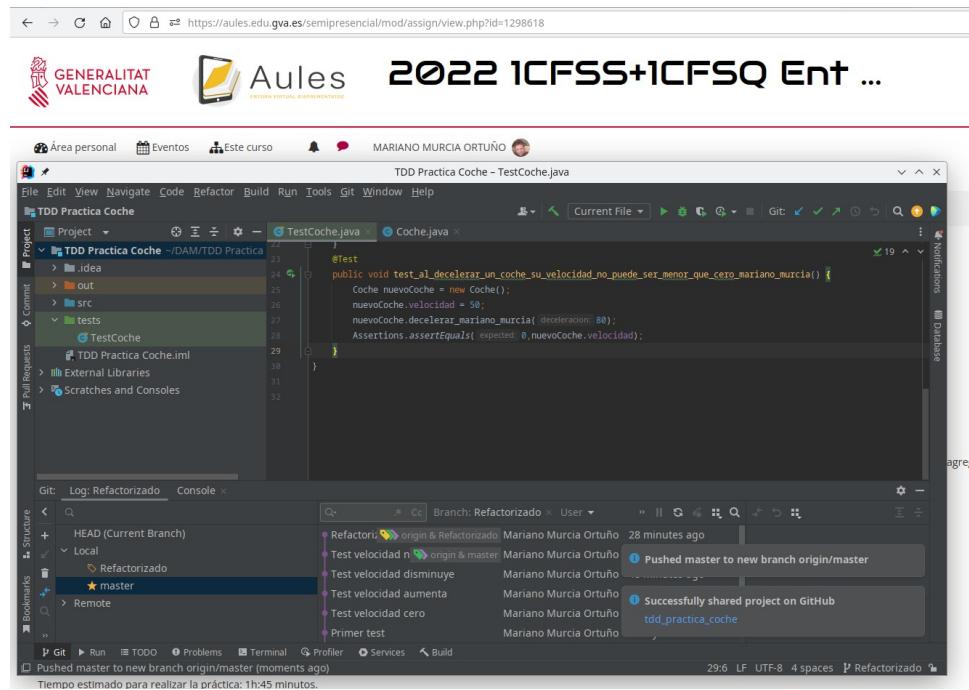
Refactorizamos las funciones, haciendo un “Rename” a todos los métodos, añadiendo mi nombre(mariano_murcia) al final de todas ellas.

The screenshot shows a Java development environment within the Aules platform. The code editor displays two files: `TestCoche.java` and `Coche.java`. `TestCoche.java` contains several test methods for a `Coche` class, including tests for acceleration and deceleration. The terminal below shows the output of a test run, indicating 4 tests passed in 20ms. A commit message at the bottom states "2 files committed: Refactorizado".

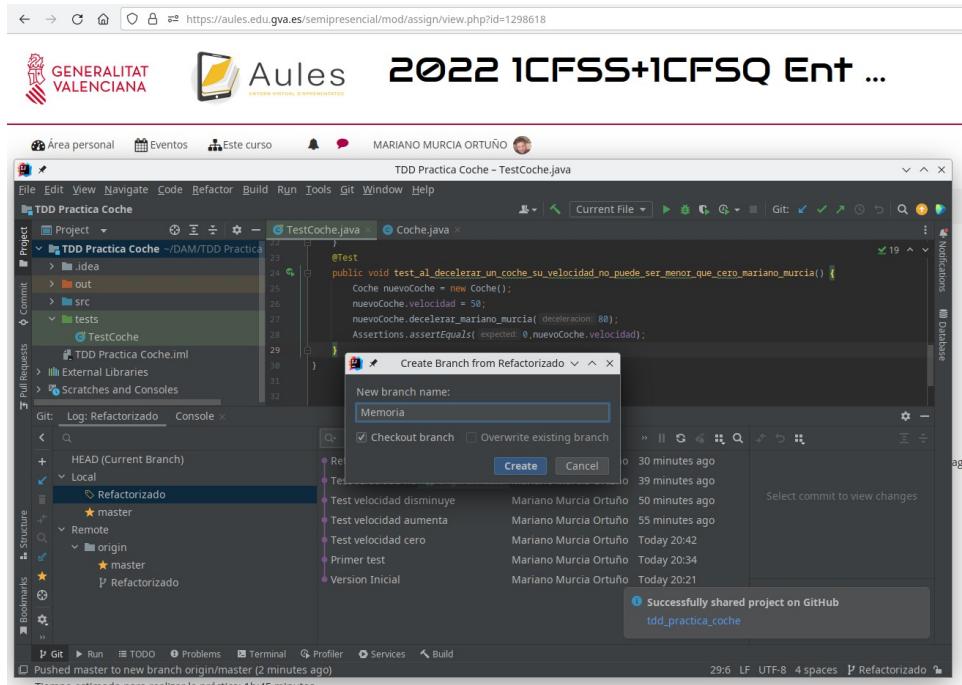
Comprobamos que los test no dan ningún error, y realizamos el Commit.



