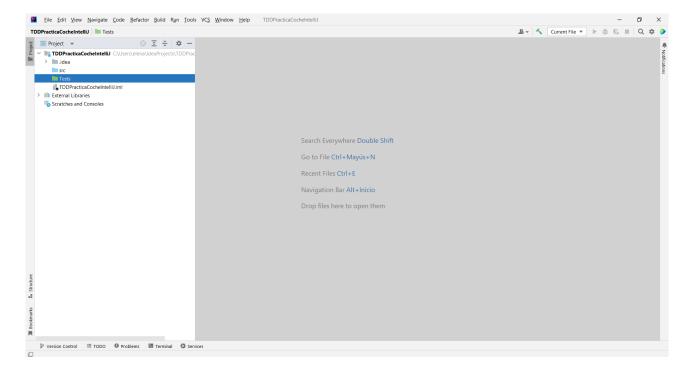
Memoria: Mi Primer TDD (con IntelliJ)

Para empezar a trabajar, abrimos IntelliJ y creamos un nuevo proyecto mediante: File-New-Project, en este caso, lo llamaremos TDDPracticaCocheIntelliJ.

Hacemos click derecho sobre el nombre del proyecto, TDDPracticaCoche, y creamos un nuevo directorio mediante: New-Directory, lo llamaremos Tests.

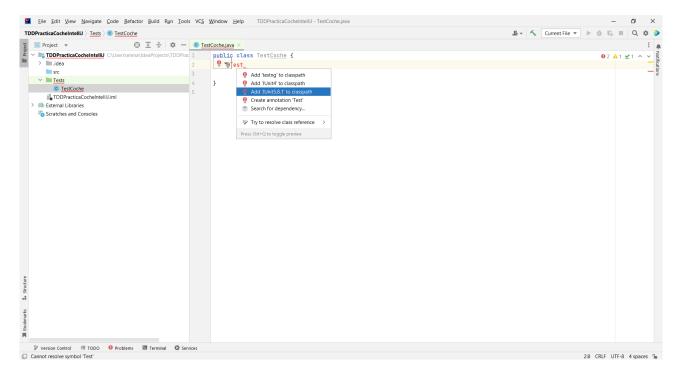
Entonces hacemos click derecho sobre el directorio Tests para marcarlo como directorio de tests mediante: Mark Directory as-Tests Sources Root.

Quedaría así:

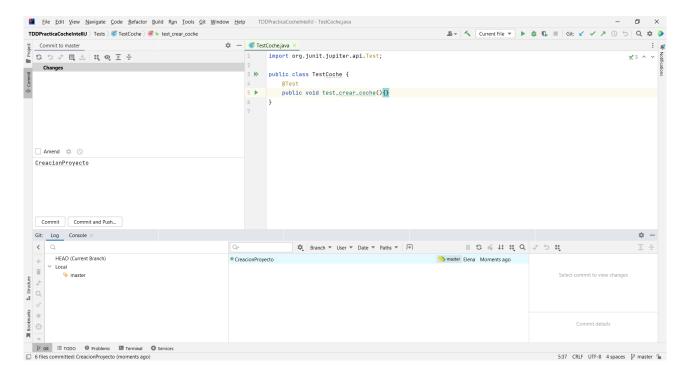


Hacemos click derecho sobre directorio Tests para crear una clase mediante: New-Java Class, la llamaremos TestCoche.

Para crear el test dentro de esta clase: Primero escribiremos @Test, que se nos marcará en rojo, entonces, con el cursor en sobre @Test, pulsamos Alt+Enter y en el menú que se despliega hacemos click sobre Add 'JUnit 5,8,1' to classpath para importar las librerías de Junit5. Sería así:

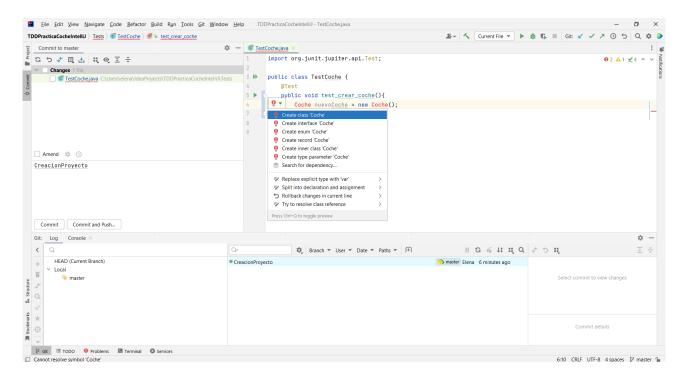


Luego ya podemos declarar el test, al que llamaremos, en este caso test_crear_coche. Quedaría así:

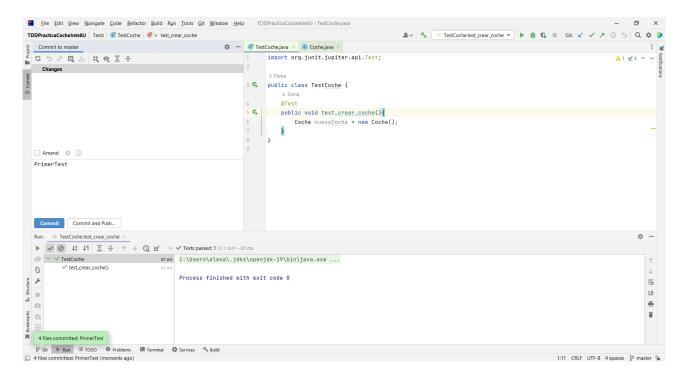


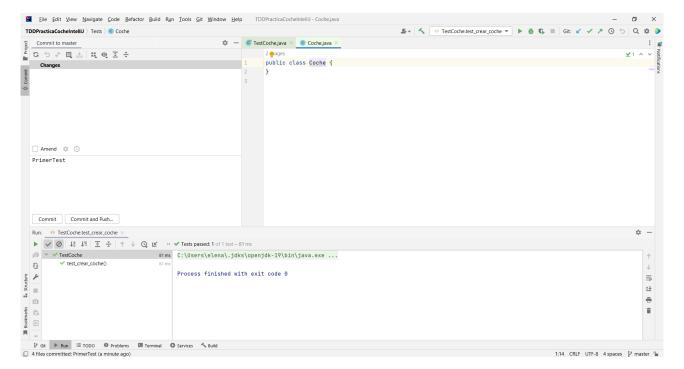
Ahora crearemos el primer test: En la clase TestCoche introducimos una linea de código para crear un objeto Coche de nombre nuevoCoche, dentro de test crear coche.

En este punto, podemos observar que al intentar crear el objeto nuevoCoche, Coche se nos marca en rojo. Nos está avisando de que no existe una clase Coche para poder crear objetos de ella, y que debemos crearla, así que desplegamos la bombilla roja y hacemos click en Create class 'Coche'. De esta forma:



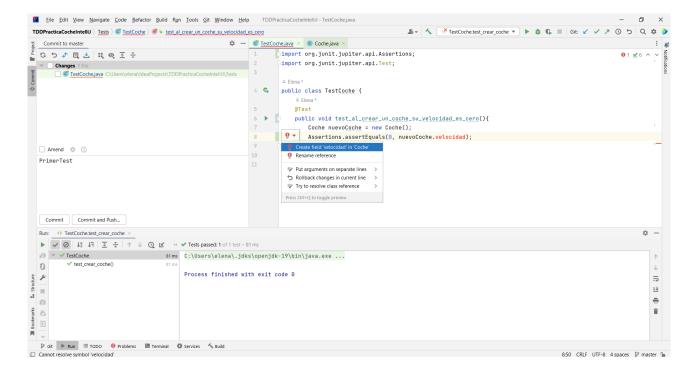
Este es el test creado, así que ahora le damos a Run 'test_crear_coche()', que aparece en la linea de código 5, correspondiente a la clase TestCoche, y el test queda pasado. Quedaría así:



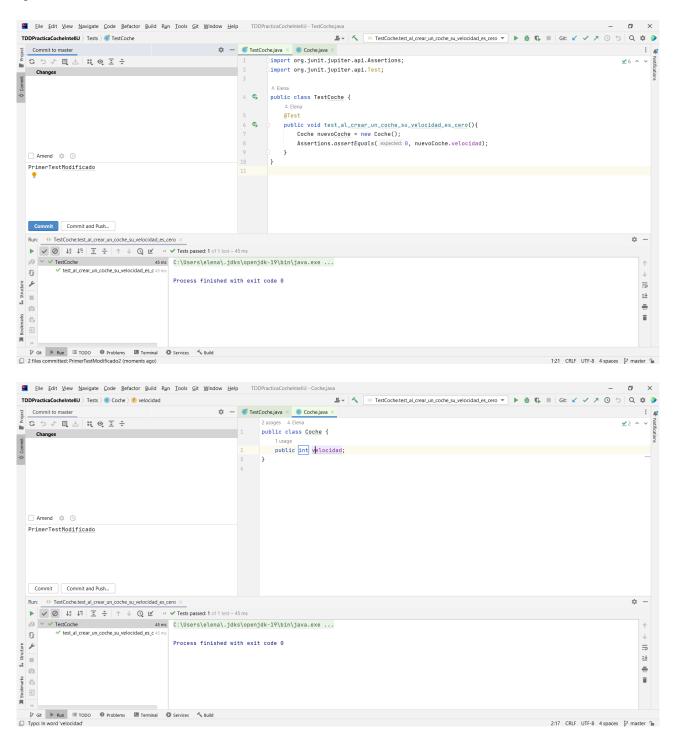


Ahora modificaremos el primer test para hacerlo un poco más complejo: Cambiamos el nombre del test a: test_al_crear_un_coche_su_velocidad_es_cero. En la clase TestCoche introducimos una linea de código para que la velocidad del objeto nuevoCoche sea 0, dentro de test_al_crear_un_coche_su_velocidad_es_cero. Nos pedirá que importemos las librerías Assertions. En esta ocasión nos marca en rojo velocidad. Nos avisa de que no existe la variable velocidad en la clase Coche y que debemos crearla, así que desplegamos la bombilla roja y hacemos click en Create field 'velocidad' in 'Coche'.

De esta forma:



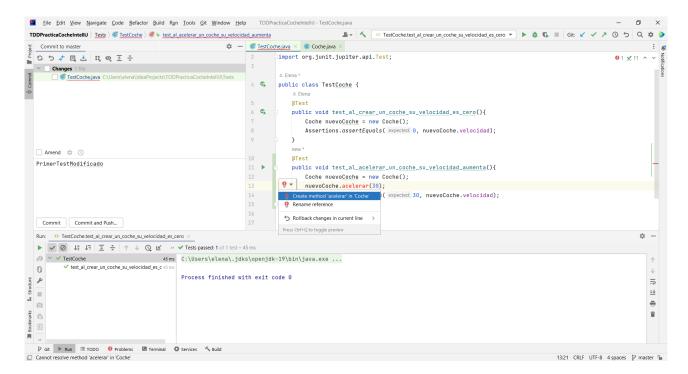
Le damos a Run 'test_al_crear_un_coche_su_velocidad_es_cero()', que aparece en la linea de código 6, correspondiente a la clase TestCoche, y el test queda pasado. Quedaría así:



Ahora crearemos el segundo test: En la clase TestCoche introducimos un nuevo test, al que llamaremos test_al_acelerar_un_coche_su_velocidad_aumenta, y escribiremos una linea de código para llamar al método acelerar.

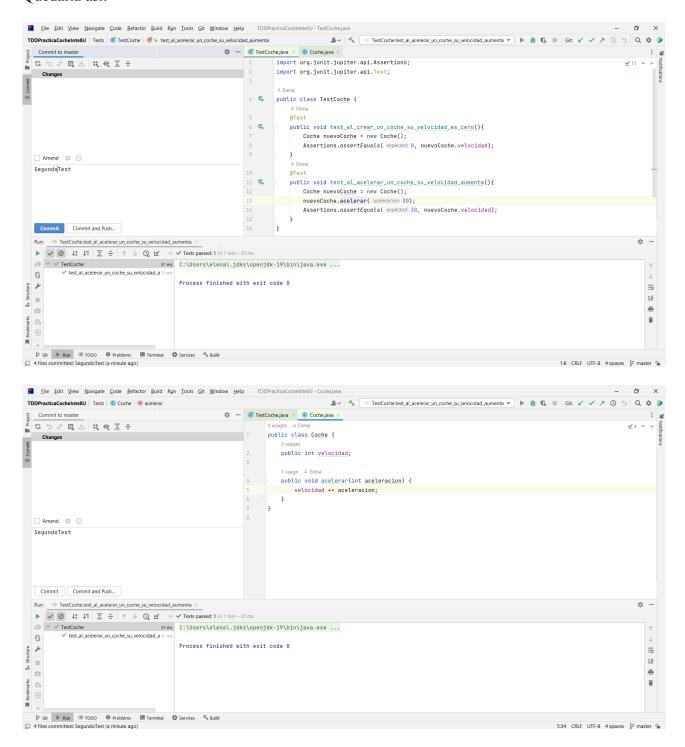
En esta ocasión nos marca en rojo acelerar. Nos avisa de que no existe el método acelerar en la clase Coche y que debemos crearlo, así que desplegamos la bombilla roja y hacemos click en Create method 'acelerar' in 'Coche'.

De esta forma:



Como hemos añadido un método a la clase Coche, escribimos el código con lo que hará, en este caso a la velocidad se le sumará la variable aceleracion.

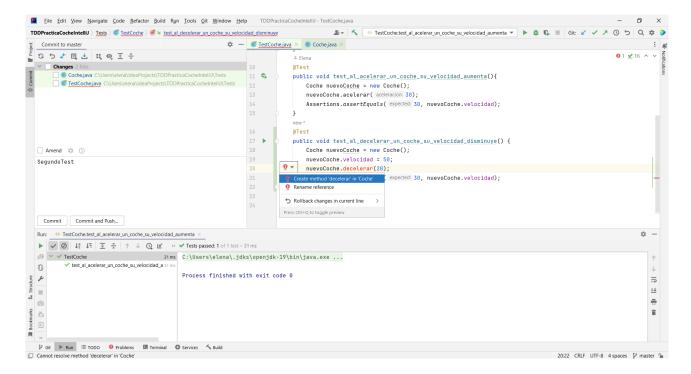
Le damos a Run 'test_al_acelerar_un_coche_su_velocidad_aumenta()', que aparece en la linea de código 11, correspondiente a la clase TestCoche, y el test queda pasado. Quedaría así:



Ahora crearemos el tercer test: En la clase TestCoche introducimos un nuevo test, al que llamaremos test_al_decelerar_un_coche_su_velocidad_disminuye, y escribiremos una linea de código para llamar al método decelerar.

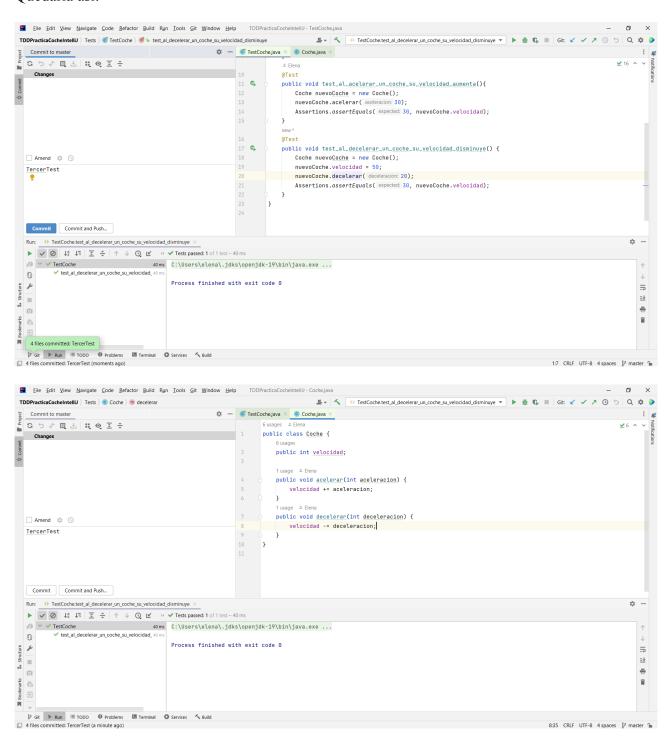
En esta ocasión nos marca en rojo decelerar. Nos avisa de que no existe el método decelerar en la clase Coche y que debemos crearlo, así que desplegamos la bombilla roja y hacemos click en Create method 'decelerar' in 'Coche'.

De esta forma:



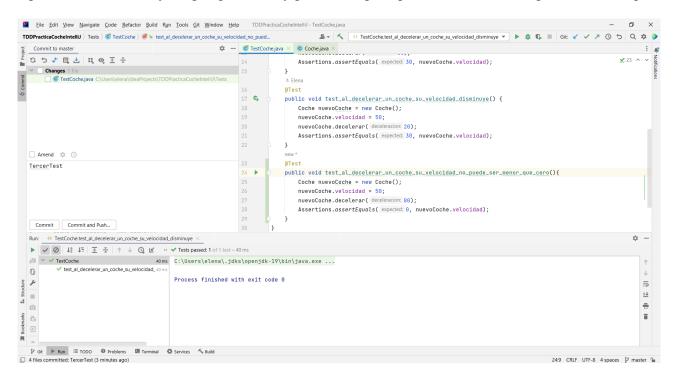
Igual que en el test anterior, hemos añadido un método a la clase Coche, así que escribimos el código con lo que hará, y en este caso a la velocidad se le restará la variable deceleracion.

Le damos a Run 'test_al_decelerar_un_coche_su_velocidad_disminuye()', que aparece en la linea de código 17, correspondiente a la clase TestCoche, y el test queda pasado. Quedaría así:

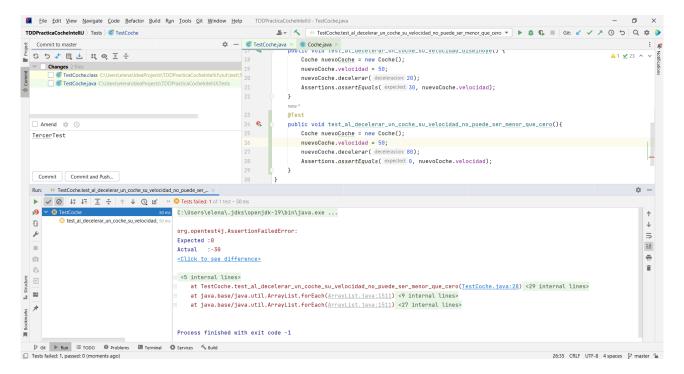


Ahora crearemos el cuarto y último test: En la clase TestCoche introducimos un nuevo test, al que llamaremos test_al_decelerar_un_coche_su_velocidad_no_puede_ser_menor_que_cero, y escribiremos una linea de código para evitar que, al restarle la decelaración a la velocidad, nos quede un resultado en negativo.

Aparentemente, no hay ningún problema y puede compilar perfectamente, como podemos ver aquí:



Pero al ejecutarlo, ocurre lo siguiente:



Nos avisa de que, a pesar de que el resultado esperado que hemos especificado en el test es 0, el resultado real es -30. Y con esta información ya podemos solucionar el problema, en este caso escribiendo una linea de código en el método decelerar para que, cuando la velocidad pretenda ser

menor que 0, la iguale a 0.

Ahora si, le damos a Run

'test_al_decelerar_un_coche_su_velocidad_no_puede_ser_menor_que_cero()', que aparece en la linea de código 24, correspondiente a la clase TestCoche, y el test queda pasado. Quedaría así:

