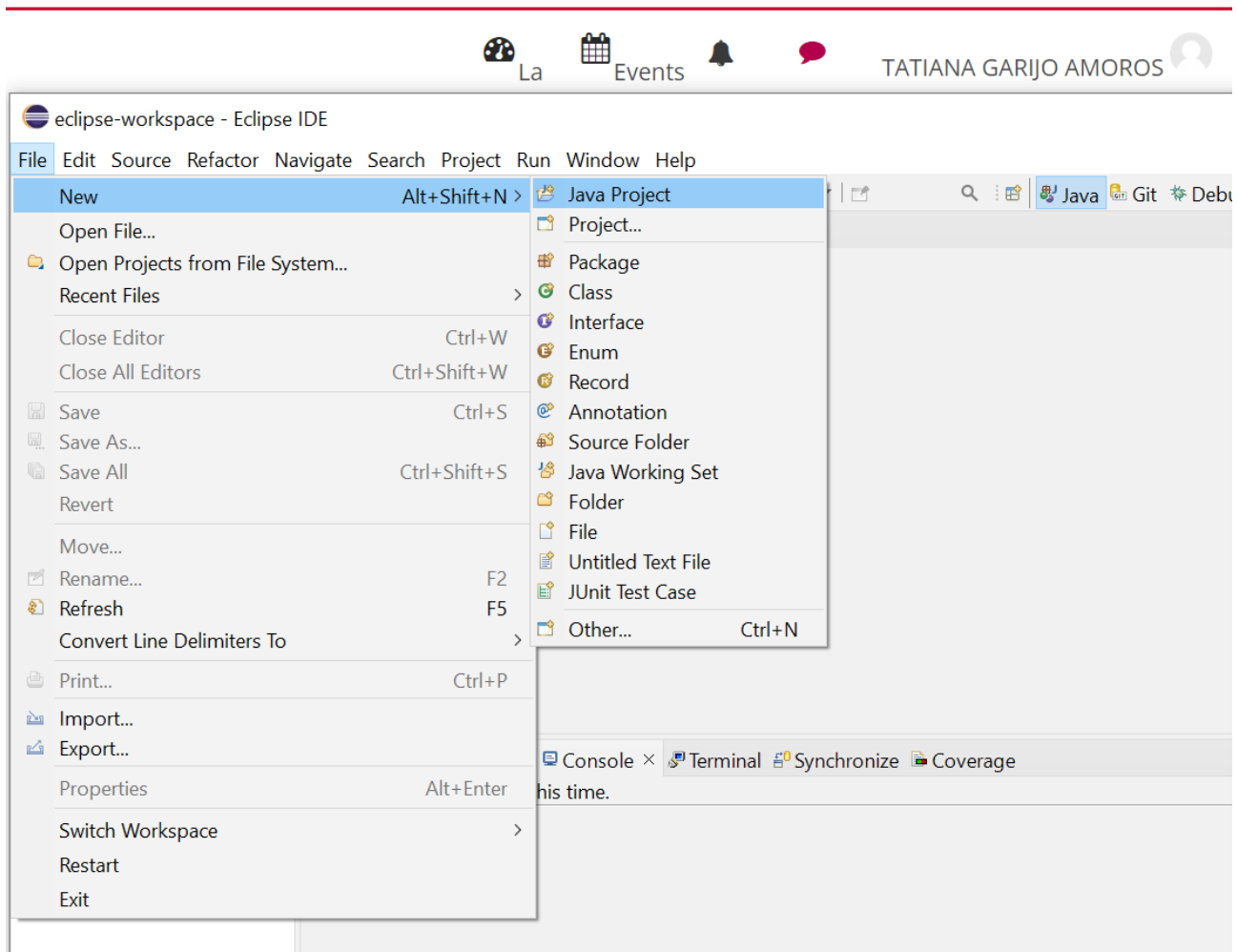


PRÁCTICA

Mi primer TDD v2.0

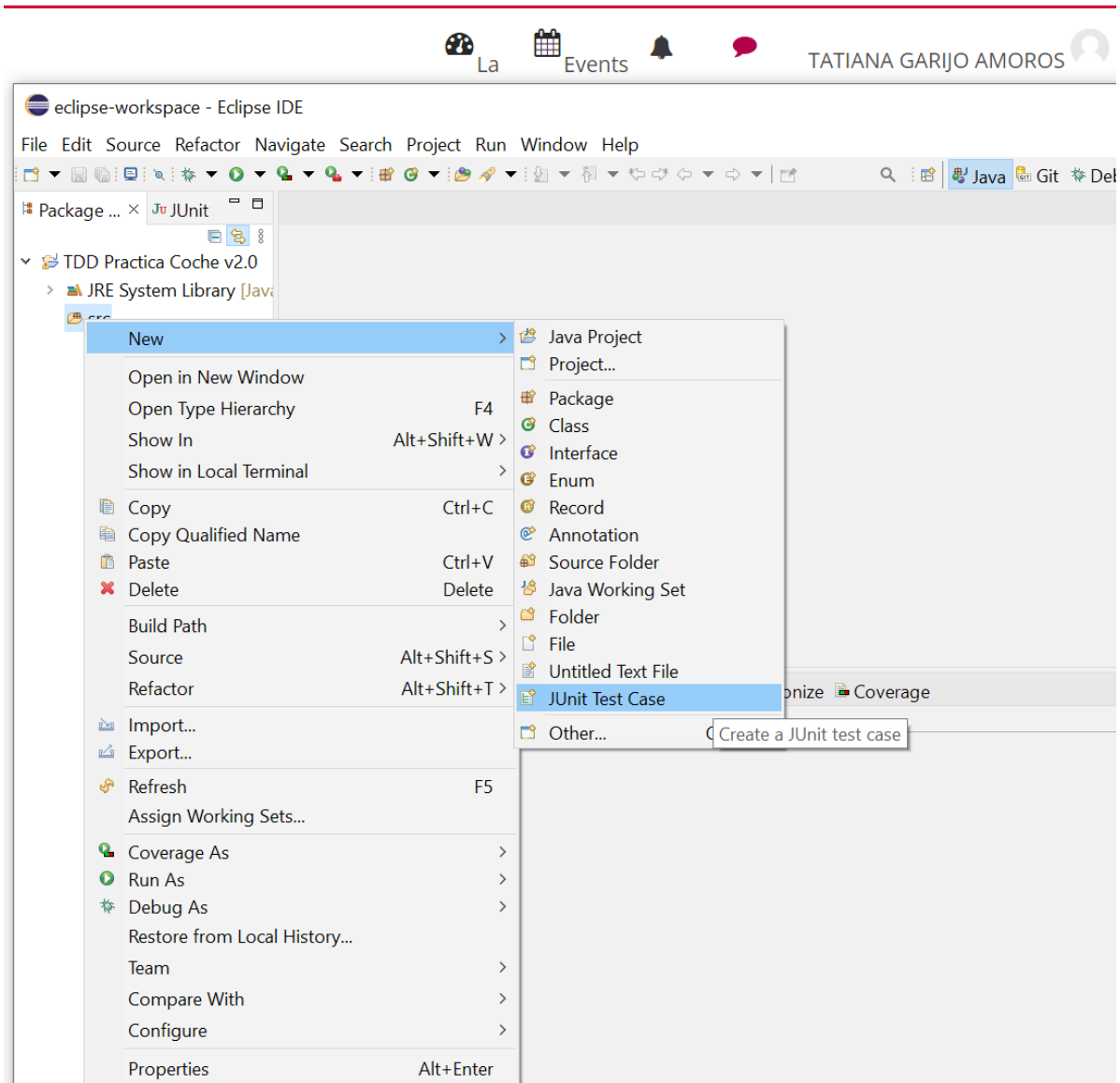
- New Project Eclipse



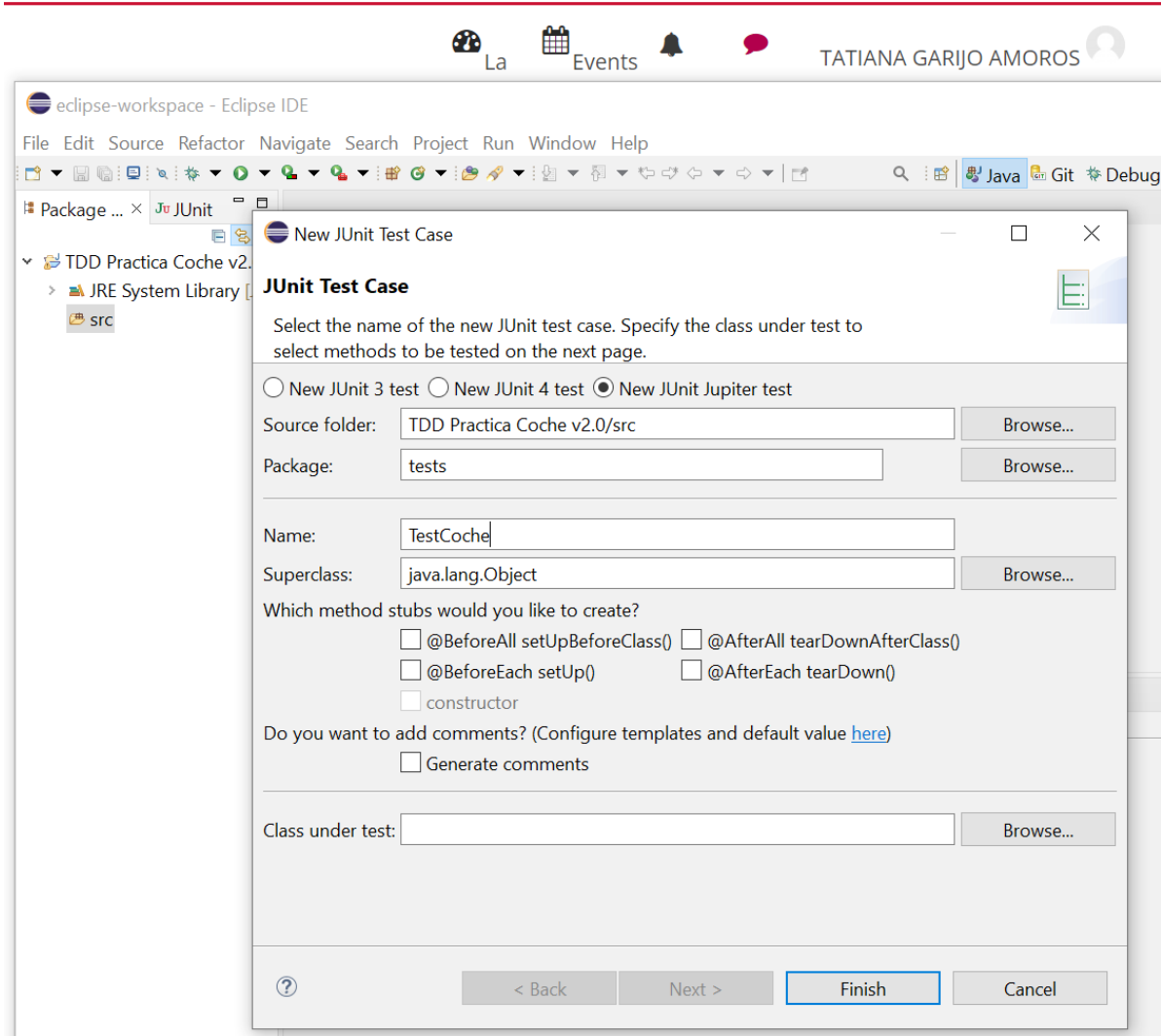
- Abrir Eclipse
- Seleccionar en el menú superior File -> New -> Java Project
- Configurar las características del proyecto

- En mi caso he seleccionado las opciones por defecto
 - He cambiado el nombre del proyecto
 - Utilizar un entorno de ejecución JRE → JavaSE-18
 - crear carpetas separadas para fuentes y archivos de clase
 - Working set de mi elección

- Crear un nuevo JUnit Test Case

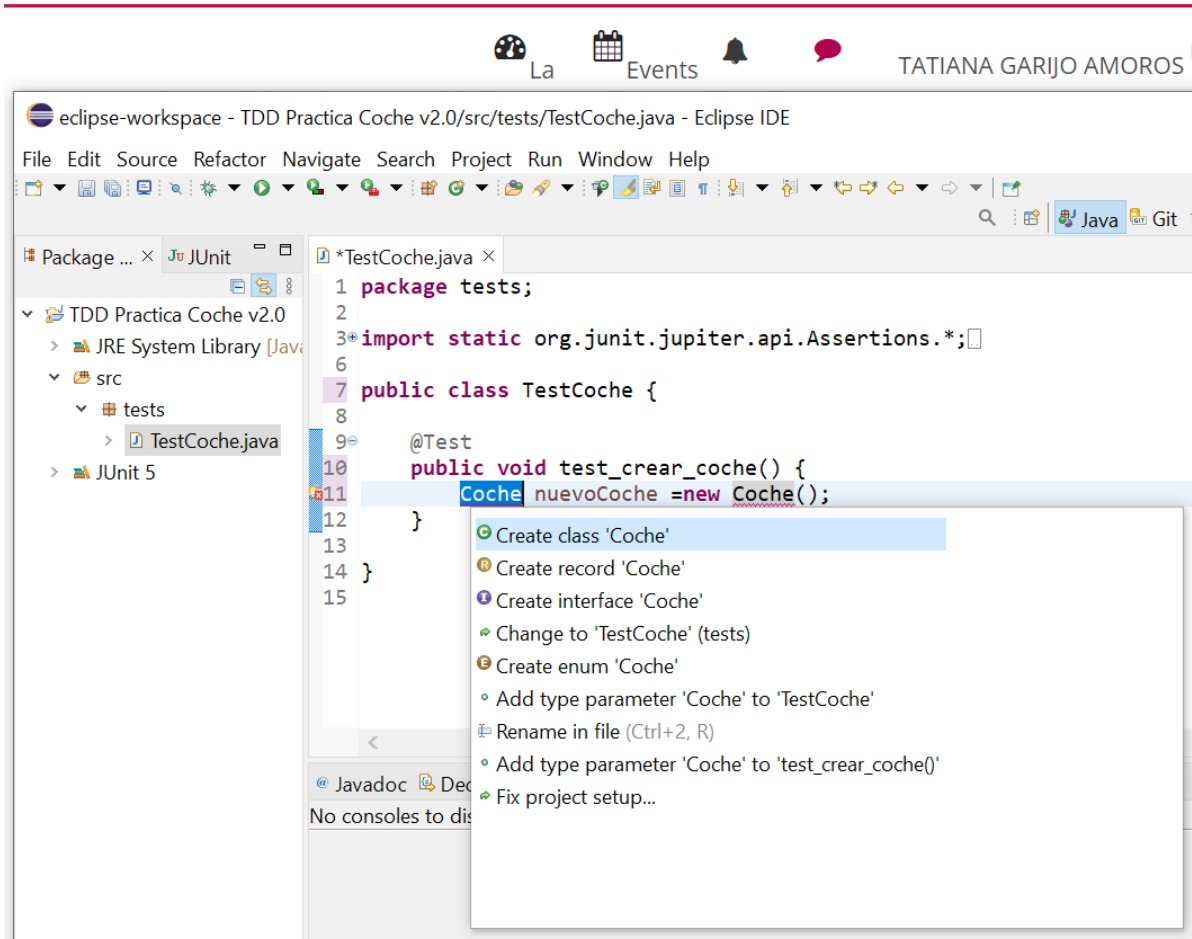
GENERALITAT
VALENCIANAAules
ENTORN VIRTUAL D'APRENTATGE

- Hacer clic con botón derecho sobre el directorio src → New → JUnit Test Case

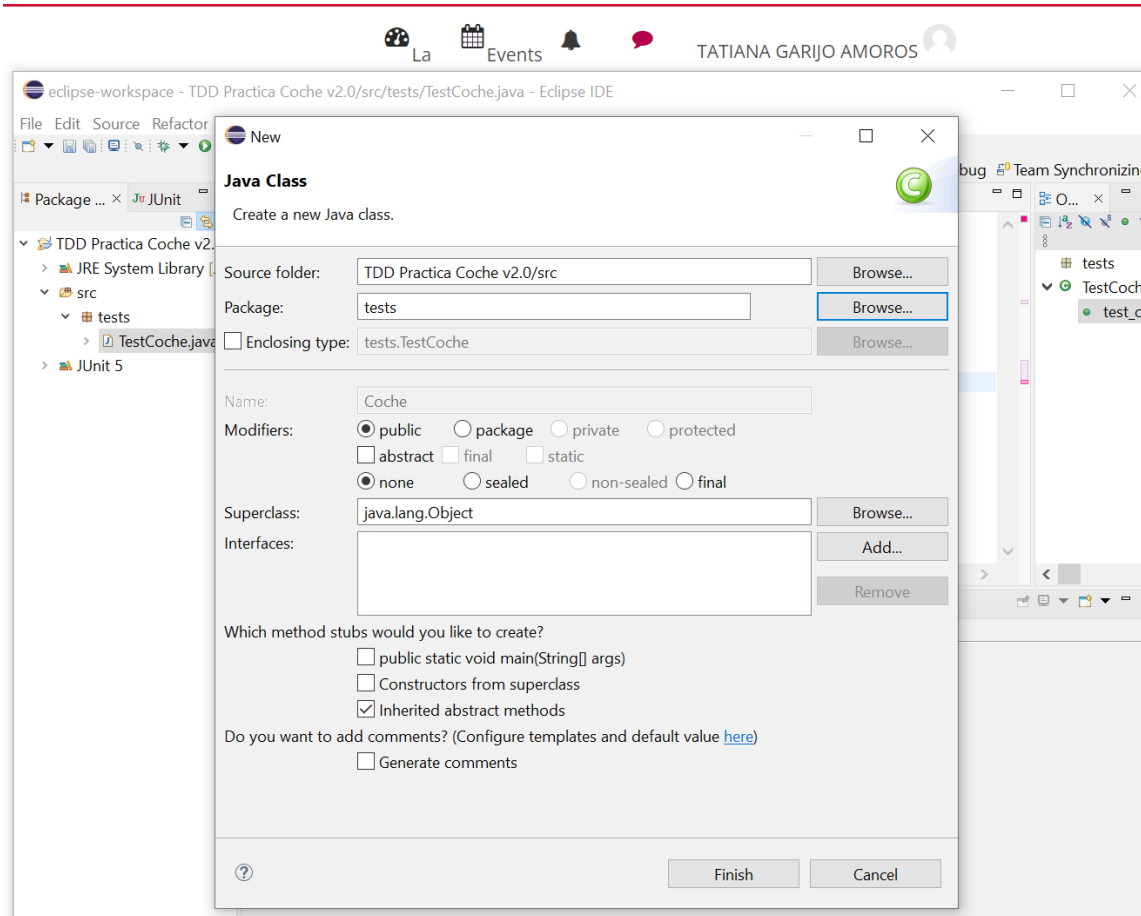


- En la ventana emergente que aparece configuramos el nombre del package, el nombre del test y en este caso seleccionamos que sea de tipo JUnit Jupiter test (JUnit 5)
- Hacer clic en Finish
- Nos puede aparecer una ventana emergente que nos pregunta si queremos añadir la librería JUnit 5 en el caso de no tenerla añadida previamente.

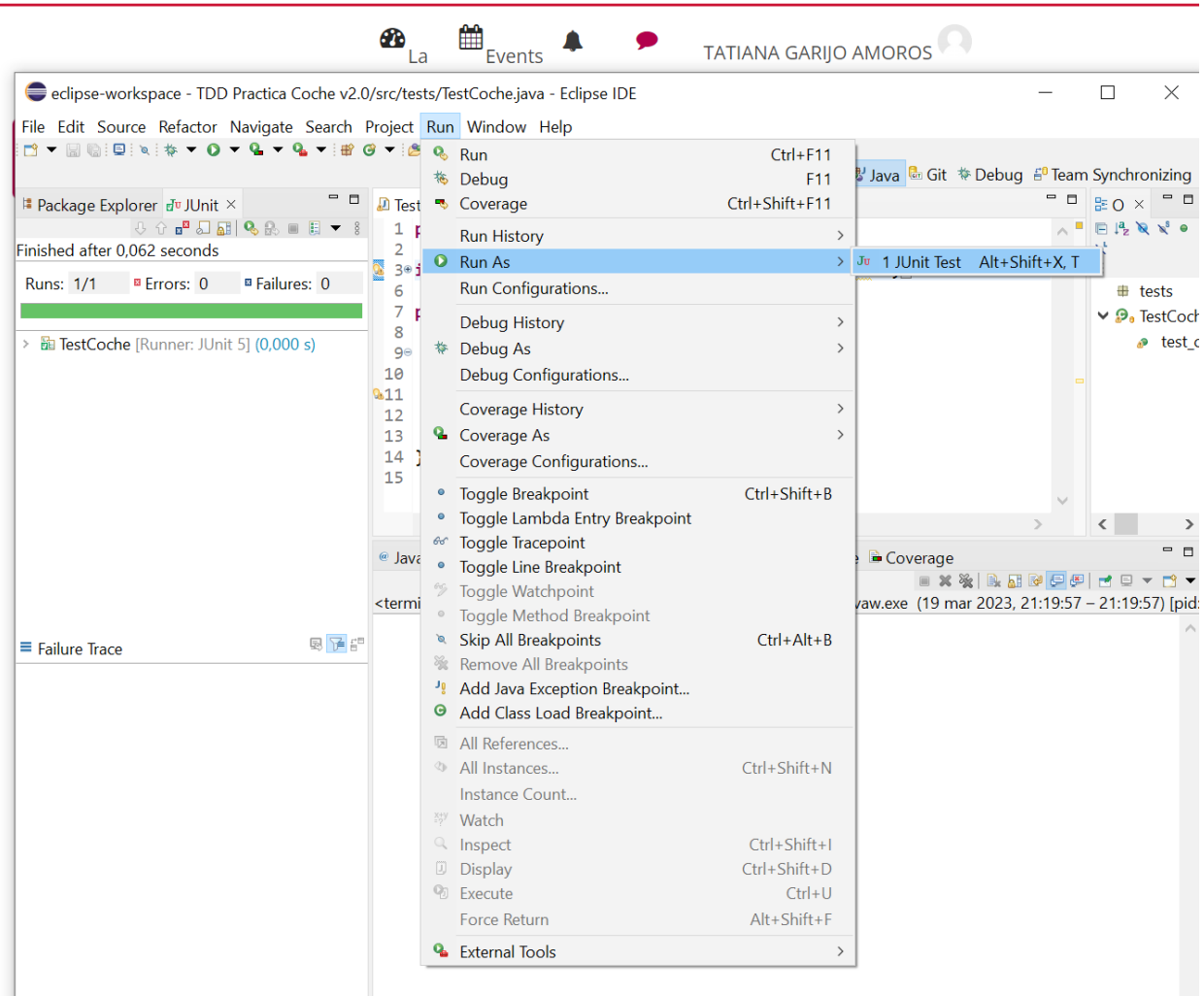
- Crear primer test



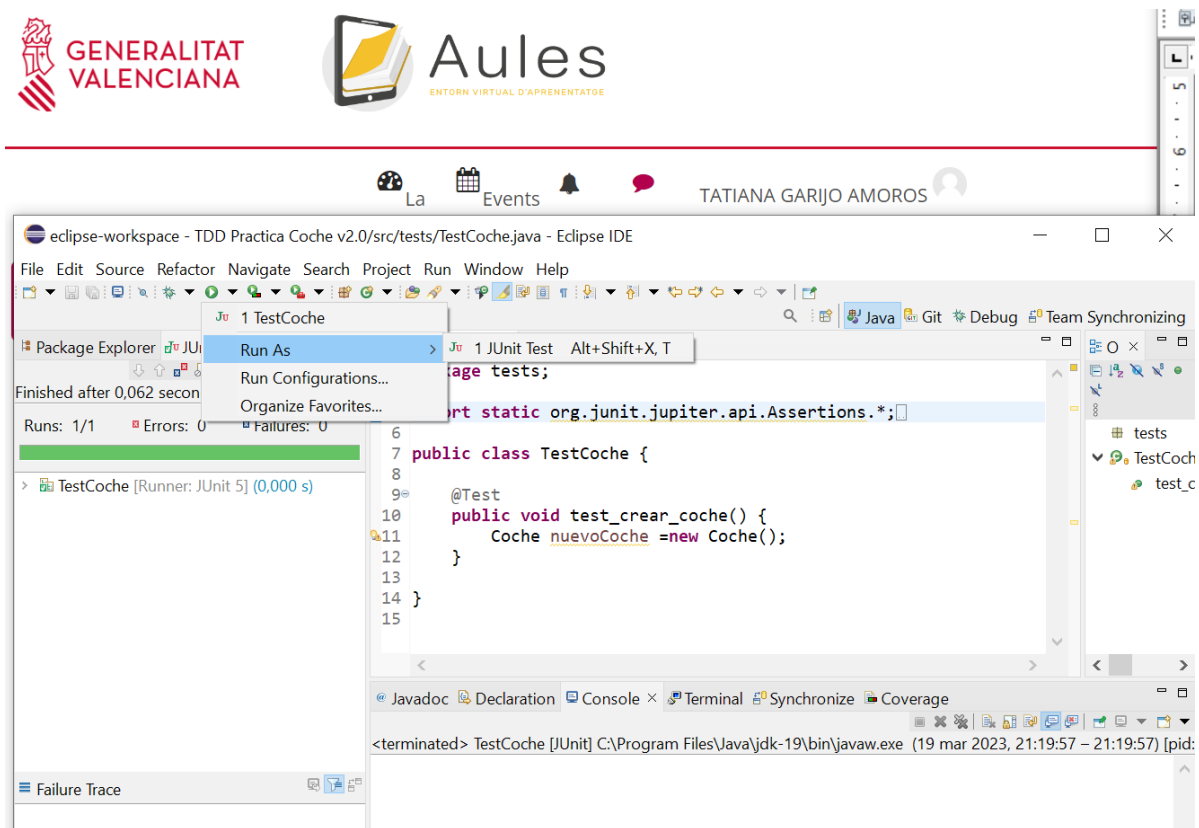
- Anotar antes del método @Test
- Nombrar el método
- Crear un objeto (el objeto pertenece a una clase que está pendiente de ser creada)
- Crear clase
 - Hacer clic sobre el icono que aparece junto a los números de línea o posicionando el cursor sobre el nombre de la clase → Create class 'nombreClase'



