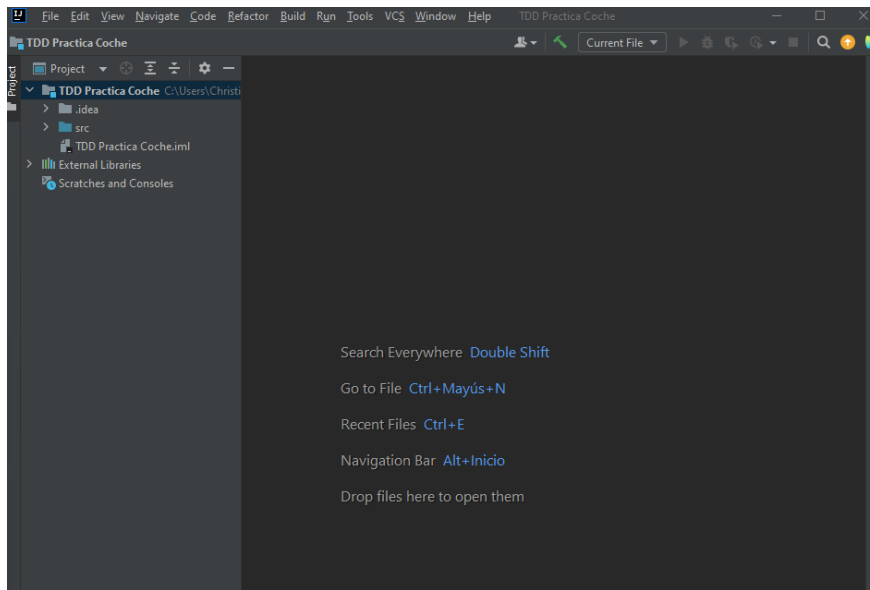


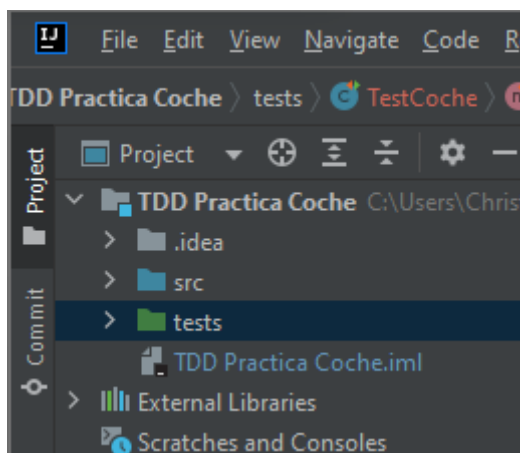
MEMORIA MI PRIMER TDD

Laura María Pinedo Puertas
Entornos de Desarrollo, 1ºQ

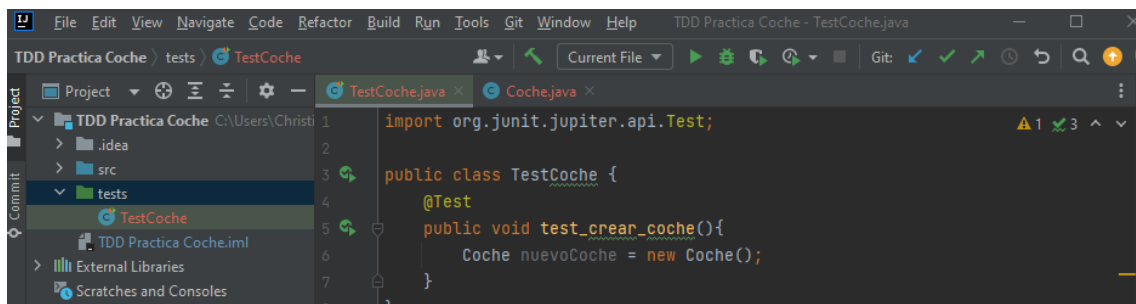
Primero creamos un nuevo proyecto Java en IntelliJ llamado “TDD Practica Coche”.



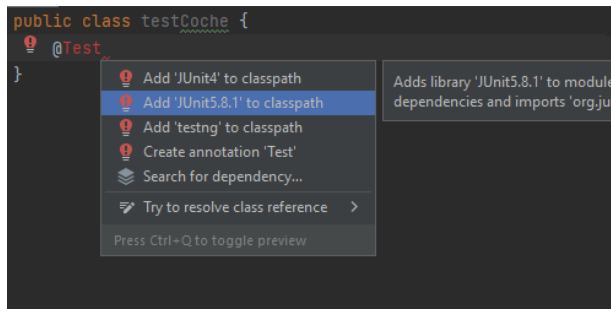
Creamos un nuevo directorio para realizar nuestros tests y lo marcaremos como tal.



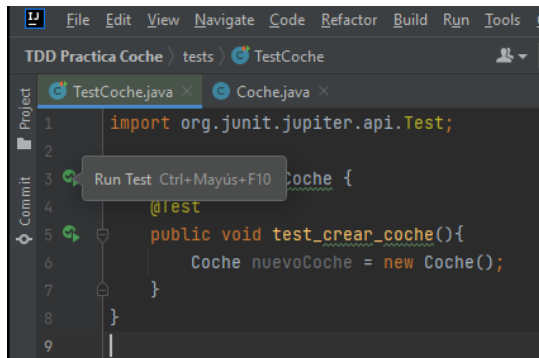
Crearemos una clase de Java dentro del directorio “tests” que se llamará TestCoche, y para evitar que nos dé error, ya que no hemos creado todavía ninguna clase, crearemos la clase “Coche”.



También cabe destacar que debemos indicar que utilice Junit 5 tal y como se especifica en la práctica.



Ahora hacemos click en “Run test” para ejecutarlo.



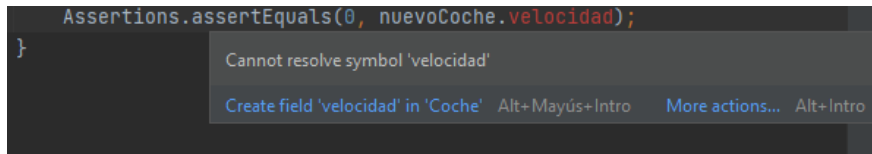
Y vemos que hemos pasado nuestro primer test.



Mejoramos nuestro test, y para ello utilizamos una Assertion.

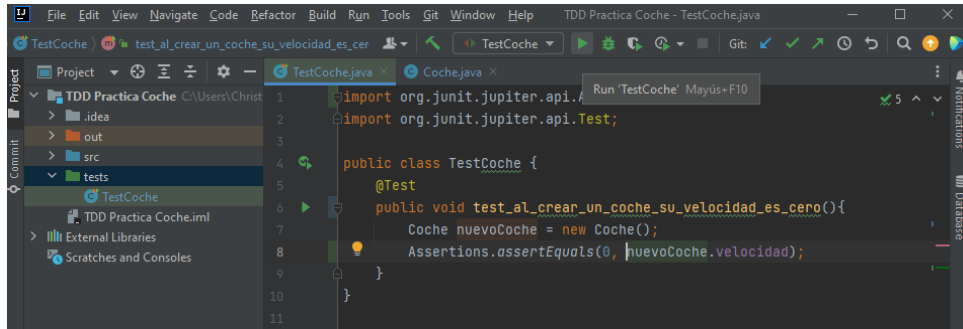


Como vemos, “velocidad” nos aparece en rojo, al igual que nos pasaba antes con la clase “Coche”, debido a que no existe todavía, así que hacemos click en “Create field ‘velocidad’ in ‘Coche’” y la creamos.

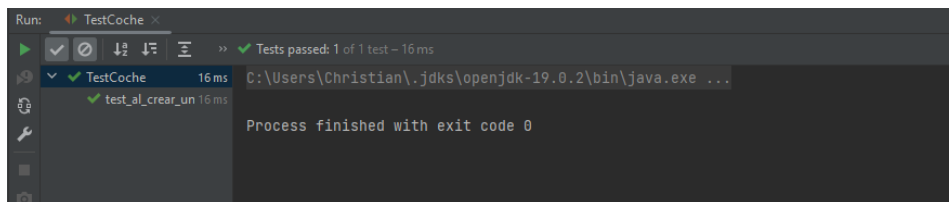


```
1 public class Coche {  
2     public int velocidad;  
3 }
```

Ahora hacemos click en “Run” para ejecutar nuestro test.



