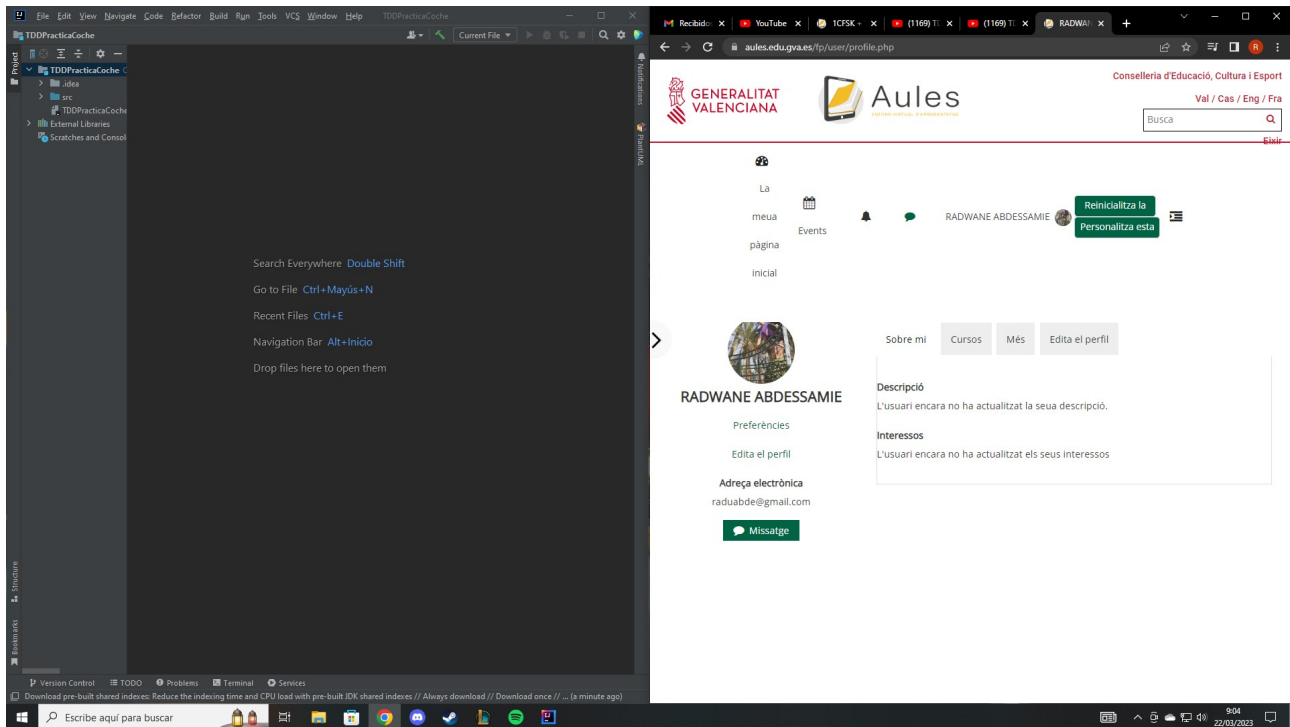
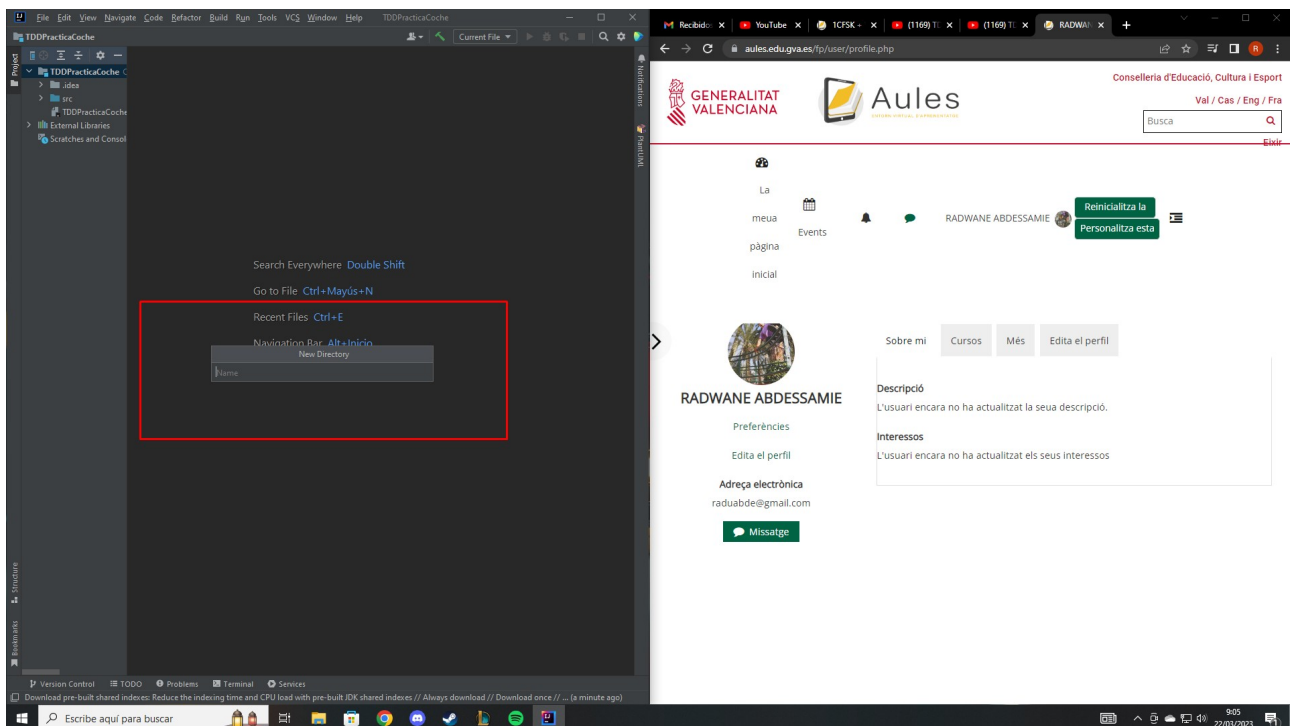