- Ejecutar Test Junit

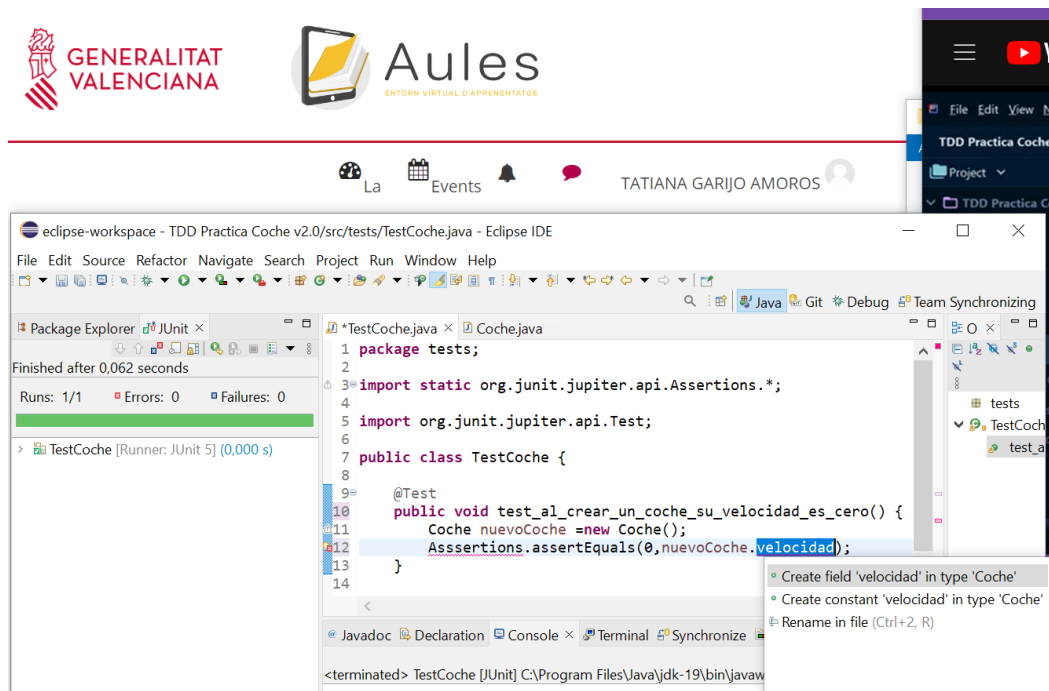


- Ejecutar Test
 - Acceder al menú superior Run → Run as → JUnit Test
 - Alt + Shift + X, T
 - Hacer clic sobre el icono play verde superior izquierda
 - Ejecutar 'nombreTest'
 - Run as → Junit Test

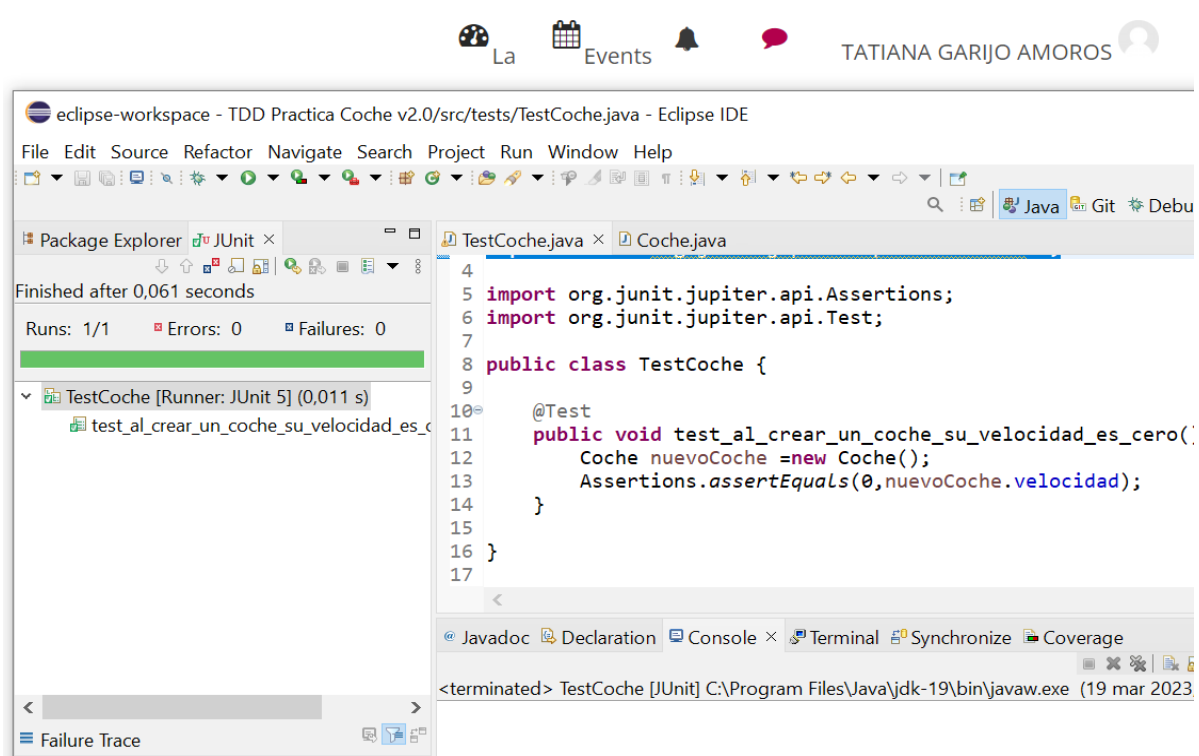


• Modificar Test (valor inicial)

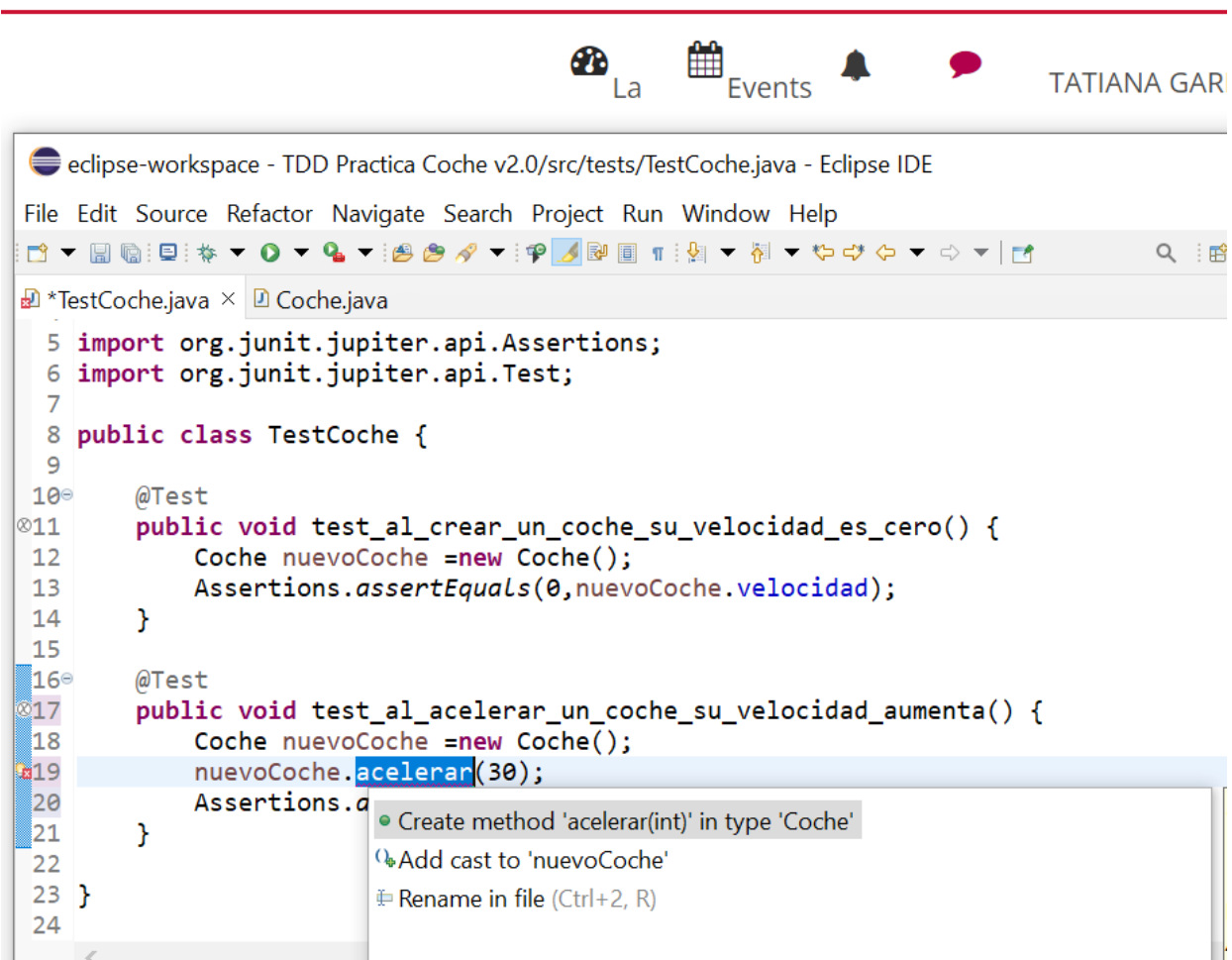
- Cambiamos el nombre del método
- Añadir Assertion
 - Importar clase Assertion, poner cursor sobre la palabra Assertion → 'Import Assertions'
 - `Assertions.assertEquals(0, nuevoCoche.velocidad);`
 - El primer valor introducido (0) corresponde al valor esperado y el valor (nuevoCoche.velocidad) es el real.
 - Estamos haciendo referencia a un atributo .velocidad que no ha sido creado, para crearlo hacer clic sobre la palabra .velocidad → Create field 'velocidad' in type '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



- Estamos haciendo referencia a un método que no ha sido creado, para crearlo hacer clic sobre el icono que aparece junto al número de línea o poner el cursor sobre la palabra el nombre del método(marcado en rojo) → Create method 'acelerar(int)' in type 'Coche'
 - En este caso elegimos que el valor que recibe el método sea de tipo entero (int), public y cambiamos el nombre del parámetro 'aceleracion'.



The screenshot shows the Eclipse IDE interface. The top bar includes icons for 'La', 'Events', a bell, a speech bubble, and the name 'TATIANA GARIJO AMOROS'. Below this is the Eclipse workspace title bar: 'eclipse-workspace - TDD Practica Coche v2.0/src/tests/Coche.java - Eclipse IDE'. The menu bar includes 'File', 'Edit', 'Source', 'Refactor', 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. The toolbar contains various icons for file operations, search, and development. The editor shows two tabs: '*TestCoche.java' and '*Coche.java'. The code in '*Coche.java' is as follows:

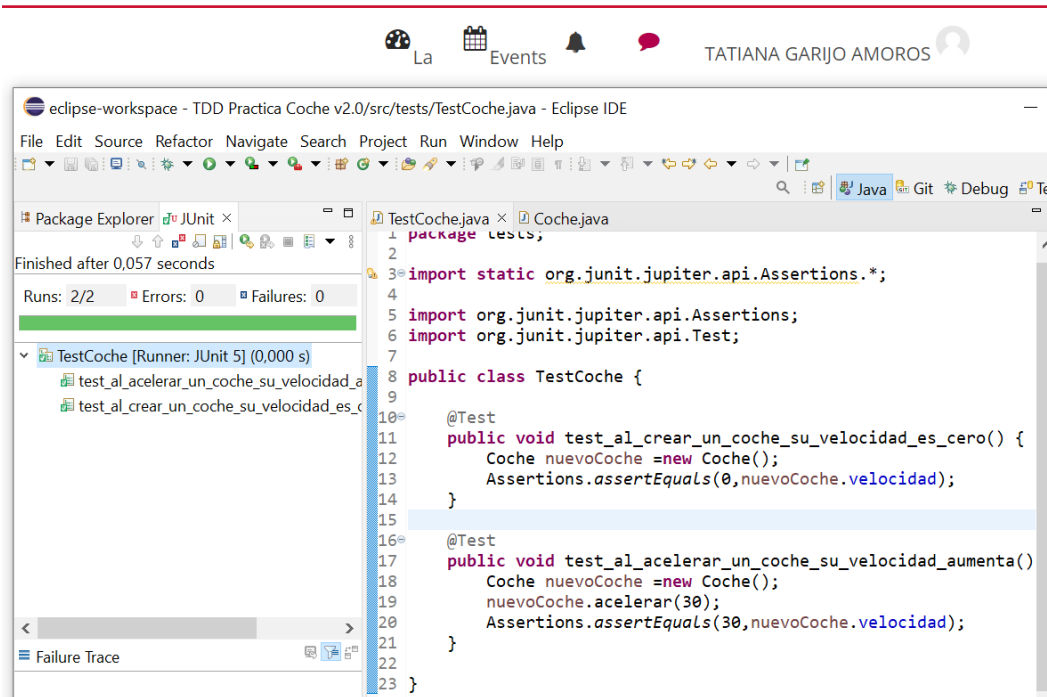
```
1 package tests;
2
3 public class Coche {
4     public int velocidad;
5
6     public void acelerar(int aceleracion) {
7         velocidad += aceleracion;
8     }
9 }
10
11
12
```

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

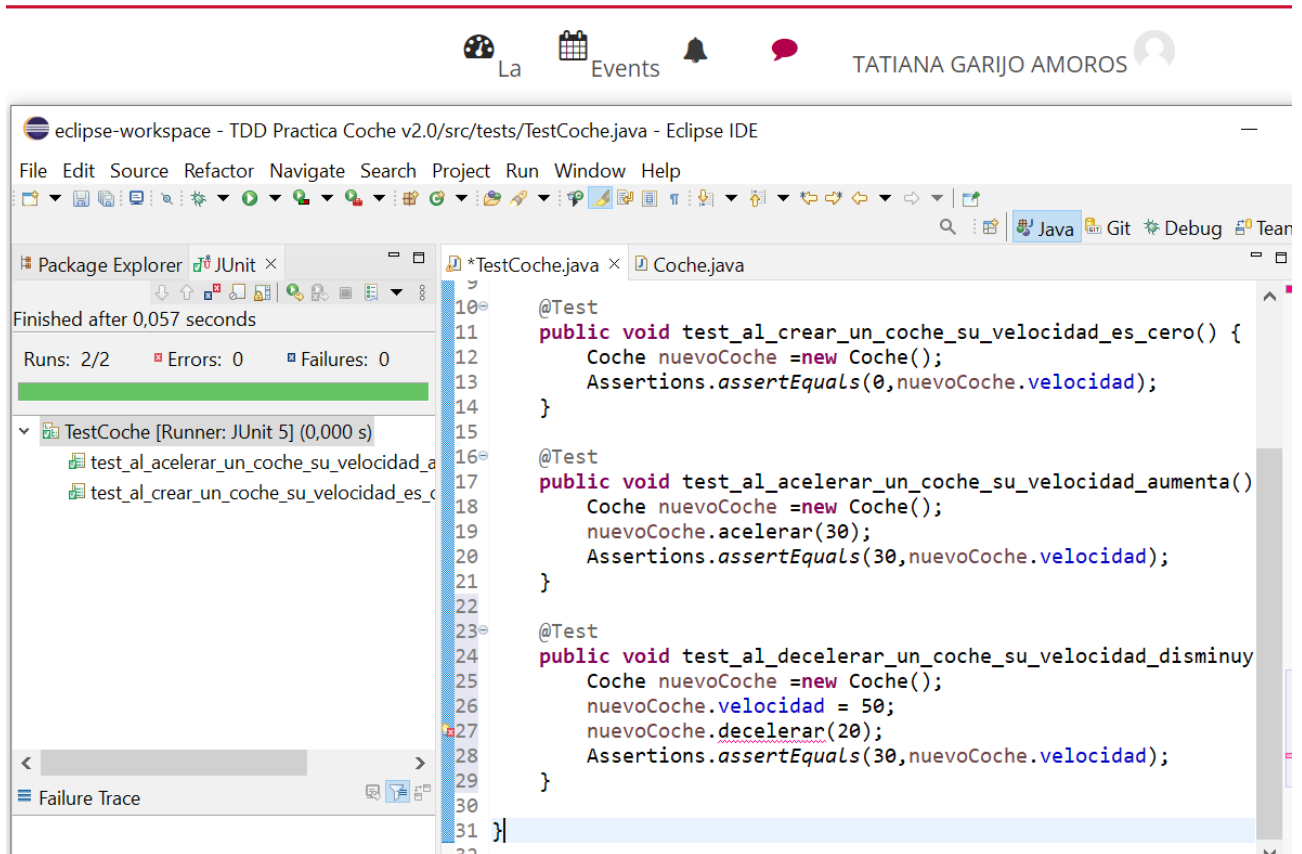
- **Añadir Assertion**

- 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.

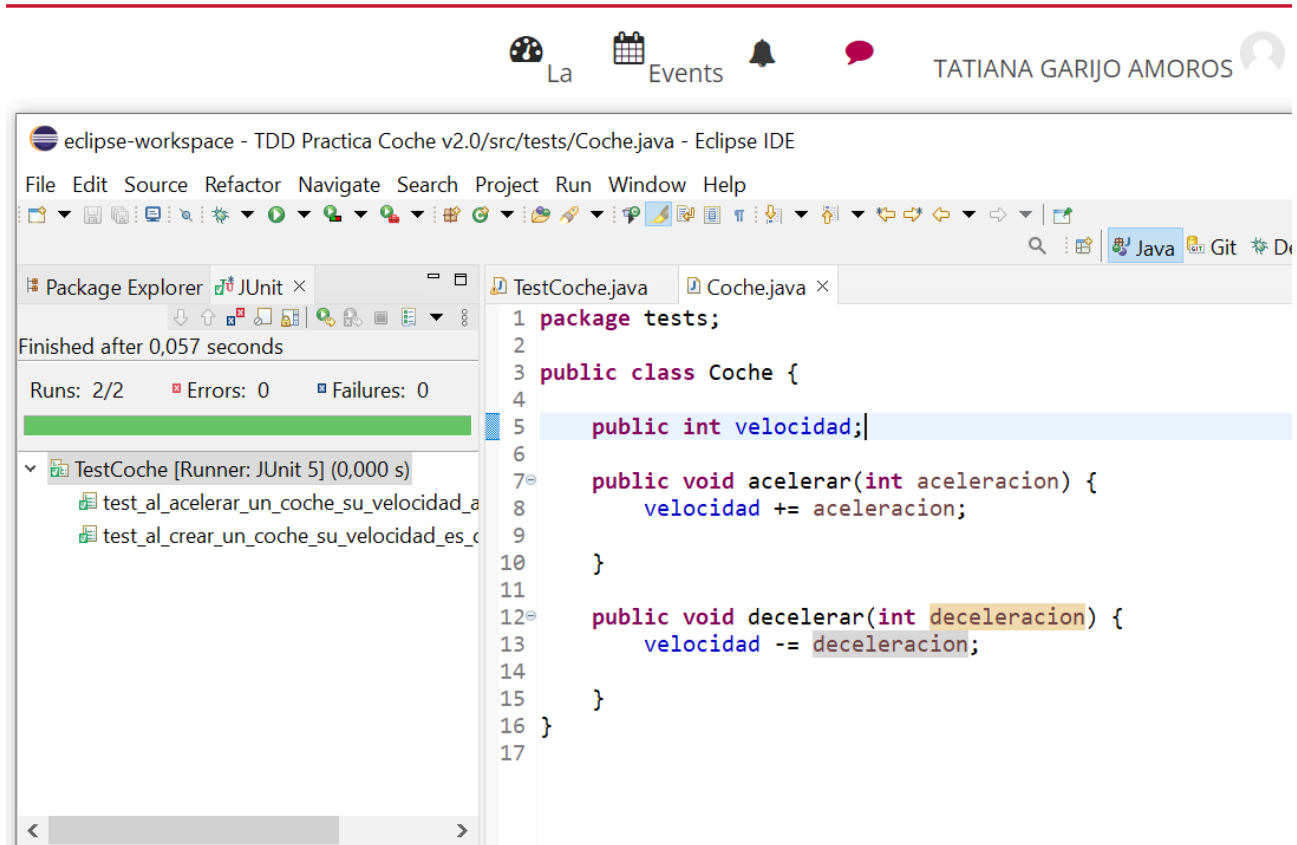
- Ejecutar Test



- 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



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

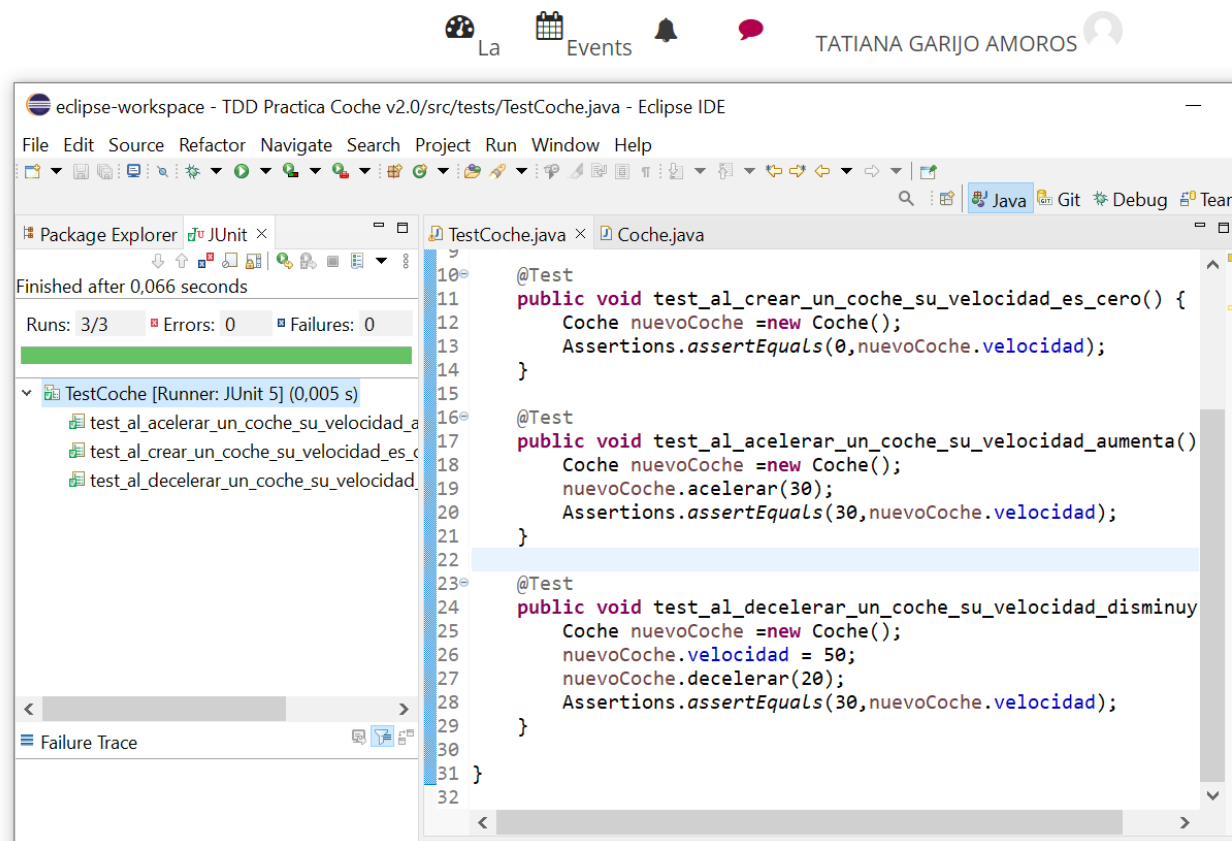


• Añadir Assertion

- 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

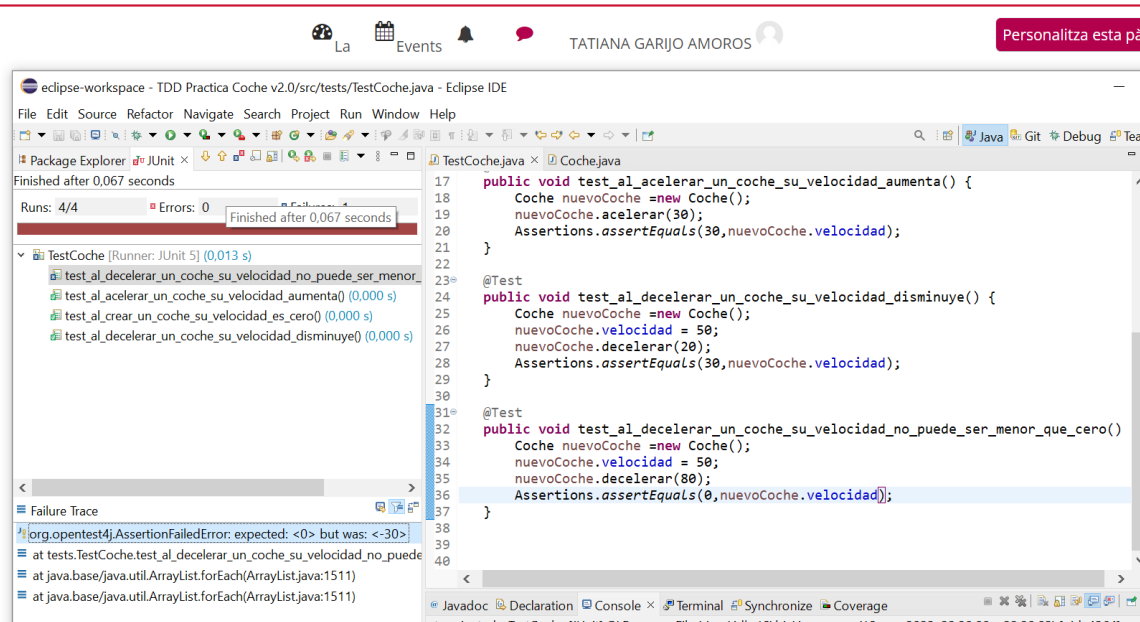
```
@Test
public void test_al_decelerar_un_coche_su_velocidad_dismin
    Coche nuevoCoche = new Coche();
    nuevoCoche.velocidad = 50;
    nuevoCoche.decelerar(20);
    Assertions.assertEquals(30, nuevoCoche.velocidad);
}
```

- Ejecutar Test



- 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



- 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



La Events TATIANA GARIJO AMOROS

eclipse-workspace - TDD Practica Coche v2.0/src/tests/Coche.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit x

Finished after 0,06 seconds

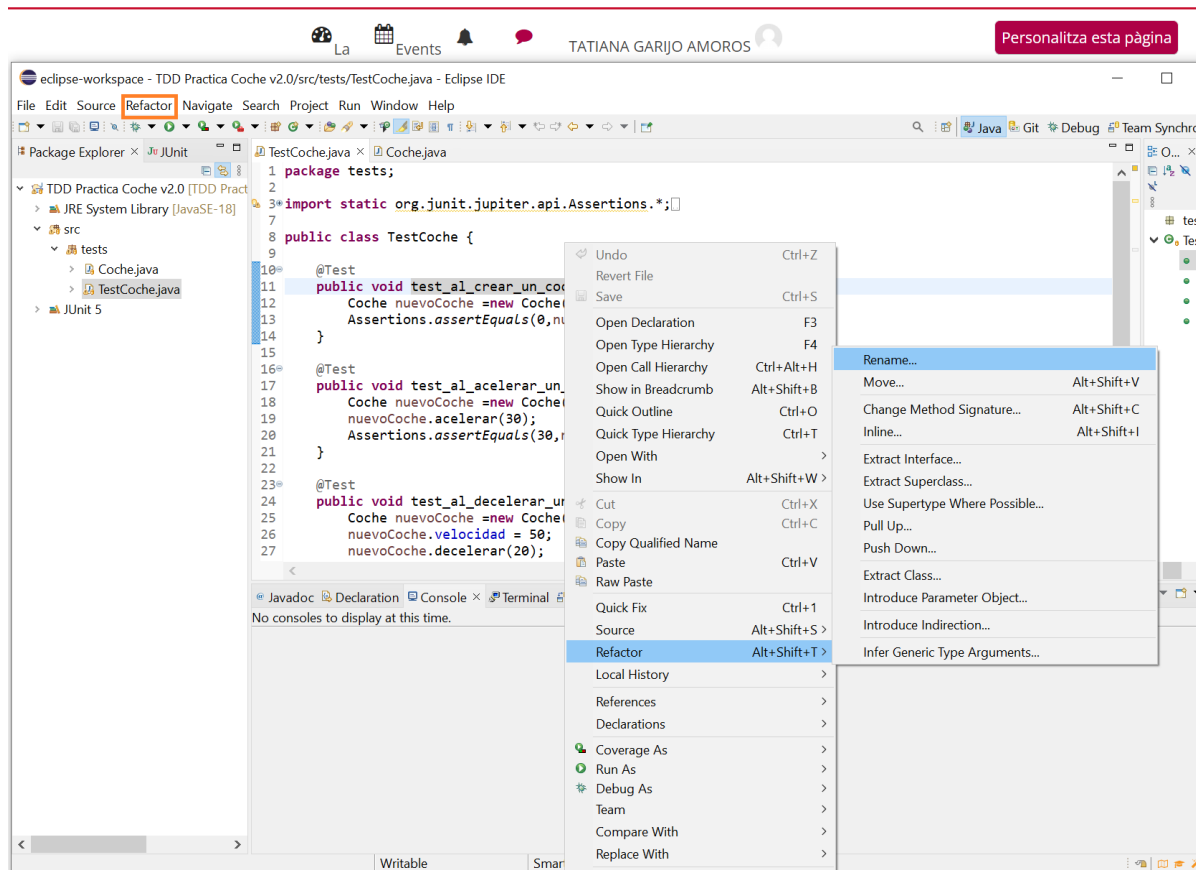
Runs: 4/4 Errors: 0 Failures: 0

TestCoche [Runner: JUnit 5] (0,010 s)

- test_al_decelerar_un_coche_su_velocidad_no_puede_ser_menor_ (0,001 s)
- test_al_acelerar_un_coche_su_velocidad_aumenta() (0,001 s)
- test_al_crear_un_coche_su_velocidad_es_cero() (0,001 s)
- test_al_decelerar_un_coche_su_velocidad_disminuye() (0,000 s)

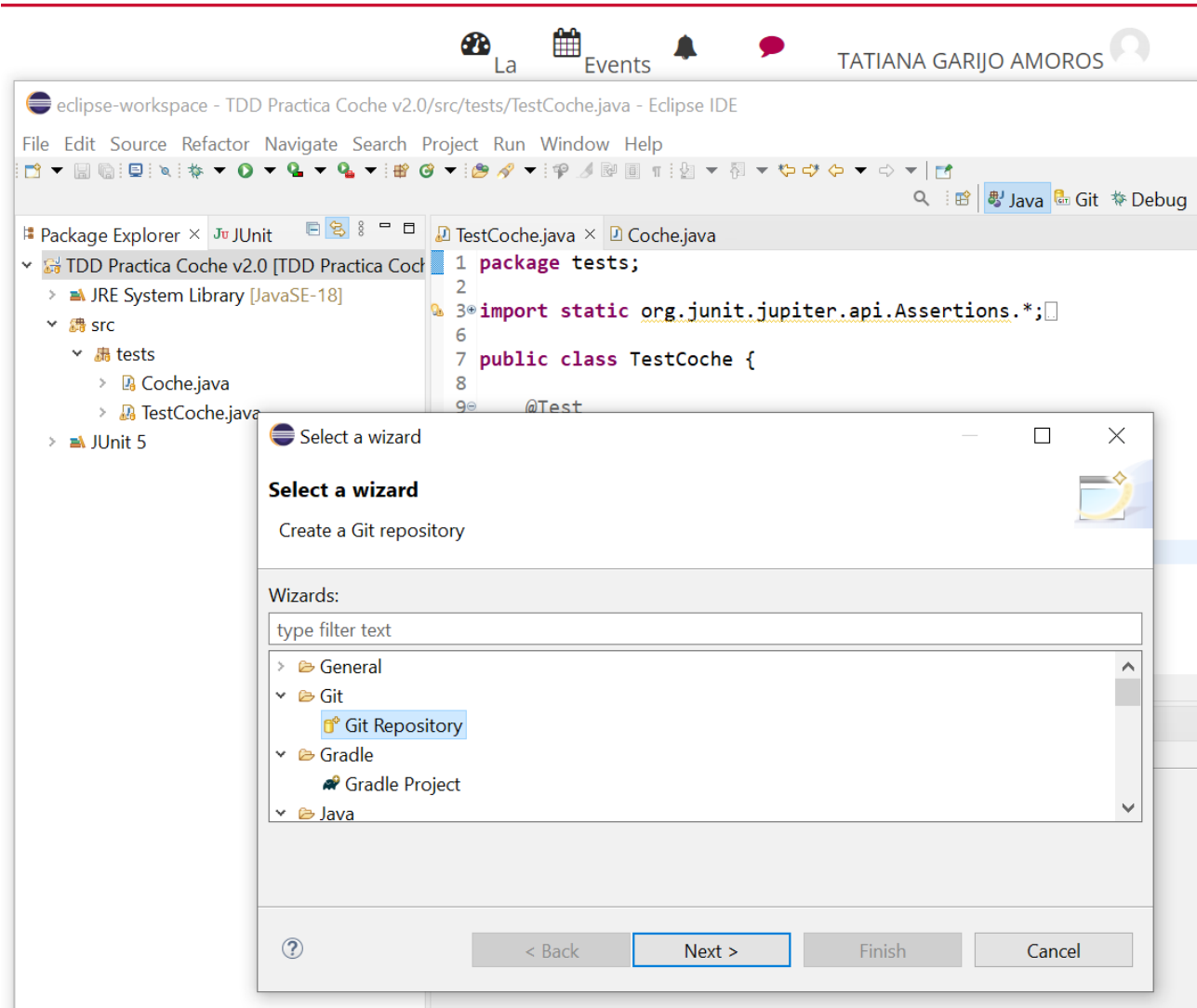
```
1 package tests;
2
3 public class Coche {
4
5     public int velocidad;
6
7     public void acelerar(int aceleracion) {
8         velocidad += aceleracion;
9     }
10
11
12     public void decelerar(int deceleracion) {
13         velocidad -= deceleracion;
14         if(velocidad<0) velocidad=0;
15     }
16 }
17
18
```


- Refactorizar en Eclipse
- Hay varias formas de refactorizar:
 - Hacer clic sobre el menú superior **Refactor** → Rename
 - Alt + Shift + T → Rename
 - Hacer clic botón derecho sobre el elemento que queremos refactorizar → Refactor → Rename

