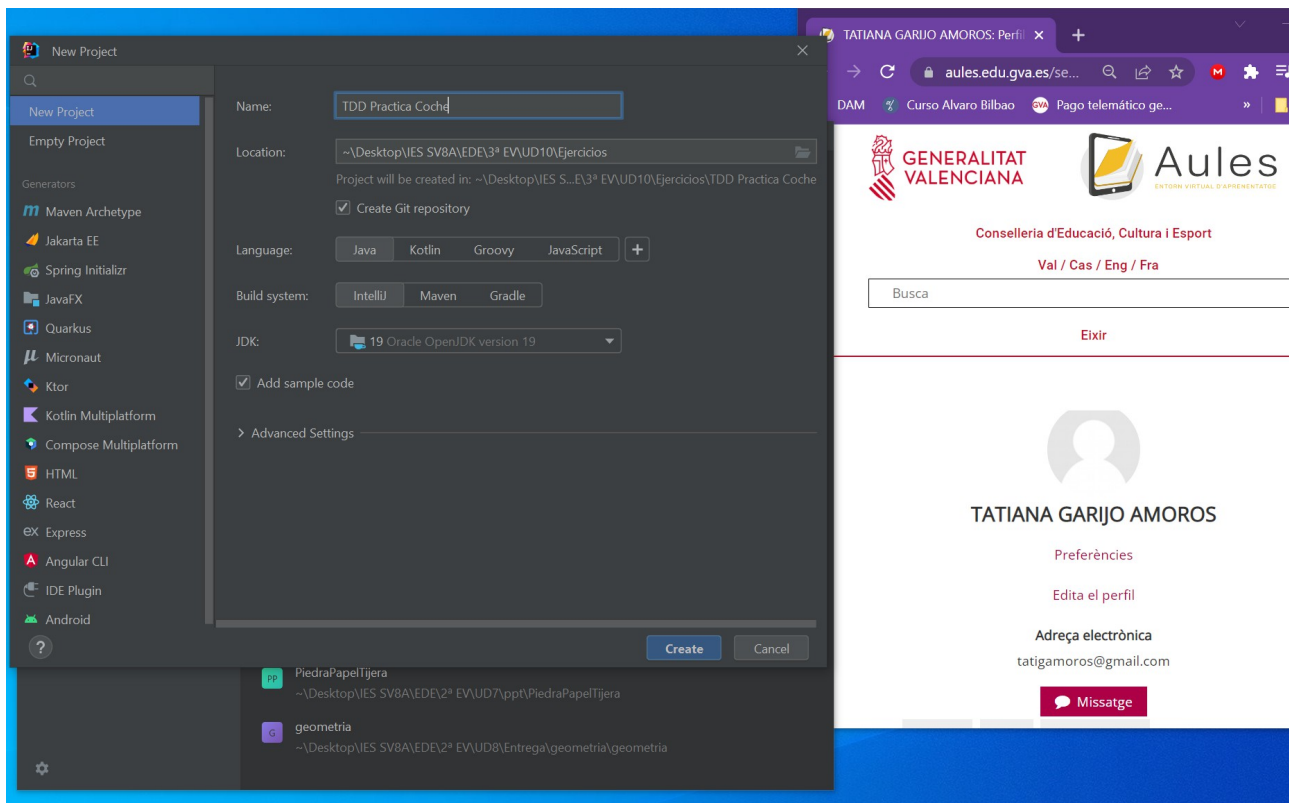


PRÁCTICA

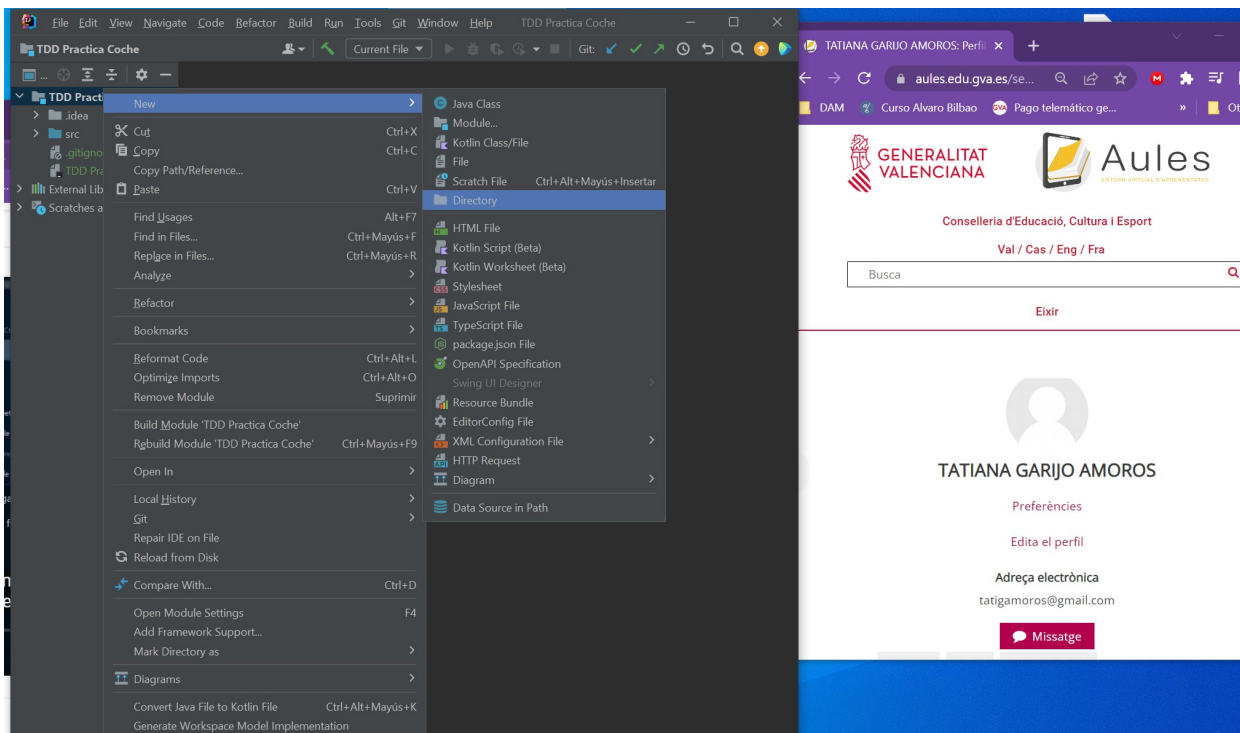
Mi primer TDD

- New Project IntelliJ



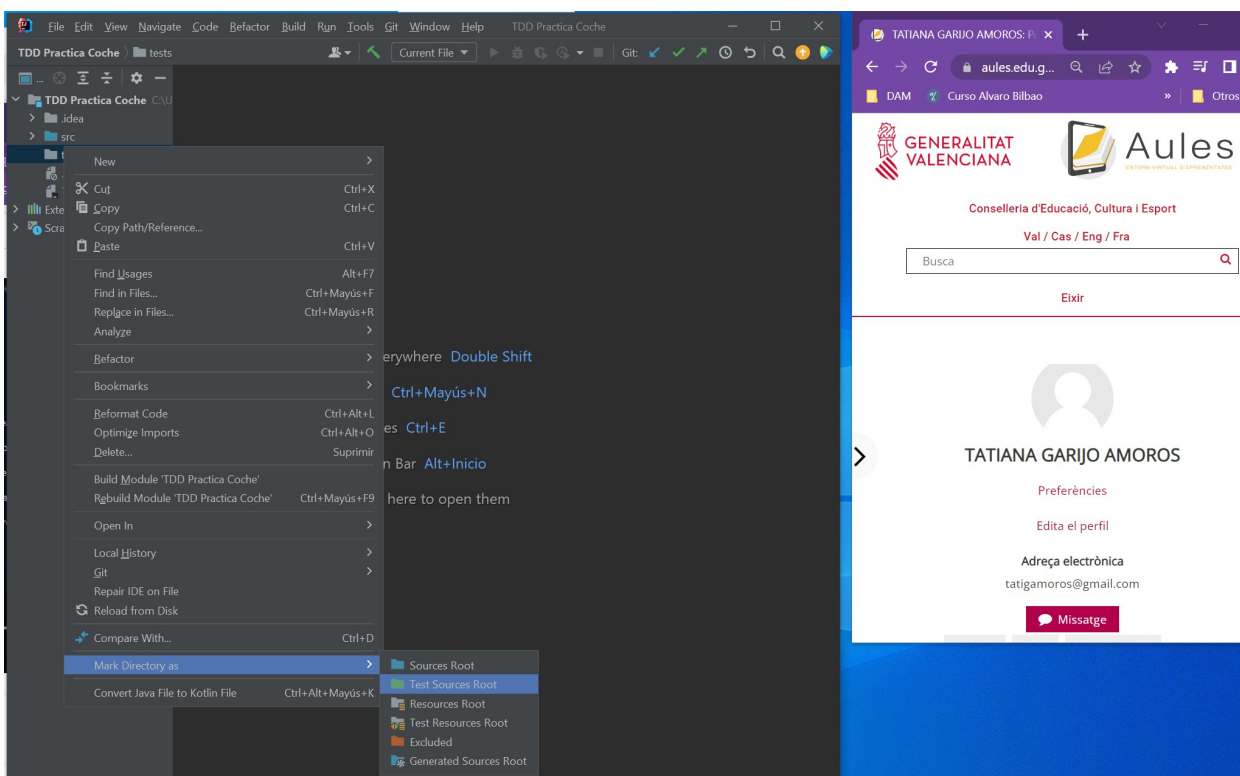
- Abrir IntelliJ
- Seleccionar la opción New Project
- Configurar las características del proyecto
 - En mi caso he seleccionado
 - Nombre del proyecto
 - Lenguaje: Java
 - Build System: IntelliJ
 - version 19 JDK
 - Create Git repository