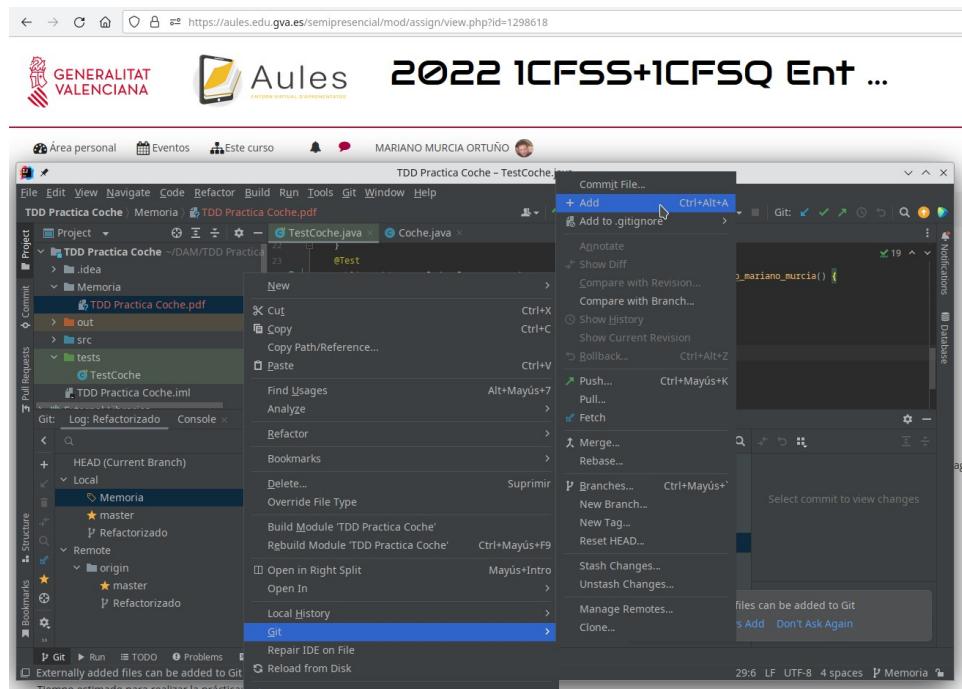
Compartimos nuestro proyecto en GitHub



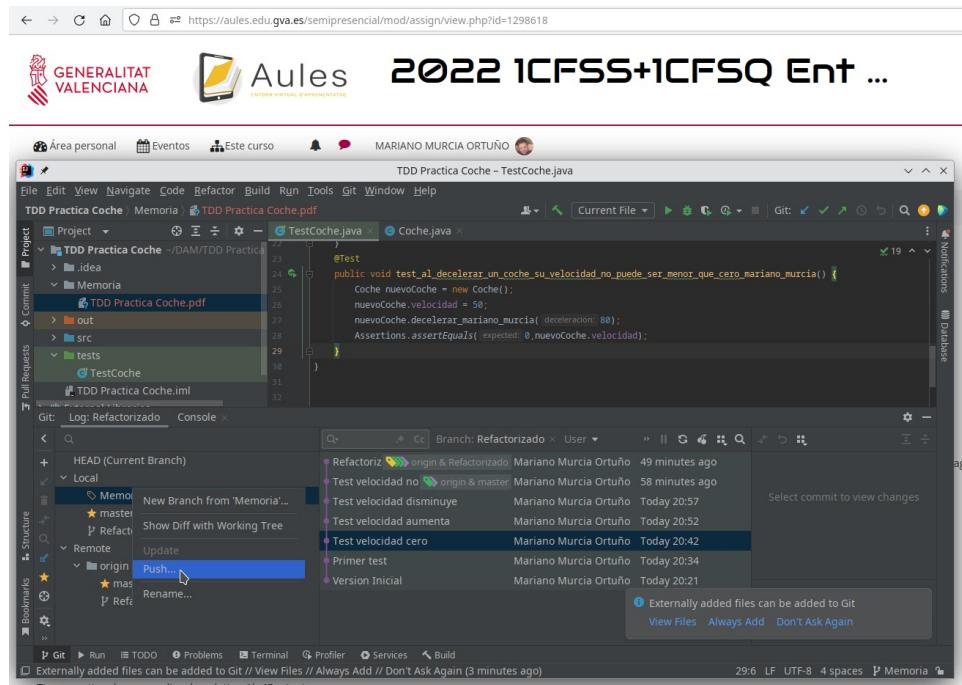
Y subimos tambien la rama master.



Creamos la rama “Memoria” y creamos una carpeta memoria donde guardaremos este documento, y lo subiremos a GitHub, de igual manera que hemos hecho con otros archivos.



Añadimos el archivo al repositorio



Y lo subimos al repositorio.