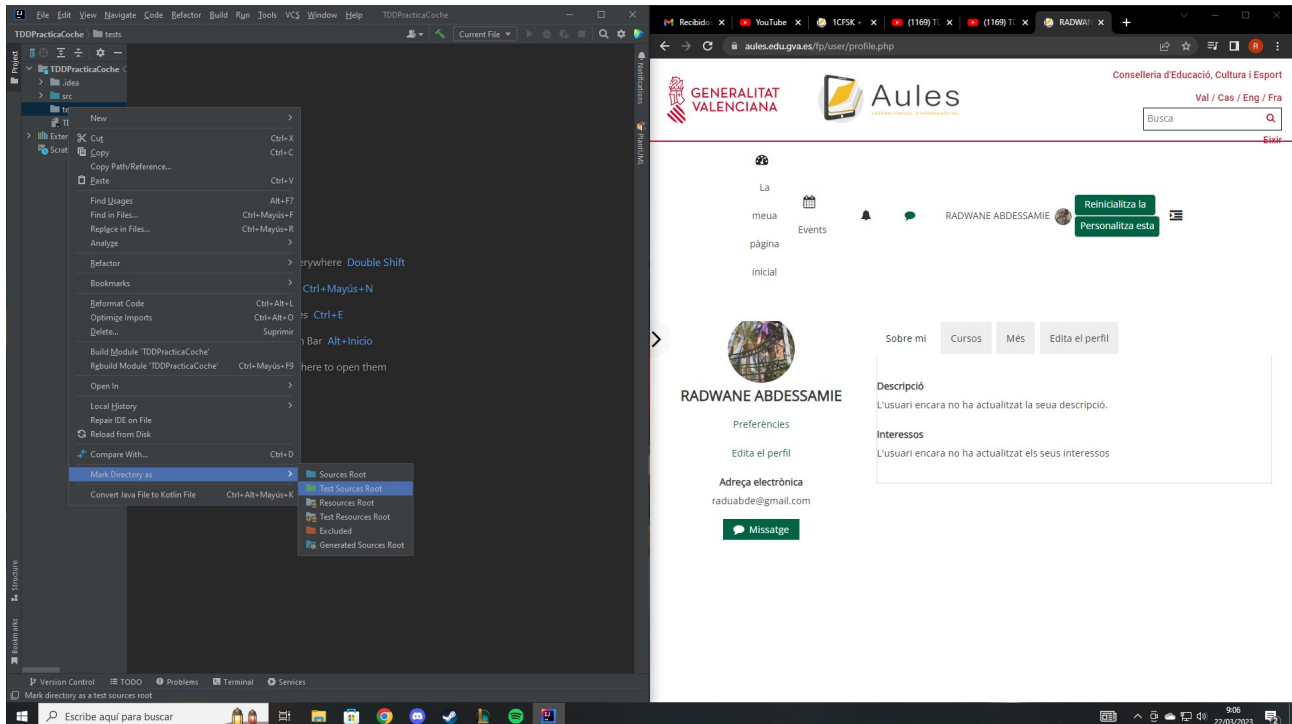
1. Creamos un proyecto con el nombre de TDDPracticaCoche



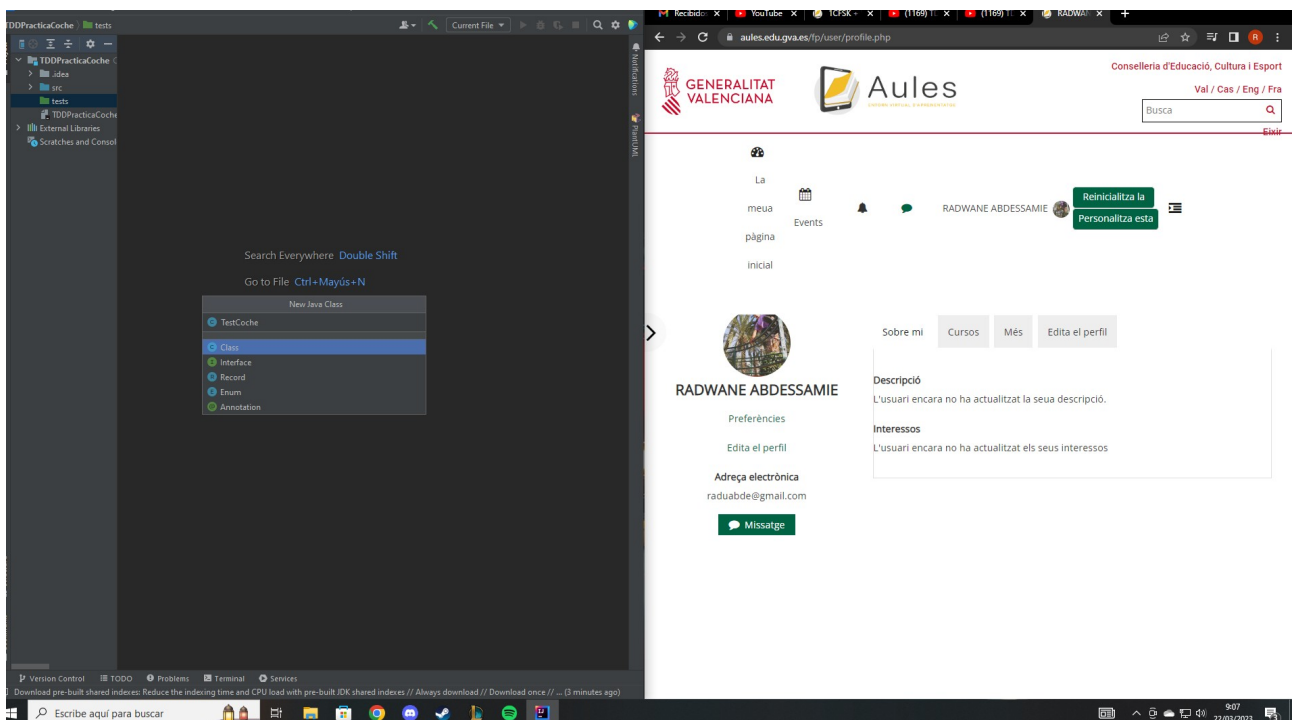
2. Creamos un directorio con el nombre de Tests