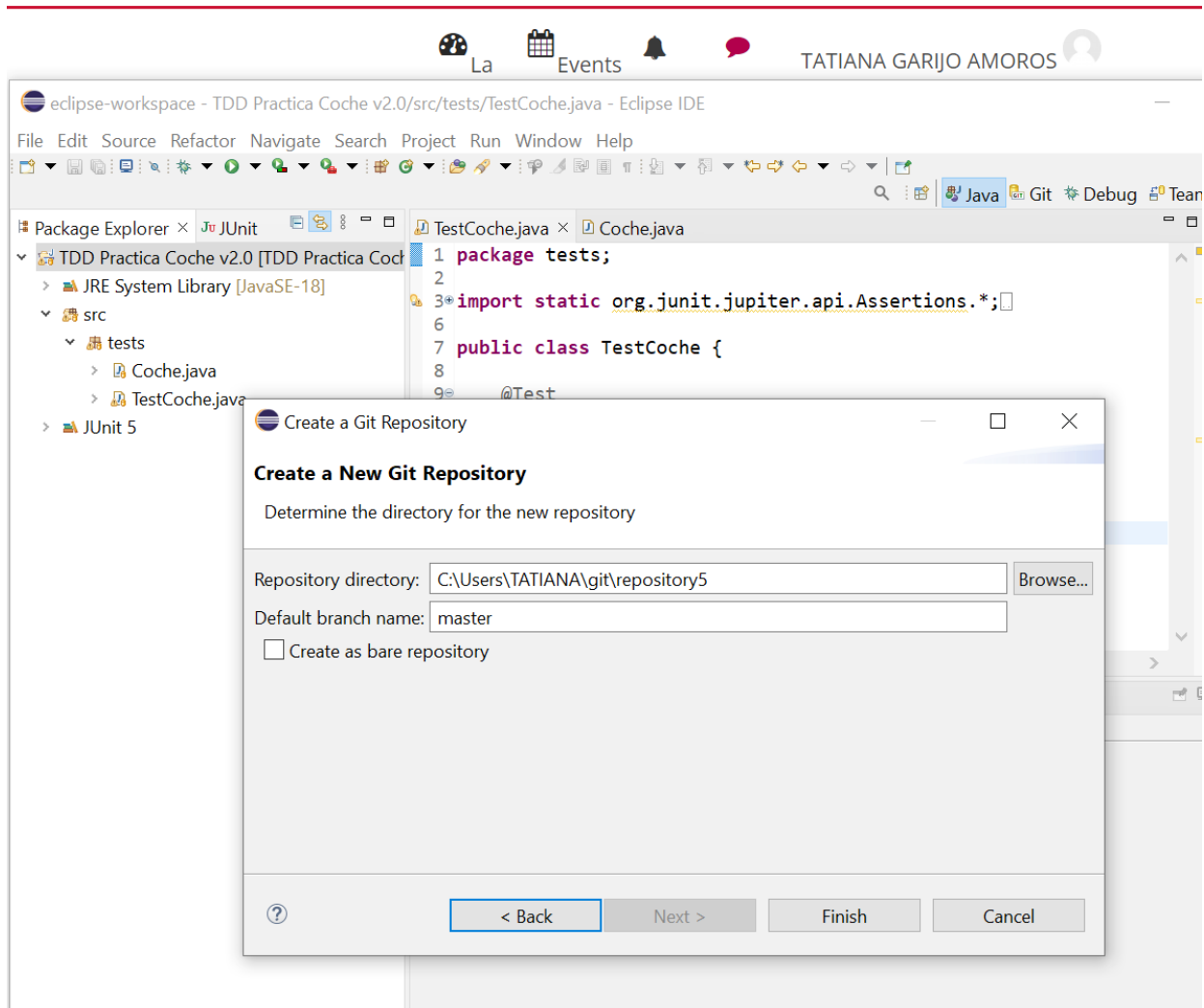


- Después de renombrar pulsar → Enter

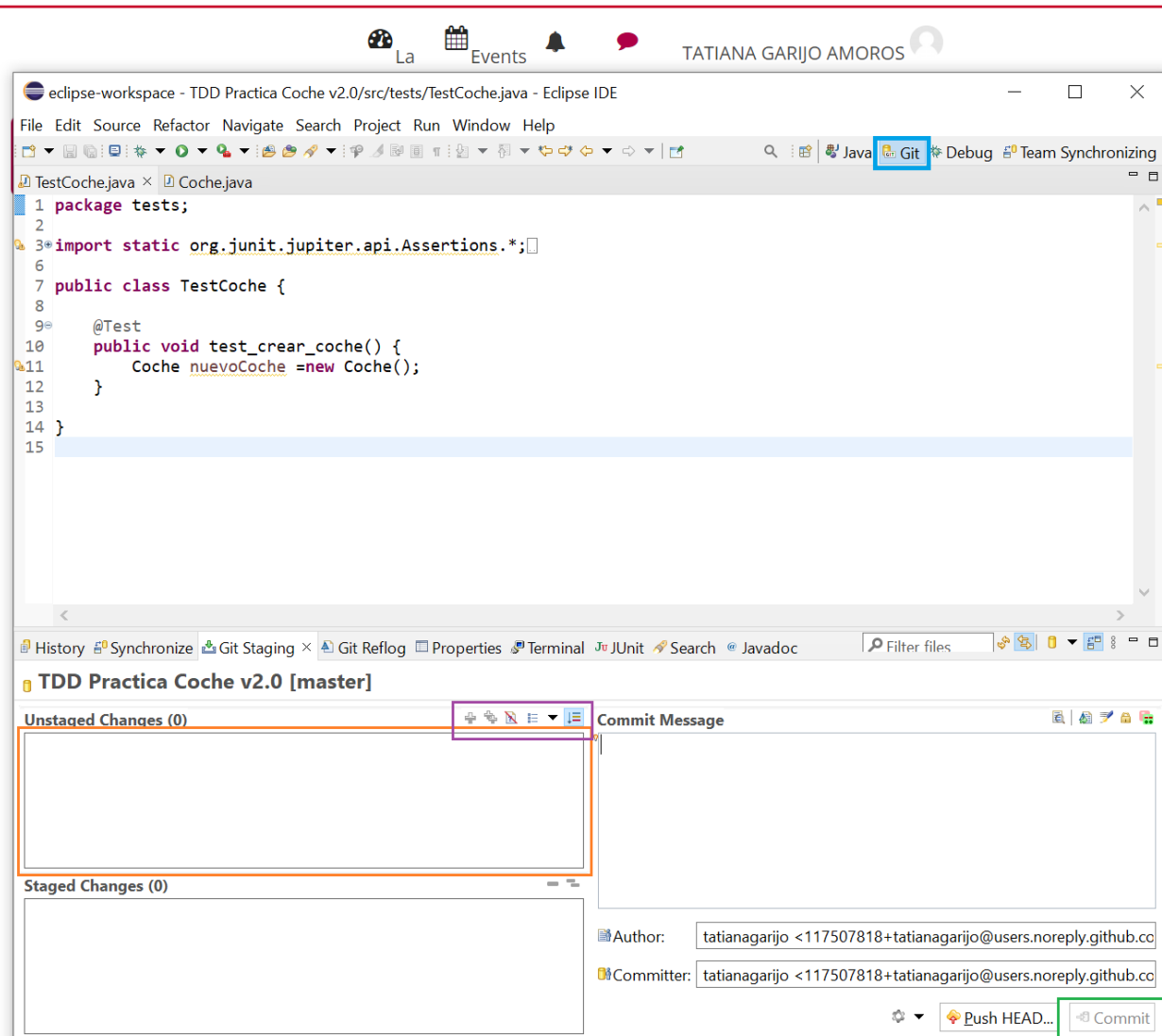
- VSC



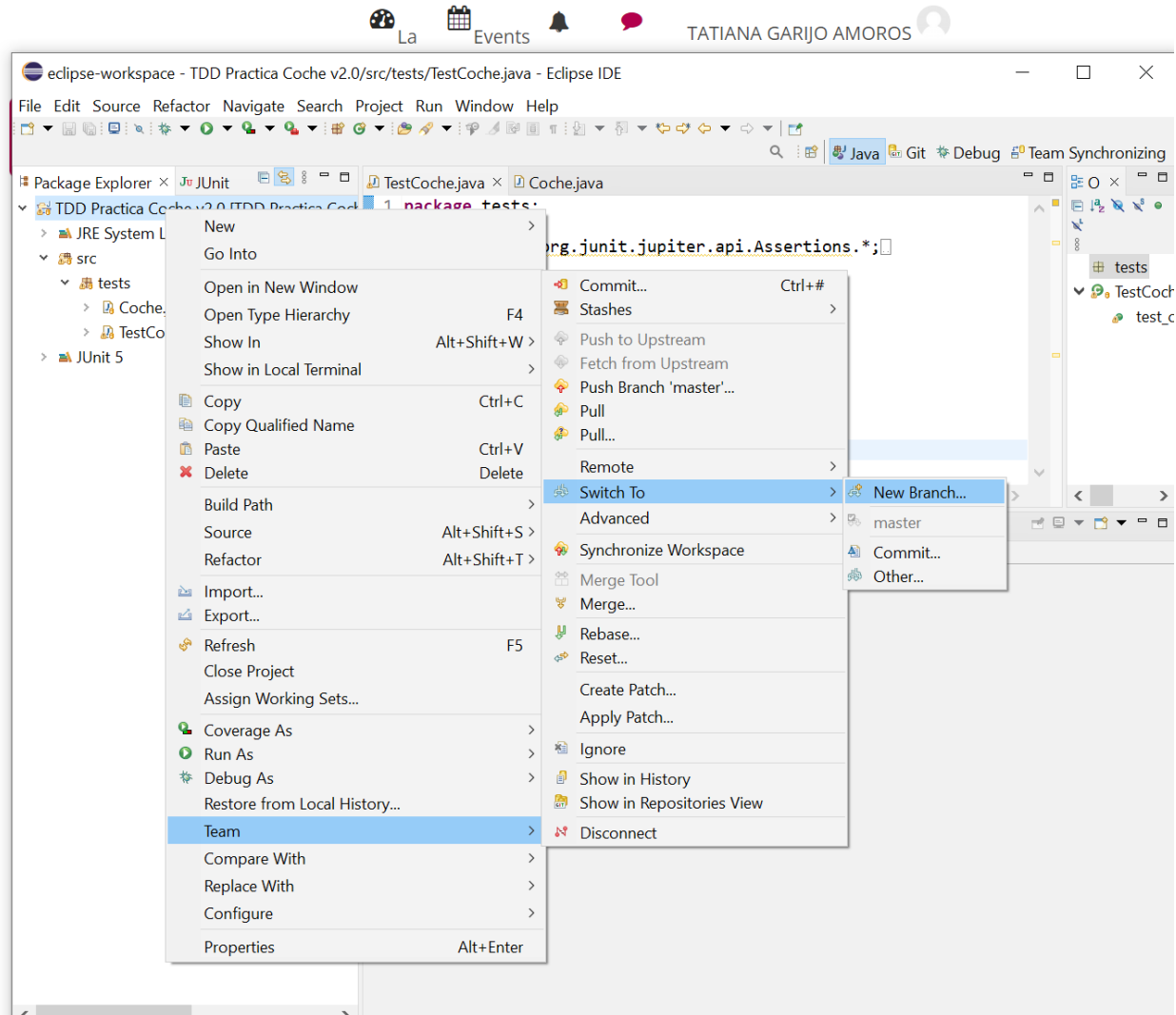
- Crear el repositorio haciendo clic con el botón derecho sobre el proyecto



- Seleccionar la ruta del repositorio y hacer clic botón Finish



- Podemos seleccionar los archivos que queramos **añadir** de **Unstaged** a **Staged**, añadir un mensaje y **hacer commit** accediendo al panel/menú Git desde el **icono 'Git'** superior derecho (en mi caso)



- Haciendo clic con botón derecho sobre el proyecto y accediendo a la opción 'Team' podemos acceder a las opciones relacionadas con Git, como crear y cambiar de rama, commit, merge...
- Para subir el proyecto a Github se puede hacer botón derecho sobre el proyecto → Team → Remote → push → configurar la dirección del repositorio y nuestros datos de usuario en la ventana emergente y seleccionar todas las ramas.

• Resolución de Errores

Al seleccionar desde el test que la clase Coche se cree, se crea dentro del mismo paquete tests.

Para resolverlo he tenido que Refactorizar.

- Clic botón derecho sobre la clase coche → Refactor o Alt+Shift+T → Move o Alt+Shift+V