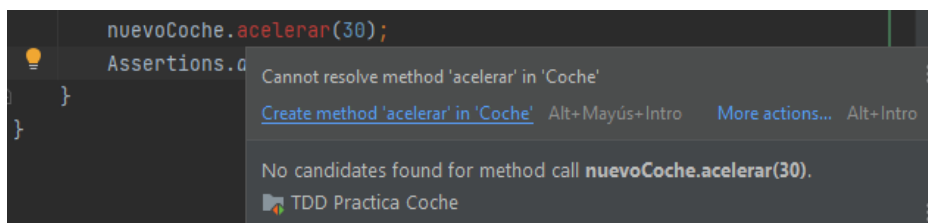
Y, de nuevo, vemos que lo pasa sin problemas.

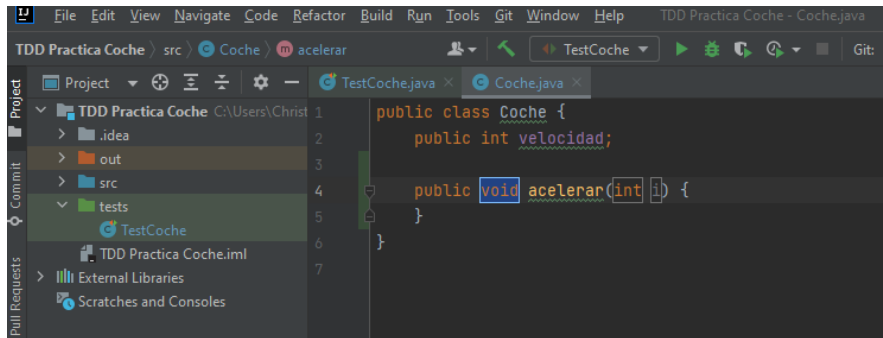


Creamos un nuevo test para “acelerar”.

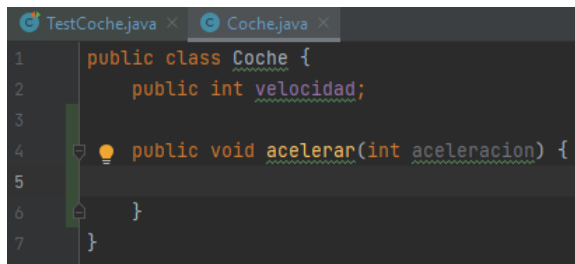
```
@Test  
public void test_al_acelerar_un_coche_su_velocidad_aumenta(){  
    Coche nuevoCoche = new Coche();  
    nuevoCoche.acelerar(30);  
    Assertions.assertEquals(30, nuevoCoche.velocidad);  
}
```

Tal y como pasaba antes, volvemos a tener el problema en “acelerar”, ya que todavía no la hemos creado, por lo tanto, lo creamos.