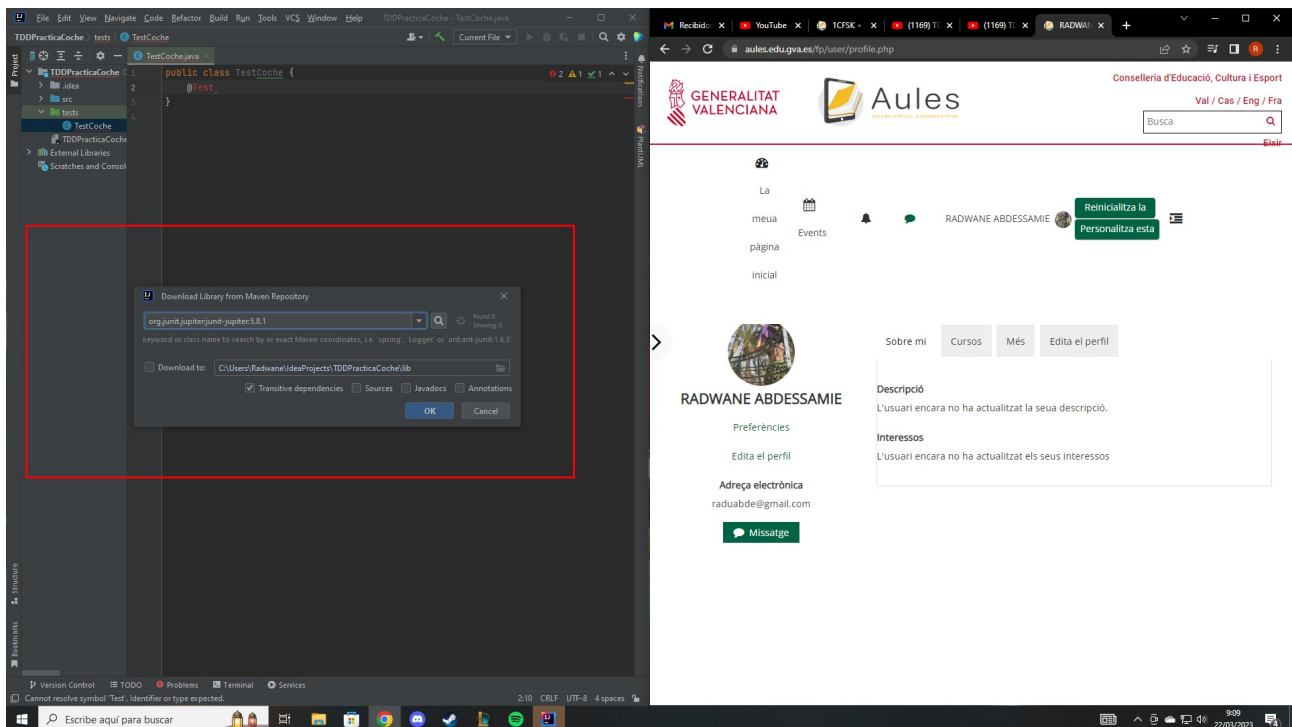
3. Marcamos el directorio Tests como fuente de tests, con click derecho Mark directory as Test Source Root



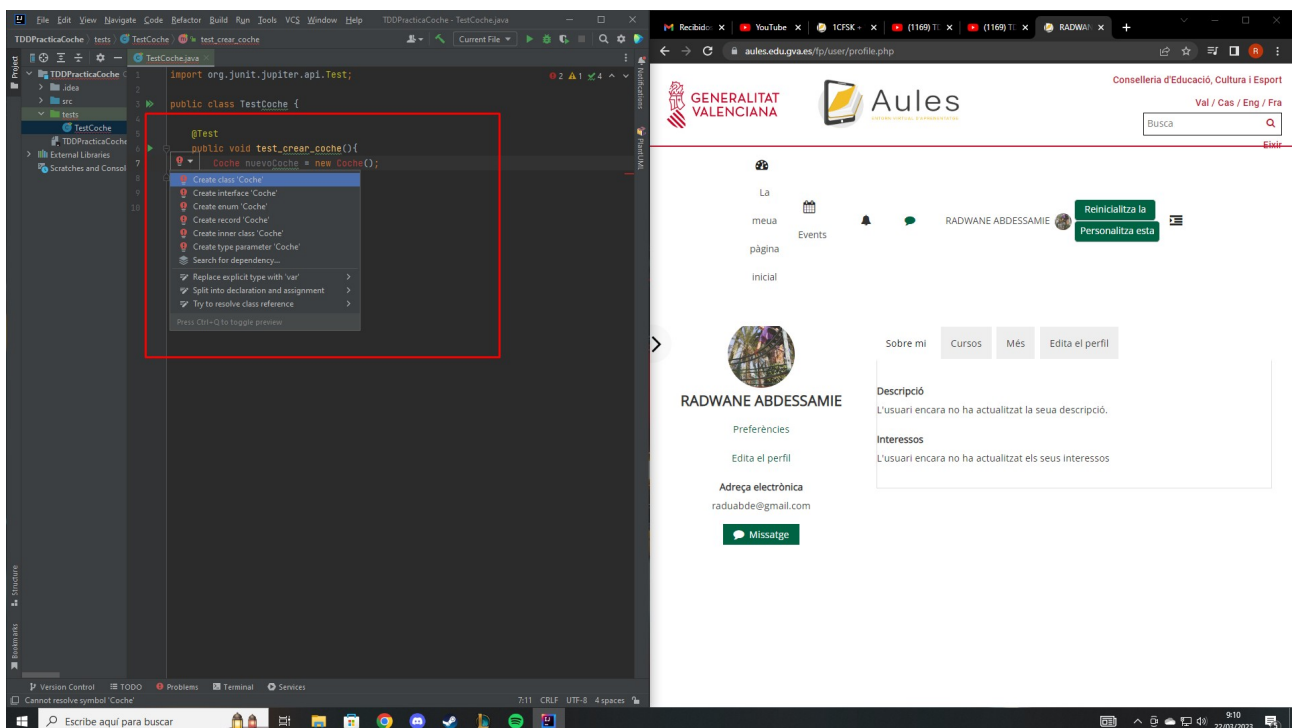
4. Creamos la clase TestCoche dentro de tests



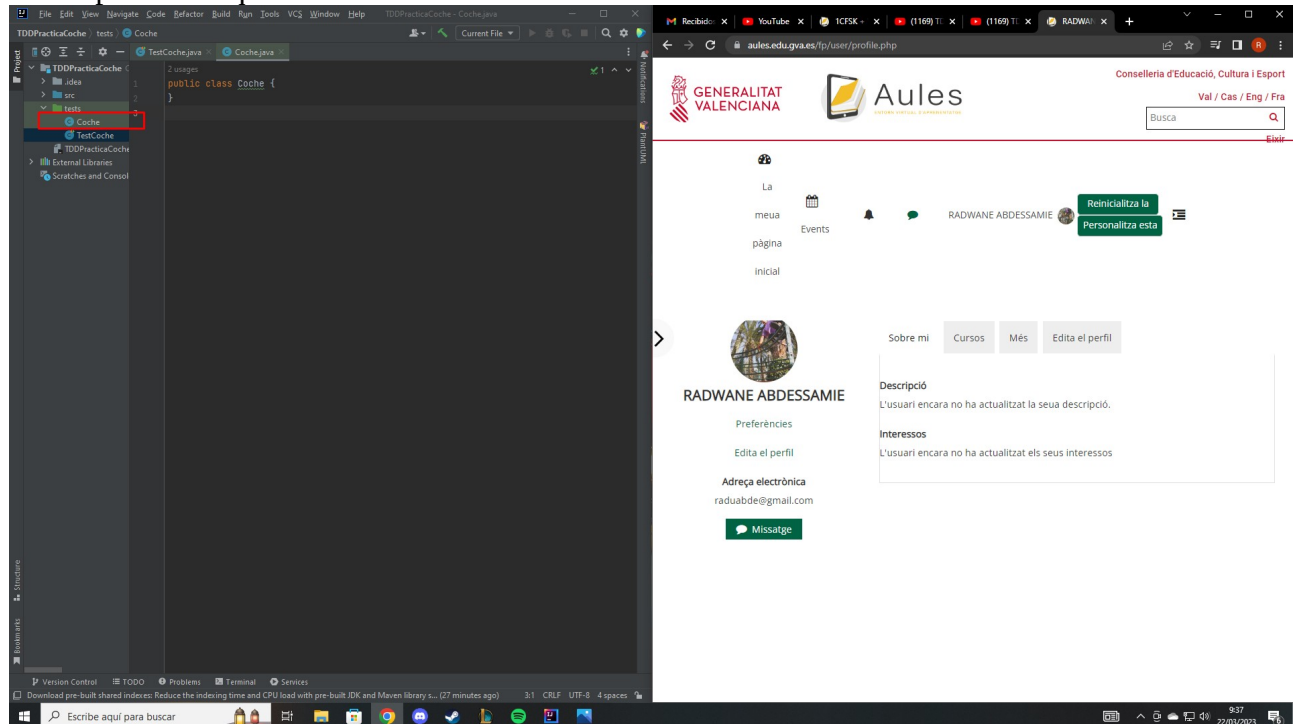
5. Escribimos la notación @Test, i para importar Junit, pulsamos la combinación de teclas alt + intro y elegimos la versión que deseamos.



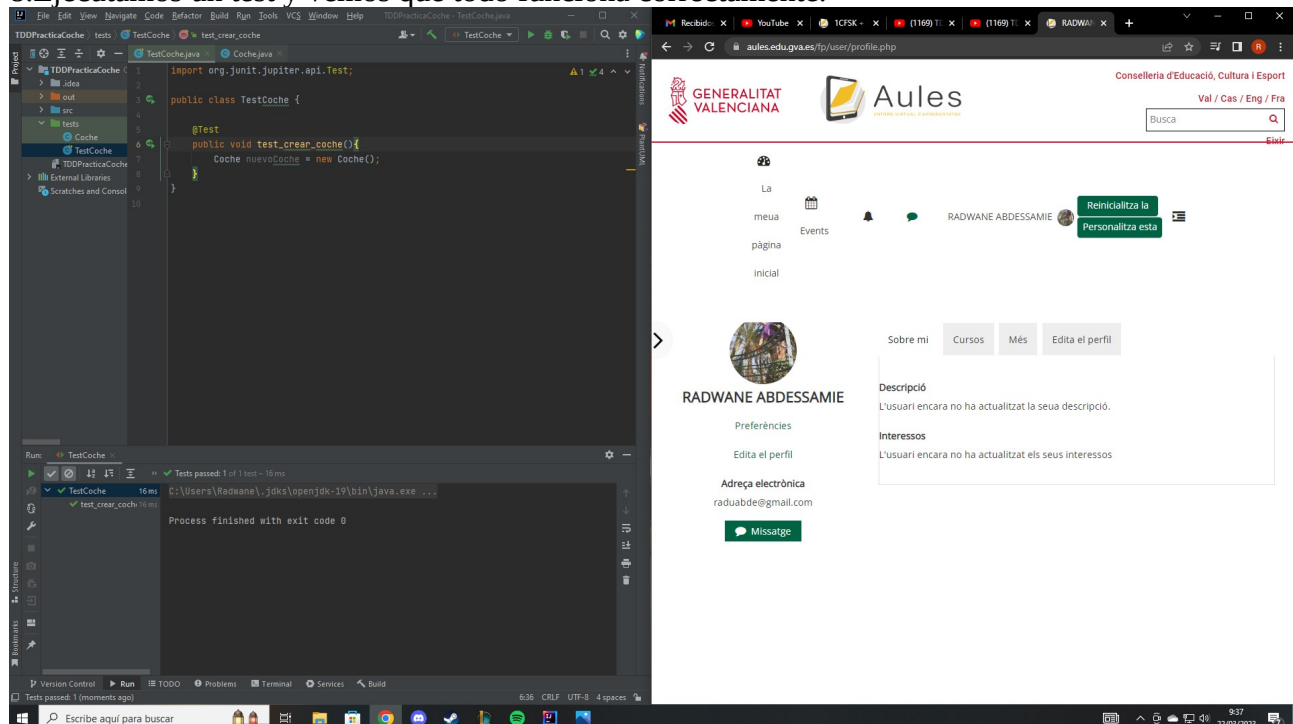
6. Creamos un método de test y creamos un coche, al ver que la clase coche no esta definida, nos lo marca en rojo, asi que para crearla nos situamos sobre la alerta y elegimos la primera opcion ya que coche debe ser un clase.



7. Comprobamos que la clase coche a sido creada.



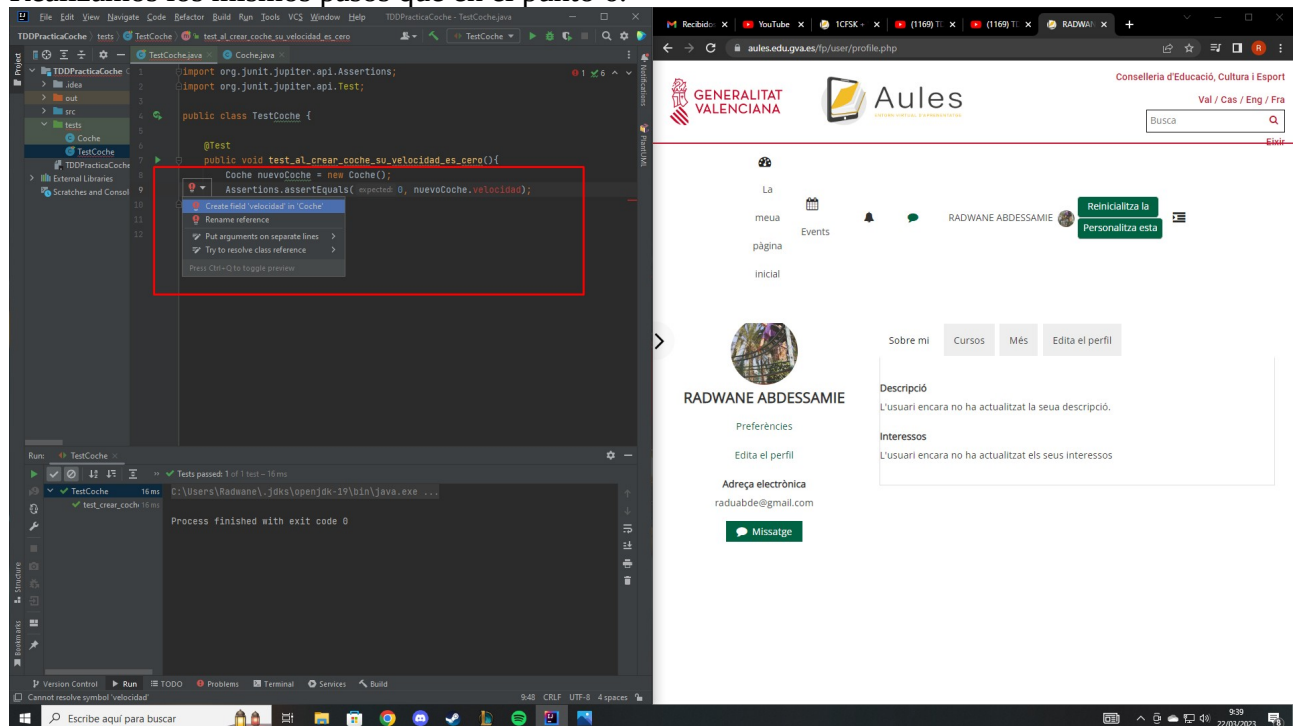
8. Ejecutamos un test y vemos que todo funciona correctamente.



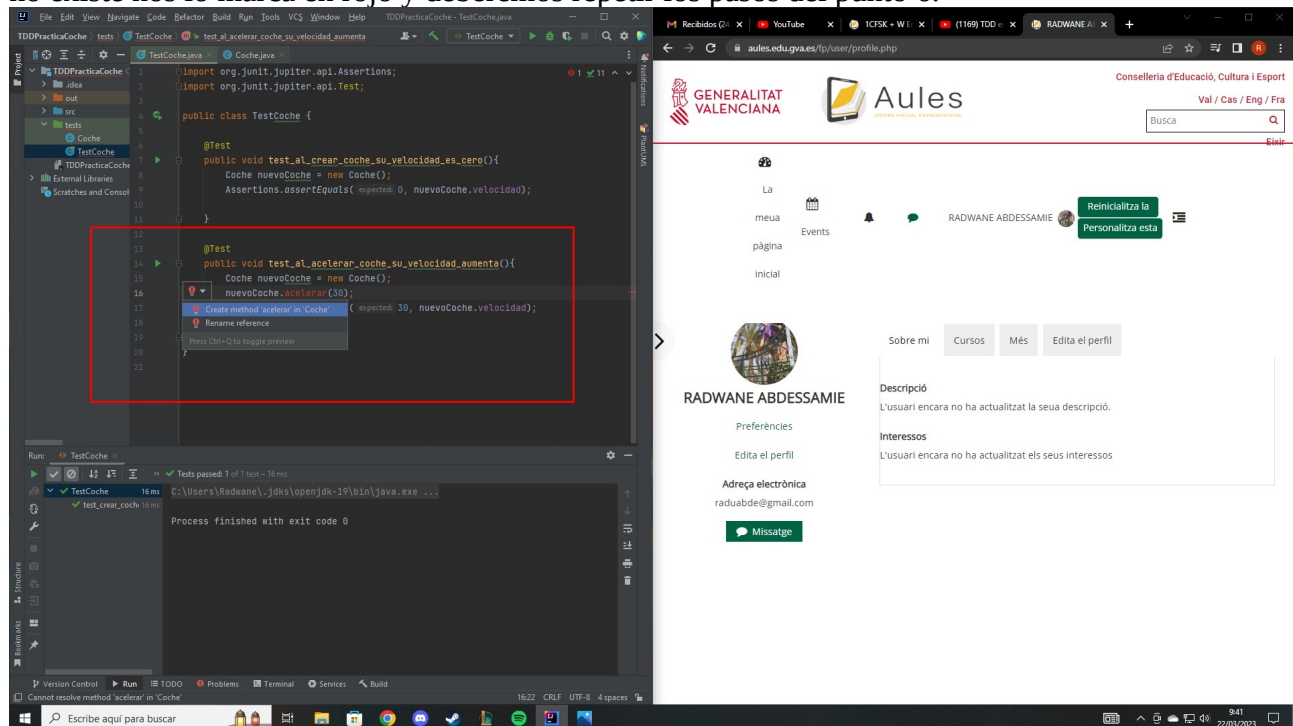
9. Modificamos el test anterior ya que era demasiado simple y añadimos que al crear un coche este debe tener una velocidad de 0.

Nos marca en rojo el campo de velocidad ya que este campo no existe.

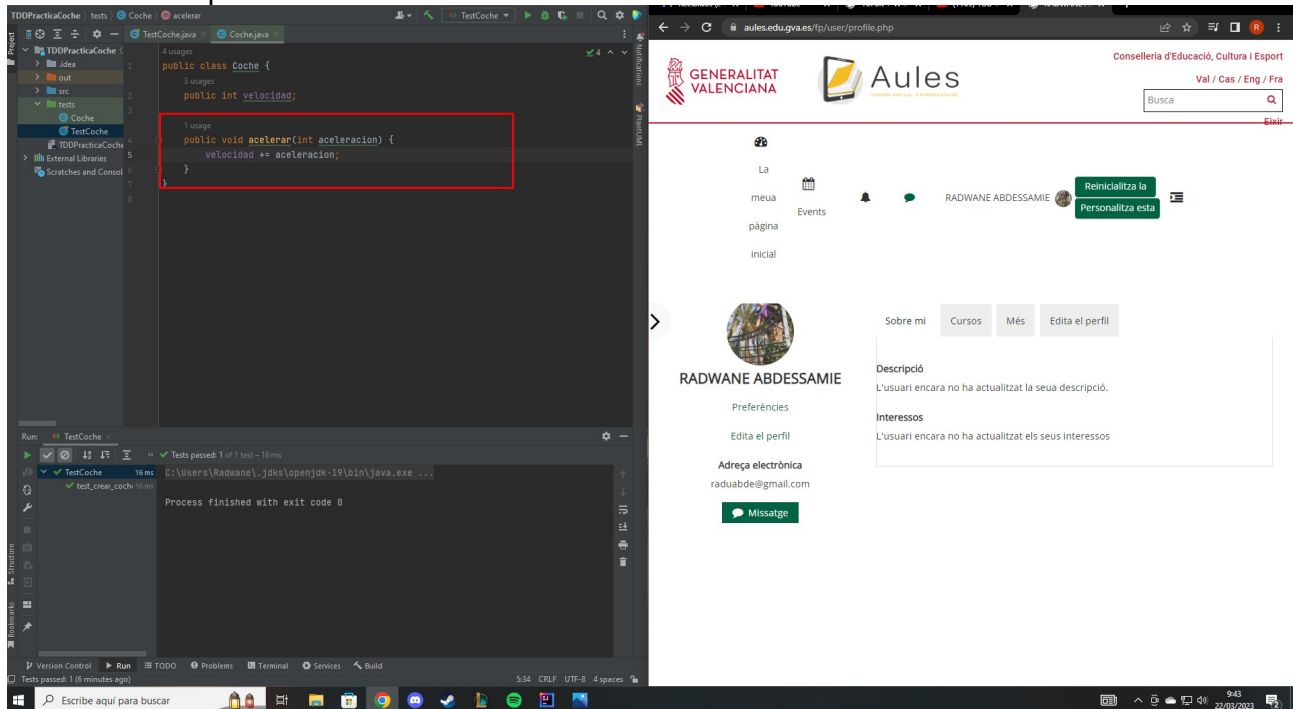
Realizamos los mismos pasos que en el punto 6.



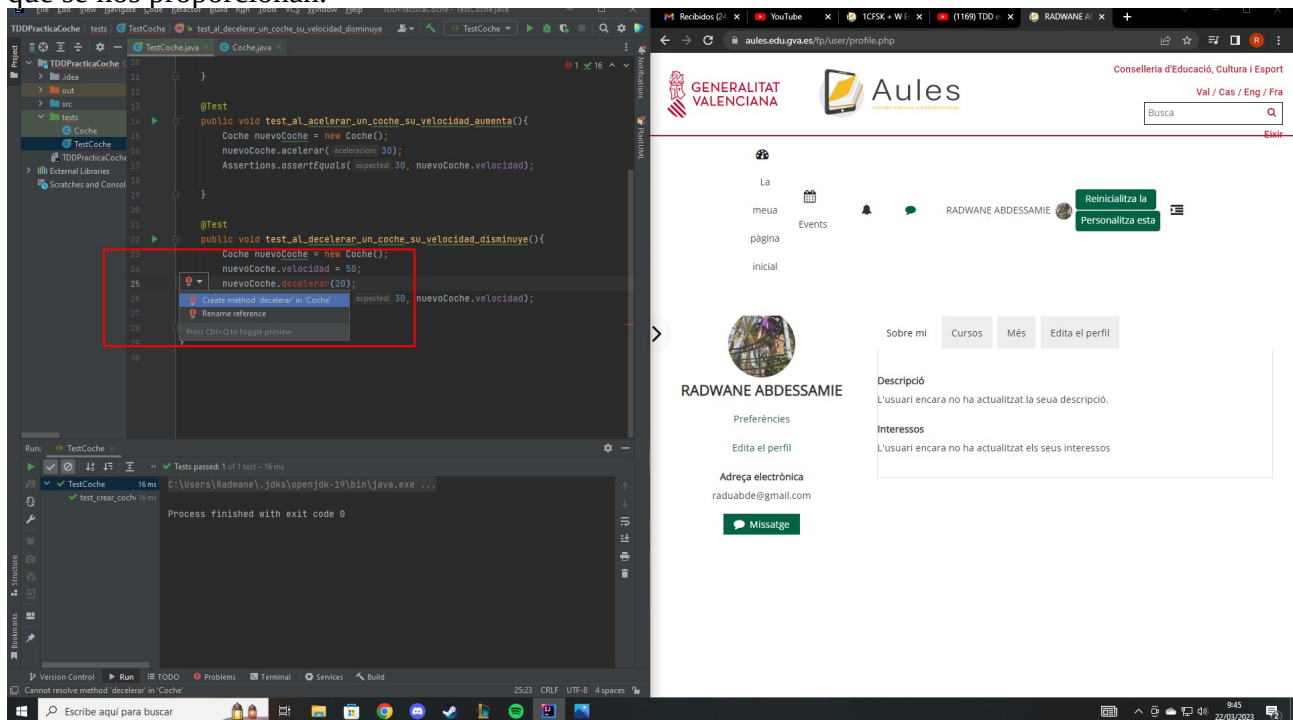
10. Creamos otro test para que al acelerar un coche su velocidad aumente. Como el método acelerar no existe nos lo marca en rojo y deberemos repetir los pasos del punto 6.



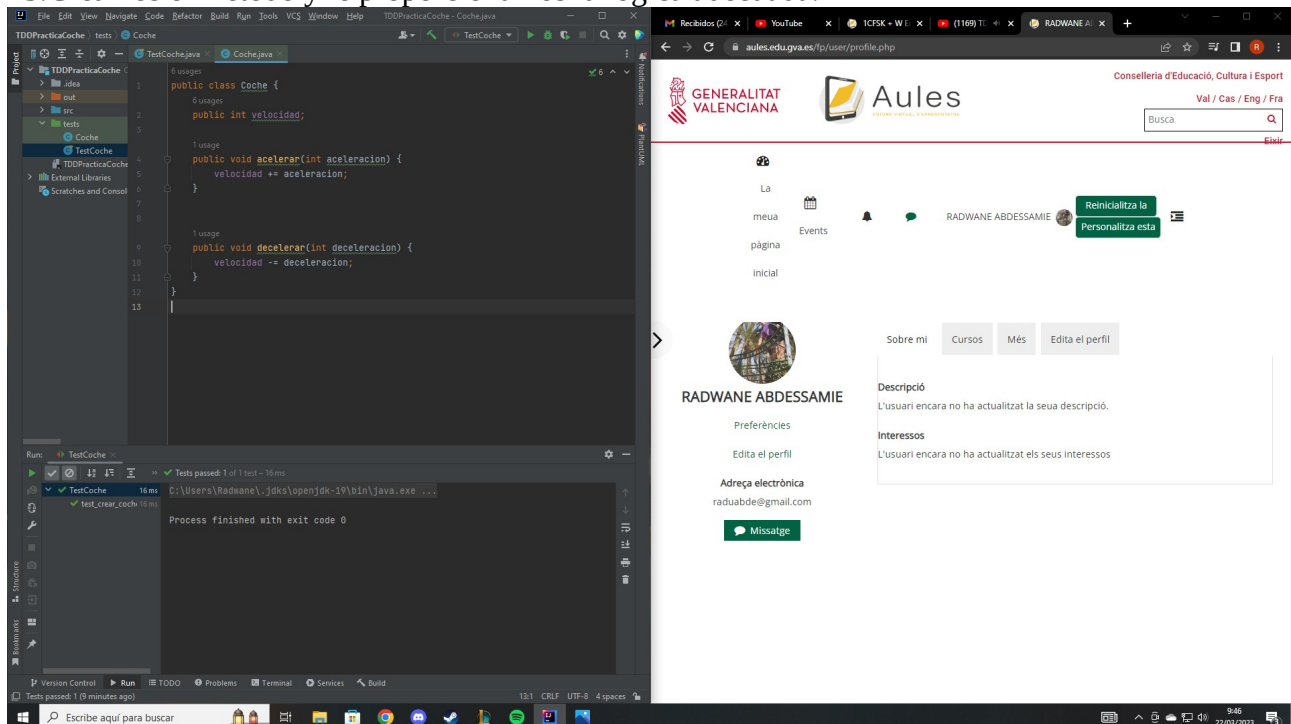
11. Comprobamos que el método acelerar a sido creado correctamente y le damos la lógica que consideremos oportuna.



