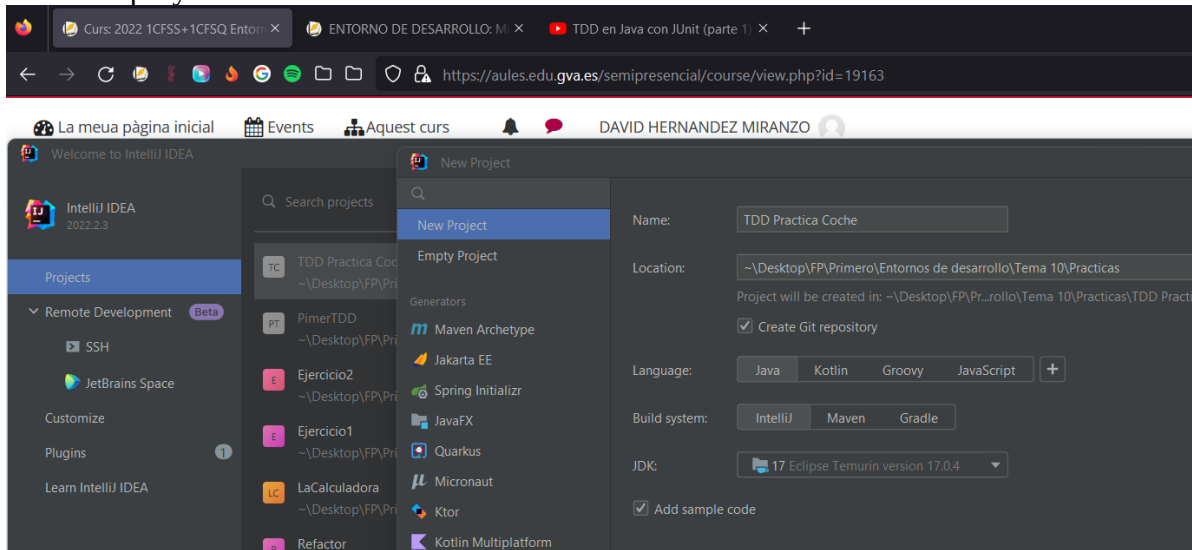


David Hernández Miranzo

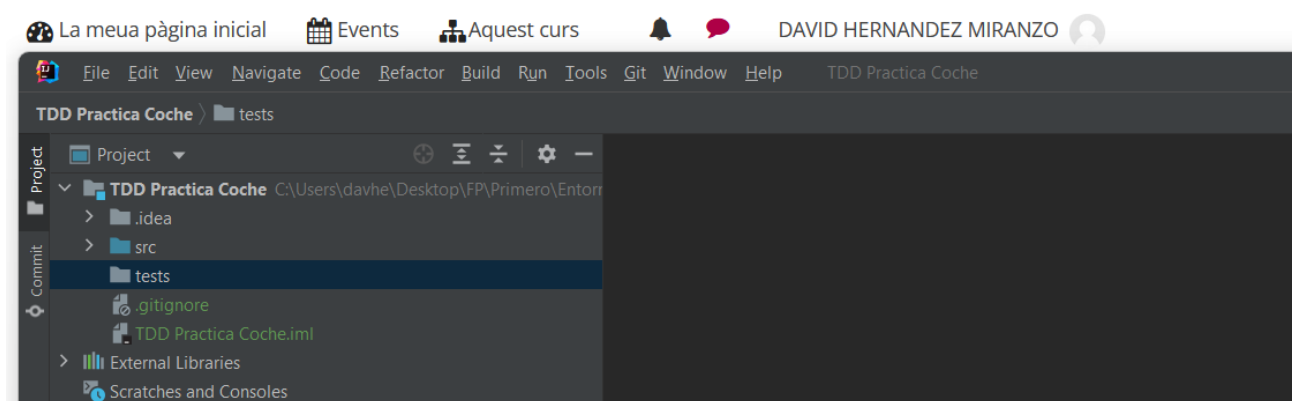
# **Mi primer TDD**

## PARTE 1

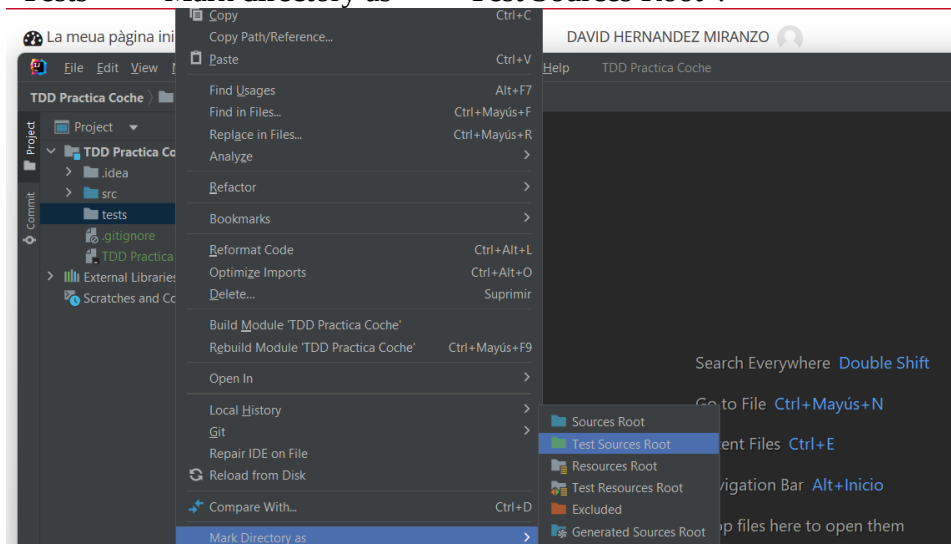
### Creación proyecto:



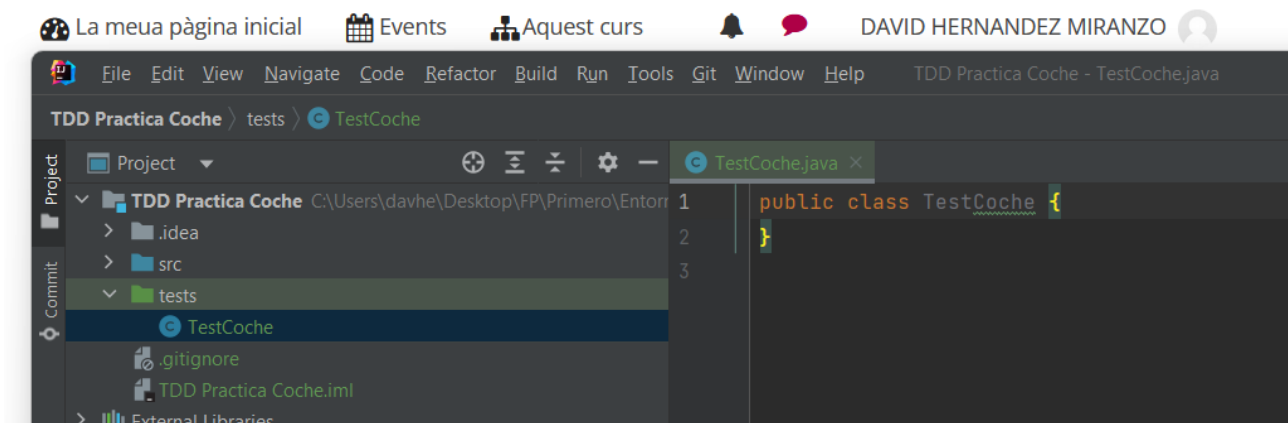
Creación de nuevo directorio para los test haciendo click derecho sobre el nombre del proyecto → new → directory:



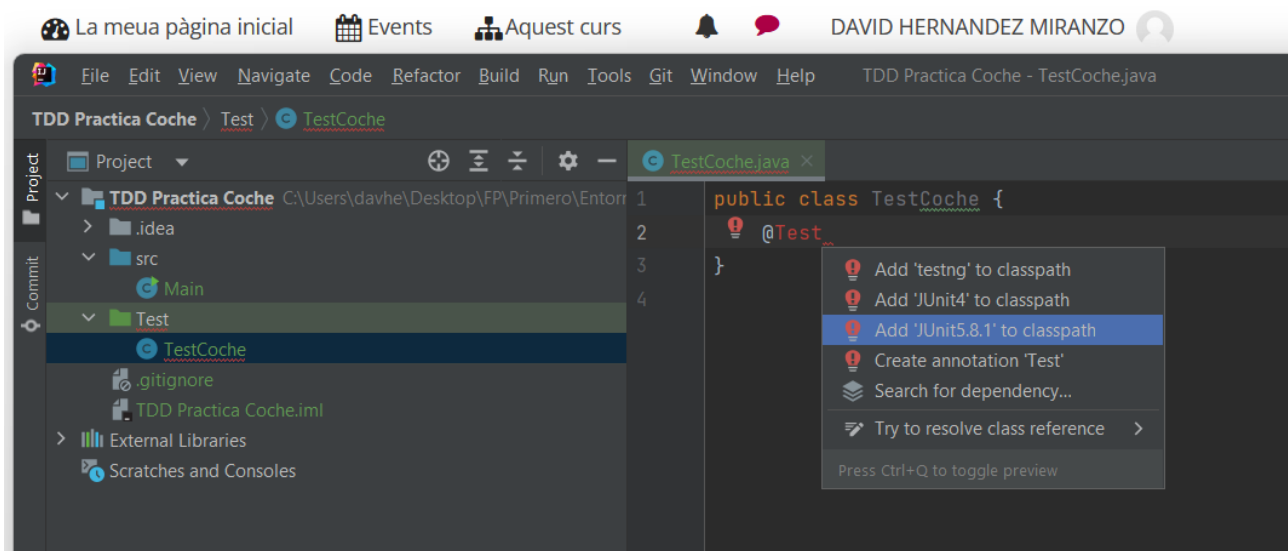
Ahora se marca el directorio como un directorio de test haciendo click derecho sobre la carpeta “Tests” → “Mark directory as” → “Test Sources Root”:



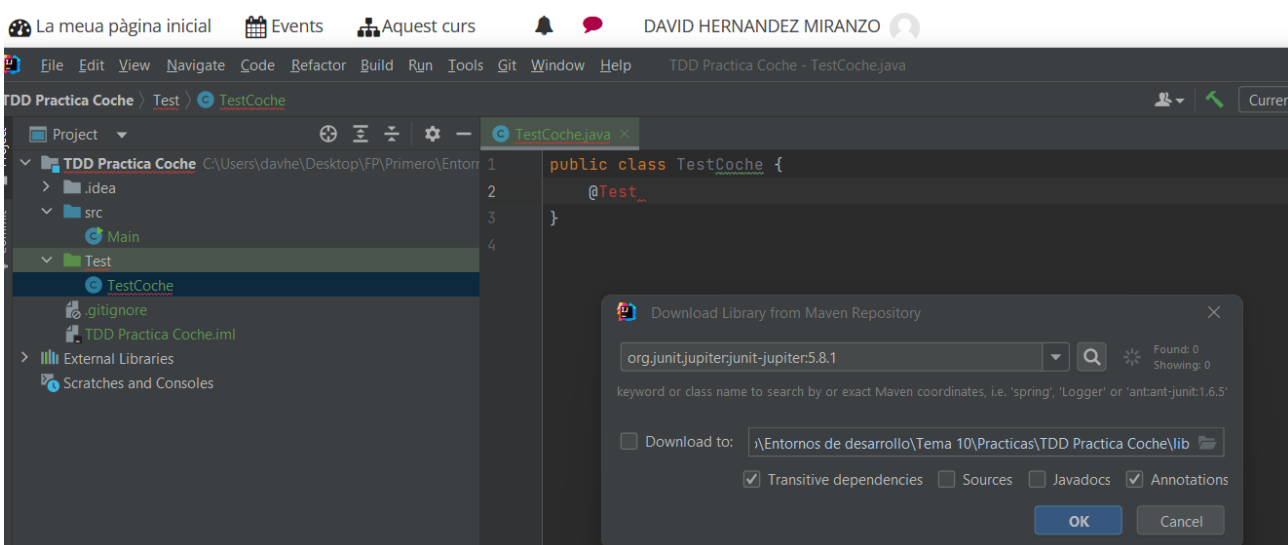
Creación de nueva clase CocheTest dentro de la carpeta Tests y añadido a git.



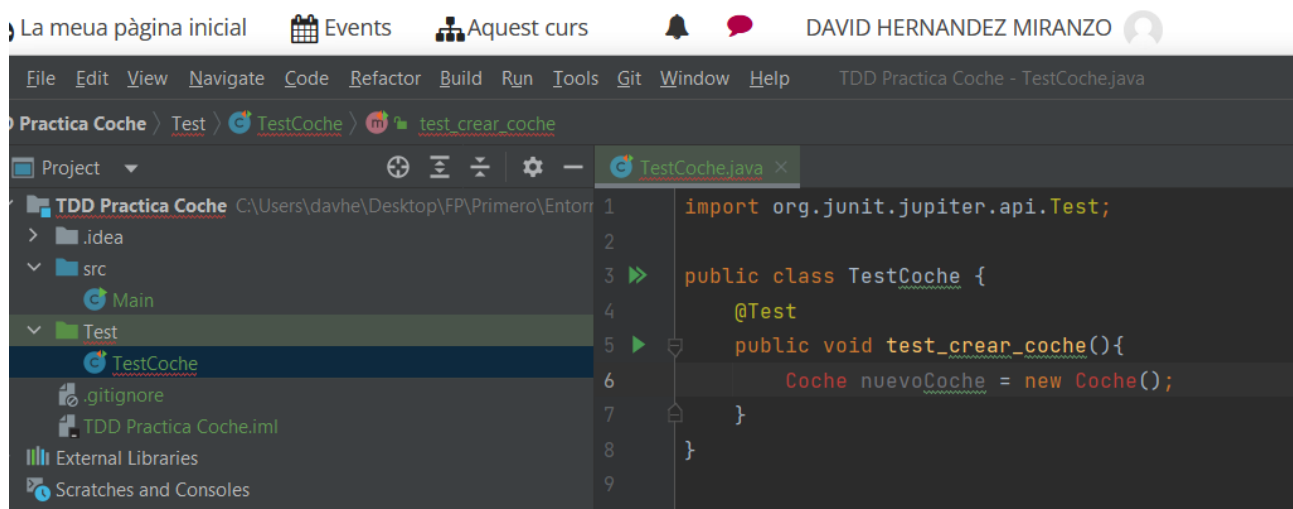
Escribimos la etiqueta @Test y se presiona Alt+Enter para importar Junit5:



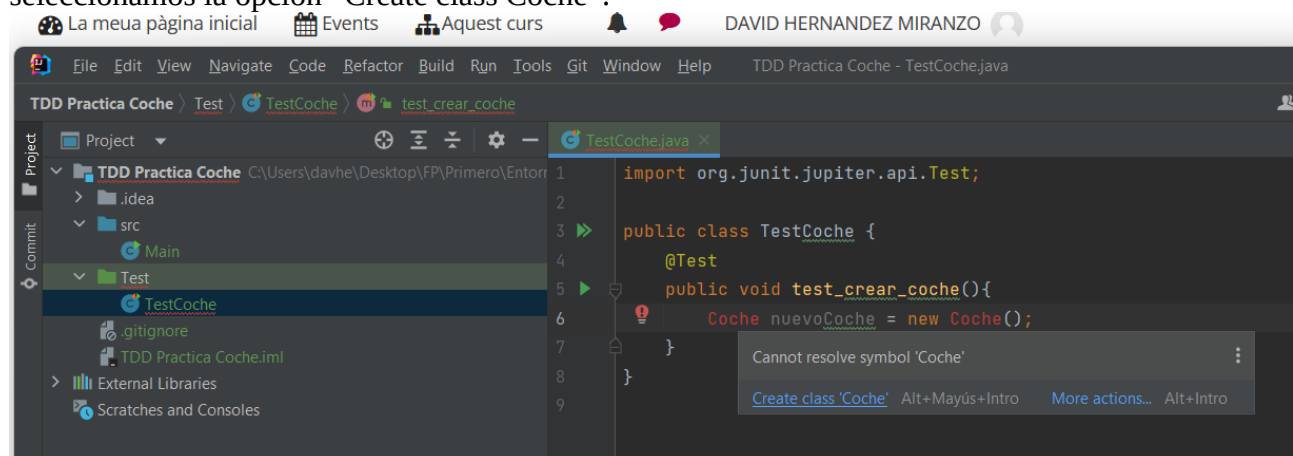
Marcamos la casilla “anotaciones” en la ventana, que descargará el contenido del repositorio de Maven:



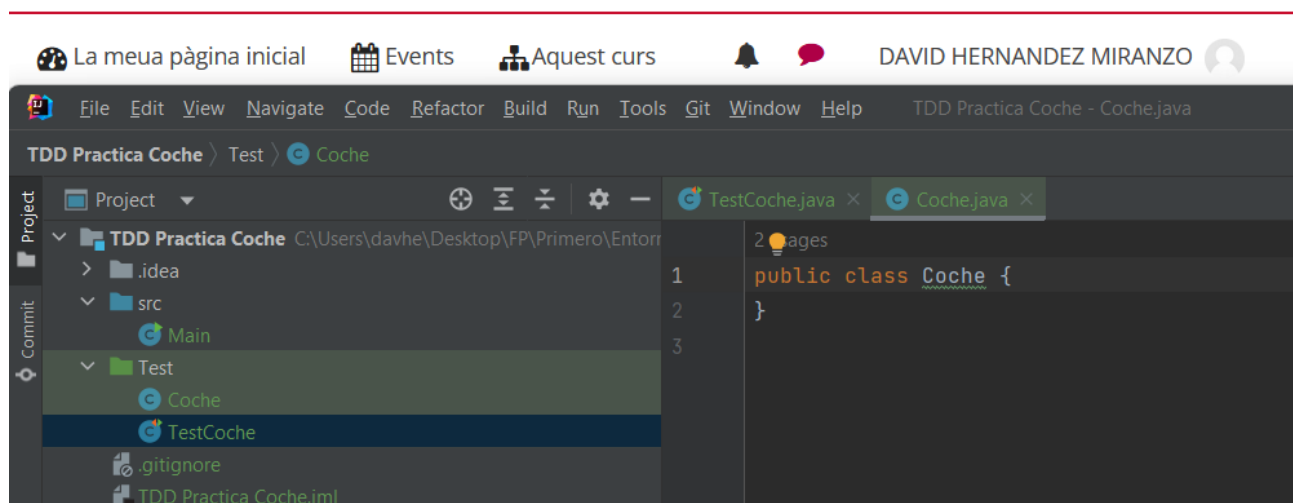
Creación de primer método con una clase inexistente todavía llamada “Coche”:



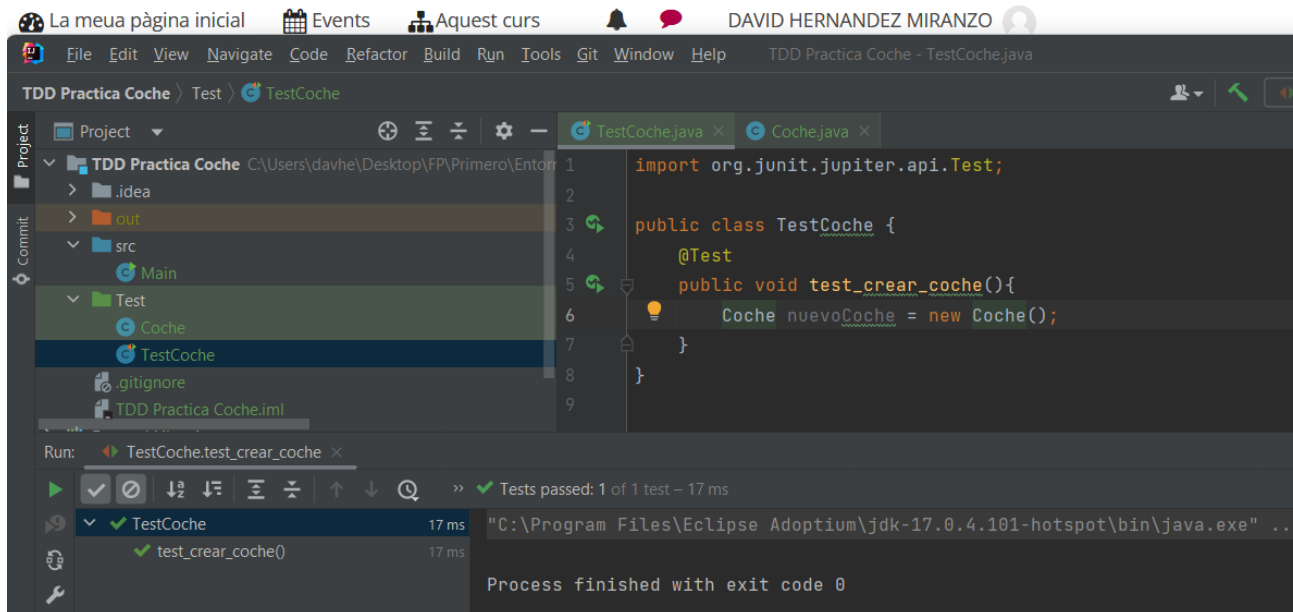
A continuación se crea la clase poniendo el cursor sobre la clase Coche marcada en rojo y seleccionamos la opción “Create class Coche”:



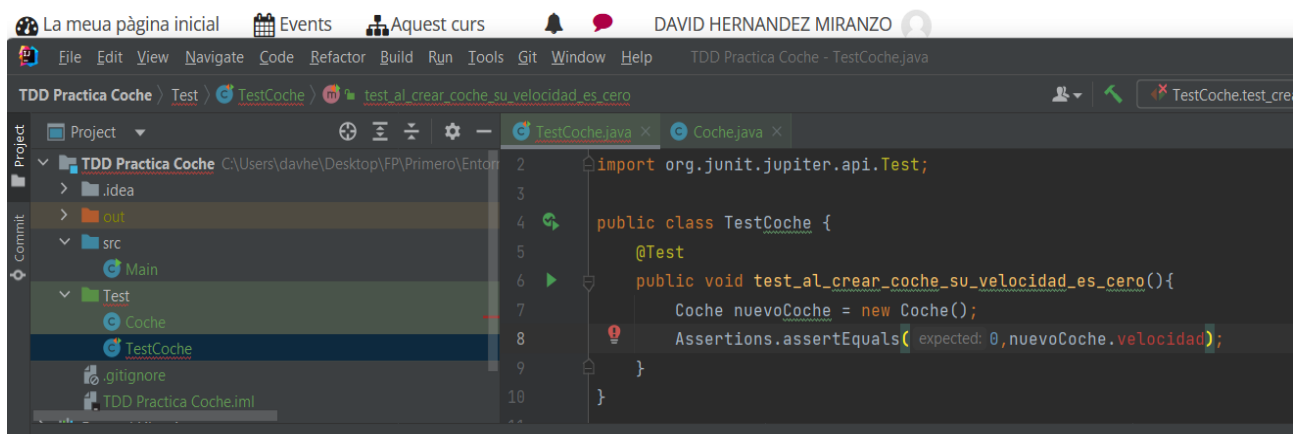
Clase creada:



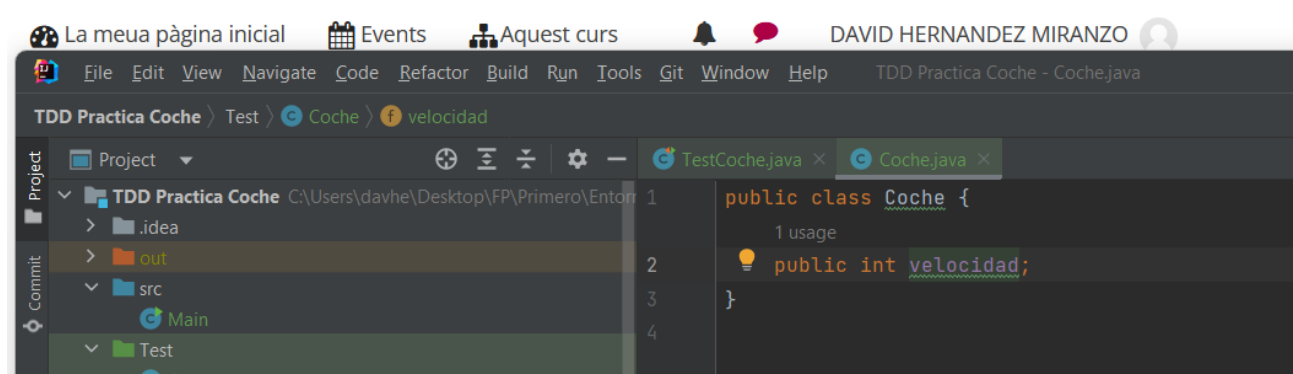
Se hace un run del test y vemos que tiene éxito:



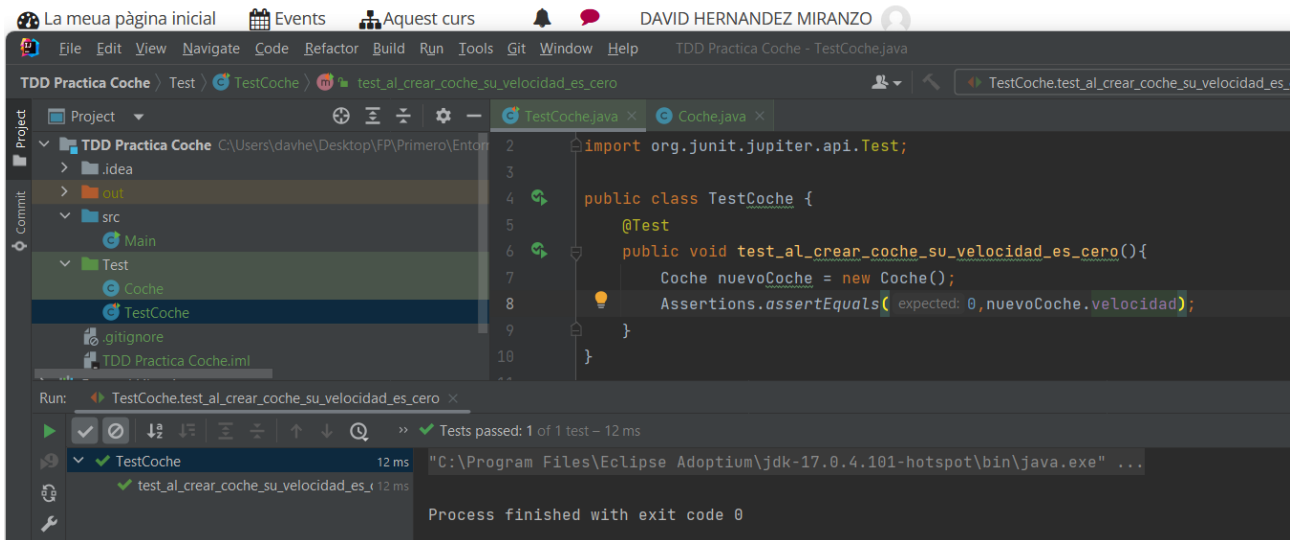
A continuación se modifica el test para comprobar el valor de un atributo. Al poner el atributo y no existir este todavía, el IDE te da la opción de crearlo en la clase que se está probando:



De este modo se crea automáticamente el nuevo campo:



Al ejecutar de nuevo el test con las modificaciones, vuelve a tener éxito:



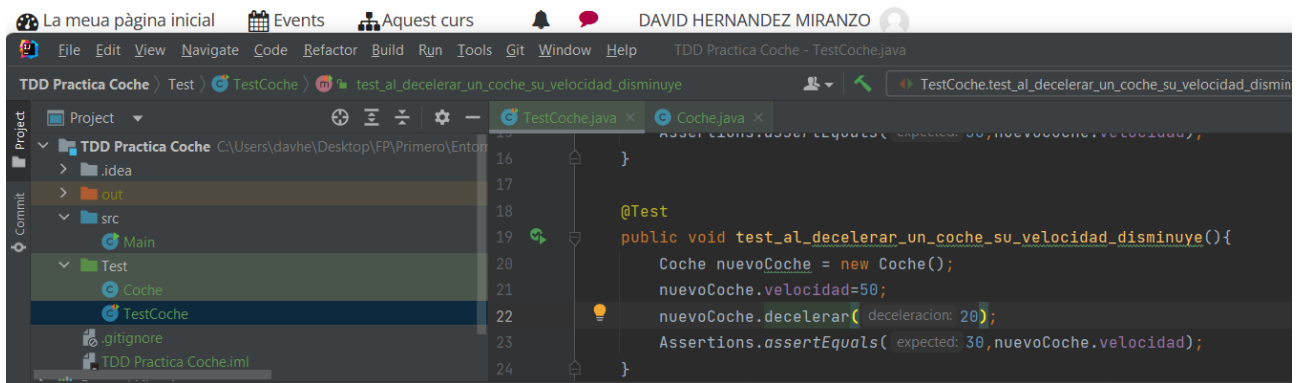
The screenshot shows an IDE with the following components:

- Project Explorer (Left):** Displays the project structure for "TDD Practica Coche". The "Test" directory is expanded, showing "Coche" and "TestCoche".
- Editor (Center):** Shows the code for "TestCoche.java". The code includes:
 

```

      8  Coche nuevoCoche = new Coche();
      9  Assertions.assertEquals( expected: 0, nuevoCoche.velocidad);
      10 }
      11
      12 @Test
      13 public void test_al_acelerar_un_coche_su_velocidad_aumenta(){
      14     Coche nuevoCoche = new Coche();
      15     nuevoCoche.acelerar( 30);
      16     Assertions.assertEquals( expected: 30, nuevoCoche.velocidad);
      17 }
      
```
- Run Console (Bottom):** Shows the execution of the test "TestCoche.test\_al\_acelerar\_un\_coche\_su\_velocidad\_aumenta". The output indicates:
  - Tests passed: 1 of 1 test – 16 ms
  - TestCoche: 16 ms
  - test\_al\_acelerar\_un\_coche\_su\_velocid: 16 ms
  - Process finished with exit code 0