- Crear un nuevo directorio dentro del proyecto



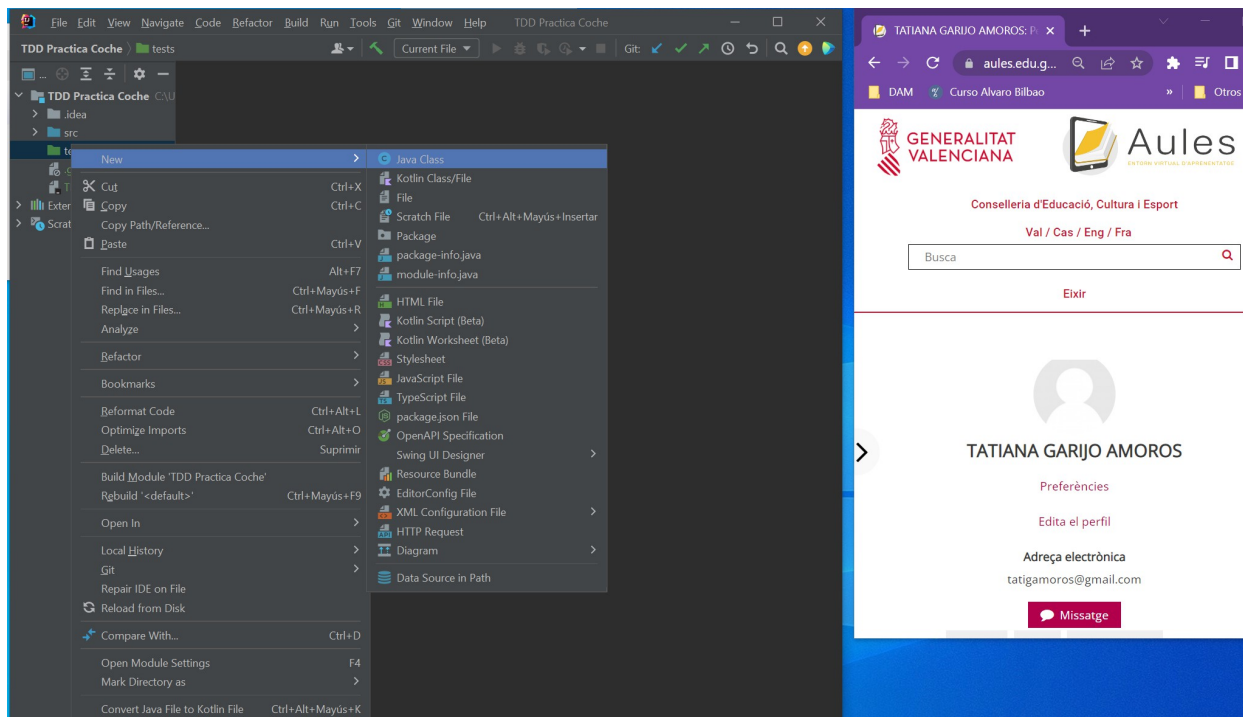
- Hacer clic con botón derecho sobre el proyecto → New → Directory

- Marcar el nuevo directorio como directorio de Test



- Hacer clic con botón derecho sobre el directorio → Mark Directory as → Test Sources Root

- Crear una clase Java dentro del directorio tests



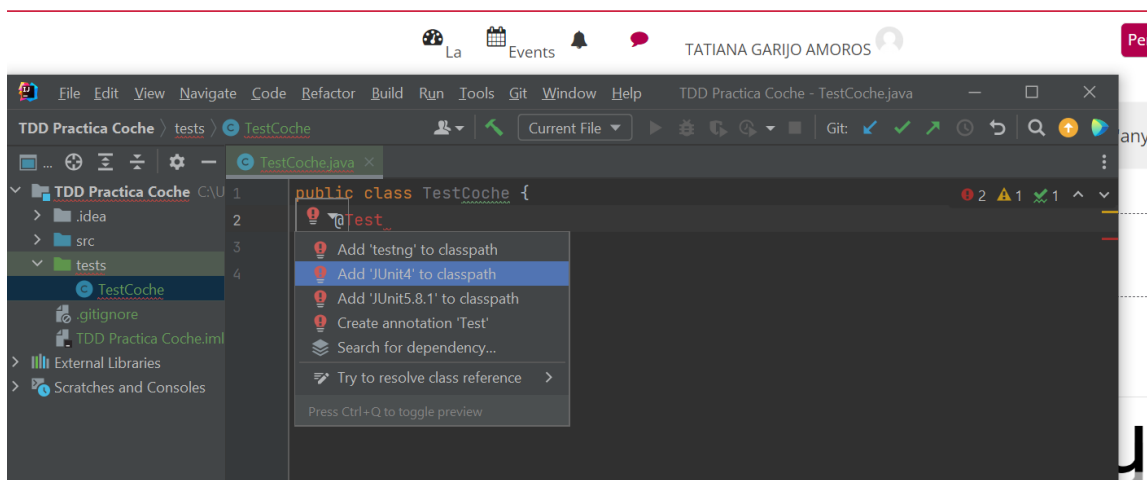
- Hacer clic con botón derecho sobre el directorio → New → Java Class

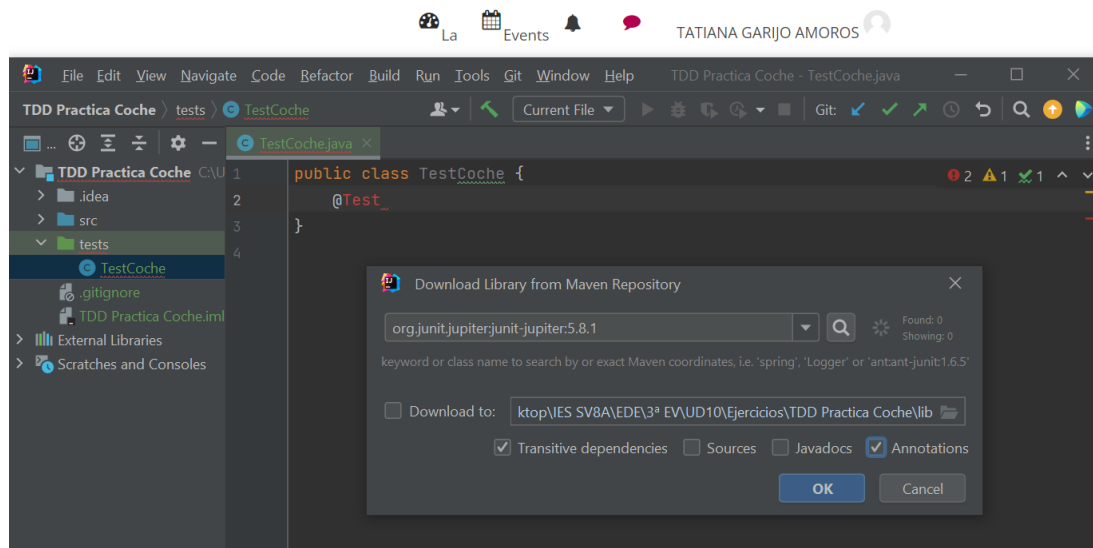
- Nombrar la clase

- Indicar que se va a crear un test y la versión de JUnit

- Dentro de la clase anotamos @Test

- Hacer clic sobre el icono de la bombilla roja o pulsando Alt + Enter → Add 'JUnit5.8.1' to classpath

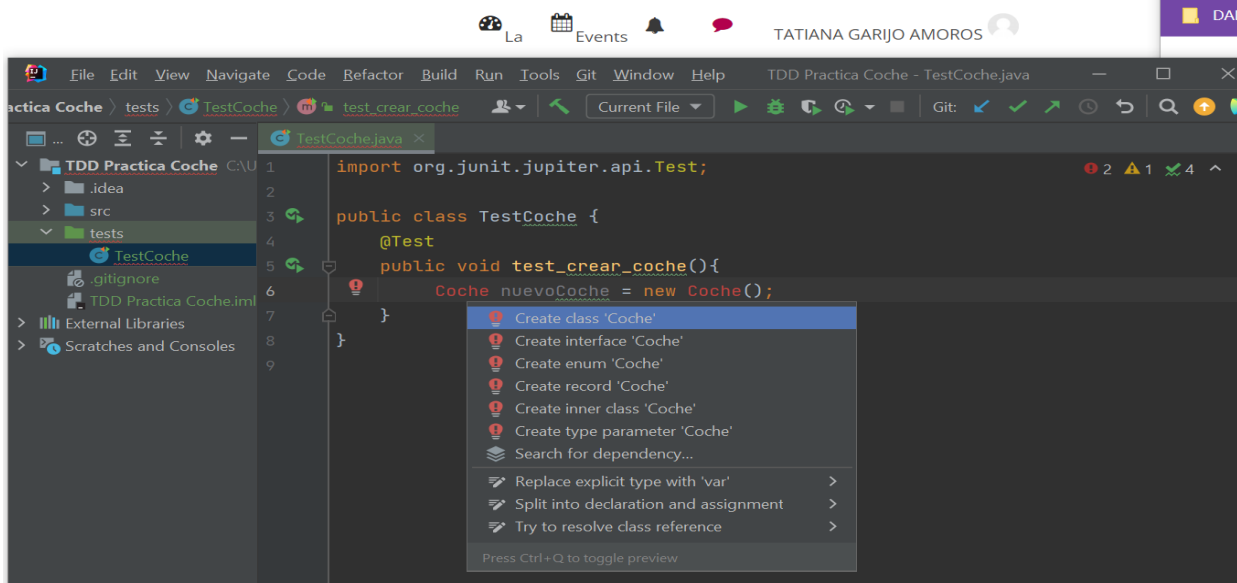




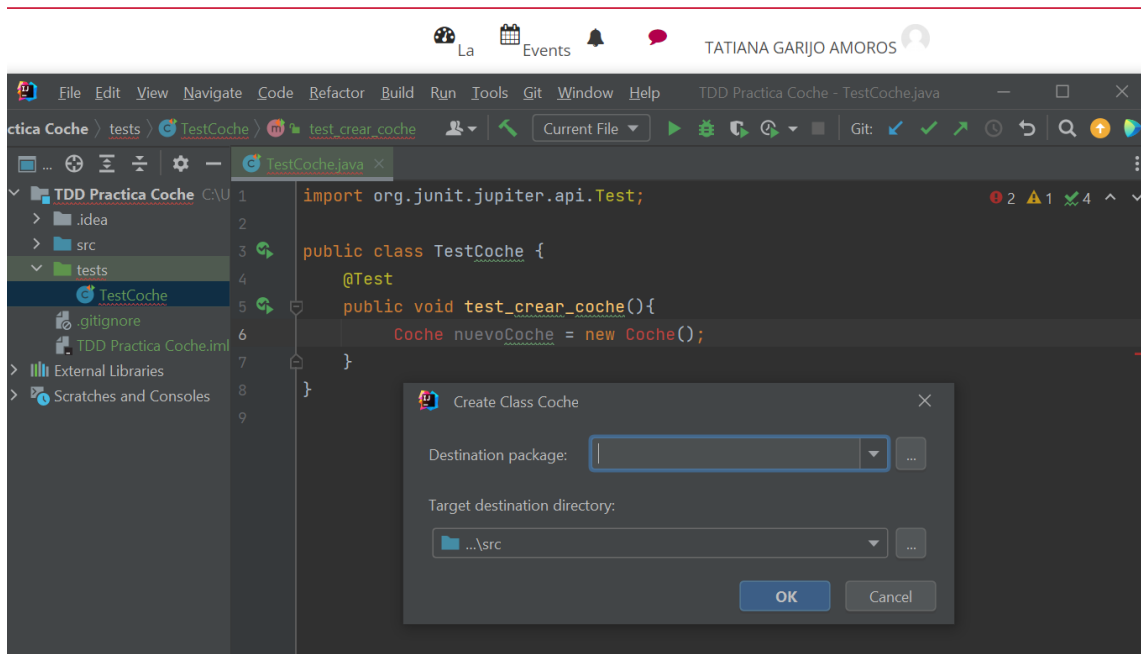
- En la ventana emergente marcar las opciones
 - Transitive dependencies
 - Annotations

- Crear primer test

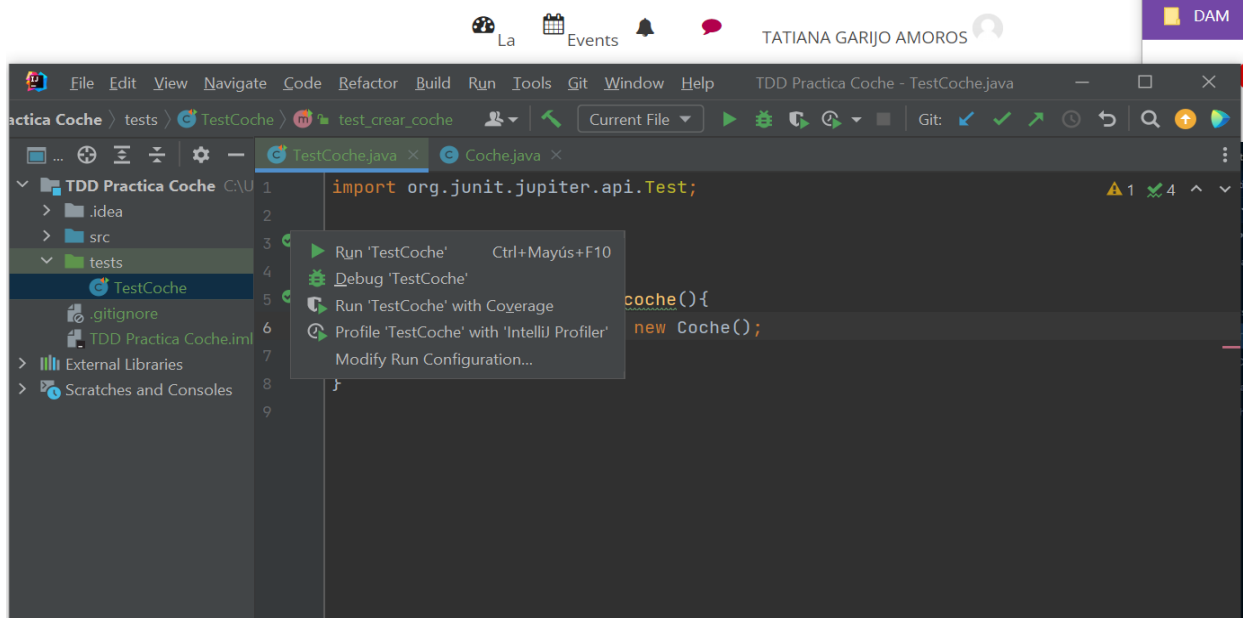
-



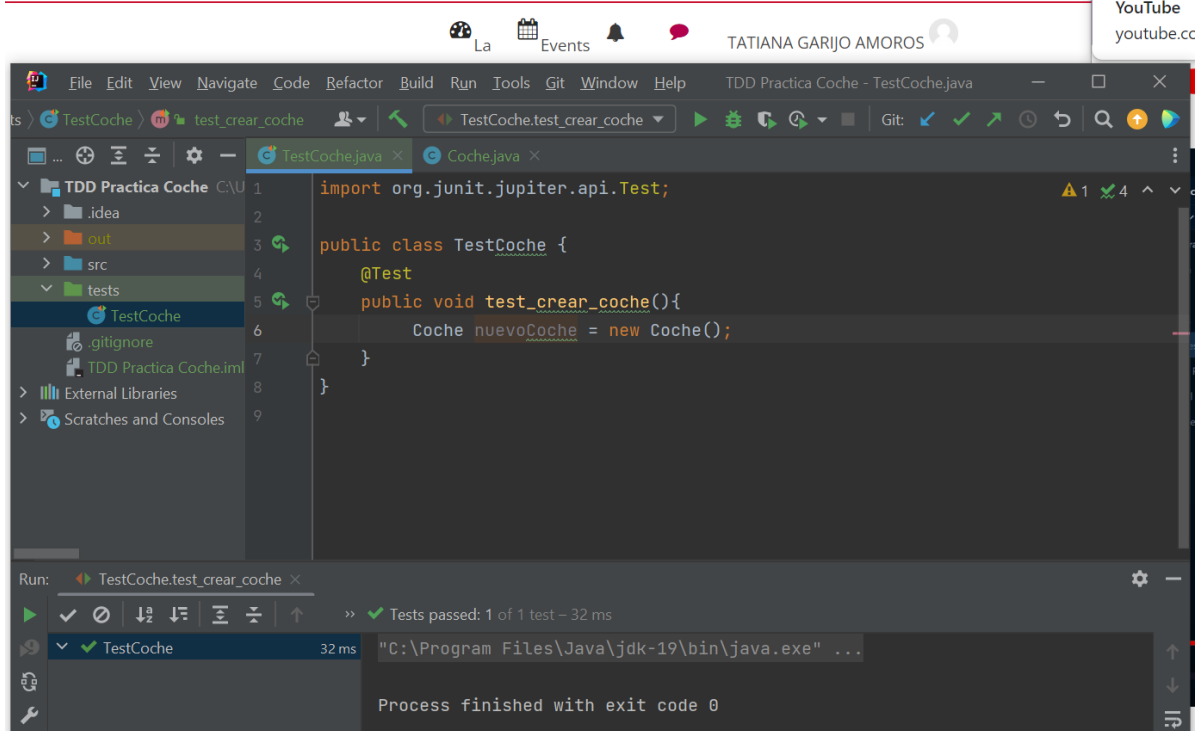
- Nombrar el método
- Crear un objeto (el objeto pertenece a una clase que está pendiente de ser creada)
- Crear clase
 - Hacer clic sobre el icono de la bombilla roja o pulsando Alt + Enter → Create class 'nombreClase'
 - Crear la clase dentro del directorio src



- Ejecutar Test

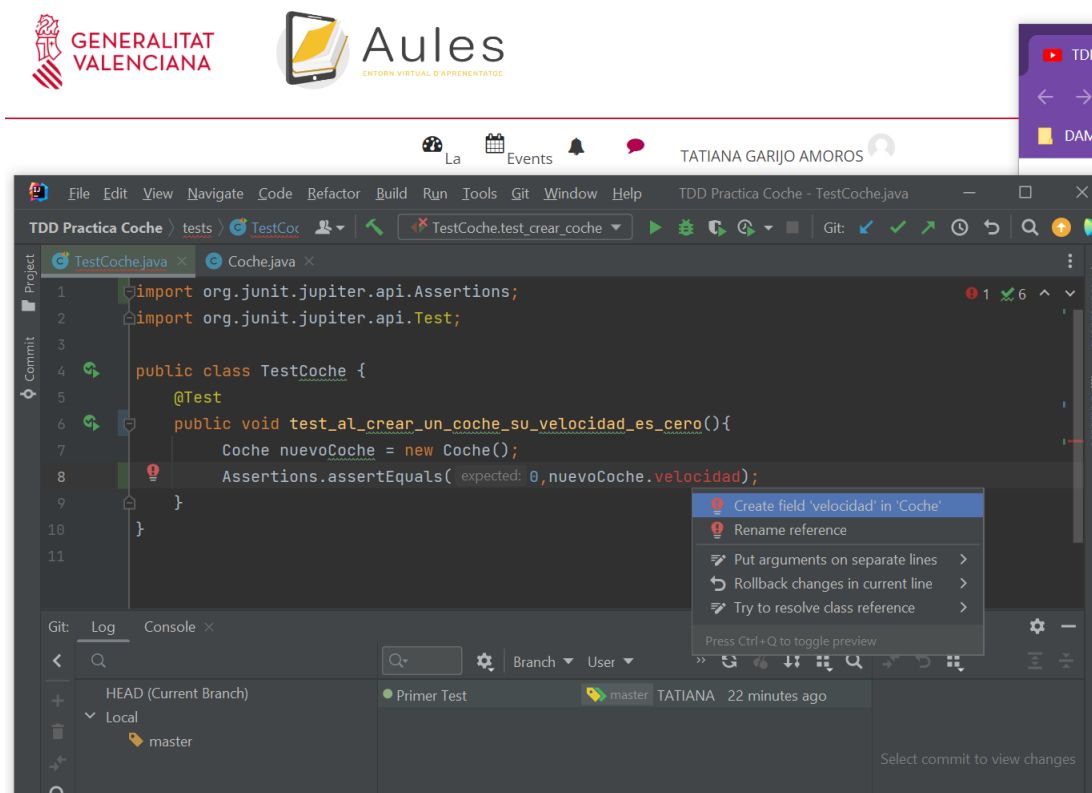


- Hacer clic sobre el icono verde que aparece junto al número de línea → Run 'nombreTest' o pulsando Ctrl + Mayus +F10

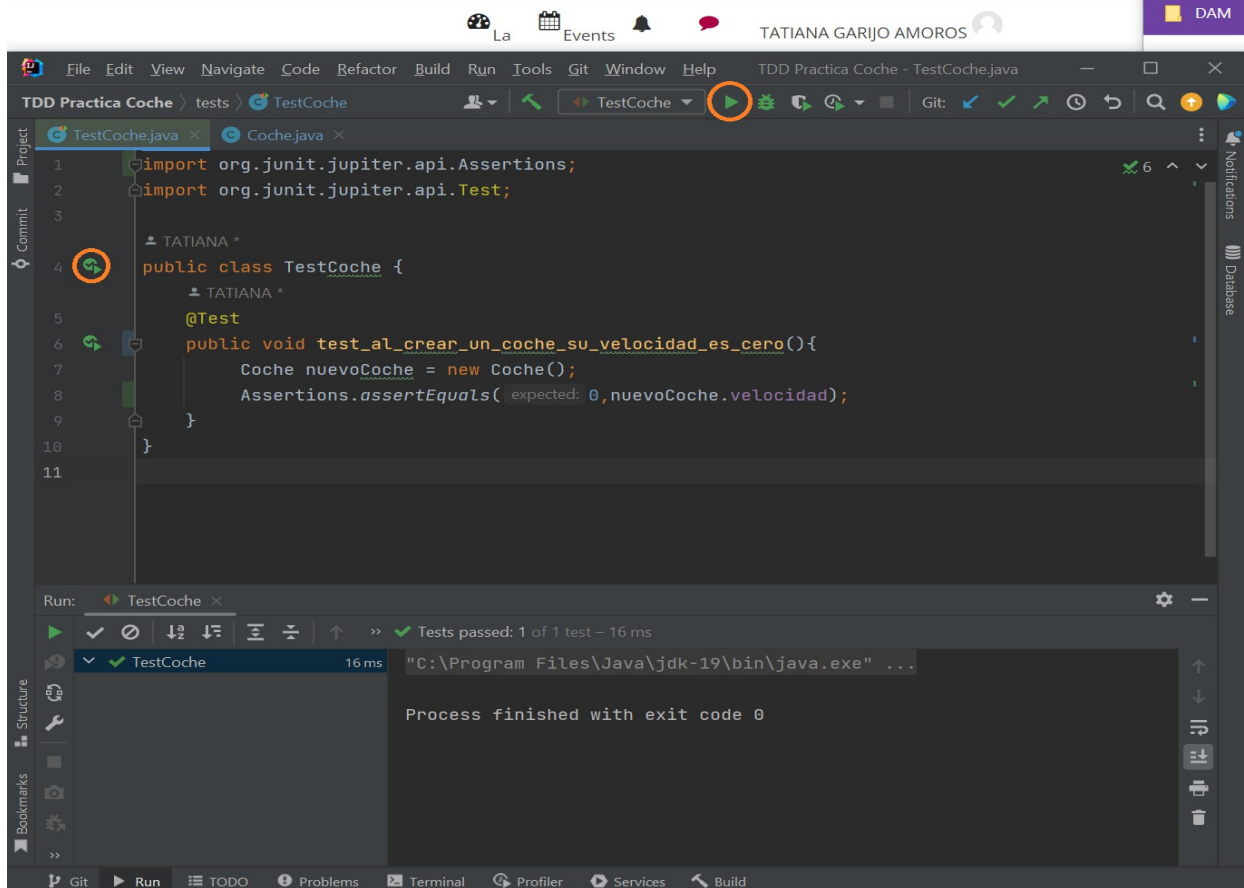


- **Modificar Test (valor inicial)**

- Cambiamos el nombre del método
- Añadir Assertion
 - Importar clase Assertion, clic sobre la palabra Assertion o Alt+Enter → 'Import class'
 - `Assertions.assertEquals(0,nuevoCoche.velocidad);`
 - El primer valor introducido (0) corresponde al valor esperado y el valor (`nuevoCoche.velocidad`) es el real.
 - Estamos haciendo referencia a un atributo `.velocidad` que no ha sido creado, para crearlo hacer clic sobre la palabra `.velocidad` o Alt+Enter → Create field 'velocidad' in 'Coche'
 - En este caso elegimos que sea de tipo entero (int)



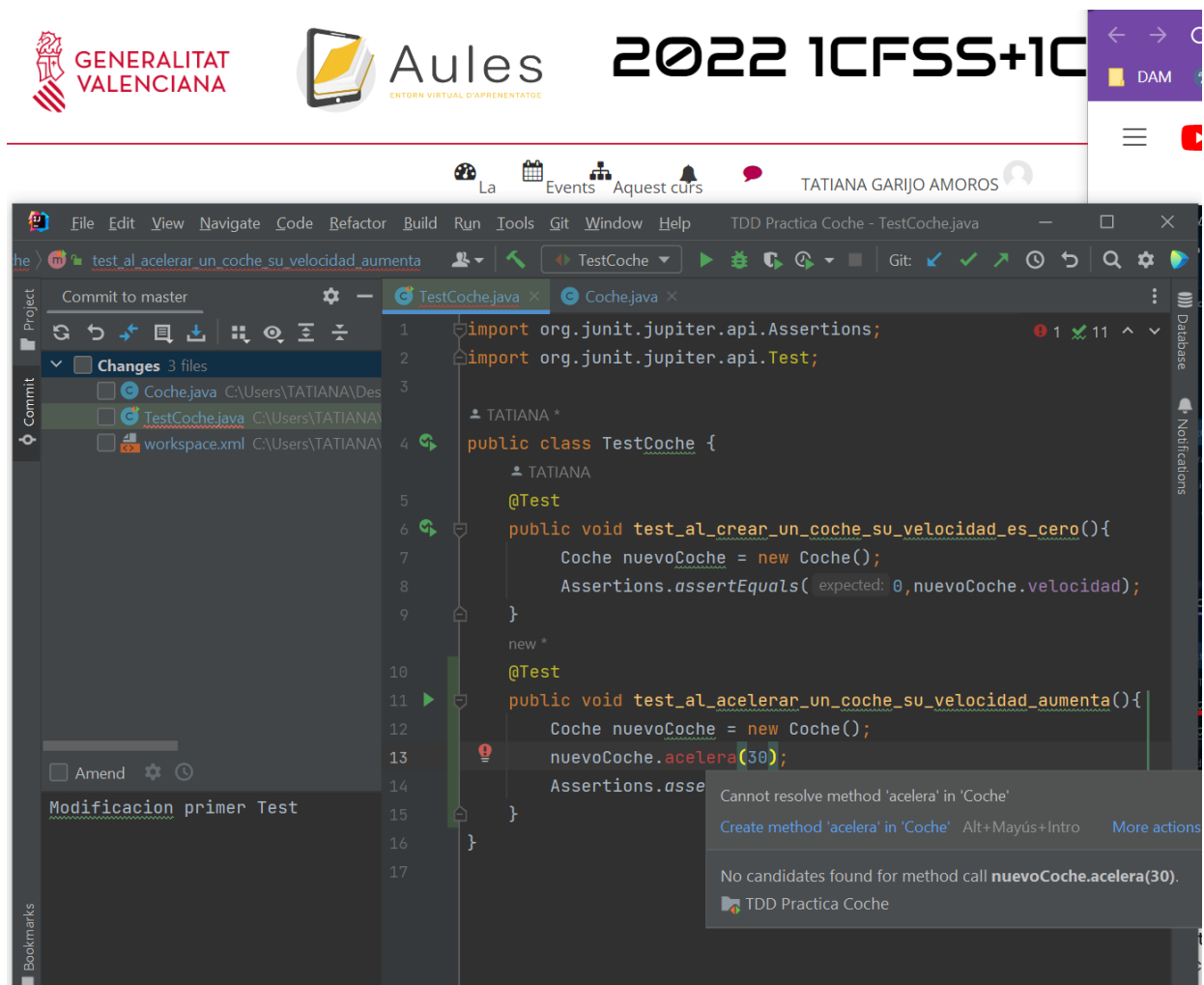
- Ejecutar Test



- Crear nuevo método(test incremento, suma)
- Hacer copy-paste del código del test que está creado
- Cambiar el nombre
- El método debe contener un objeto
- Una llamada a un método de la clase coche, al que se le pasa un valor de aceleración.



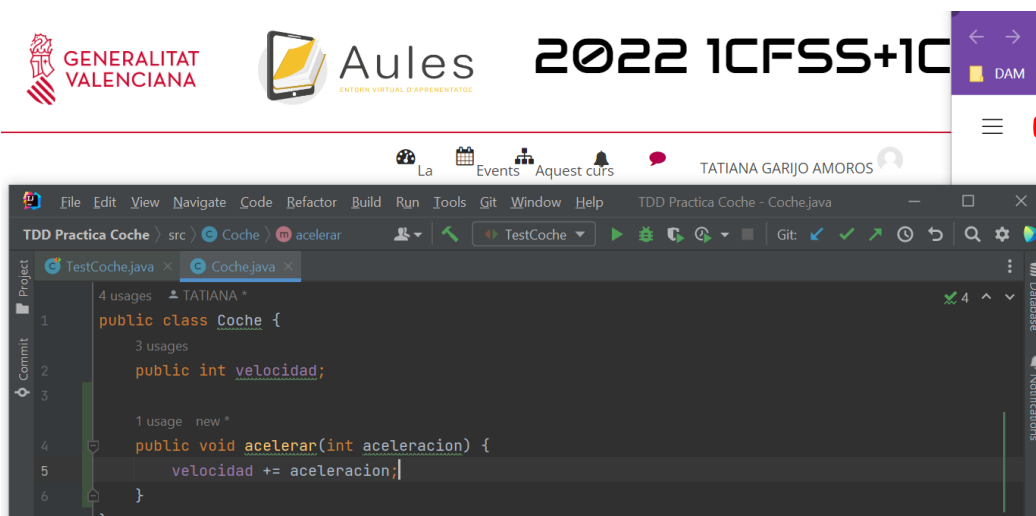
2022 1CFSS+1C



- Estamos haciendo referencia a un método que no ha sido creado, para crearlo hacer clic sobre la palabra el nombre del método (marcado en rojo) o Alt+Enter → Create method 'acelerar' in 'Coche'
 - En este caso elegimos que el valor que recibe el método sea de tipo entero (int) y cambiamos el nombre del parámetro.



2022 1CFSS+1C

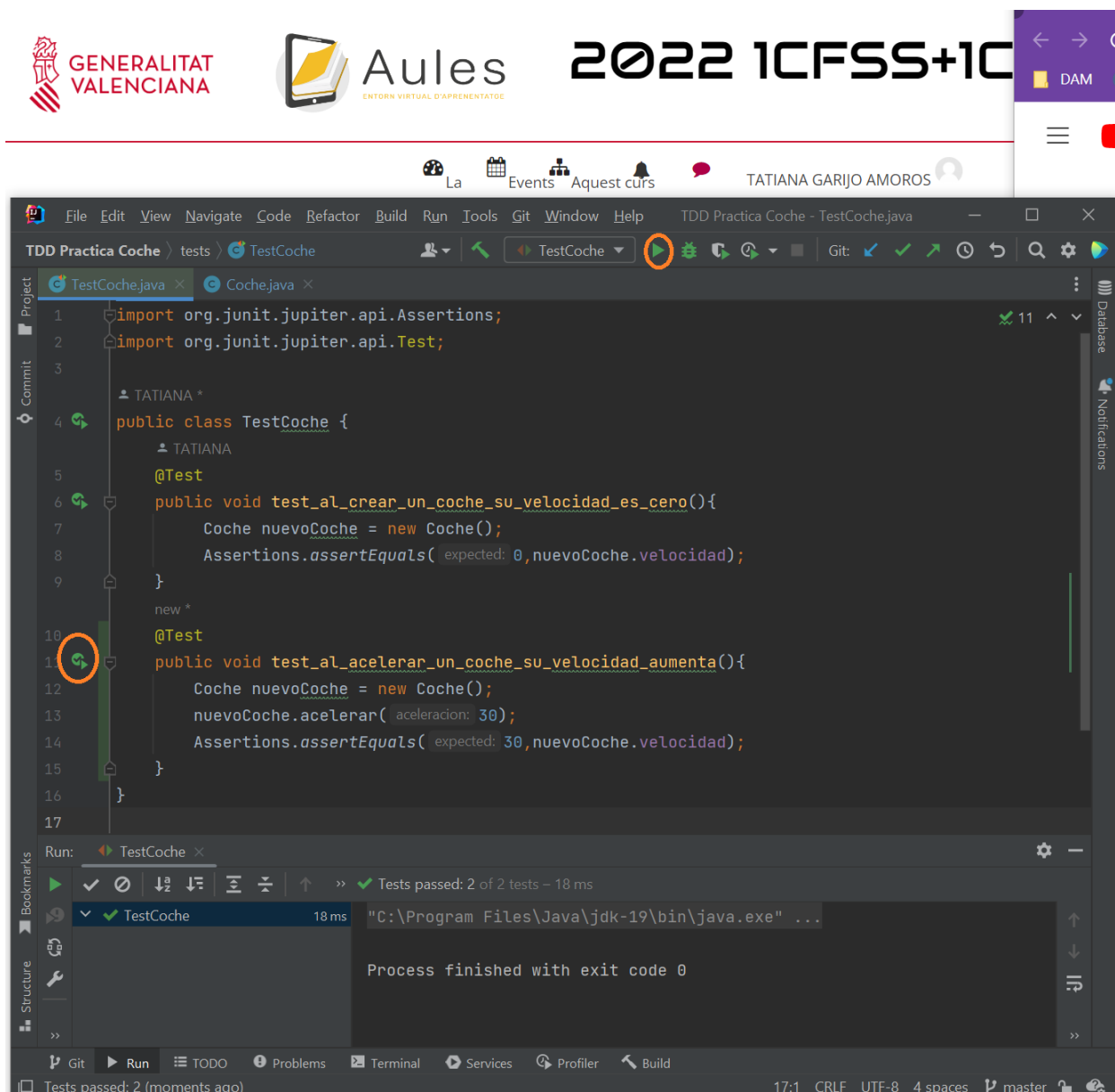


- En el método acelerar aumentar el valor de la variable local 'velocidad' con el valor recibido 'aceleracion'

• Añadir Assertion

- El primer valor introducido (30) corresponde al valor esperado y el valor (nuevoCoche.velocidad) es el real, en este caso el valor de velocidad se ha aumentado con el valor pasado en la llamada al método acelerar.

• Ejecutar test

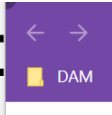


- Crear nuevo método(test decremento, resta)
- Hacer copy-paste del código del test que está creado
- Cambiar el nombre

- El método debe contener un objeto
- Dar a la variable velocidad un valor superior al decremento
- Una llamada a un método de la clase coche, al que se le pasa un valor de deceleración



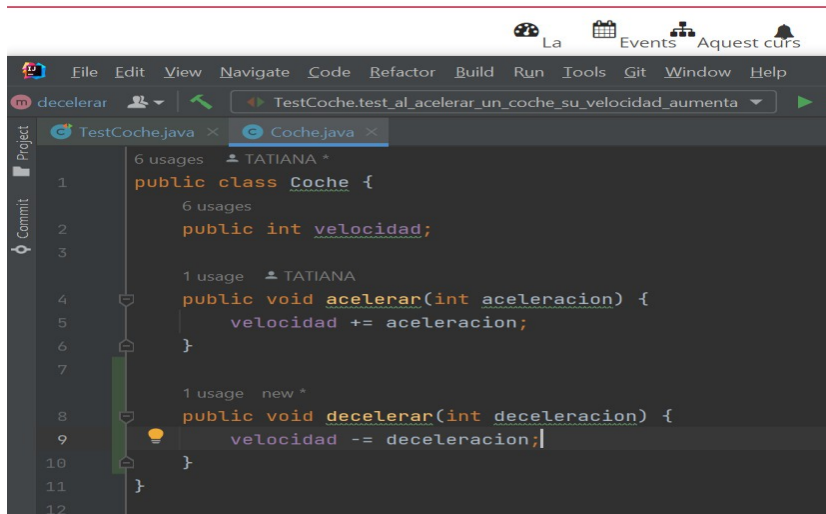
2022 1CFSS+1C



```

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    @Test
11    public void test_al_acelerar_un_coche_su_velocidad_aumenta(){
12        Coche nuevoCoche = new Coche();
13        nuevoCoche.acelerar( aceleracion: 30);
14        Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
15    }
16    @Test
17    public void test_al_decelerar_un_coche_su_velocidad_disminuye(){
18        Coche nuevoCoche = new Coche();
19        nuevoCoche.velocidad= 50;
20        nuevoCoche.decelera(20);
21        Assertions.asse
22    }
23 }
  
```

- Estamos haciendo referencia a un método que no ha sido creado, para crearlo hacer clic sobre la palabra el nombre del método(marcado en rojo) o Alt+Enter → Create method 'decelerar' in 'Coche'
 - En este caso elegimos que el valor que recibe el método sea de tipo entero (int) y cambiamos el nombre del parámetro
- En el método decelerar disminuir el valor de la variable local 'velocidad' con el valor recibido 'deceleracion'



• Añadir Assertion

- El primer valor introducido (30) corresponde al valor esperado, es el cálculo de restar el **valor que se le pasa al método decelerar** al **valor asignado a velocidad**
- El valor (nuevoCoche.velocidad) es el real, en este caso el valor de velocidad se ha disminuido con el valor pasado en la llamada al método acelerar.

```

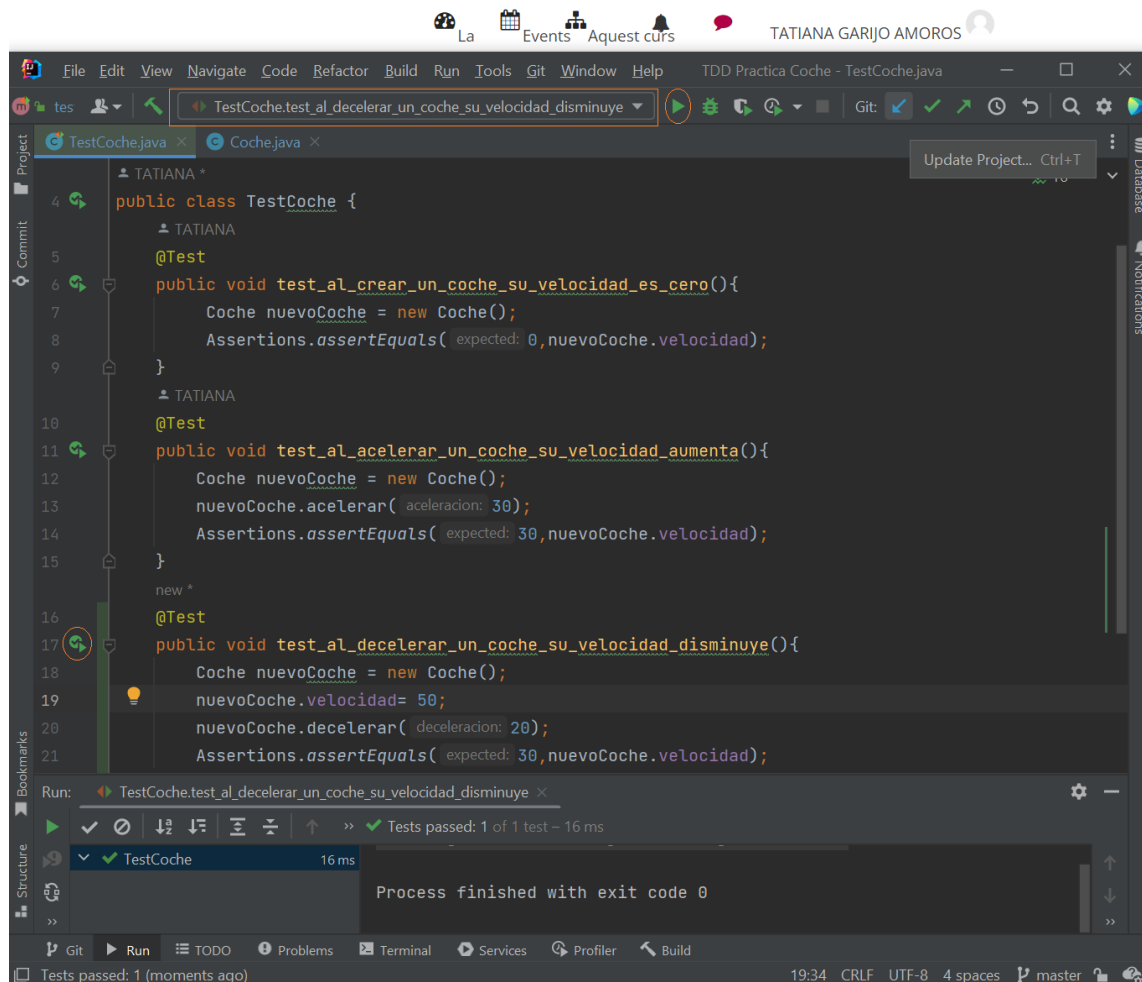
@Test
public void test_al_decelerar_un_coche_su_velocidad_disminuye(){
    Car nuevoCoche = new Car();
    nuevoCoche.velocidad= 50;
    nuevoCoche.decelerar( deceleracion: 20);
    Assertions.assertEquals( expected: 30 ,nuevoCoche.velocidad);
}

```

- Ejecutar Test



2022 1CFSS+1CFSC



- Crear nuevo método (decelerar no devuelva negativo)
- Hacer copy-paste del código del test 'decelerar'
- Cambiar el nombre
- El método debe contener un objeto
- Dar a la variable velocidad un valor inferior al decremento
- Una llamada al método 'decelerar' de la clase coche, al que se le pasa un valor de deceleración superior al valor de 'velocidad'

• Añadir Assertion

- El primer valor introducido (0) corresponde al valor esperado, no queremos que devuelva un valor negativo
- El valor (`nuevoCoche.velocidad`) es el real, en este caso el valor de velocidad se ha disminuido con el valor pasado en la llamada al método `acelerar`.

• Ejecutar Test

The screenshot shows an IDE with the following content:

GENERALITAT VALENCIANA **Aules** **2022 1CFSS+1C**

TestCoche.java

```

18 Coche nuevoCoche = new Coche();
19 nuevoCoche.velocidad= 50;
20 nuevoCoche.decelerar( deceleracion: 20);
21 Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
22 }
23
24 @Test
25 public void test_al_decelerar_un_coche_su_velocidad_no_puede_ser_menor_que_cero(){
26     Coche nuevoCoche = new Coche();
27     nuevoCoche.velocidad= 50;
28     nuevoCoche.decelerar( deceleracion: 80);
29     Assertions.assertEquals( expected: 0,nuevoCoche.velocidad);
30 }
31

```

Run: TestCoche

Tests failed: 1, passed: 3 of 4 tests - 16 ms

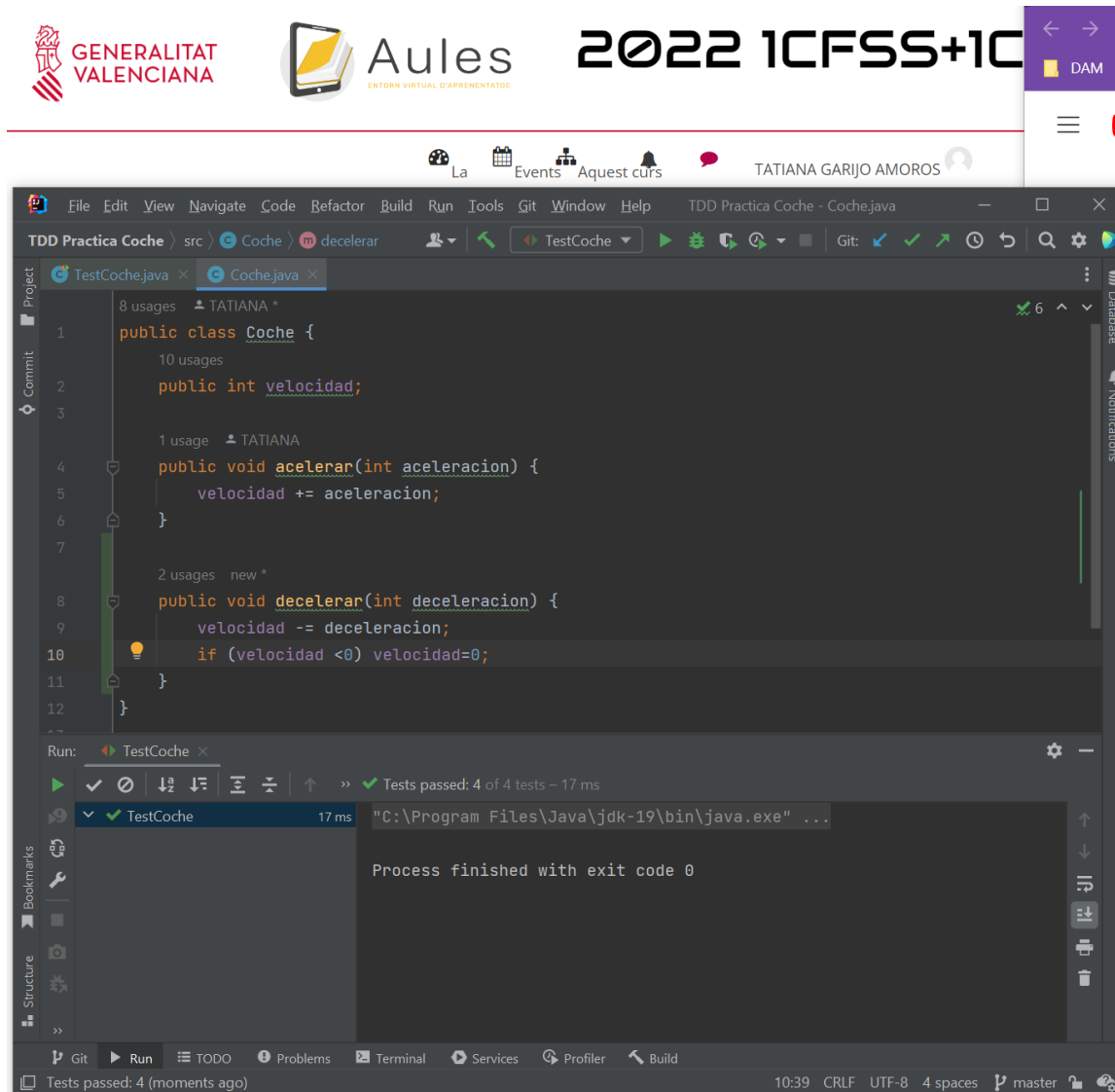
TestCoche 16 ms

test_al_decelerar_un_coche 16 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_m

- No pasa el test, el valor real no es 0 es -30
- En el método `decelerar` de la clase `Coche` debemos programar el código necesario para controlar que no devuelva un valor negativo cuando el valor 'deceleracion' sea mayor a 'velocidad', debe devolver 0
- En este caso:
 - *if (velocidad < 0) velocidad=0;*

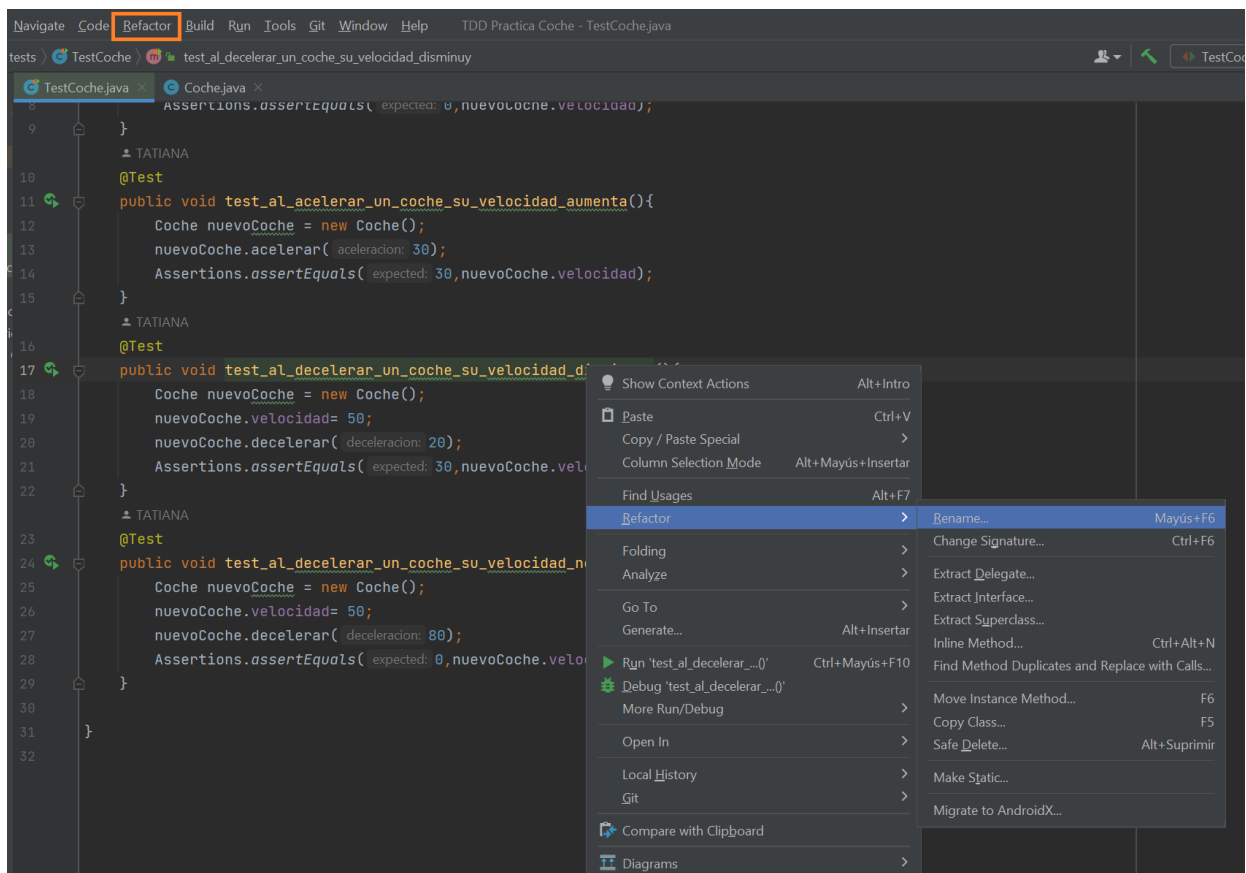
- Ejecutar Test



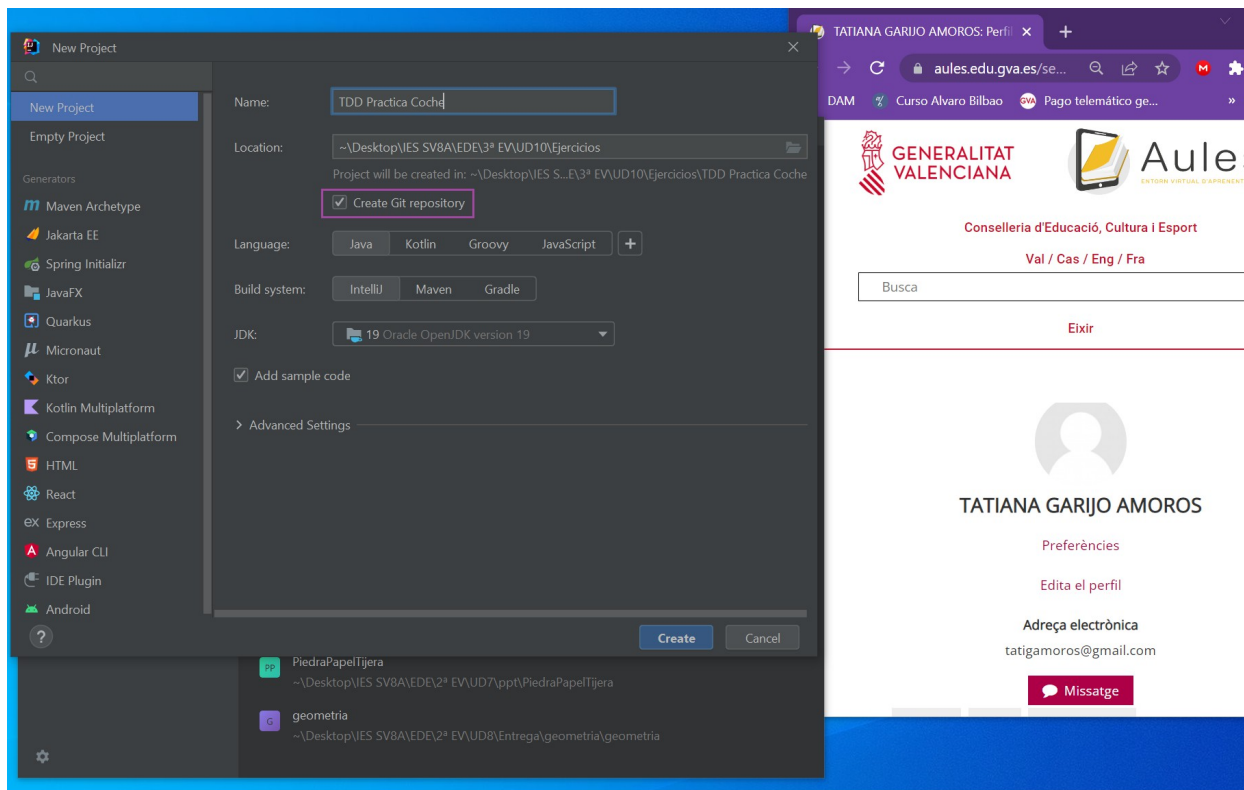
- Refactorizar en IntelliJ

- Hay varias formas de refactorizar:

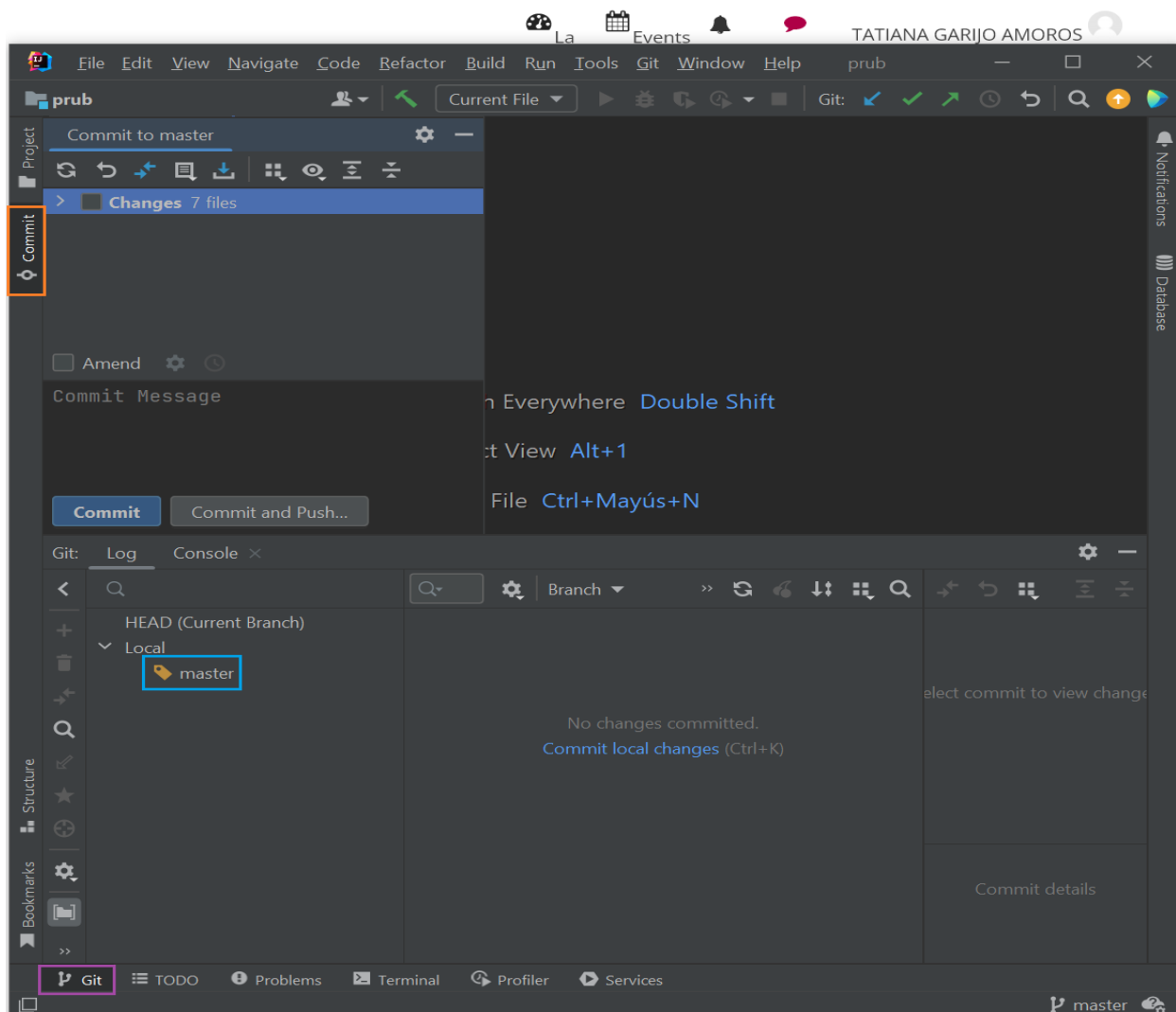
- Hacer clic sobre el menú superior **Refactor** → Rename
- Mayús + F6
- Hacer clic botón derecho sobre el elemento que queremos refactorizar → Refactor → Rename



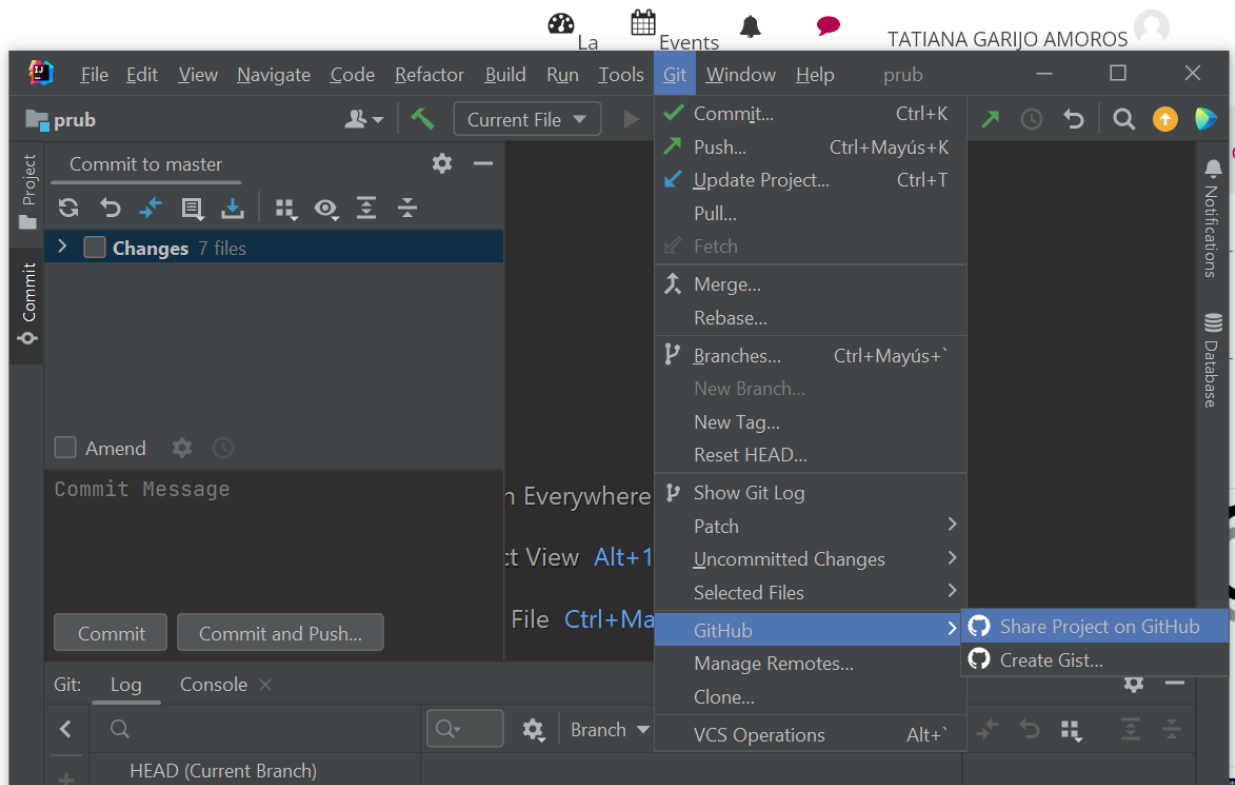
- VSC
- Al crear el proyecto he seleccionado la opción de crear un repositorio Git



- Durante el proceso realizar commit, haciendo uso del **panel lateral izquierdo 'commit'**
 - Seleccionar los archivos modificados y los que se vayan a incluir en el commit
 - Añadir mensaje del commit
 - Hacer clic sobre el botón commit
- Haciendo uso del **panel inferior izquierdo 'Git'** podemos crear nuevas ramas y hacer cambios entre ramas
 - Hacer clic con **botón derecho sobre la rama** y elegir la opción deseada 'new branch from nombreRama' o 'checkout'



- Subir repositorio a Github



- Hacer clic sobre el menú superior 'Git' → GitHub → Share Project on GitHub

- Dar nombre al repositorio y hacer clic sobre el botón 'Share'