12. Creamos un método test para desacelerar un coche y ocurre lo que ya venimos viendo , que al no estar creado dicho método, este es marcado en rojo y debemos crearlo seleccionando las opciones que se nos proporcionan.

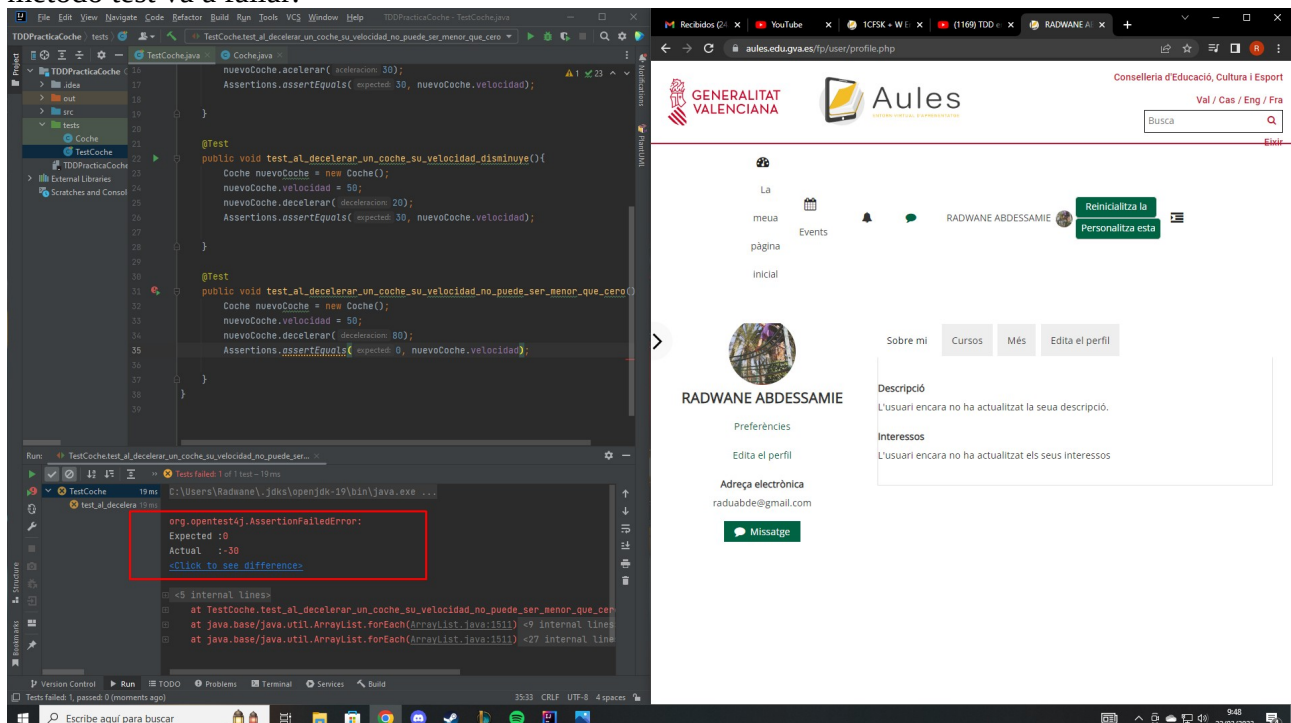


13. Creamos el método y le proporcionamos la lógica adecuada.



14. Cambiamos los valores del test anterior para comprobar que las velocidades al frenar no deben ser negativas.

Claramente como la implementación del método desacelerar no a sido totalmente correcta este método test va a fallar.



15. Solucionamos el error de lógica en el método.

The screenshot shows two windows. On the left, an IDE (IntelliJ IDEA) displays the `Coche` class with the `decelerar` method. The method contains a red box highlighting the logic error: `if (velocidad < 0) velocidad = 0;`. Below the code, the Run window shows a test failure: `org.opentest4j.AssertionFailedError: Expected :0 Actual :-30`. On the right, a web browser shows the Aules profile page for RADWANE ABDESSAMIE, with a search bar and navigation links.

16. Comprobamos que todo funciona correctamente tras hacer dichos cambios.

The screenshot shows two windows. On the left, the IDE displays the `TestCoche` class with four test methods. The Run window shows that all tests passed: `Tests passed: 4 (moments ago)`. On the right, the web browser shows the Aules profile page for RADWANE ABDESSAMIE, with a search bar and navigation links.

REFACTORIZACION

