

IES SEVERO OCHOA

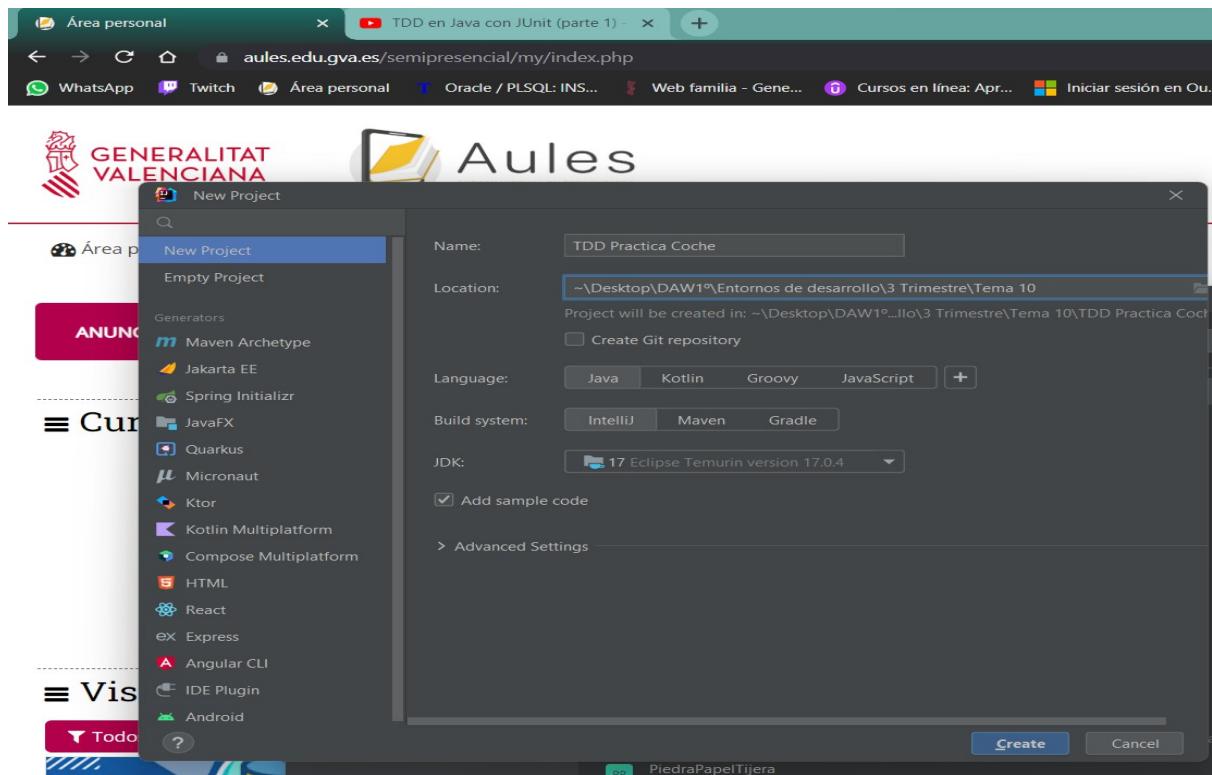
# ENTORNOS DE DESARROLLO



DIEGO PEREZ ANTON  
2022/2023

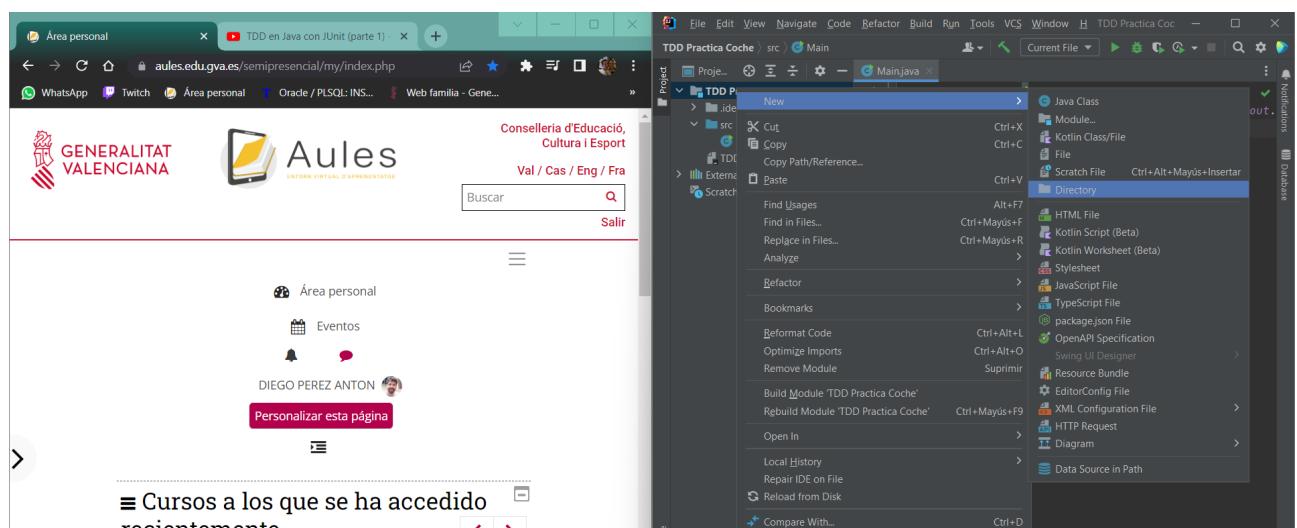
# MI PRIMER TDD

-Inicio IntelliJ IDEA y creo un nuevo proyecto llamado TDD Practica Coche y le doy a crear

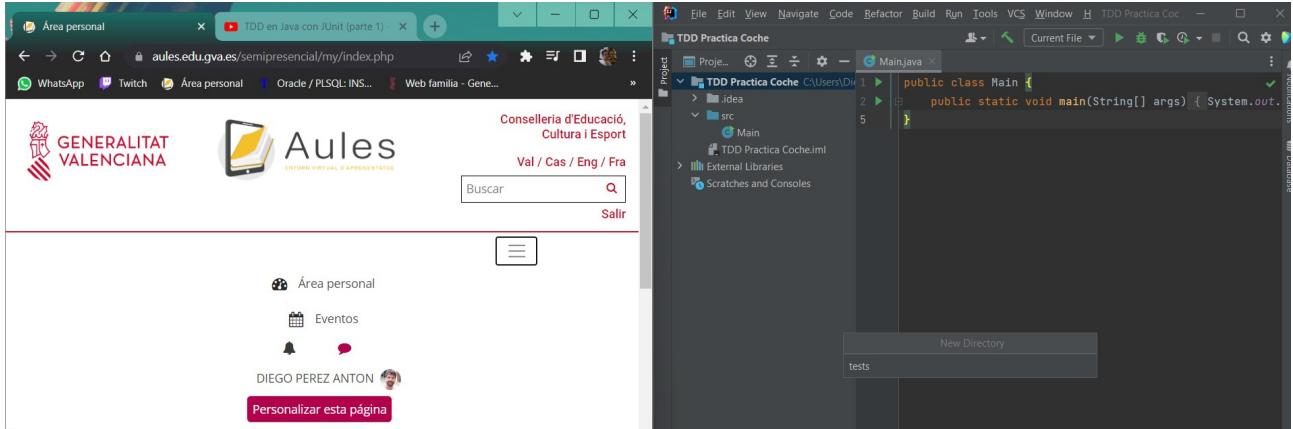


-Creo un directorio para los test:

botón derecho del ratón en la carpeta del proyecto/new/Directory:

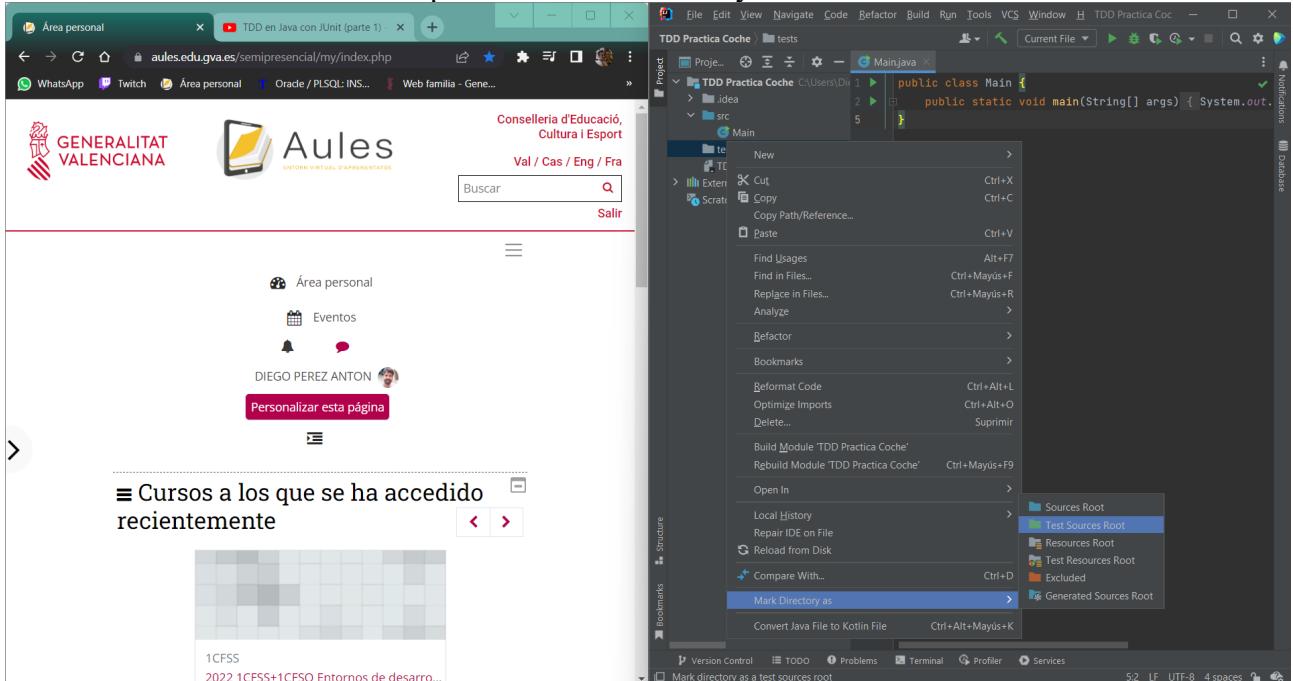


-Pongo como nombre tests:



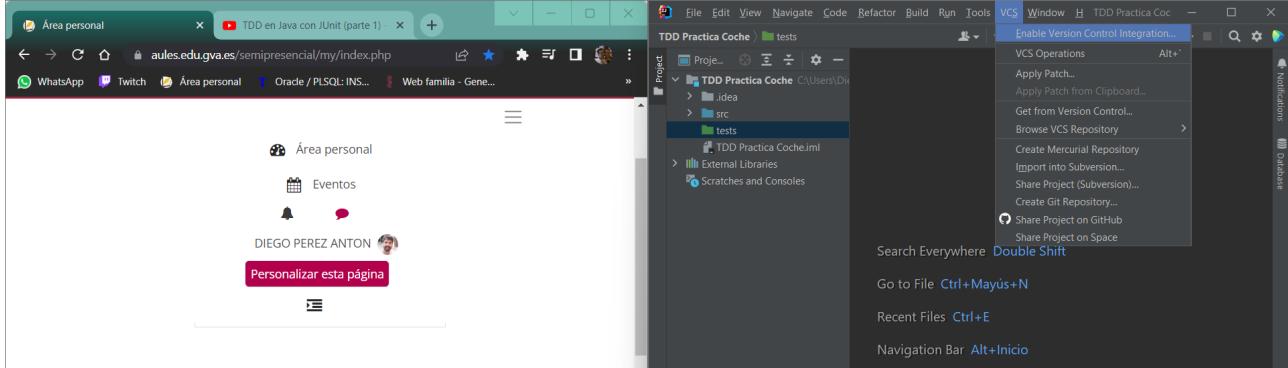
-Marco el directorio tests, como directorio de test:

Botón derecho del ratón en la carpeta test/Marck directory as/Test Sources Root

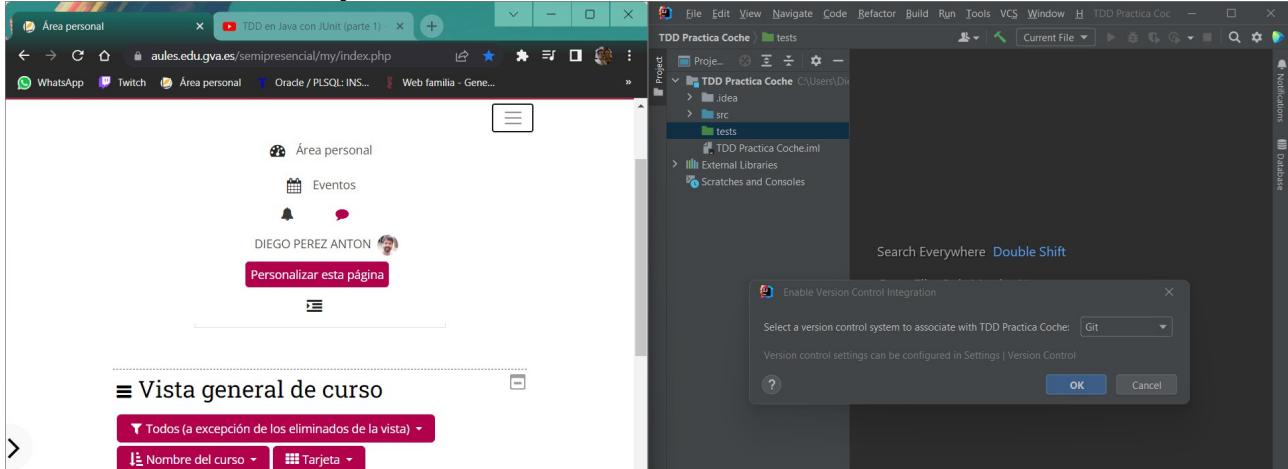


## Activo Git:

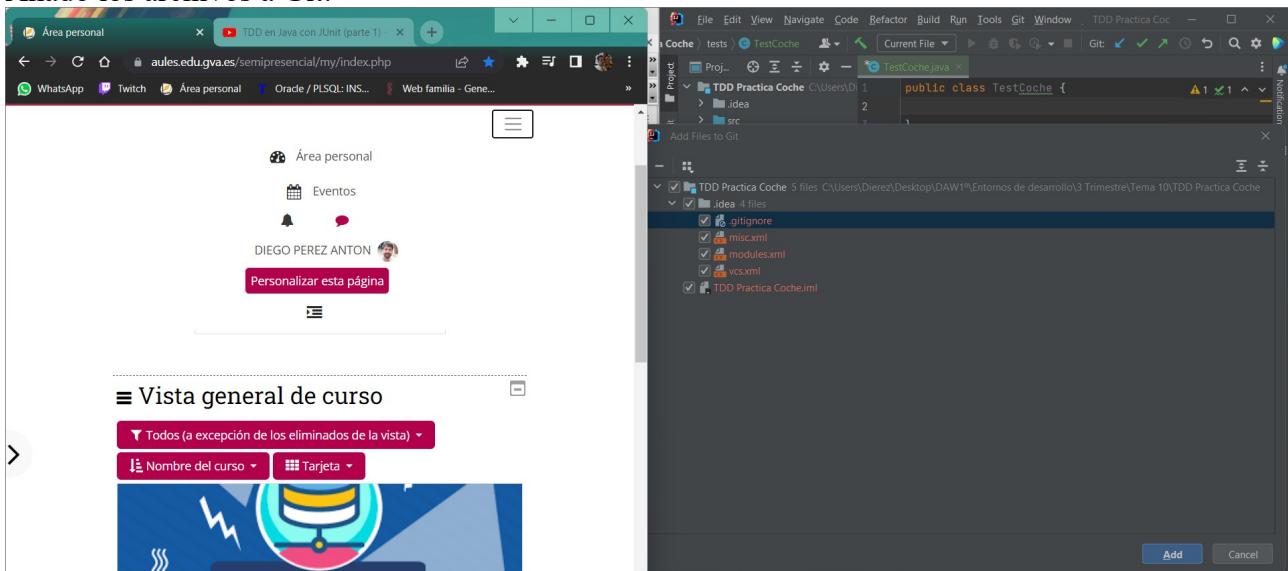
selecciono la pestaña VCS/Enabla Version Control Integration



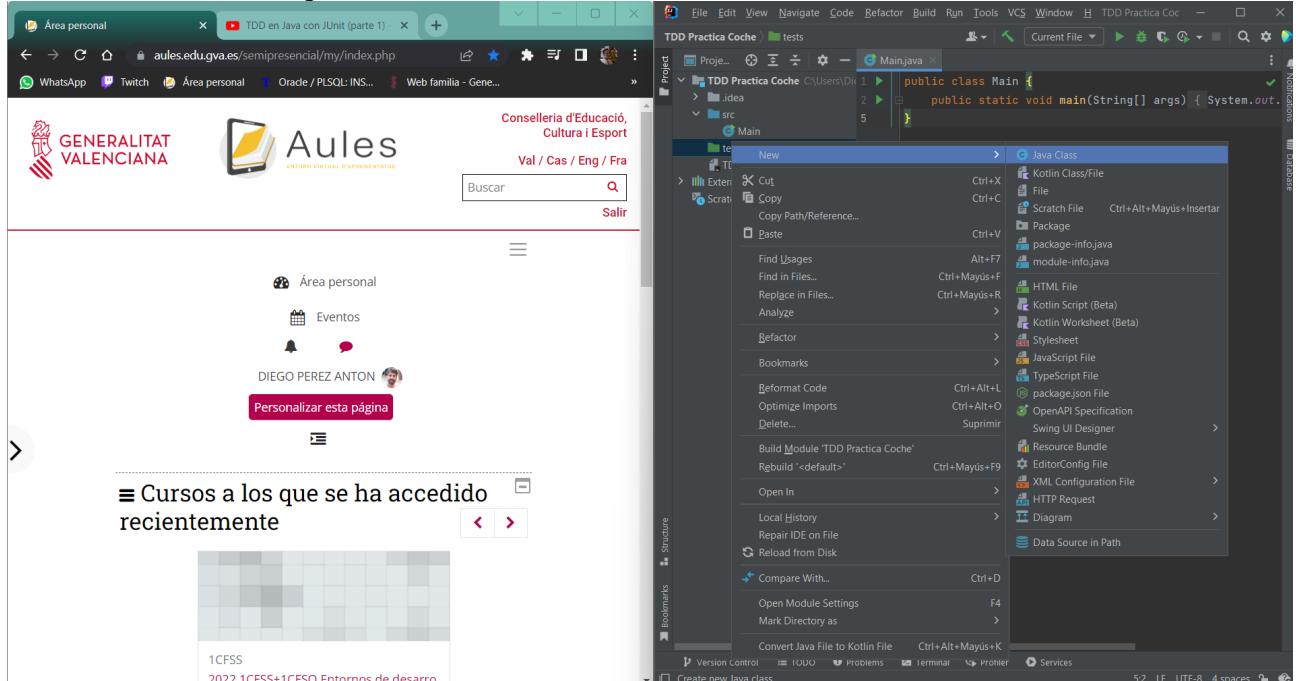
-Con Git seleccionado, doy a ok:



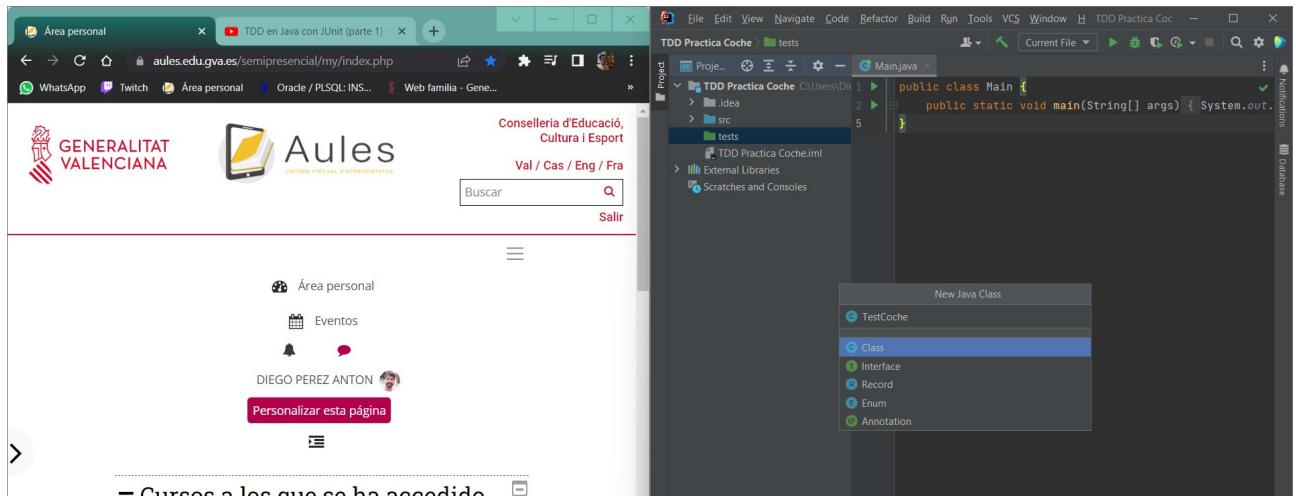
Añado los archivos a Git:



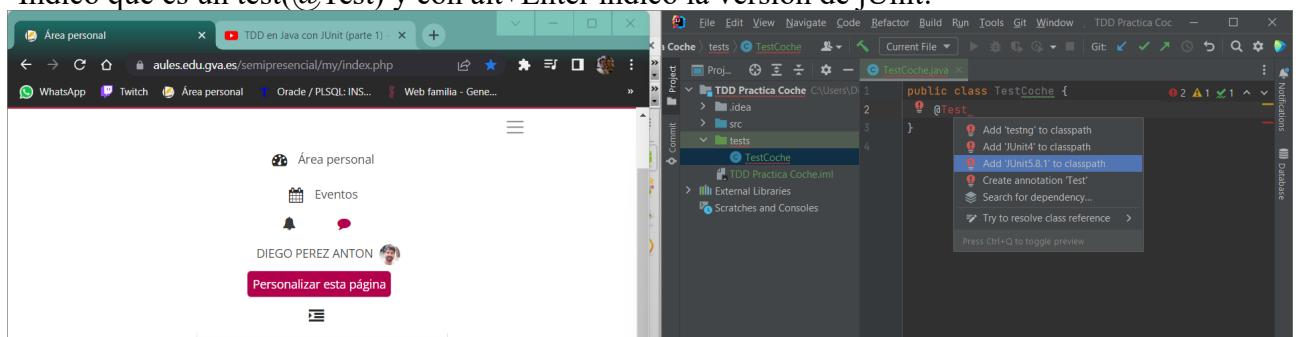
-En la carpeta de tests creo una nueva clase:  
botón derecho en la carpeta tests/News/Java Class



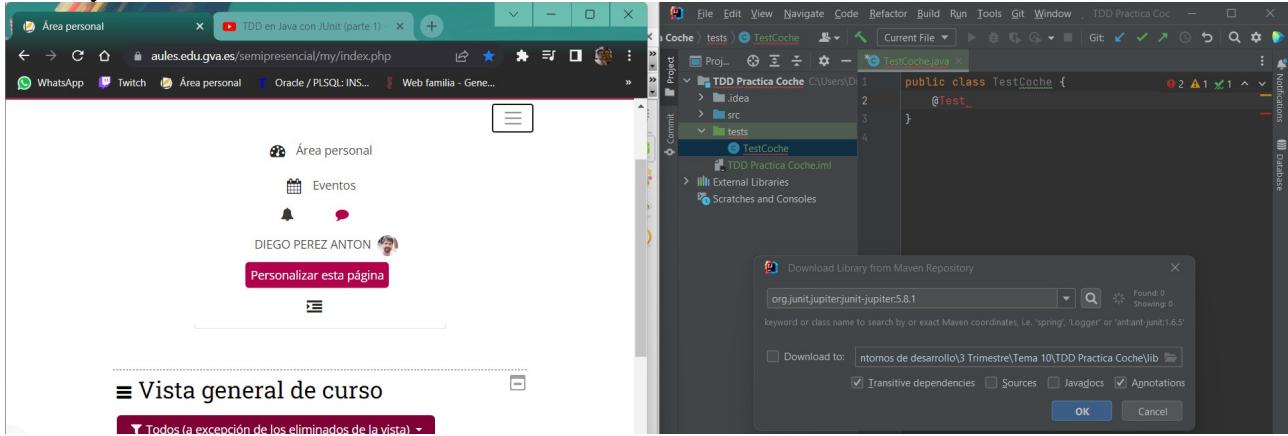
-La llamo TestCoche:



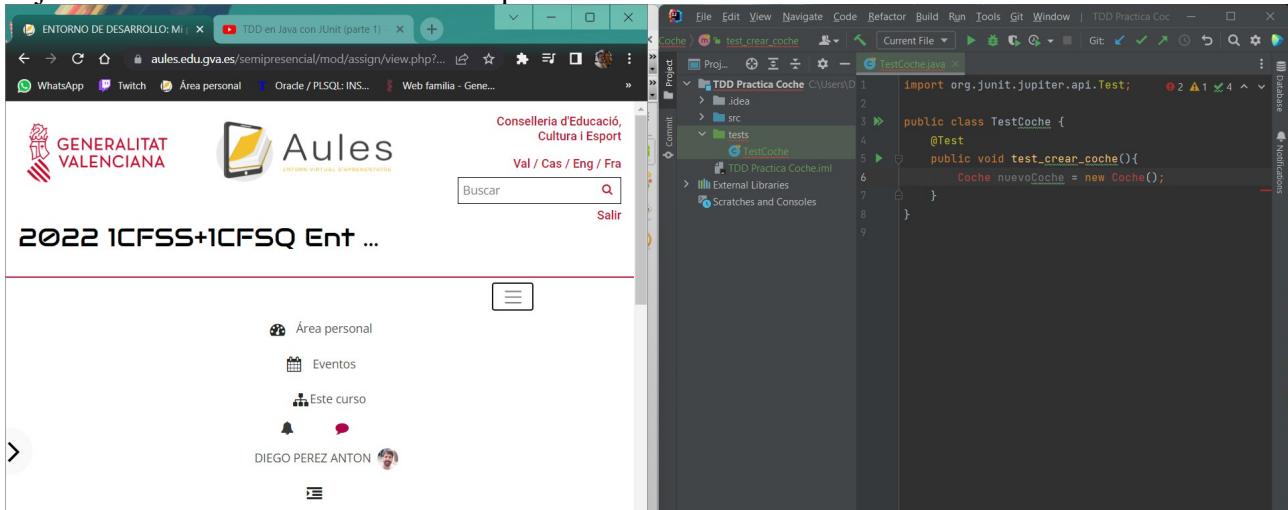
-Indico que es un test(@Test) y con alt+Enter indico la versión de jUnit:



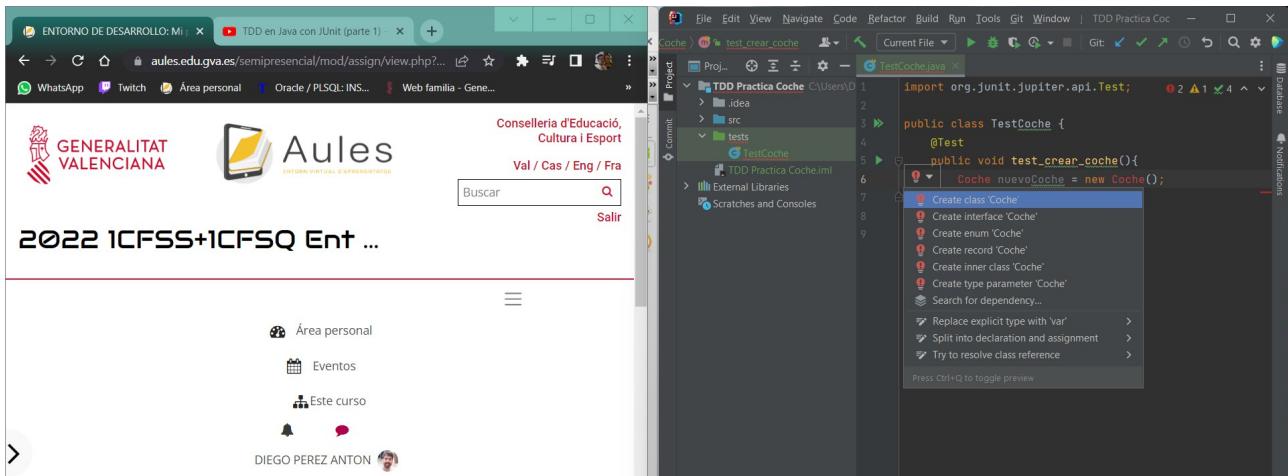
-Le doy a ok:



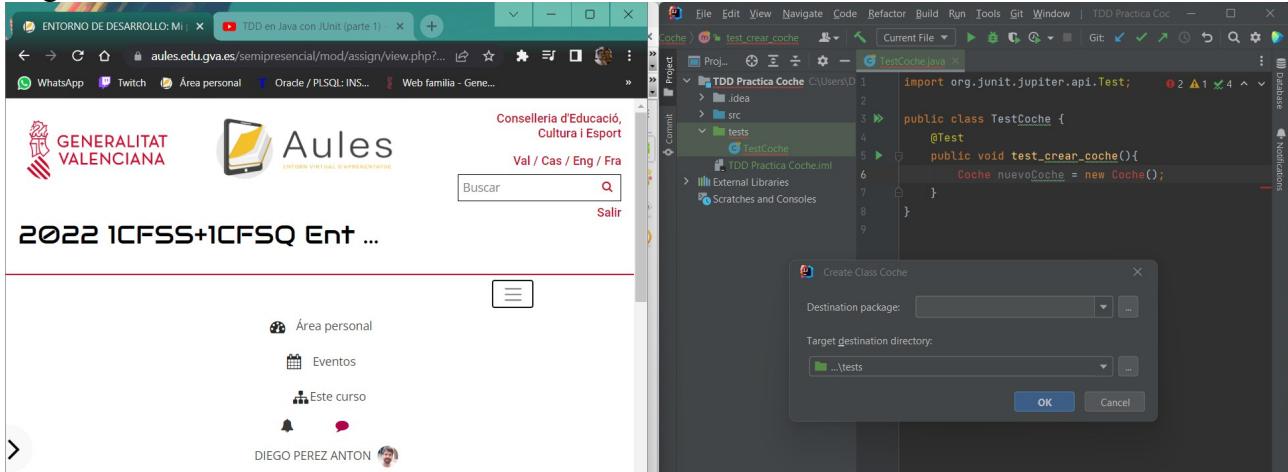
-Ahora creo mi primer test (`public void test_crear_coche()`), y a continuación creo el objeto nuevoCoche de la clase Coche que aun no está creada:



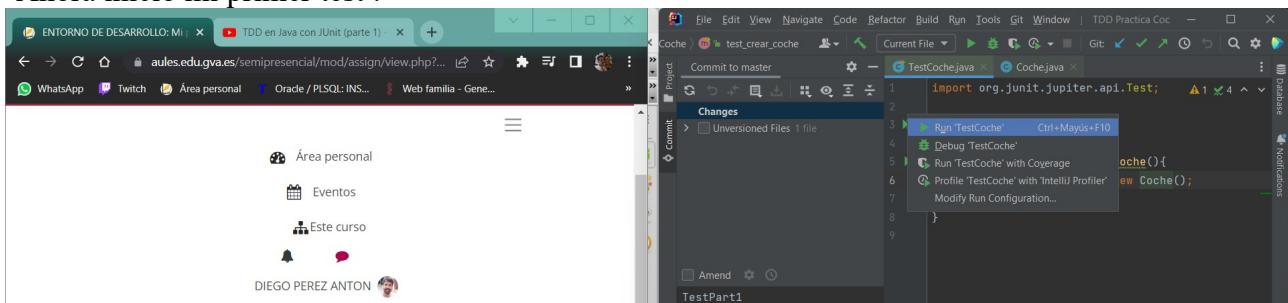
-Me avisa de que la clase Coche no está creada por lo que en Coche, alt + Enter y selecciono crear la clase:



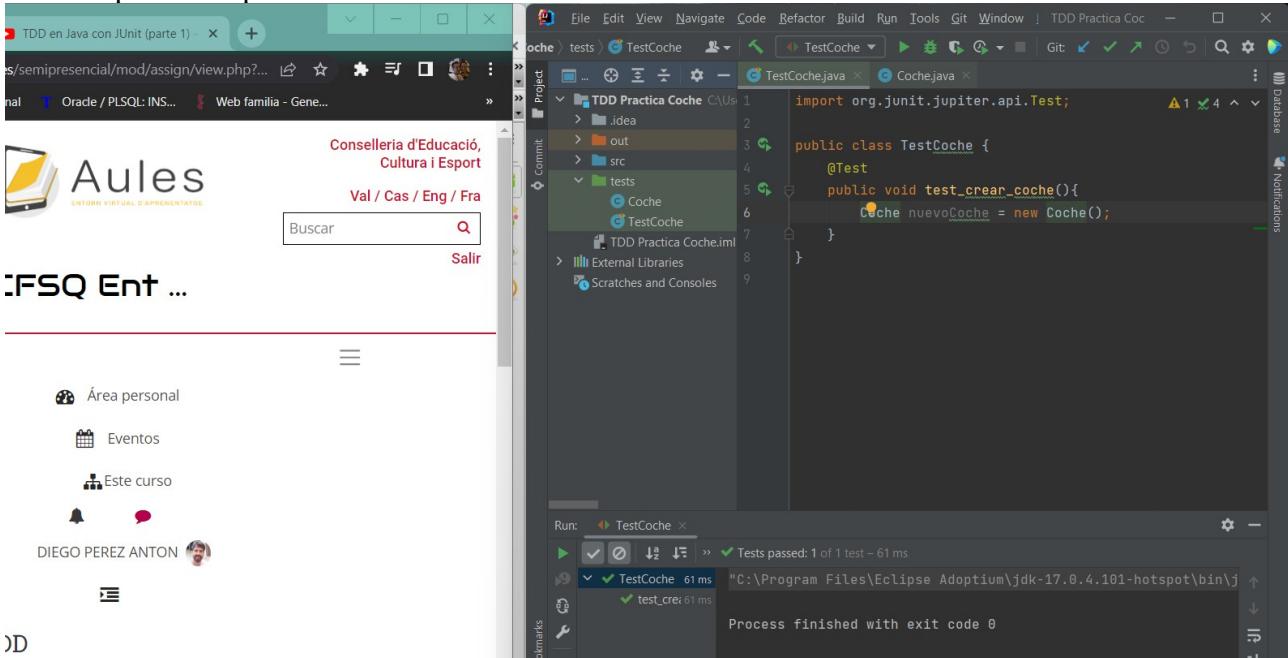
-Siguiente ventana ok:



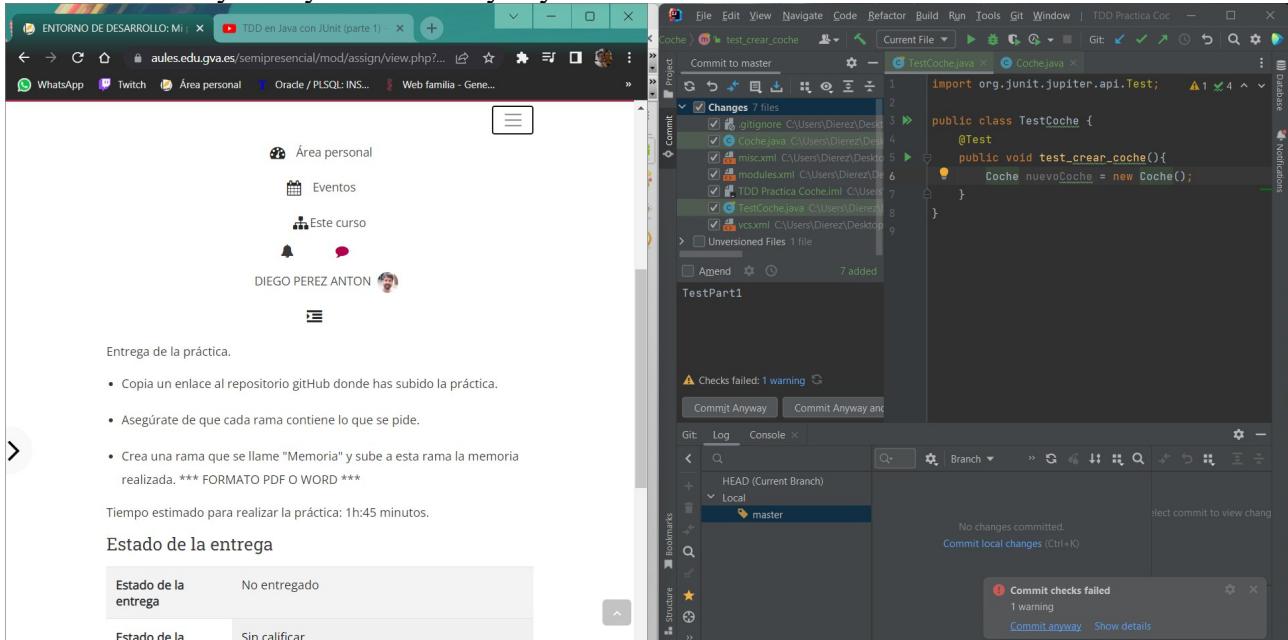
-Ahora inicio mi primer test :



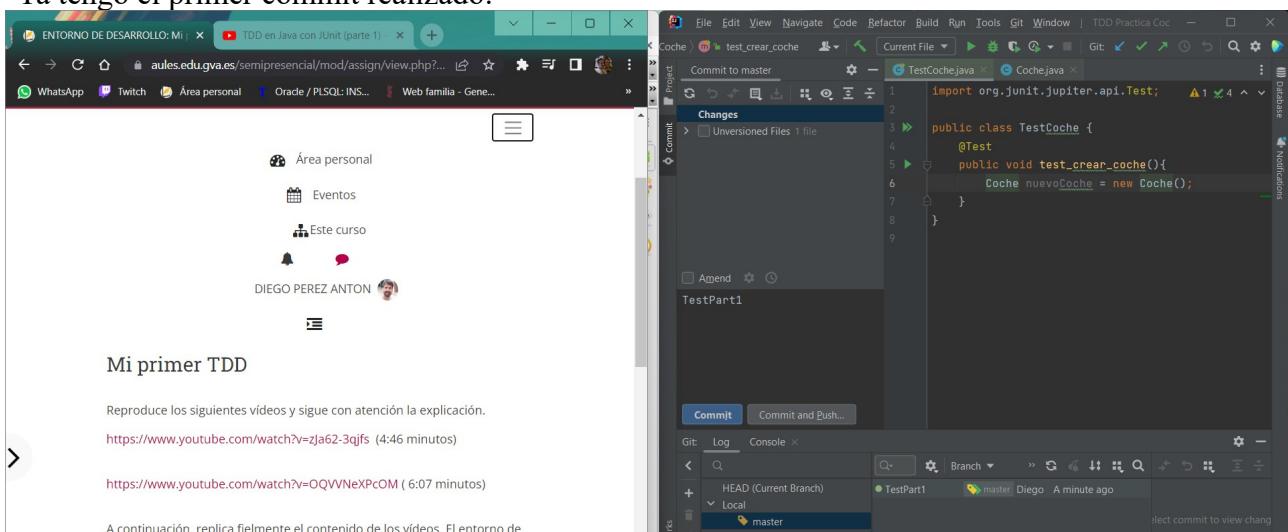
-Hemos pasado el primer test:



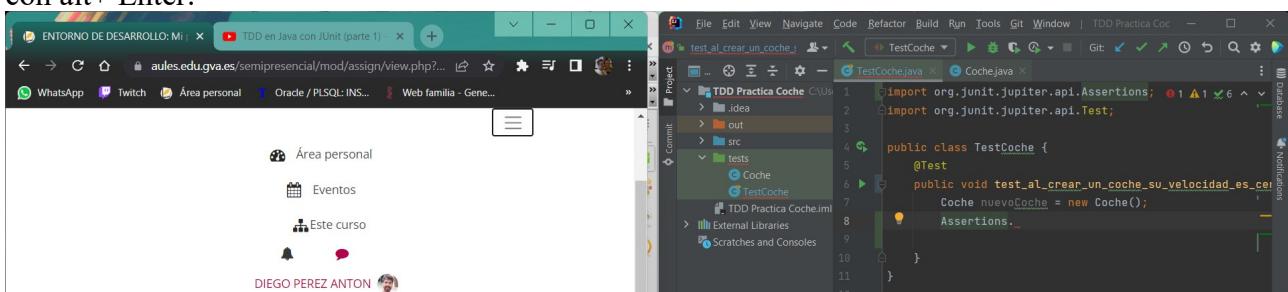
-Ahora hago mi primer commit, dando a la pestaña de la parte izq llamada Commit, selecciono los archivos, pongo una frase descriptiva y doy a commit, seguidamente me sale una ventana en la parte inferior derecha y le doy a commit anyway:



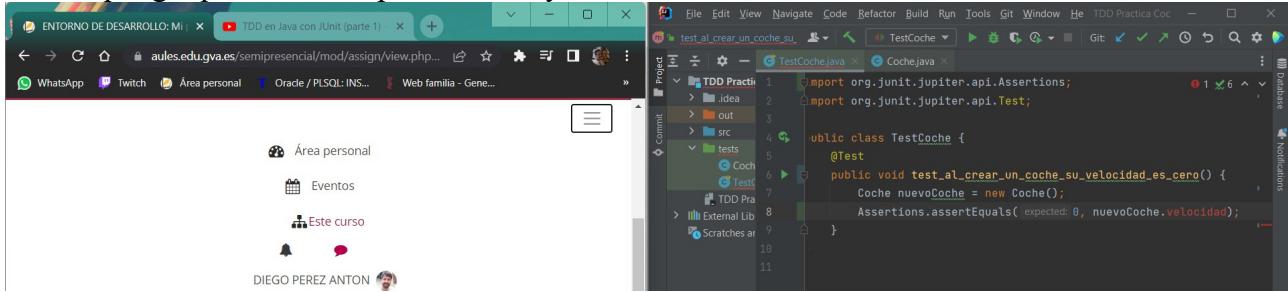
-Ya tengo el primer commit realizado:



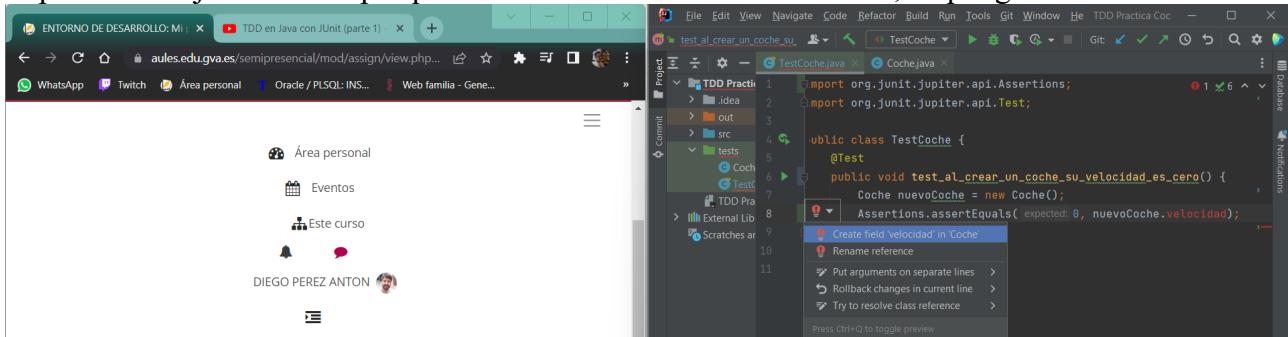
-Ahora indico con test\_al\_crear\_un\_coche\_su\_velocidad\_es\_cero que la velocidad sea cero, para eso hay que indicarlo con assertions y lo importo dándole a la bombilla o con alt+ Enter:



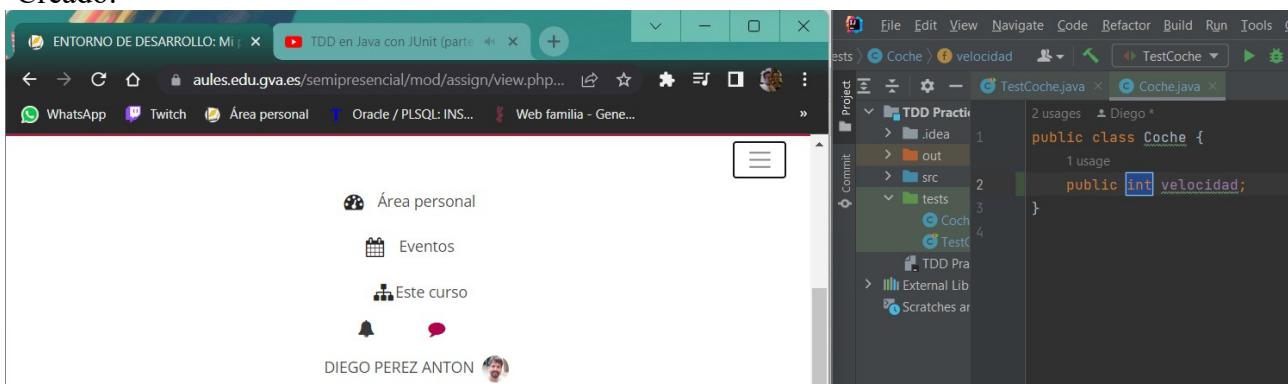
-Ahora pongo que el valor esperado sea 0 y el valor real será nuevoCoche.velocidad:



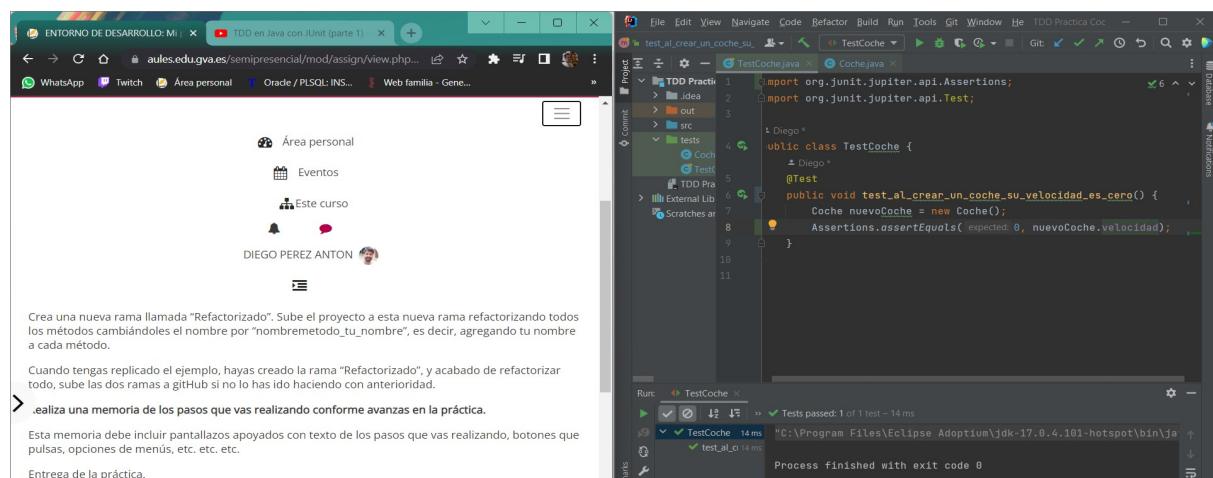
-Aparece en rojo velocidad porque no tenemos aun creado el símbolo, dispongo a crearlo:



-Creado:



-Inicio dándole al play para ejecutar los test y ha pasado correctamente:



-Ahora hago el segundo commit de los cambios realizados:

The screenshot shows an IDE interface with a commit message in the center. The message reads: "Crea una nueva rama llamada 'Refactorizado'. Sube el proyecto a esta nueva rama refactorizando todos los métodos cambiándoles el nombre por 'nombremetodo\_tu\_nombre', es decir, agregando tu nombre a cada método." Below this, another message says: "Cuando tengas replicado el ejemplo, hasayas creado la rama 'Refactorizado', y acabado de refactorizar todo, sube las dos ramas a GitHub si no lo has ido haciendo con anterioridad." At the bottom, there's a list of instructions for handing in the practical work.

Crea una nueva rama llamada "Refactorizado". Sube el proyecto a esta nueva rama refactorizando todos los métodos cambiándoles el nombre por "nombremetodo\_tu\_nombre", es decir, agregando tu nombre a cada método.

Cuando tengas replicado el ejemplo, hasayas creado la rama "Refactorizado", y acabado de refactorizar todo, sube las dos ramas a GitHub si no lo has ido haciendo con anterioridad.

> .realiza una memoria de los pasos que vas realizando conforme avanzas en la práctica.

Esta memoria debe incluir pantallazos apoyados con texto de los pasos que vas realizando, botones que pulsas, opciones de menús, etc. etc. etc.

Entrega de la práctica.

- Copia un enlace al repositorio GitHub donde has subido la práctica.
- Asegúrate de que cada rama contiene lo que se pide.
- Crea una rama que se llame "Memoria" y sube a esta rama la memoria realizada. \*\*\* FORMATO PDF O WORD \*\*\*

<https://aulas.edu.gva.es/semipresencial/mod/assign/view.php?id=1500619>

-Creo otro test con `public void test_al_acelerar_un_coche_su_velocidad_aumenta()` "sin es"  
le digo que nuevo.coche tenga la velocidad de 30 y la Aserción que tenga como velocidad 30:

The screenshot shows an IDE interface with a code editor. A new test method is being created: `public void test_al_acelerar_un_coche_su_velocidad_aumenta()`. The code inside the method initializes a new car with speed 30 and asserts that its speed is 30. A tooltip appears over the `acelerar` method call.

```
public class TestCoche {  
    public void test_al_crear_un_coche_su_velocidad_es_cero() {  
        Coche nuevoCoche = new Coche();  
        Assertions.assertEquals(0, nuevoCoche.velocidad);  
    }  
    @Test  
    public void test_al_acelerar_un_coche_su_velocidad_es_aumenta() {  
        Coche nuevoCoche = new Coche();  
        nuevoCoche.acelerar(30);  
        Assertions.assertEquals(30, nuevoCoche.velocidad);  
    }  
}
```

-Acelerar aparece en rojo porque no existe por lo que creo el método:

The screenshot shows an IDE interface with a tooltip for the `acelerar` method. It suggests creating a new method in the `Coche` class. The tooltip options include "Create method 'acelerar' in 'Coche'" (highlighted), "Rename reference", and "Rollback changes in current line".

A continuación, replica fielmente el contenido de los videos. El entorno de desarrollo utilizado es IntelliJ Idea y el plugin JUnit 5.  
Inicializa el control de versiones y ve realizando commits mientras replicas el contenido de los videos según estimes oportuno.

Crea una nueva rama llamada "Refactorizado". Sube el proyecto a esta nueva rama refactorizando todos los métodos cambiándoles el nombre por "nombremetodo\_tu\_nombre", es decir, agregando tu nombre

-Método creado:

The screenshot shows the Eclipse IDE interface. On the left, the 'Project Explorer' view displays a Java project named 'TDD Pr'. Inside the 'src' folder, there is a 'test' folder containing a 'Coche.java' file. The code editor on the right shows the following Java code:

```
public class Coche {  
    public int velocidad;  
    public void acelerar(int i) {  
        velocidad += i;  
    }  
}
```

-Cambio el nombre del parámetro por aceleracion y le digo que velocidad aumente en aceleracion:

The screenshot shows the Eclipse IDE interface. The code editor on the right now displays the following Java code after refactoring:

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

-Ejecuto y he pasado el test correctamente:

The screenshot shows the Eclipse IDE interface. The code editor on the right shows the Java code from the previous step. Below it, the 'Run' view shows the results of a JUnit test run:

```
Tests passed: 2 of 2 tests - 16 ms  
Process finished with exit code 0
```

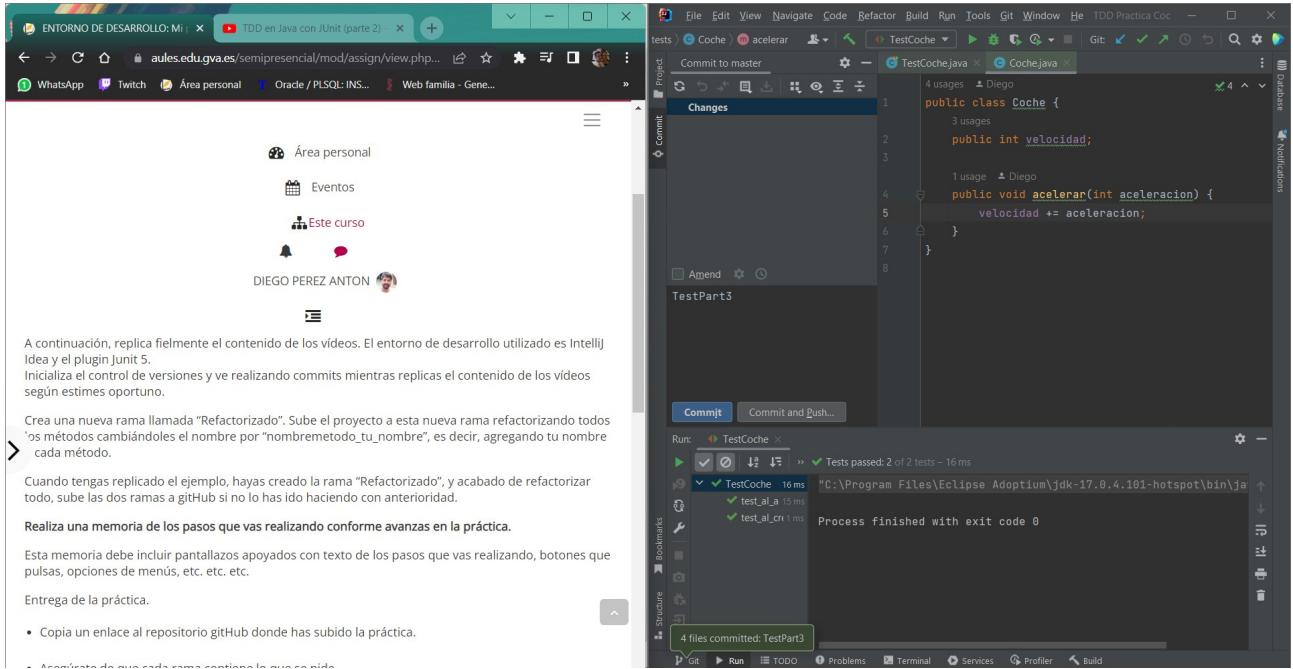
At the bottom of the screen, a commit message is displayed: "TestPart3".

-Hago el tercer commit (TestPart3):

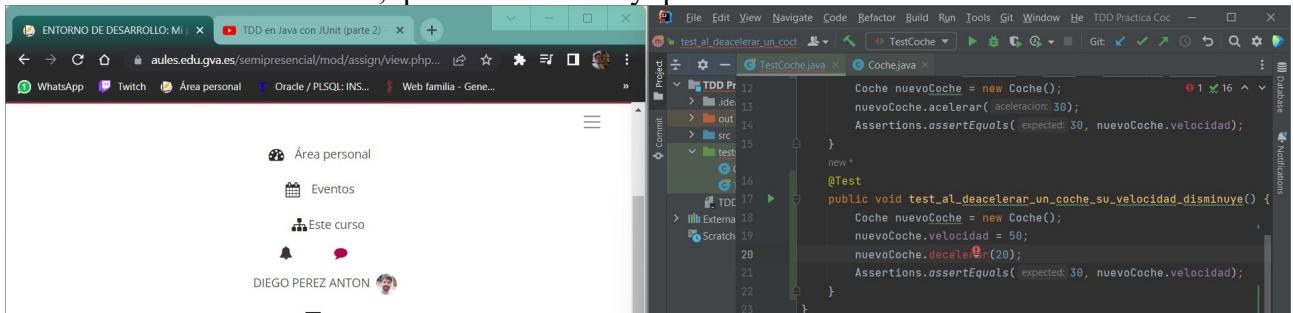
The screenshot shows the Eclipse IDE interface. The code editor on the right shows the Java code from the previous step. Below it, the 'Commit' view shows the commit message "TestPart3" and the changes made:

- Coche.class (C:\Users\Diego\IdeaProjects\tdd\src\Coche.java)
- Coche.java (C:\Users\Diego\IdeaProjects\tdd\src\Coche.java)
- TestCoche.class (C:\Users\Diego\IdeaProjects\tdd\src\test\Coche.java)
- TestCoche.java (C:\Users\Diego\IdeaProjects\tdd\src\test\Coche.java)

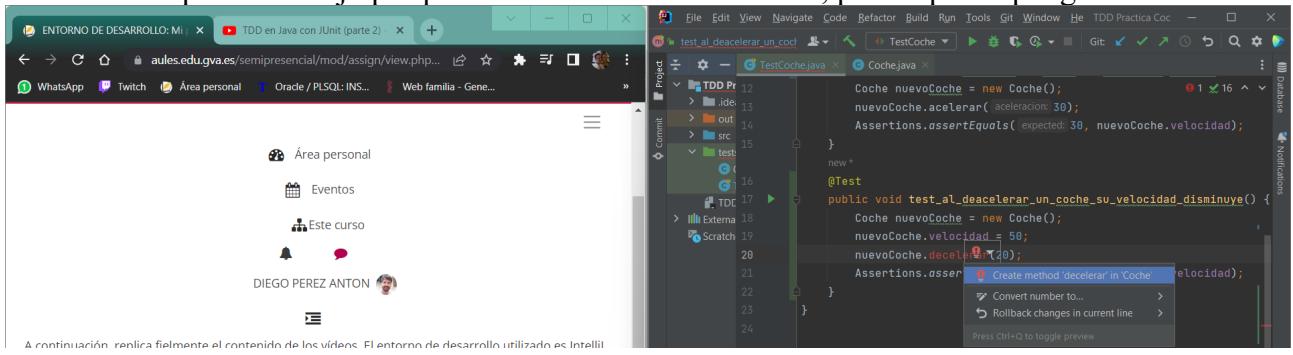
## -Commit realizado:



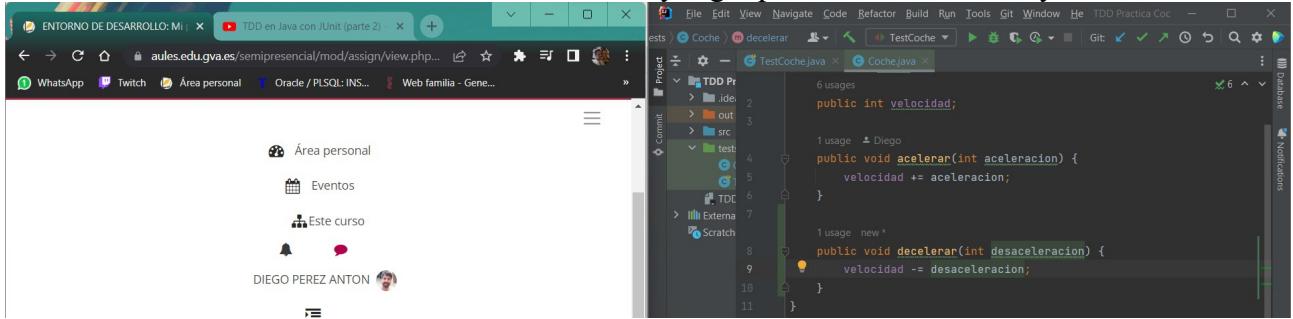
-Ahora creo otro método que sea decelerar, primero indico que la velocidad inicial sea 50, porque si parte de 0 y tiene que desacelerar no queremos que sea un numero negativo la velocidad.  
Indico la velocidad inicial 50, que desacelere 20 y que la velocidad final sea 30:



-desacelerar aparece en rojo porque aun no he creado el método, por lo que dispongo a crearlo:

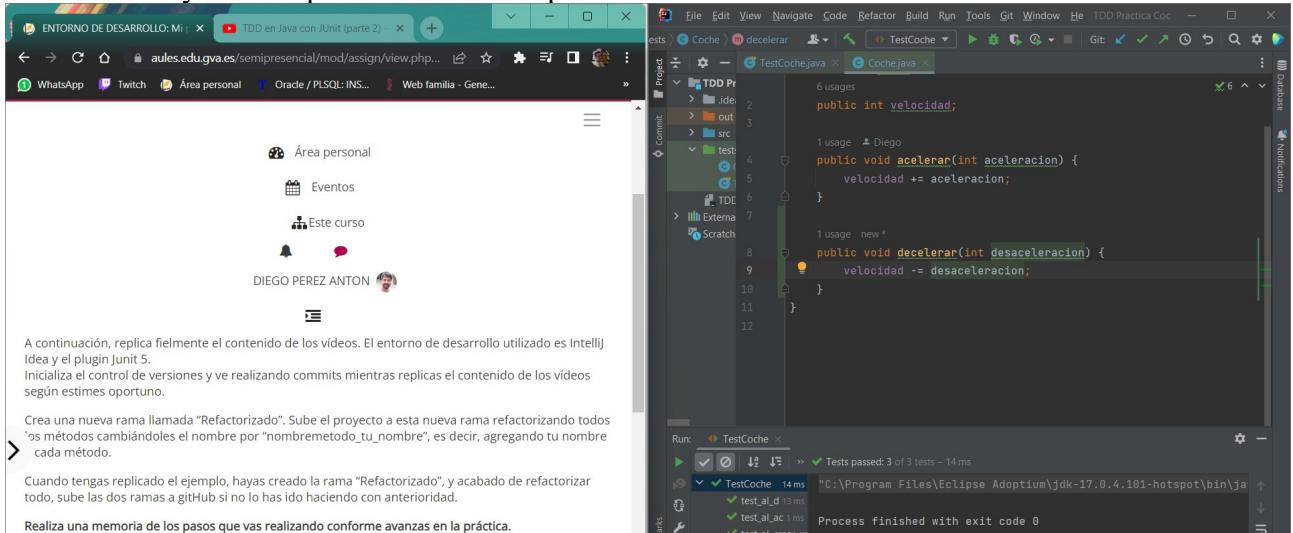


-Cambio el nombre de la variable a desaceleracion y digo que velocidad disminuya en aceleracion:



```
public int velocidad;
public void acelerar(int aceleracion) {
    velocidad += aceleracion;
}
public void decelerar(int desaceleracion) {
    velocidad -= desaceleracion;
}
```

-Inicio el test y vemos que de nuevo lo he pasado:



```
public int velocidad;
public void acelerar(int aceleracion) {
    velocidad += aceleracion;
}
public void decelerar(int desaceleracion) {
    velocidad -= desaceleracion;
}
```

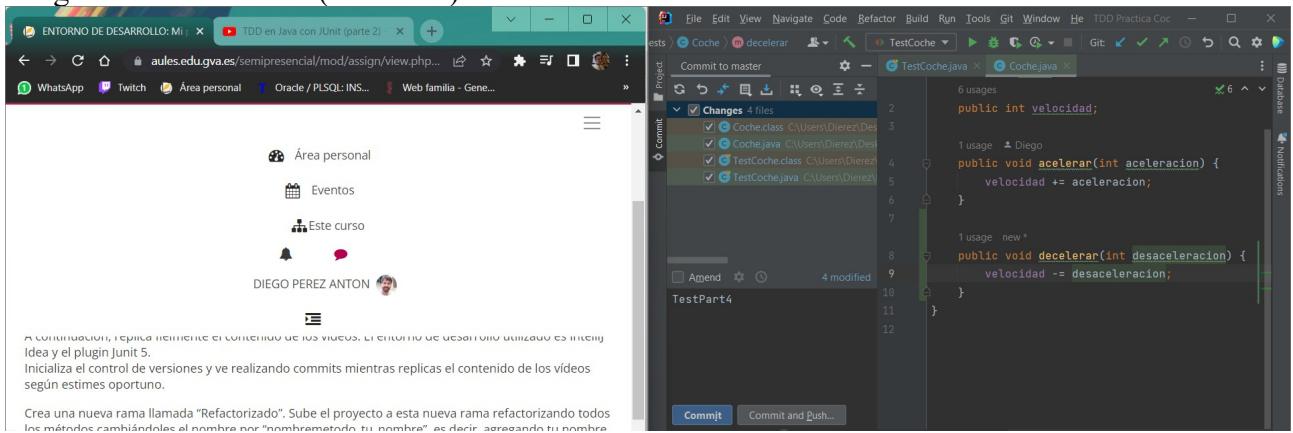
A continuación, replica fielmente el contenido de los videos. El entorno de desarrollo utilizado es IntelliJ Idea y el plugin JUnit 5.  
Iniciaiza el control de versiones y ve realizando commits mientras replicas el contenido de los videos según estimes oportuno.

Crea una nueva rama llamada "Refactorizado". Sube el proyecto a esta nueva rama refactorizando todos los métodos cambiándoles el nombre por "nombremetodo\_tu\_nombre", es decir, agregando tu nombre cada método.

Cuando tengas replicado el ejemplo, hayas creado la rama "Refactorizado", y acabado de refactorizar todo, sube las dos ramas a gitHub si no lo has ido haciendo con anterioridad.

Realiza una memoria de los pasos que vas realizando conforme avanzas en la práctica.

-Hago el cuarto commit (TestPart4):



```
public int velocidad;
public void acelerar(int aceleracion) {
    velocidad += aceleracion;
}
public void decelerar(int desaceleracion) {
    velocidad -= desaceleracion;
}
```

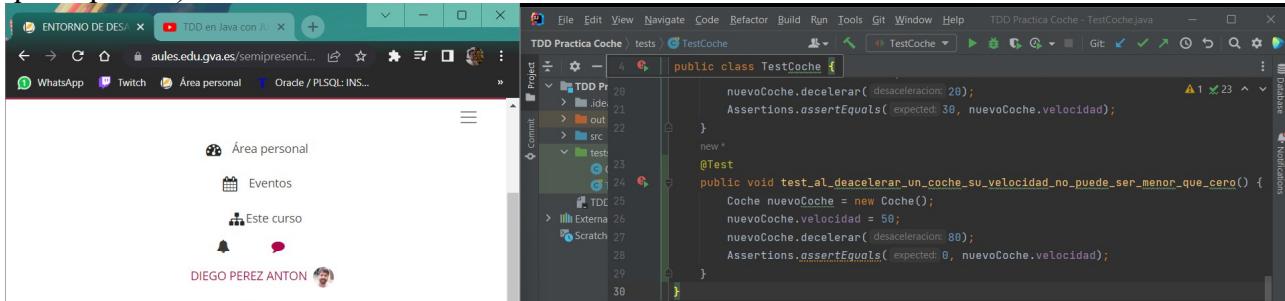
A continuación, replica fielmente el contenido de los videos. El entorno de desarrollo utilizado es IntelliJ Idea y el plugin JUnit 5.  
Iniciaiza el control de versiones y ve realizando commits mientras replicas el contenido de los videos según estimes oportuno.

Crea una nueva rama llamada "Refactorizado". Sube el proyecto a esta nueva rama refactorizando todos los métodos cambiándoles el nombre por "nombremetodo\_tu\_nombre" es decir, agregando tu nombre

-Ahora voy a crear otro test que sea void

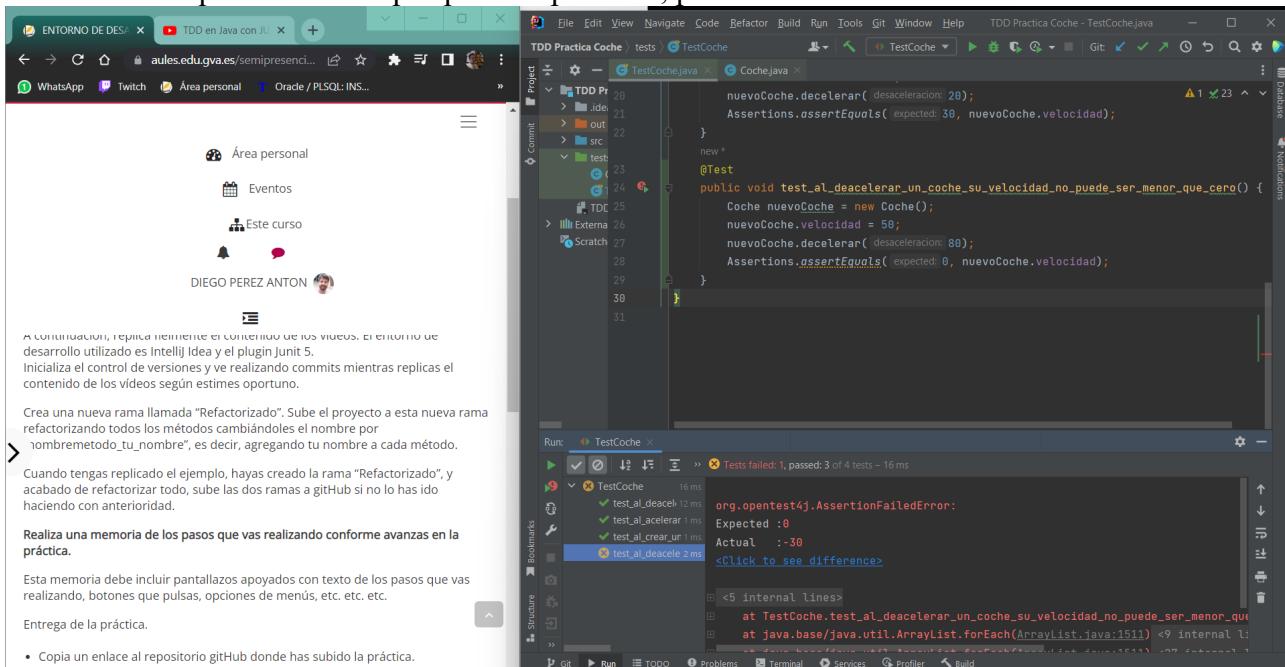
test\_al\_deacelerar\_un\_coche\_su\_velocidad\_no\_puede\_ser\_menor\_que\_ce  
ro()

e indico que la velocidad inicial sea 50, que desacelera 80 y que la velocidad final sea 0 (que se quede parado):



```
public class TestCoche {
    @Test
    public void test_al_deacelerar_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);
    }
}
```

-Podemos ver que el test falla porque se esperaba 0, pero obtiene -30:



A continuación, repite nuevamente el contenido de los videos. El entorno de desarrollo utilizado es IntelliJ Idea y el plugin JUnit 5.  
Inicializa el control de versiones y ve realizando commits mientras replicas el contenido de los videos según estimes oportuno.

Crea una nueva rama llamada "Refactorizado". Sube el proyecto a esta nueva rama refactorizando todos los métodos cambiándoles el nombre por "nombremetodo\_tu\_nombre", es decir, agregando tu nombre a cada método.

Cuando tengas replicado el ejemplo, has creado la rama "Refactorizado", y acabado de refactorizar todo, sube las dos ramas a GitHub si no lo has ido haciendo con anterioridad.

Realiza una memoria de los pasos que vas realizando conforme avanzas en la práctica.

Esta memoria debe incluir pantallazos apoyados con texto de los pasos que vas realizando, botones que pulsas, opciones de menús, etc. etc. etc.

Entrega de la práctica.

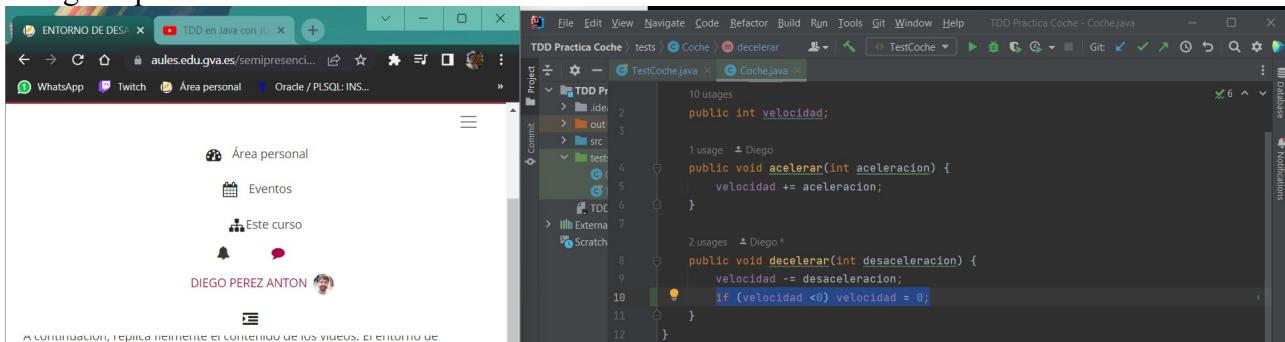
- Copia un enlace al repositorio GitHub donde has subido la práctica.

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

org.opentest4j.AssertionFailedError:  
Expected :0  
Actual :-30  
<Click to see difference>

at TestCoche.test\_al\_deacelerar\_un\_coche\_su\_velocidad\_no\_puede\_ser\_menor\_que\_cero  
at java.base/java.util.ArrayList.forEach(ArrayList.java:1511) <9 internal l:

-Voy a la clase Coche y ponemos que en desacelerar si la velocidad es menor que cero, velocidad sea igual que cero:



```
public int velocidad;
```

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

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

-Le doy a Iniciar para ejecutar los test y vemos que todos los he pasado:

The screenshot shows the IntelliJ IDEA interface. On the left, there's a sidebar with links like 'Área personal', 'Eventos', 'Este curso', and a user profile for 'DIEGO PEREZ ANTON'. The main area has two tabs open: 'TestCoche.java' and 'Coche.java'. The code editor shows Java code for a car class with methods for accelerating and decelerating. Below the code editor is the 'Run' tool window, which displays the output of running the tests: 'Tests passed: 4 of 4 tests - 16 ms' and 'Process finished with exit code 0'. The status bar at the bottom indicates '4 files committed: TestPart5'.

-Hago el quinto commit (TestPart5):

This screenshot is similar to the previous one, showing the IntelliJ IDEA interface with the same sidebar and tabs. However, the 'Commit' tool window is open in the foreground, showing a list of changes: 'Changes 4 files' with four checked files: 'Coche.class', 'Coche.java', 'TestCoche.class', and 'TestCoche.java'. Below this, it says '4 modified'. At the bottom of the commit window are 'Commit' and 'Commit and Push...' buttons.

-Commit realizado:

This screenshot shows the IntelliJ IDEA interface again. The sidebar and tabs are identical to the previous screenshots. The 'Run' tool window is open, showing the same test results: 'Tests passed: 4 of 4 tests - 16 ms' and 'Process finished with exit code 0'. The status bar at the bottom indicates '4 files committed: TestPart5'.

-En la consola, introduzco las siguientes líneas que están en github para iniciar y unir el proyecto al repositorio creado en github y el primer commit:

```

WhatsApp Twitch Área personal Oracle / PLSQL: INS...
Terminal: Local + v
PS C:\Users\Dierez\Desktop\DAW1\Entornos de desarrollo\3 Trimestre\Tema 10\TDD Practica Coche> git init
Reinitialized existing Git repository in C:/Users/Diezrez/Desktop/DAW1/Entornos de desarrollo/3 Trimestre/Tema 10/TDD Practica Coche/.git/
PS C:\Users\Dierez\Desktop\DAW1\Entornos de desarrollo\3 Trimestre\Tema 10\TDD Practica Coche> git add README.md
PS C:\Users\Dierez\Desktop\DAW1\Entornos de desarrollo\3 Trimestre\Tema 10\TDD Practica Coche> git commit -m "first commit"
[master d4a6d58] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md

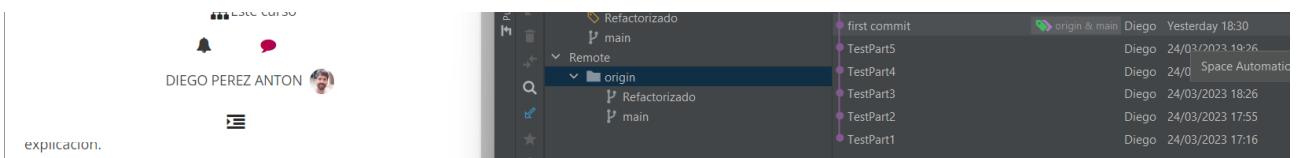
```

```

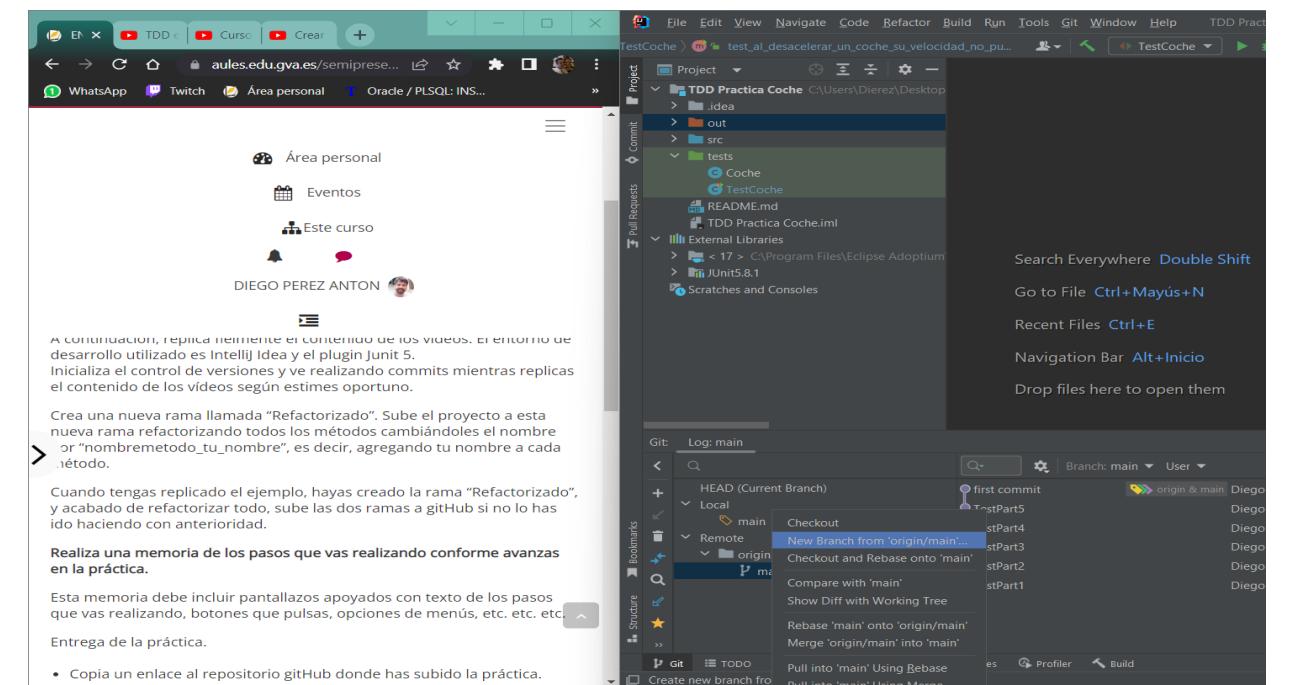
File Edit View Navigate Code Refactor Build Run Tools Git Window Help TDD Practica Coche - TestCoche.java
tests > TestCoche > test_al_crear_un_coche_su_velocidad_es_cero > MainJava > Coche.java >
PS C:\Users\Dierez\Desktop\DAW1\Entornos de desarrollo\3 Trimestre\Tema 10\TDD Practica Coche> git push -u origin main
Enumerating objects: 59, done.
Delta compression using up to 20 threads
Compressing objects: 100% (49/49), done.
Writing objects: 100% (59/59), 6.70 KiB | 2.23 MiB/s, done.
Total 59 (delta 16), reused 0 (delta 0), pack-reused 0
Compressing objects: 100% (49/49), done.
Writing objects: 100% (59/59), 6.70 KiB | 2.23 MiB/s, done.
Compressing objects: 100% (49/49), done.
Compressing objects: 100% (49/49), done.
branch 'main' set up to track 'origin/main'.

```

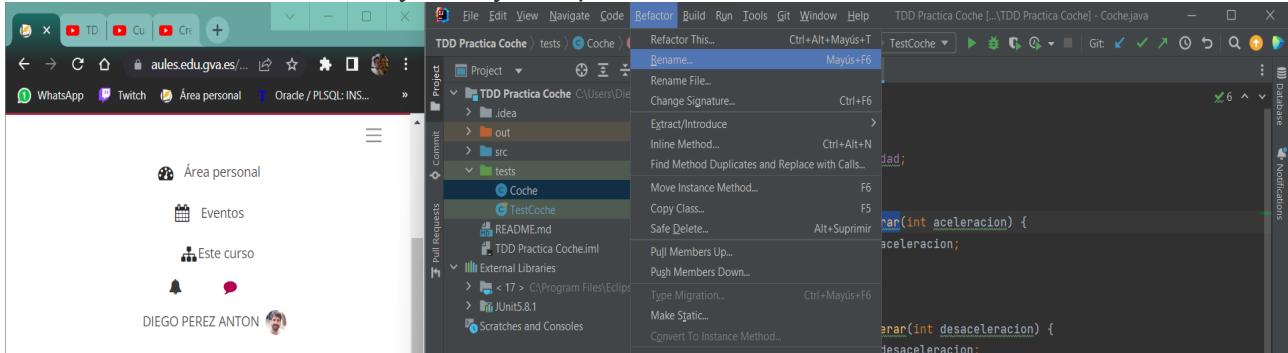
-Podemos ver que se ha subido correctamente al repositorio de github:



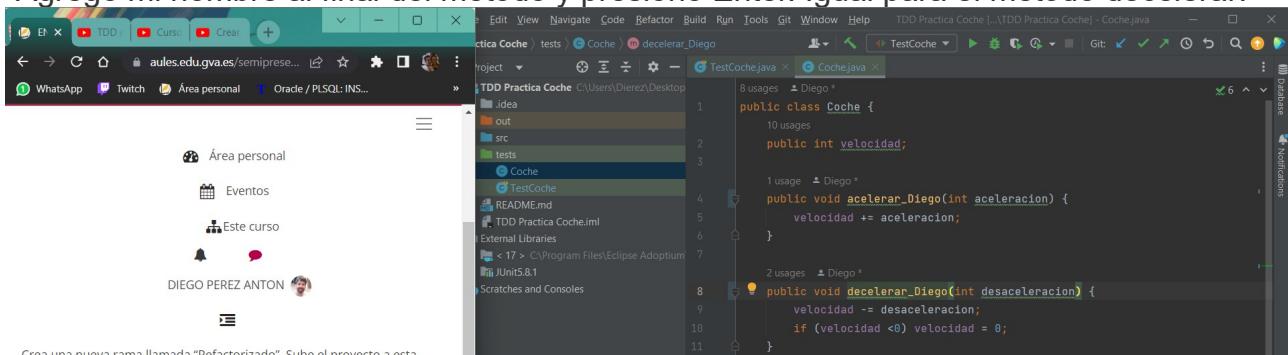
--Le doy a la pestaña de Git que está en la parte inferior izquierda y dentro de la rama main, creo la rama Refactorizado (botón derecho del ratón en main):



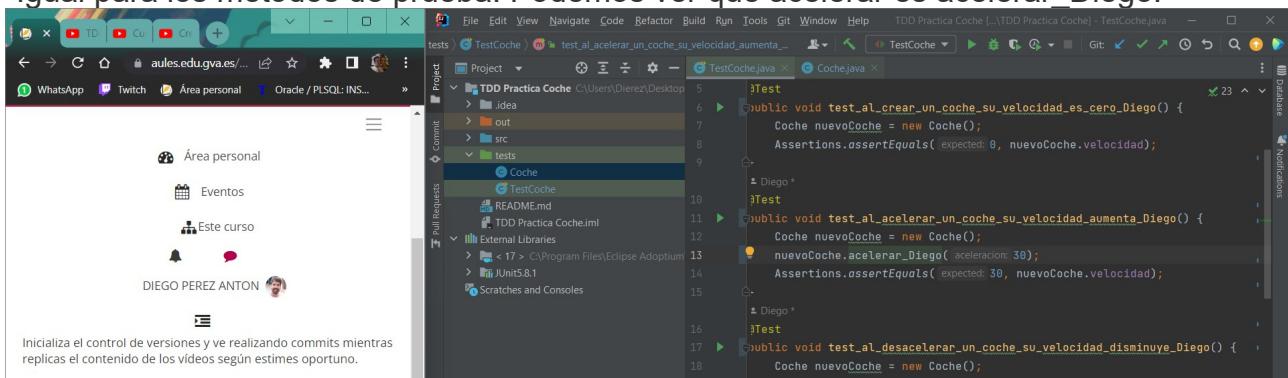
-Señalo el método acelerar y le doy a la pestaña Refactor/Rename:



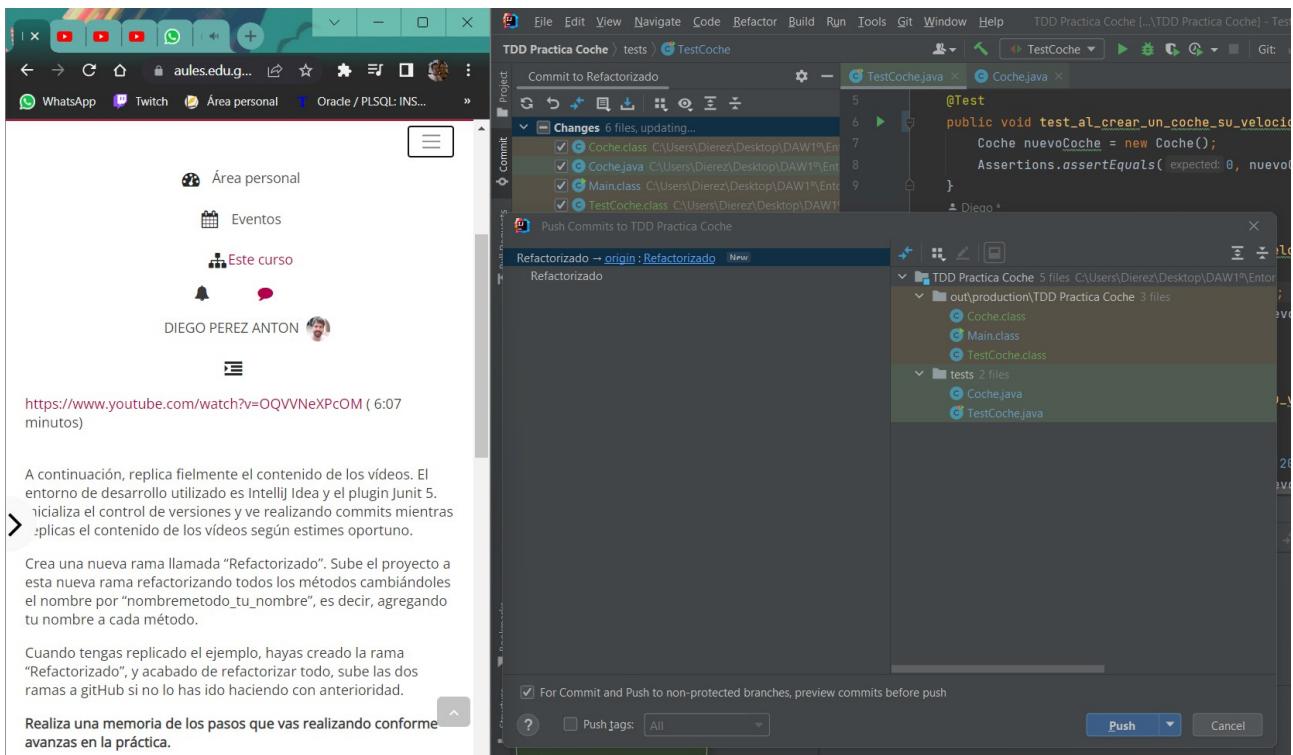
-Agrego mi nombre al final del método y presiono Enter. Igual para el método decelerar:



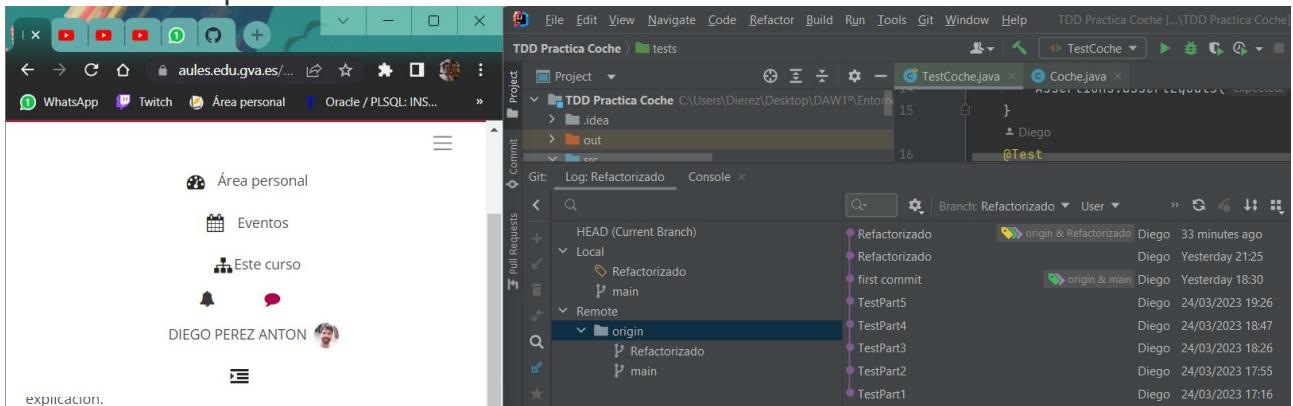
-Igual para los métodos de prueba. Podemos ver que acelerar es acelerar\_Diego:



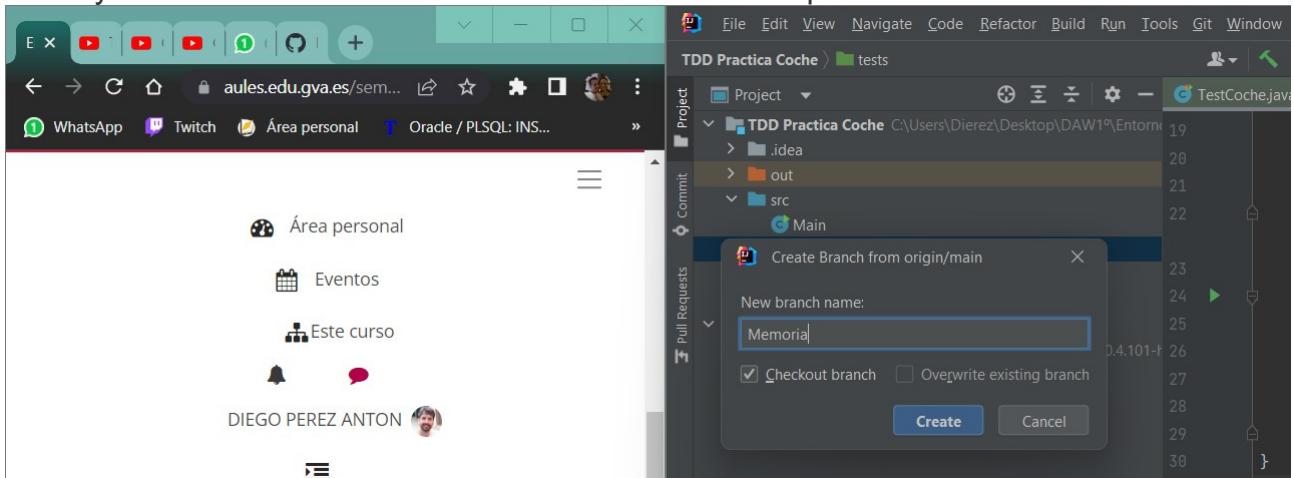
-Hago commit push para subir la rama al repositorio de github:



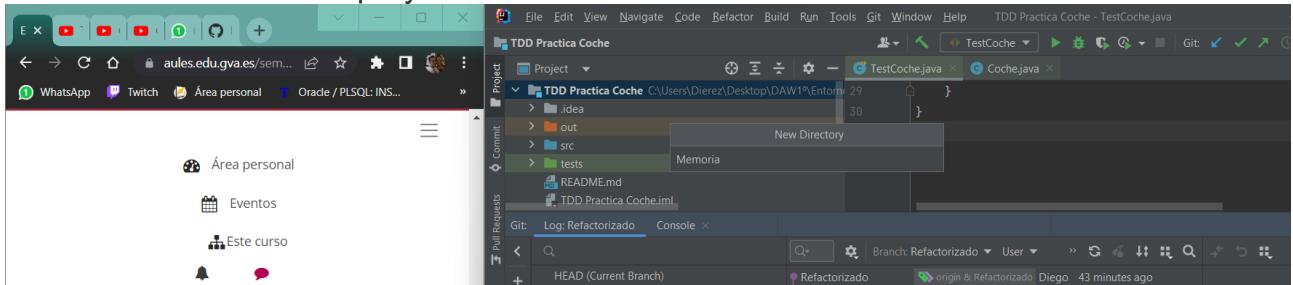
-Podemos ver que se ha subido:



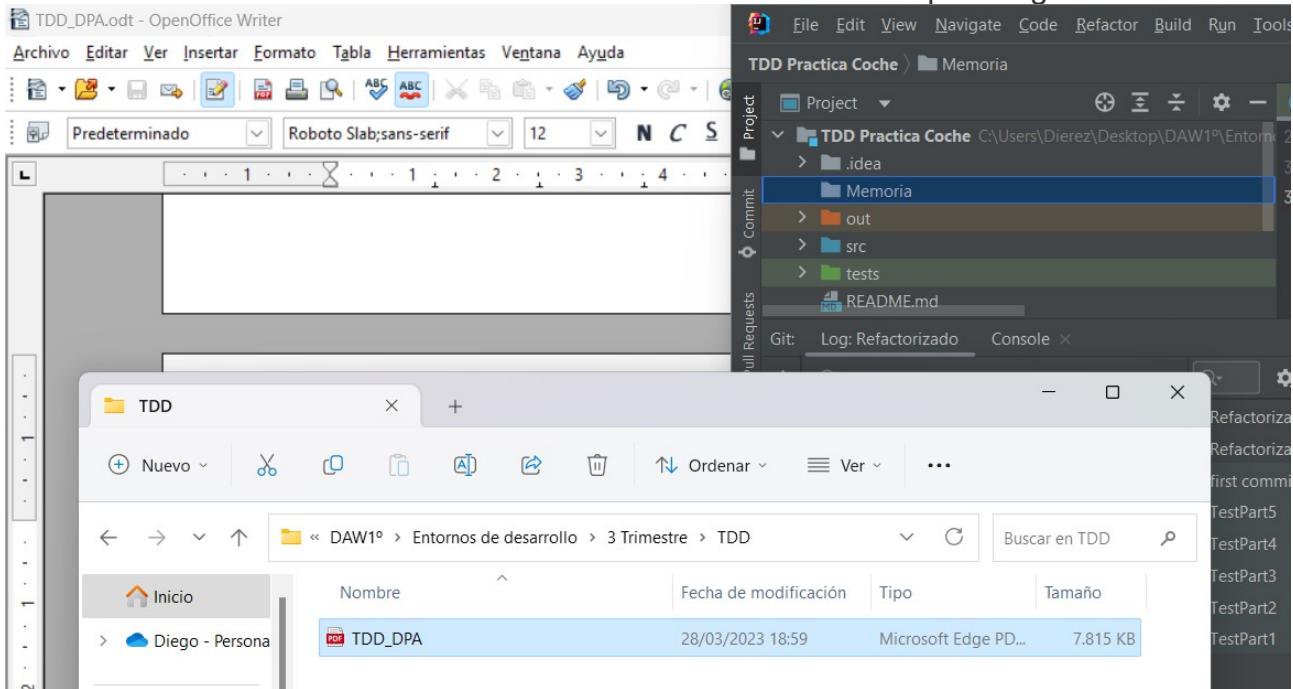
-Creo y me muevo a la rama Memoria del mismo modo que la rama refactorizado:



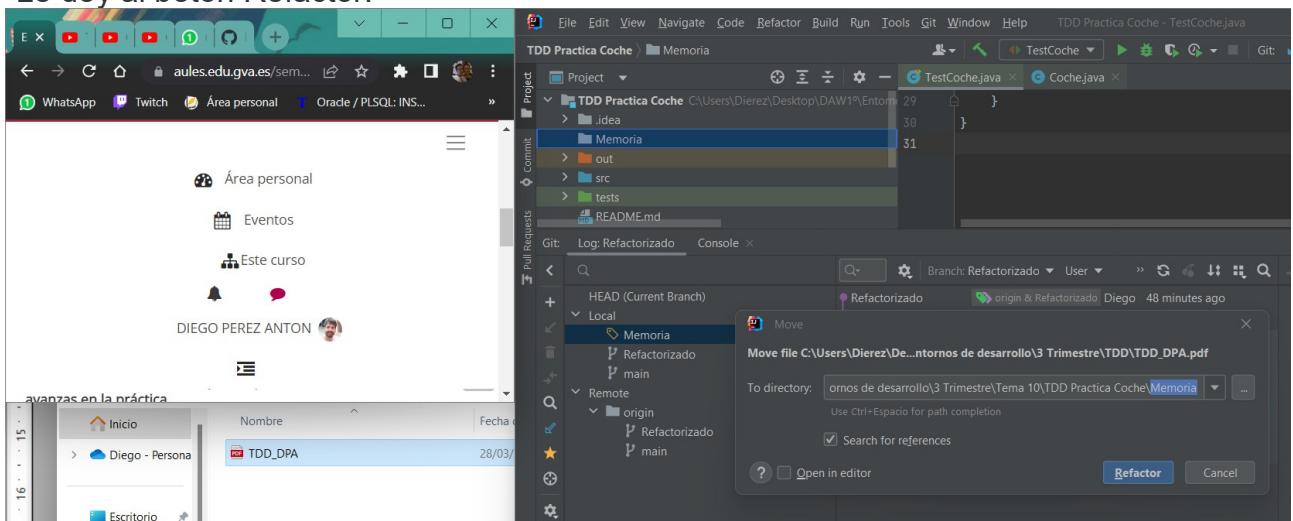
-Creo un directorio en el proyecto llamado Memoria:



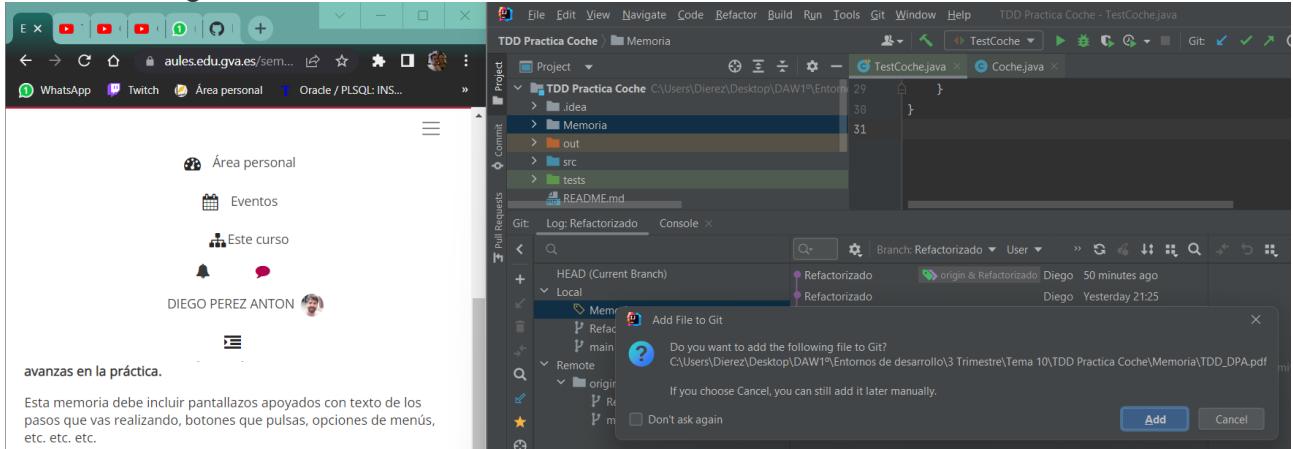
-Dentro del directorio arrastro el archivo de la memoria realizada por Diego Pérez Antón:



-Le doy al botón Refactor:



-Lo añado a git:



-Por último hago commit and push para subir los cambios en la rama Memoria.