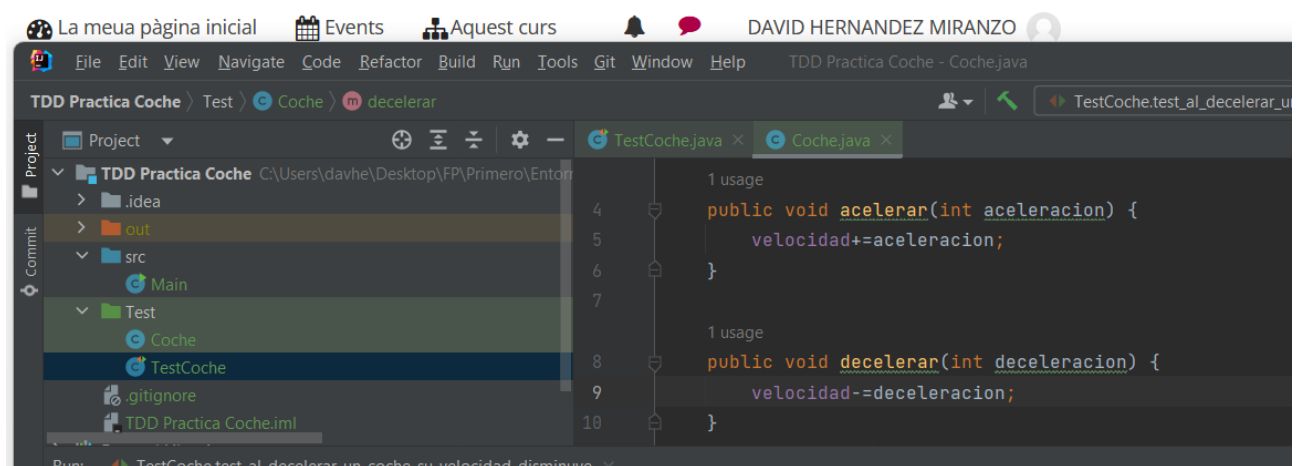
Siguiendo los pasos anteriores se crea el test para la deceleración, y se crea el método decelerar en la case:



The screenshot shows the IntelliJ IDEA interface with the 'TestCoche.java' file open. The project structure on the left shows 'TDD Practica Coche' with a 'Test' directory containing 'Coche' and 'TestCoche'. The main editor displays the following code:

```
16 }
17
18
19 @Test
20 public void test_al_decelerar_un_coche_su_velocidad_disminuye(){
21     Coche nuevoCoche = new Coche();
22     nuevoCoche.velocidad=50;
23     nuevoCoche.decelerar( deceleracion: 20);
24     Assertions.assertEquals( expected: 30,nuevoCoche.velocidad);
25 }
```

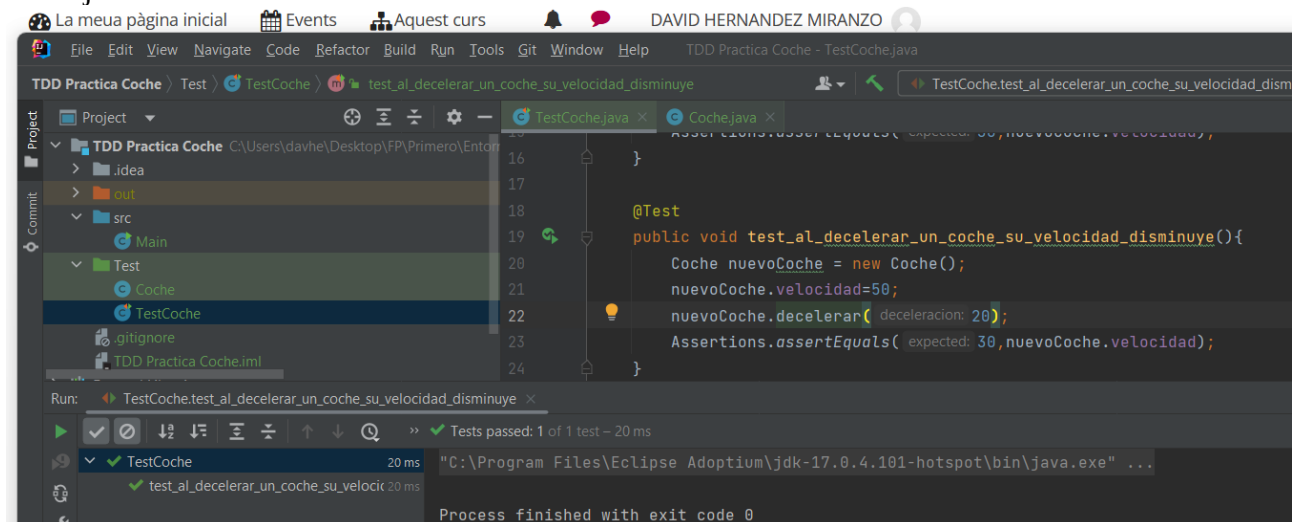
Método en la clase:



The screenshot shows the IntelliJ IDEA interface with the 'Coche.java' file open. The project structure on the left is the same. The main editor displays the following code:

```
4 public void acelerar(int aceleracion) {
5     velocidad+=aceleracion;
6 }
7
8 public void decelerar(int deceleracion) {
9     velocidad-=deceleracion;
10 }
```

Test ejecutado con éxito:

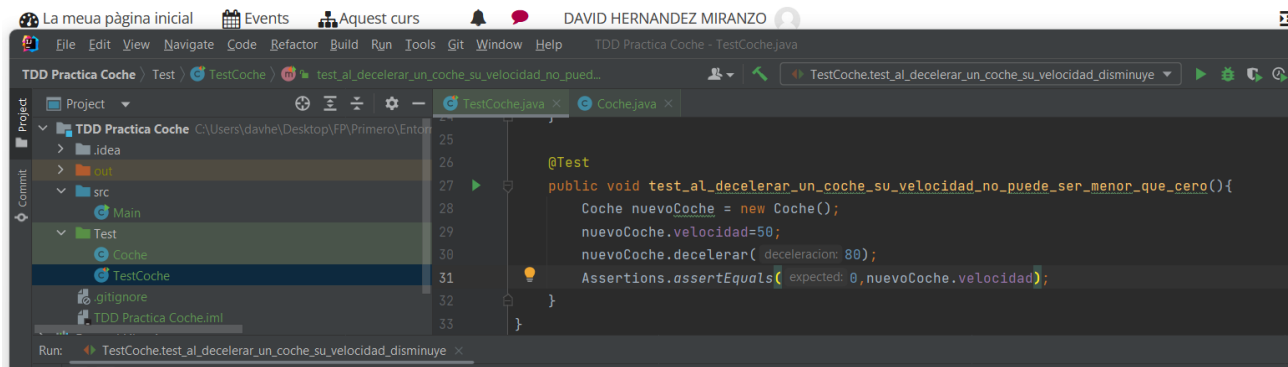


The screenshot shows the IntelliJ IDEA interface with the 'TestCoche.java' file open. The project structure on the left is the same. The main editor displays the same code as the previous screenshot. The 'Run' tab at the bottom shows the test execution results:

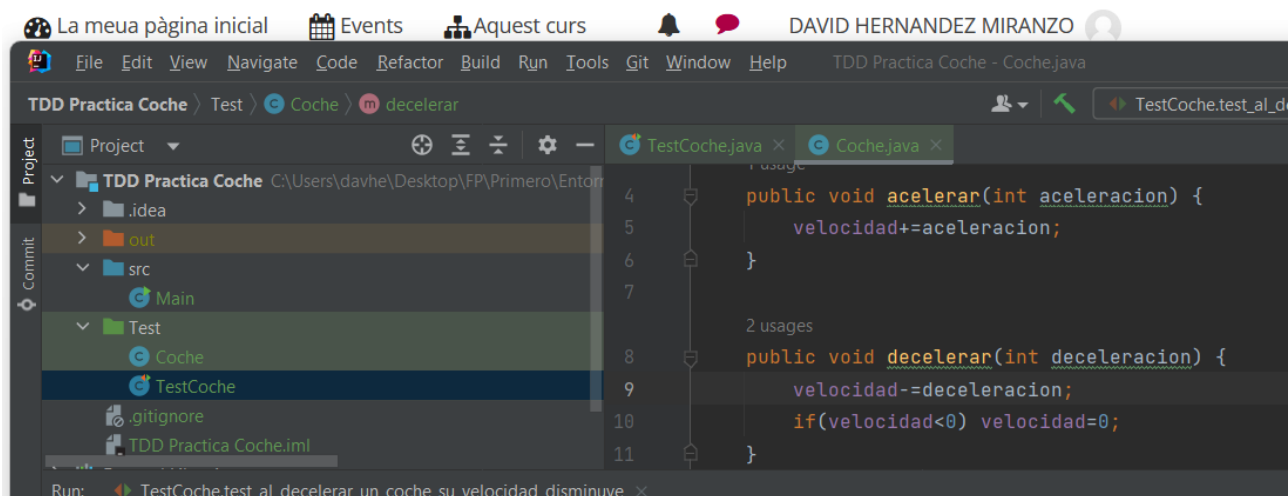
```
Run: TestCoche.test_al_decelerar_un_coche_su_velocidad_disminuye
Tests passed: 1 of 1 test - 20 ms
TestCoche 20 ms
test_al_decelerar_un_coche_su_velocid 20 ms
Process finished with exit code 0
```



Generación de nuevo método para comprobar que cuando frene el coche, no tendrá una velocidad inferior a 0:



En este caso el test falla al ejecutarlo, ya que la lógica no esta preparada para este caso, por lo que hay que modificarla en el cuerpo del método “decelerar” en la clase Coche:



De esta forma el test creado previamente ya se ejecuta correctamente:

