

UD 10:

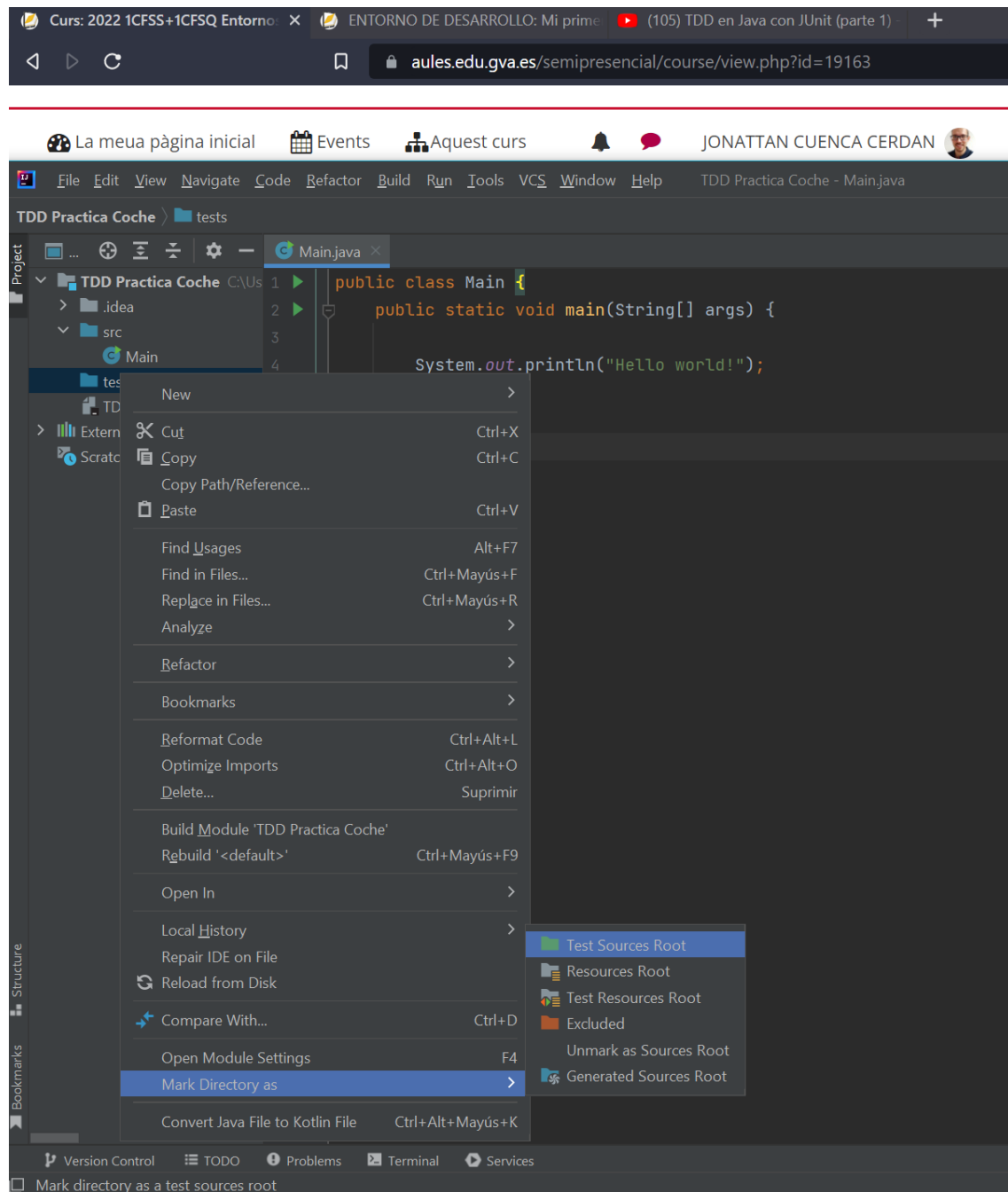
Práctica

Mi primer TDD

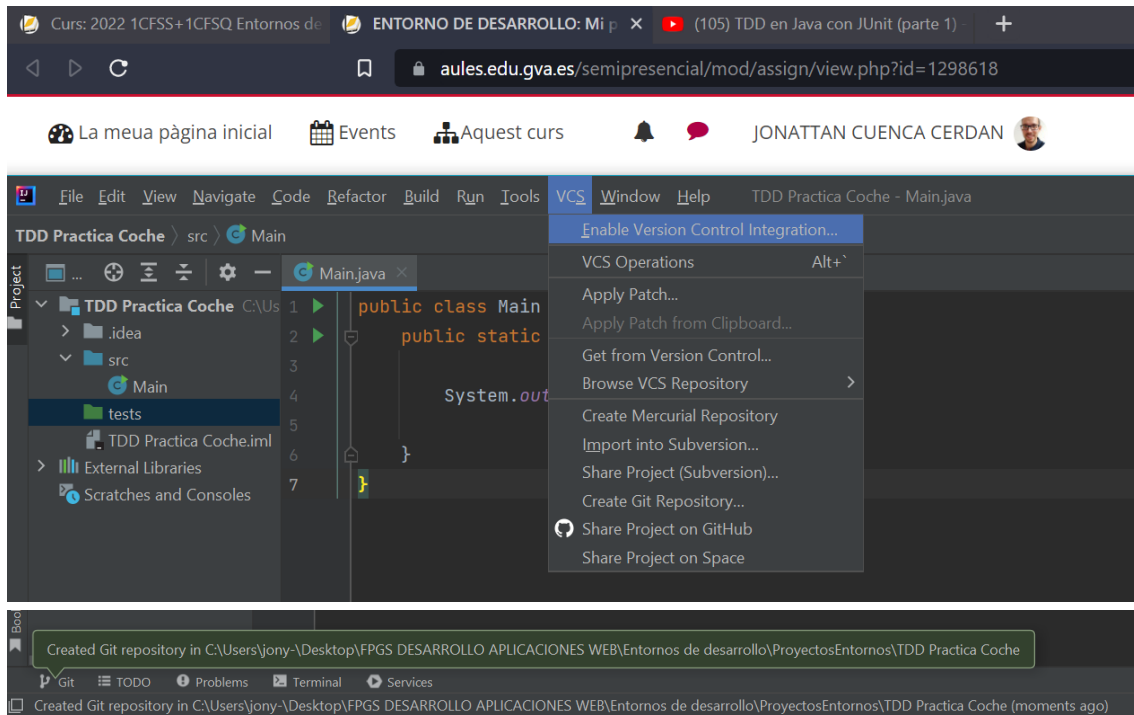
Vamos a poner en práctica el TDD en un proyecto de java utilizando el IDE de IntelliJ Idea. Para esto, creamos un nuevo proyecto y lo nombramos como TDD Practica Coche.

Añadimos un nuevo directorio donde vamos a almacenar los tests: Click Derecho del ratón en el proyecto / New / Directory.

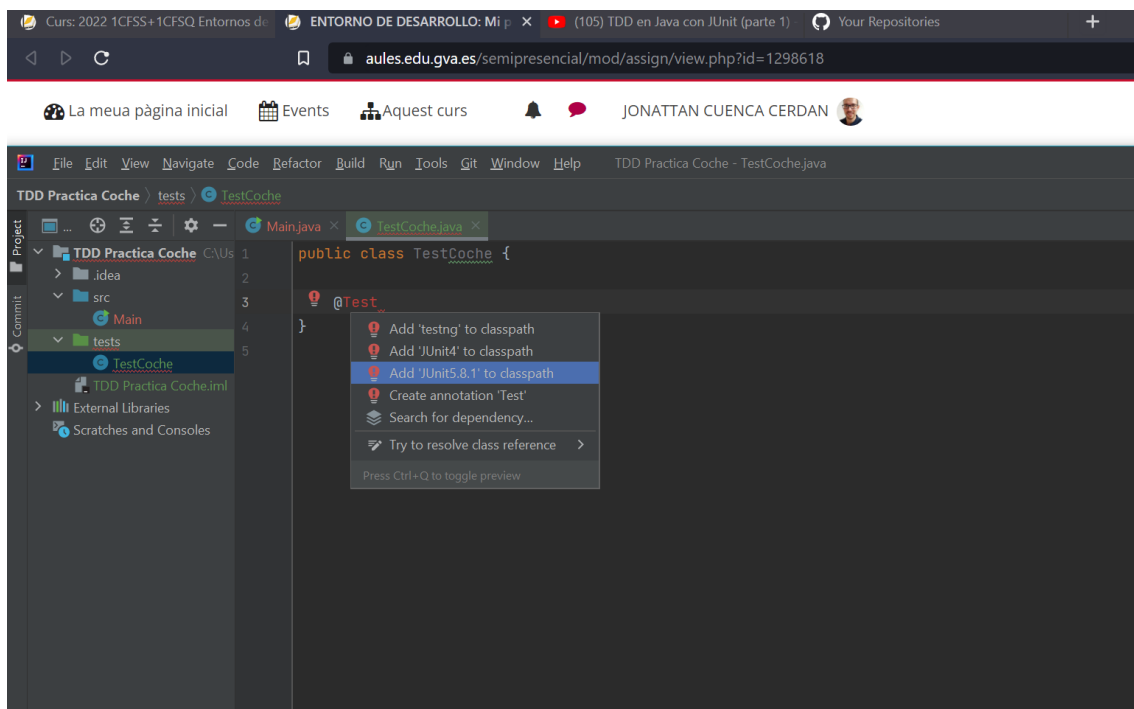
A continuación, marcamos el nuevo directorio creado como directorio de test.



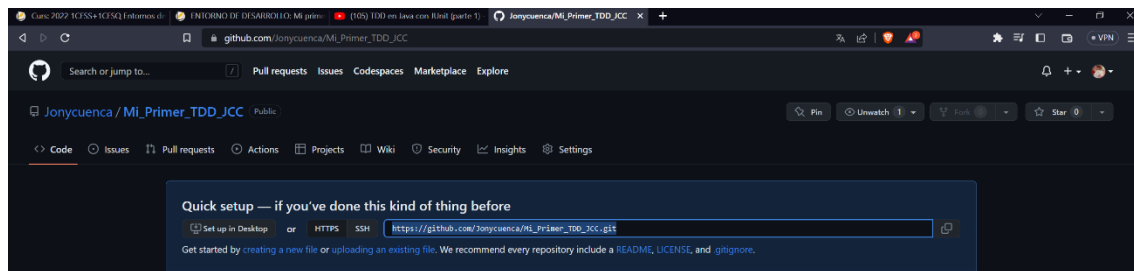
Vamos a iniciar el control de versiones con Git.



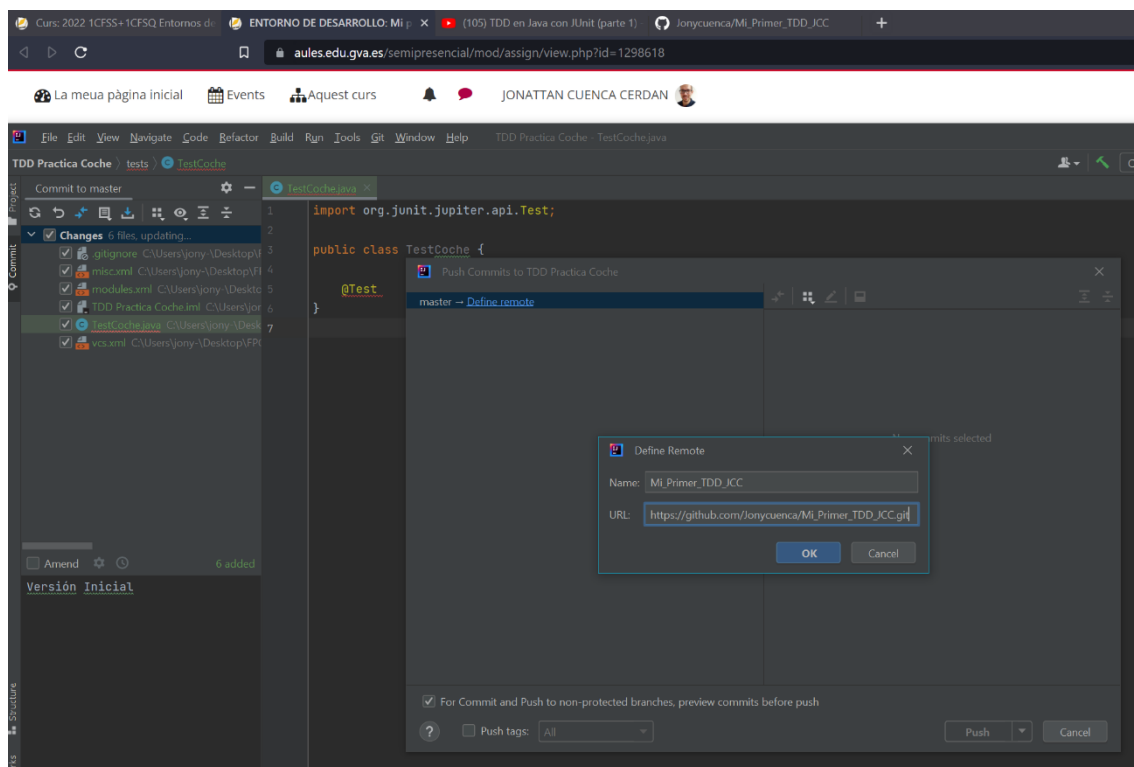
Creamos una clase en el directorio “tests” con el nombre de TestCoche e indicamos que es un test con @Test. Hay que indicar que utilice JUnit7, esto se realiza pulsando Alt+Enter sobre la palabra @Test y clicar en Add JUnit5. Añadimos la opción de Annotations.



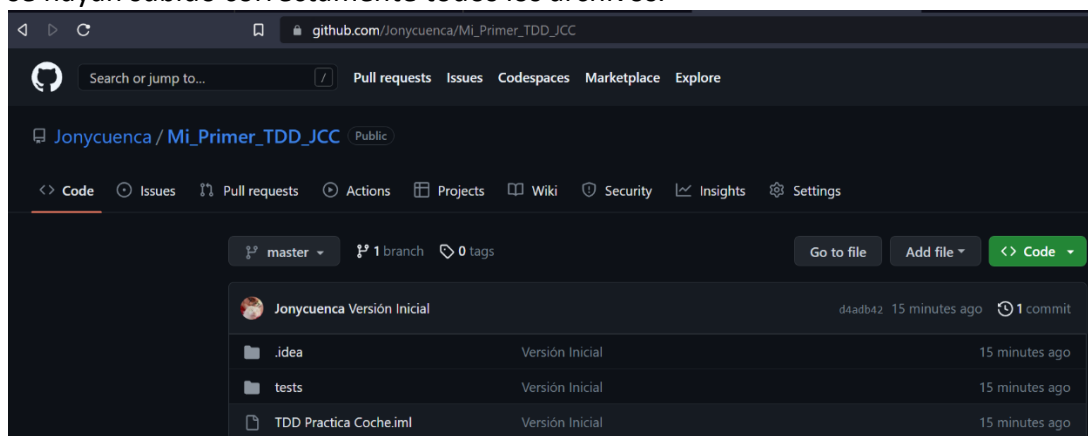
Realizamos el primer commit del proyecto ahora que tenemos preparado el proyecto para ir introduciendo los métodos. Para ello, pulsamos CTRL+K, seleccionamos todos los cambios que se han realizado, ponemos un mensaje “Version Inicial” y presionamos el botón Commit and Push para poder subirlo a nuestro directorio remoto de GitHub. Para definir el directorio remoto, accedemos a nuestra cuenta de GitHub, creamos un nuevo repositorio en el apartado de nuestros repositorios y copiamos la URL.



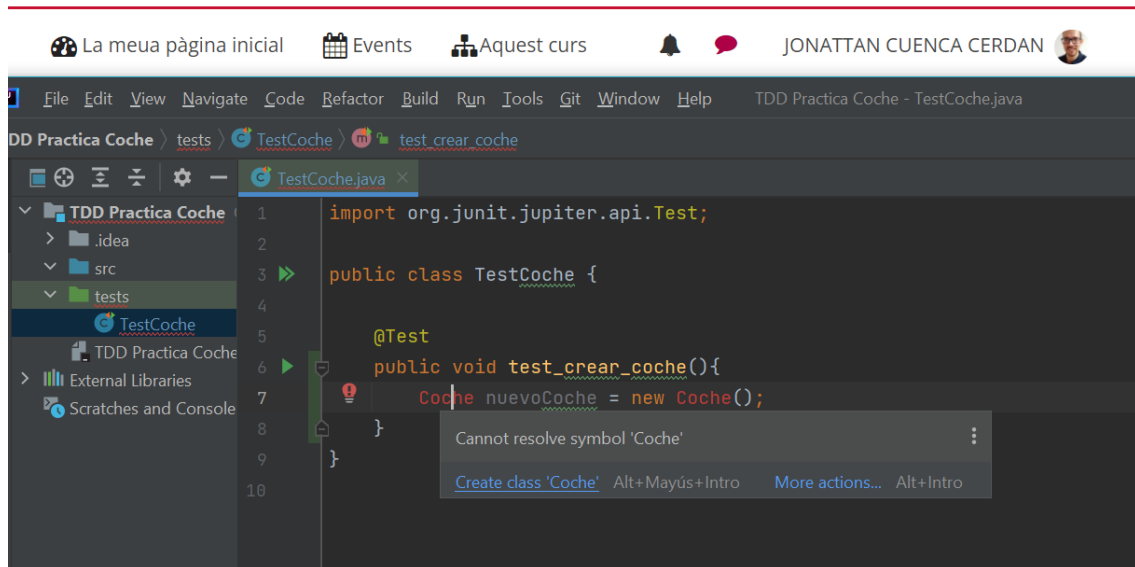
A continuación, en el IDE IntelliJ le asignamos un nombre y introducimos el enlace del repositorio de nuestra cuenta de GitHub.



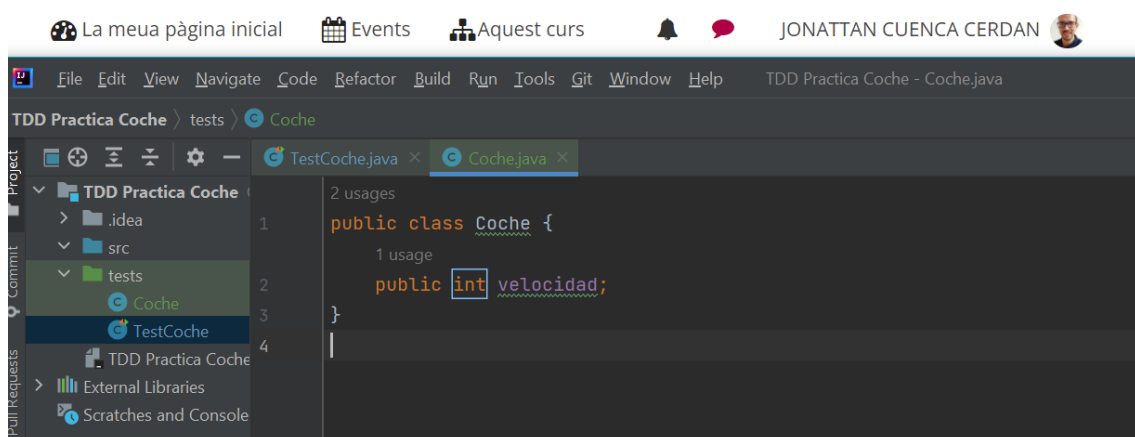
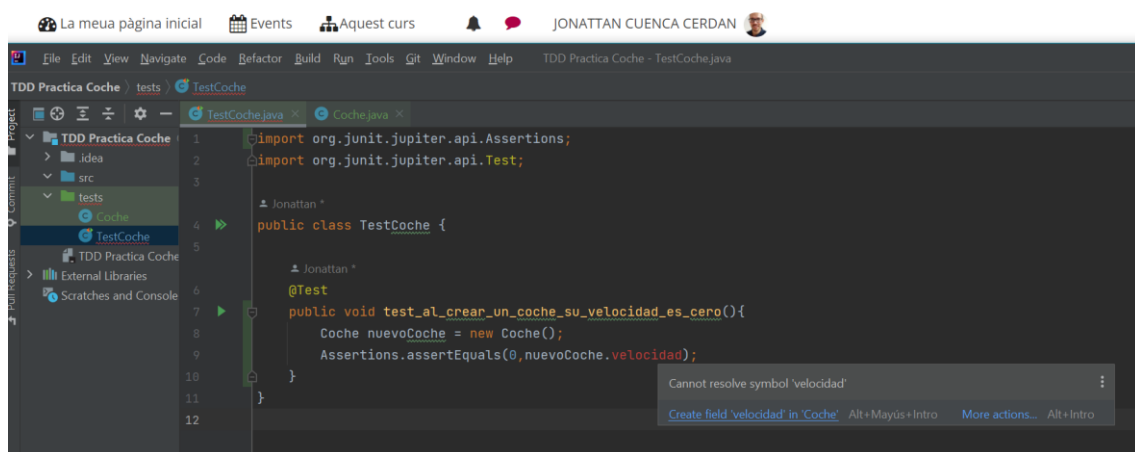
Tras esto, pre visualizamos los archivos que se van a subir al repositorio remoto y si estamos de acuerdo seleccionamos Push. Comprobamos que en el repositorio remoto se hayan subido correctamente todos los archivos.



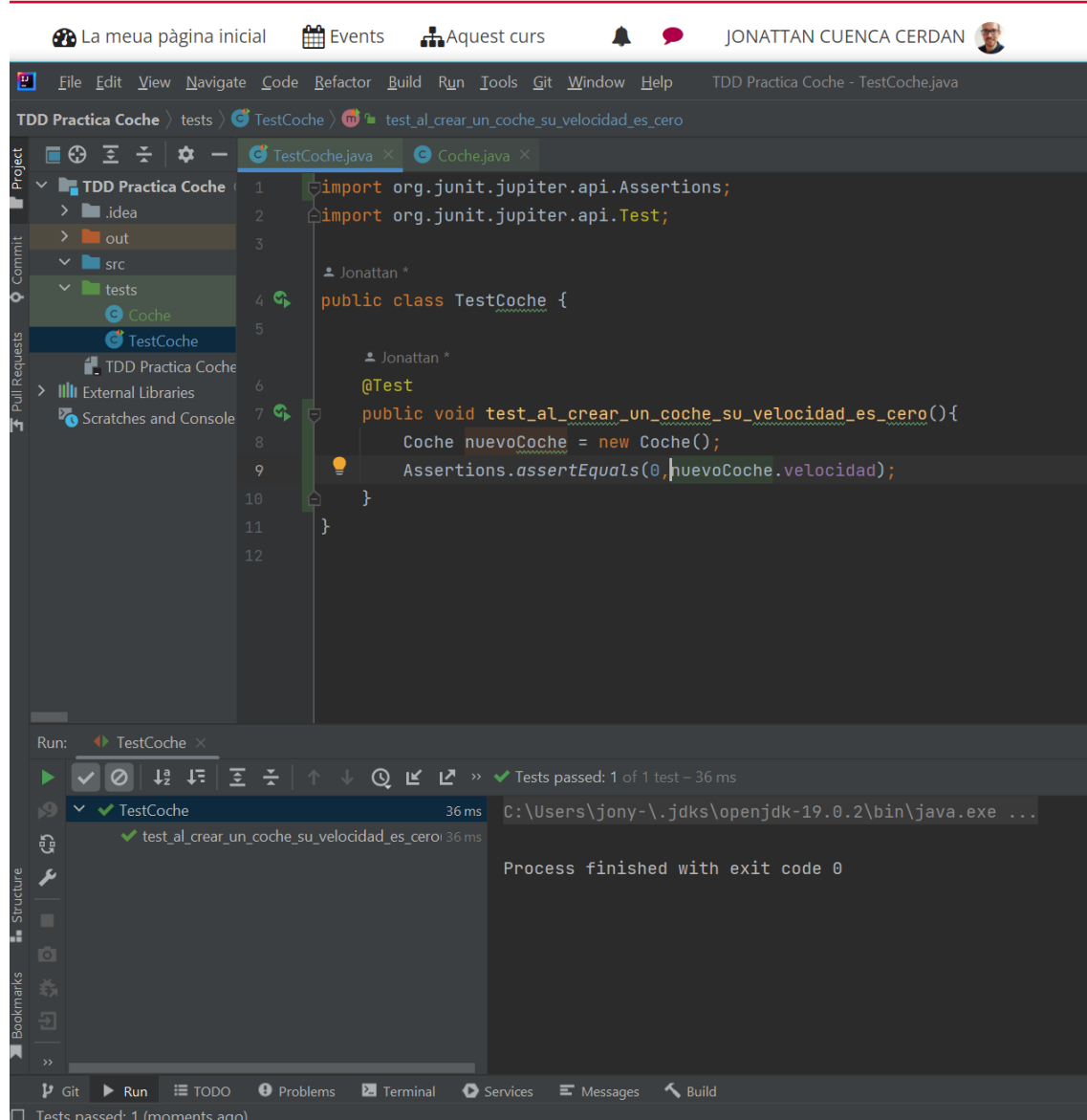
Después de esto, continuamos realizando el proyecto, añadiendo el método `test_crear_coche()`. Añadimos un nuevo objeto coche y como no tenemos la clase coche creada aparece en rojo. Añadimos la nueva clase coche poniendo el cursor encima y seleccionamos Create class Coche.



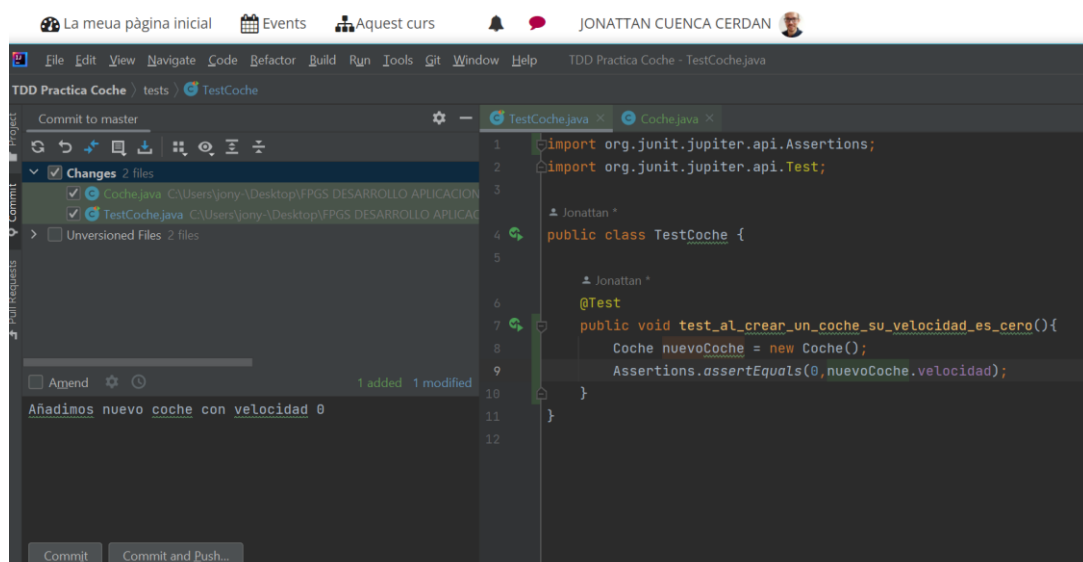
Modificamos el método para asignar al coche la velocidad de 0 cuando lo creamos. Como no tenemos el atributo de velocidad en la clase coche tenemos que añadirla poniendo el cursor encima y seleccionando Create field velocidad en coche.



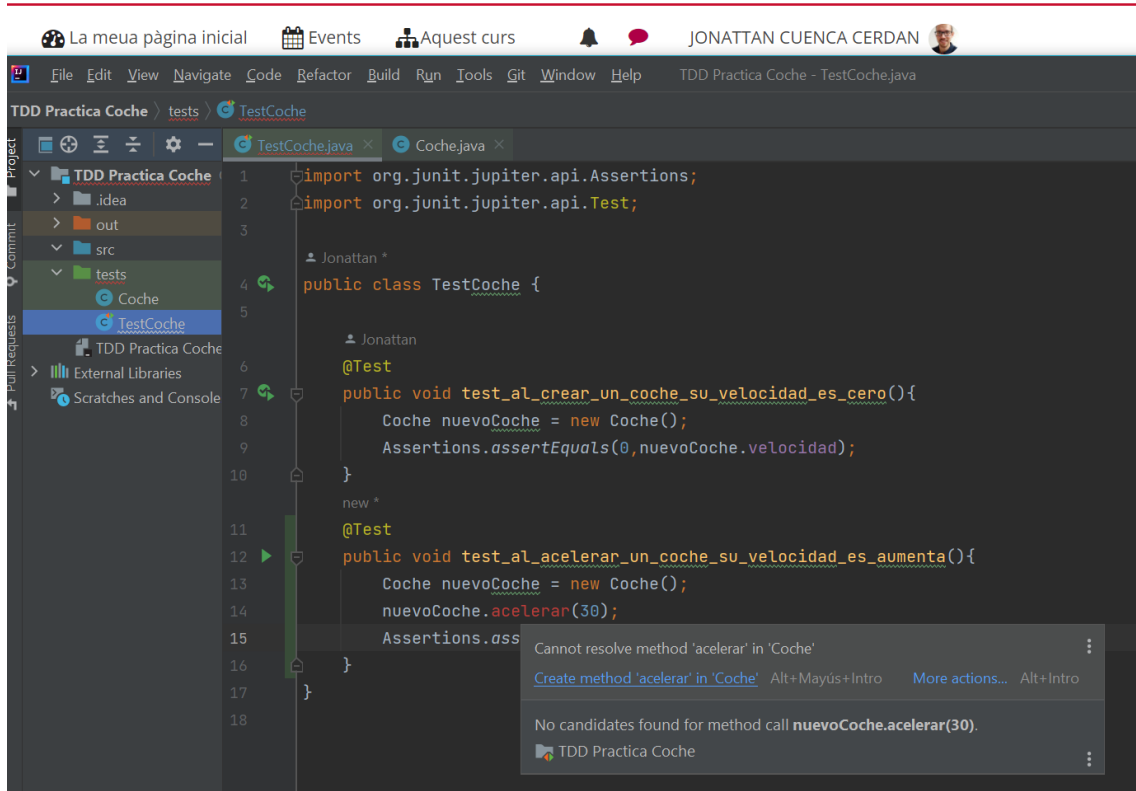
Tras realizar esto, ejecutamos el test y nos informa que ha superado el test correctamente.



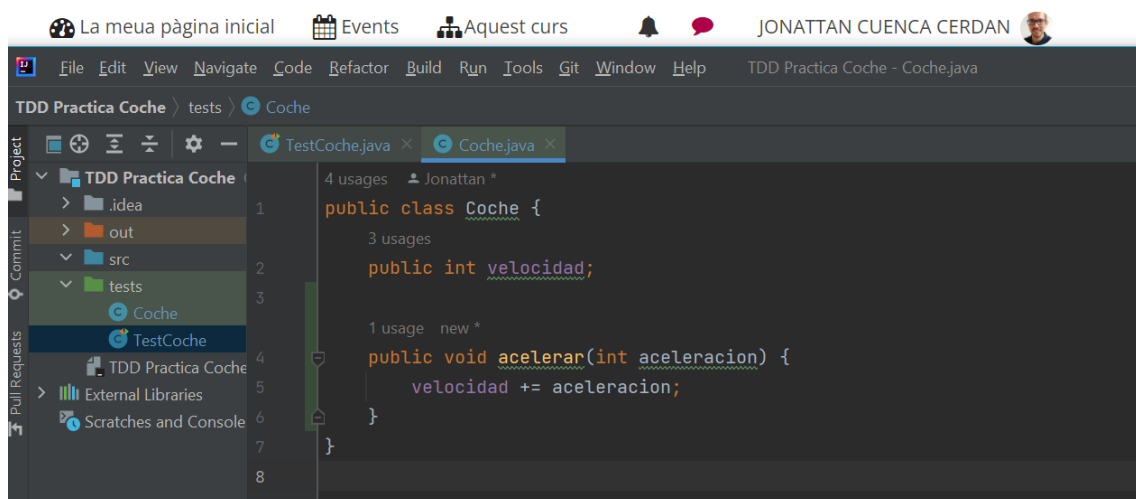
Realizamos un commit and push poniendo de comentario “Añadimos Nuevo coche con Velocidad 0”.



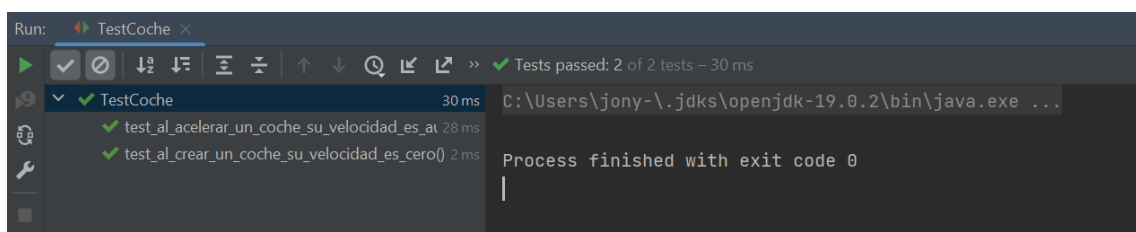
Añadimos un método para indicar que el coche acelera. Como el método acelerar no existe, lo creamos siguiendo los pasos que ya realizamos previamente para crear el atributo velocidad.



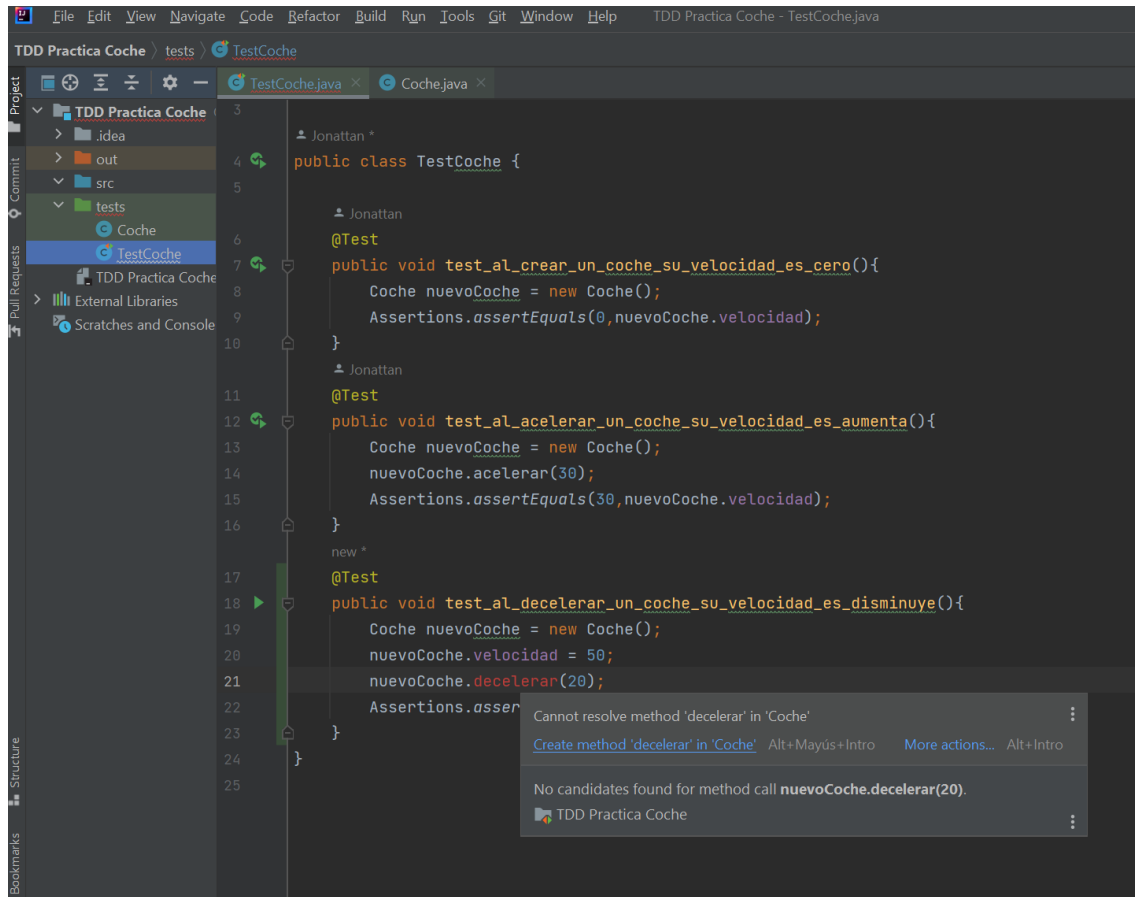
Esta vez, sustituimos el nombre del parámetro i por aceleración y le asignamos una función para que la velocidad aumente con la aceleración.



Ejecutamos el test y lo hemos superado correctamente. Realizamos commit and push con el comentario “Añadimos el método acelerar”

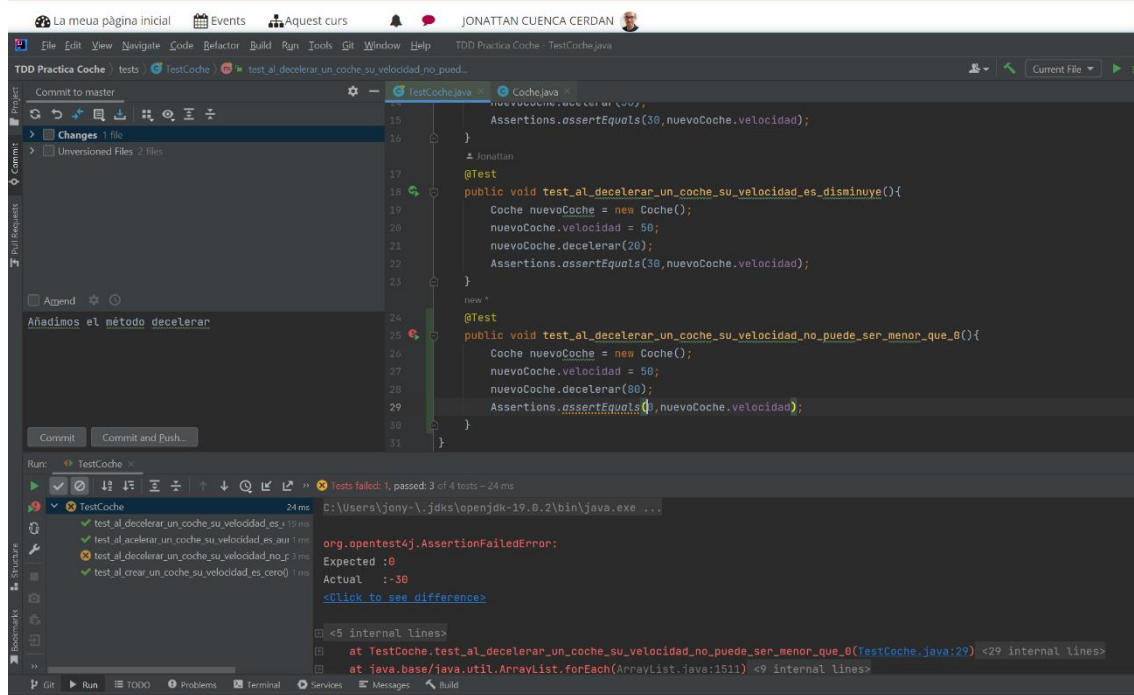


Ahora creamos un método decelerar. Realizamos el mismo proceso que hicimos para crear el método acelerar y asignamos una velocidad al coche > 0 para indicar que el coche está en movimiento y va a reducir su velocidad.

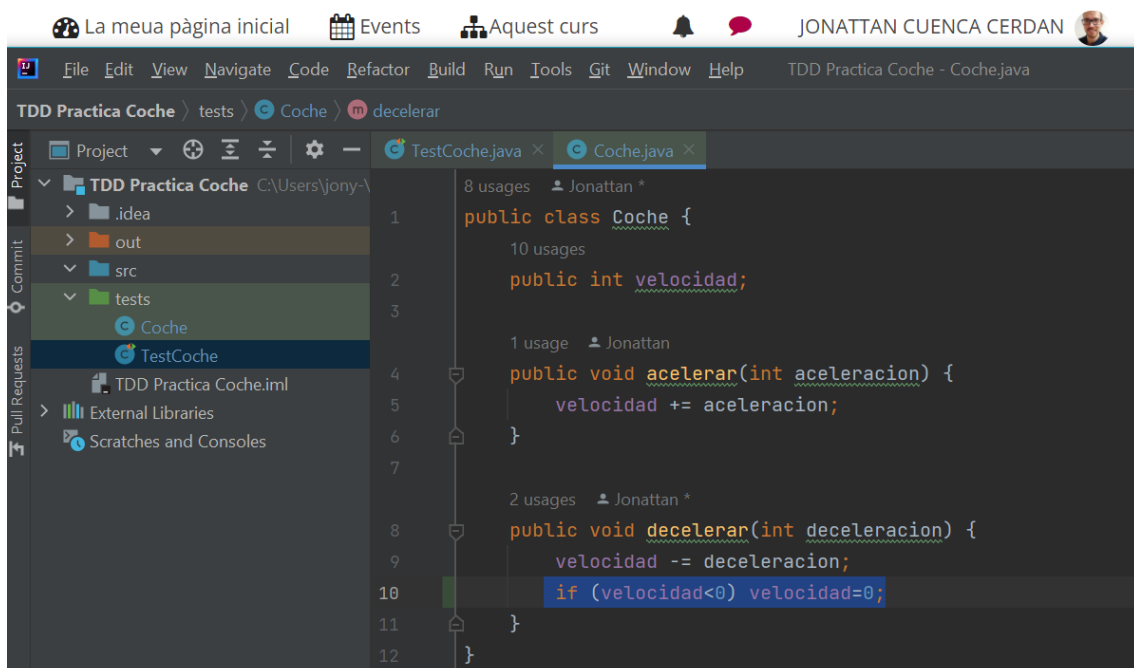


Ejecutamos el test y nos indica que ha superado los 3 procesos. Realizamos un commit and push con el comentario “Añadimos el método decelerar”.

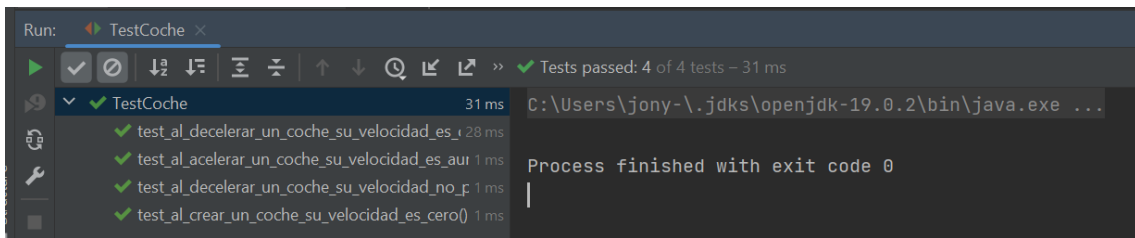
Ahora añadimos un nuevo método para indicar que la velocidad no puede ser menor que 0. En este método, introducimos valores en el que la velocidad tras decelerar quedaría negativa. Entonces, al ejecutar este método, lo compila correctamente pero el resultado no es el esperado y nos falla.



Para solucionar esto, nos dirigimos al método decelerar y añadimos una condición para indicar que la velocidad no puede ser inferior a 0.

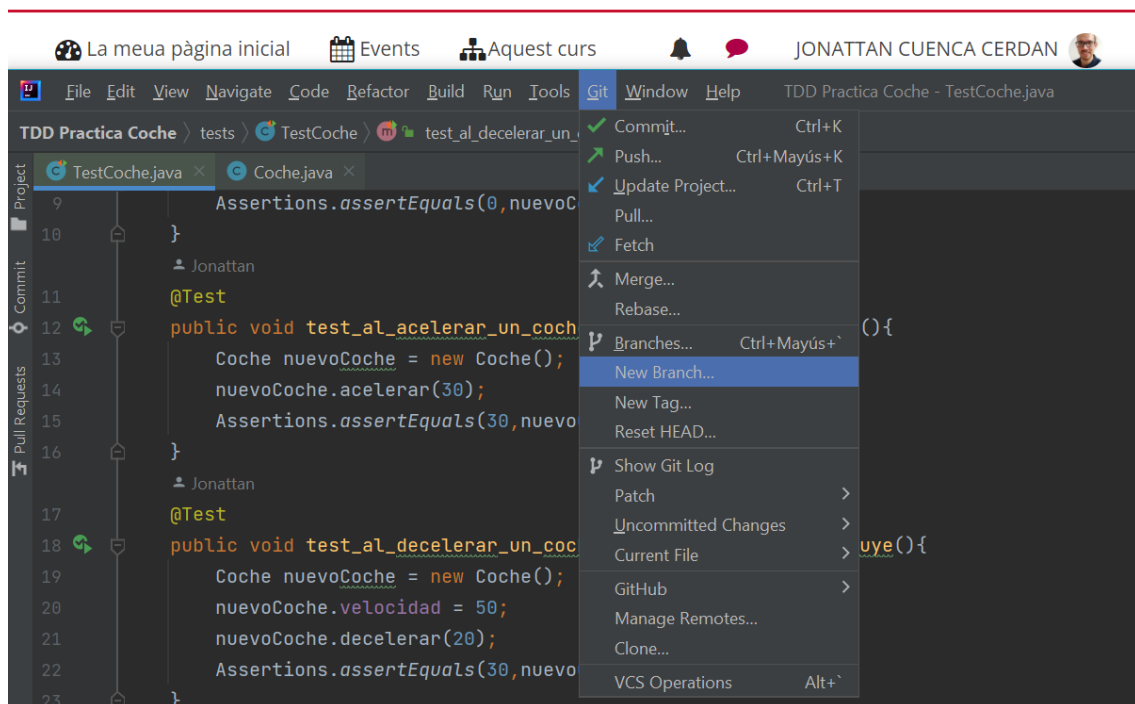


Y así no da fallo.

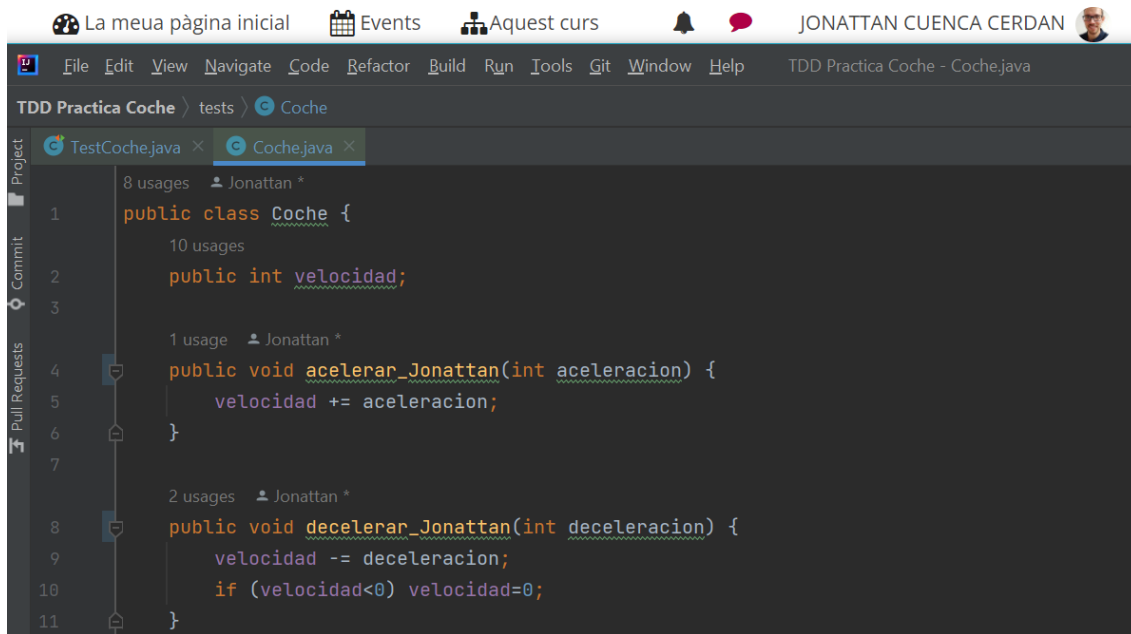


Realizamos un commit and push con el comentario “Modificamos método decelerar para velocidad no puede ser < 0”.

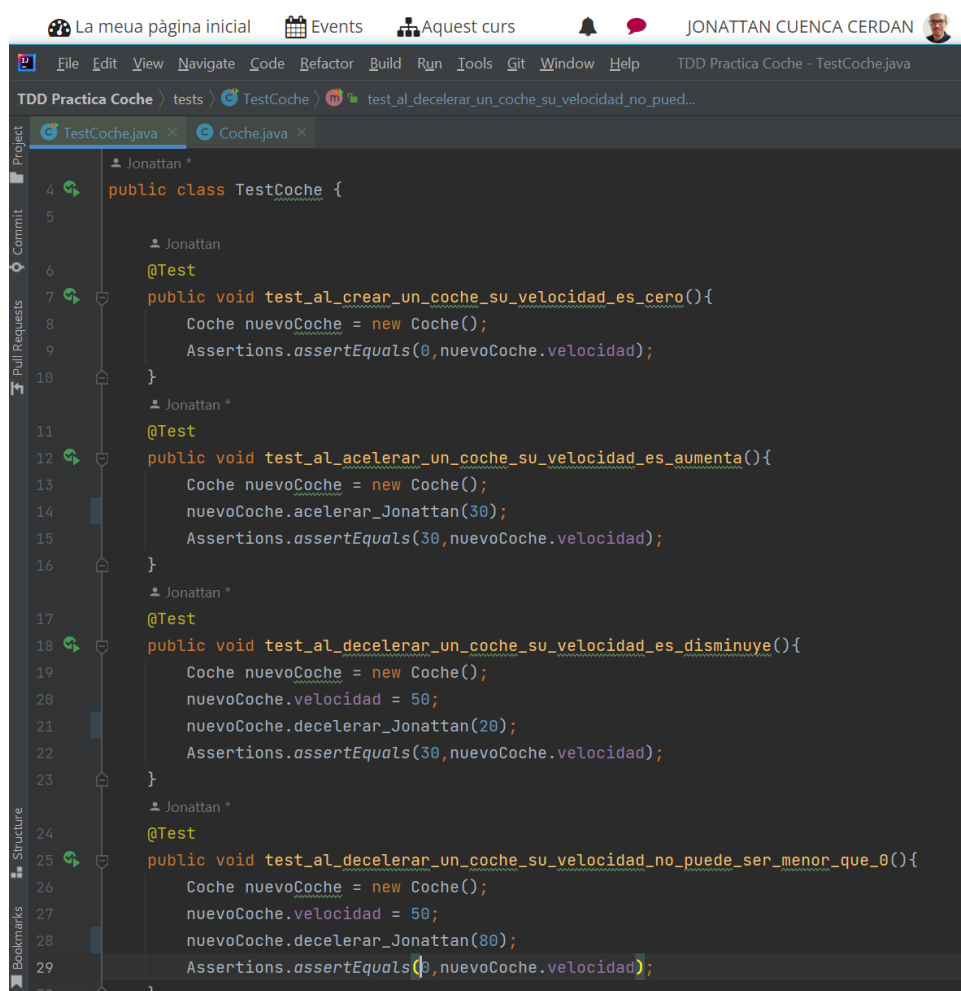
Ya hemos realizado el seguimiento de los vídeos. Ahora creo una rama nueva. Para esto me dirijo al apartado del menú Git / New Branch y le asigno el nombre “Refactorizado”



Una vez hecho esto, me sitúo a la clase Coche y posiciono el cursor encima del método que quiero modificar su nombre, clico en el botón derecho del ratón y selecciono Refactor/Rename. Tras realizar esta acción, vemos como se ha modificado automáticamente el nombre de los métodos donde se hayan utilizado.

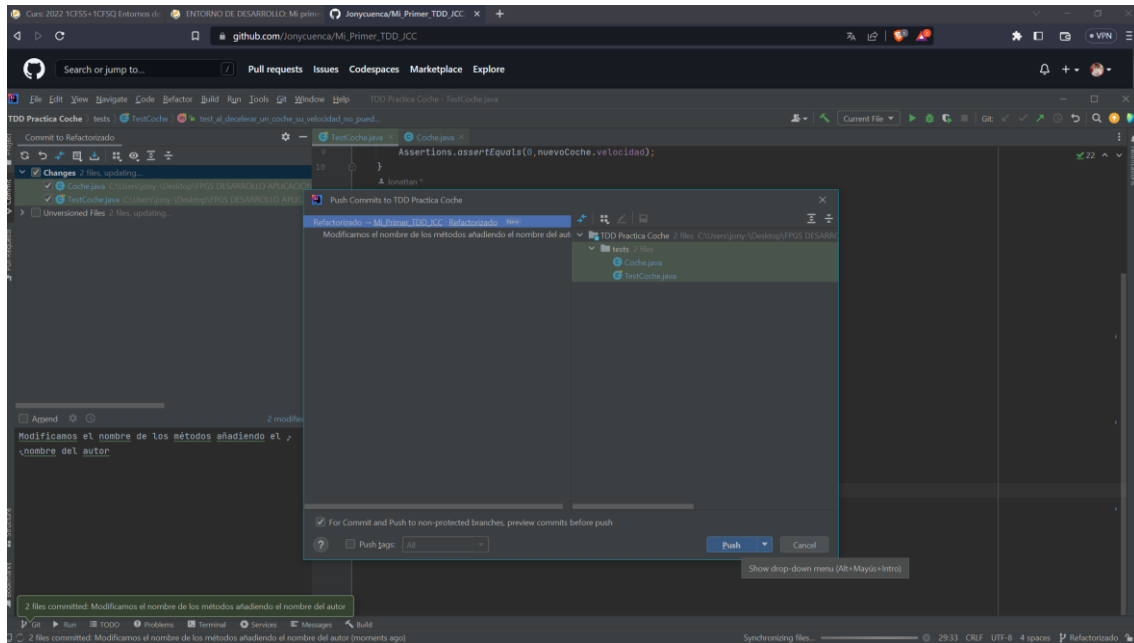


```
1 public class Coche {
2     public int velocidad;
3
4     public void acelerar_Jonattan(int aceleracion) {
5         velocidad += aceleracion;
6     }
7
8     public void decelerar_Jonattan(int deceleracion) {
9         velocidad -= deceleracion;
10        if (velocidad<0) velocidad=0;
11    }
```



```
4 public class TestCoche {
5
6     @Test
7     public void test_al_crear_un_coche_su_velocidad_es_cero(){
8         Coche nuevoCoche = new Coche();
9         Assertions.assertEquals(0,nuevoCoche.velocidad);
10    }
11
12     @Test
13     public void test_al_acelerar_un_coche_su_velocidad_es_aumenta(){
14         Coche nuevoCoche = new Coche();
15         nuevoCoche.acelerar_Jonattan(30);
16         Assertions.assertEquals(30,nuevoCoche.velocidad);
17    }
18
19     @Test
20     public void test_al_decelerar_un_coche_su_velocidad_es_disminuye(){
21         Coche nuevoCoche = new Coche();
22         nuevoCoche.velocidad = 50;
23         nuevoCoche.decelerar_Jonattan(20);
24         Assertions.assertEquals(30,nuevoCoche.velocidad);
25    }
26
27     @Test
28     public void test_al_decelerar_un_coche_su_velocidad_no_puede_ser_menor_que_0(){
29         Coche nuevoCoche = new Coche();
30         nuevoCoche.velocidad = 50;
31         nuevoCoche.decelerar_Jonattan(80);
32         Assertions.assertEquals(0,nuevoCoche.velocidad);
33    }
34 }
```

Realizamos un commit and push de la nueva rama “Refactorizado”



Tras esto, creamos una nueva rama “Memoria” y guardamos el archivo PDF de la memoria dentro del directorio del proyecto. Realizamos un commit and Push en la rama Memoria para que se quede almacenada junto con el proyecto.