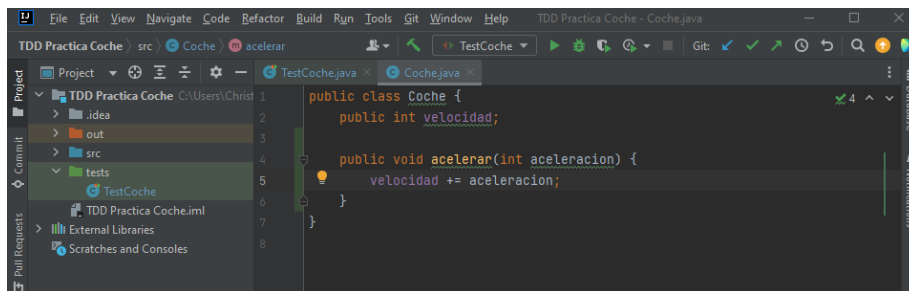




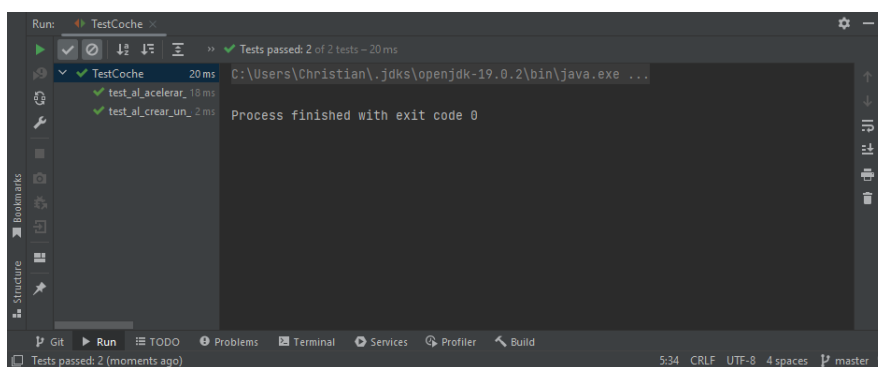
Cambiamos el nombre del argumento de “i” a “aceleracion”.



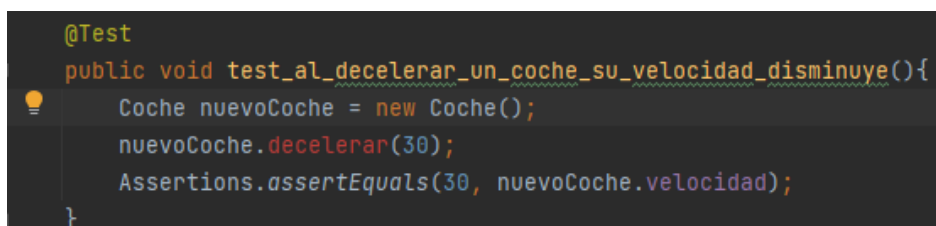
Le indicamos que su “velocidad” aumente en “aceleracion”.



Hecho esto, hacemos click en “Run” para ejecutar el test y vemos que el resultado no muestra ningún error



Creamos un nuevo test para “decelerar”.



Así sin mas no funcionaría, debido que al crear el coche sin más la velocidad estaría a 0, por lo que, antes de nada, debemos asignarle una velocidad mayor, por ejemplo 50.

```
@Test
public void test_al_decelerar_un_coche_su_velocidad_disminuye(){
    Coche nuevoCoche = new Coche();
    nuevoCoche.velocidad = 50;
    nuevoCoche.decelerar(30);
    Assertions.assertEquals(30, nuevoCoche.velocidad);
}
```

Hecho esto, indicaremos que queremos que decelere 20 y expected lo dejaremos en 30.

```
nuevoCoche.velocidad = 50;
nuevoCoche.decelerar(20);
Assertions.assertEquals(30, nuevoCoche.velocidad);
}
```

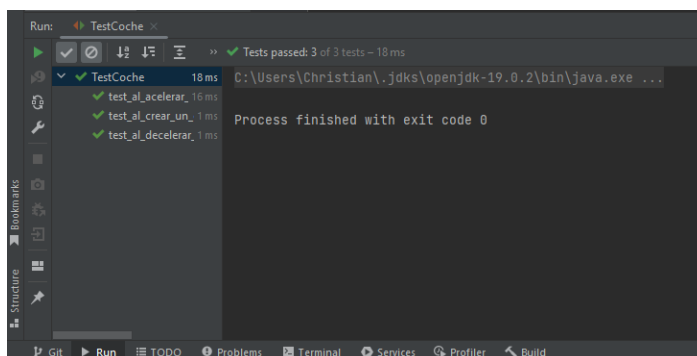
Como vemos, otra vez tenemos el problema de que no encuentra el método “decelerar”, así que lo creamos.

```
1 public class Coche {
2     public int velocidad;
3
4     public void acelerar(int aceleracion) {
5         velocidad += aceleracion;
6     }
7
8     public void decelerar(int i) {
9     }
10 }
11
```

Y cambiamos el nombre del parámetro de “i” a “deceleración”, además le indicamos que la velocidad disminuya en la deceleración.

```
8     public void decelerar(int deceleracion) {
9         velocidad -= deceleracion;
10    }
11 }
```

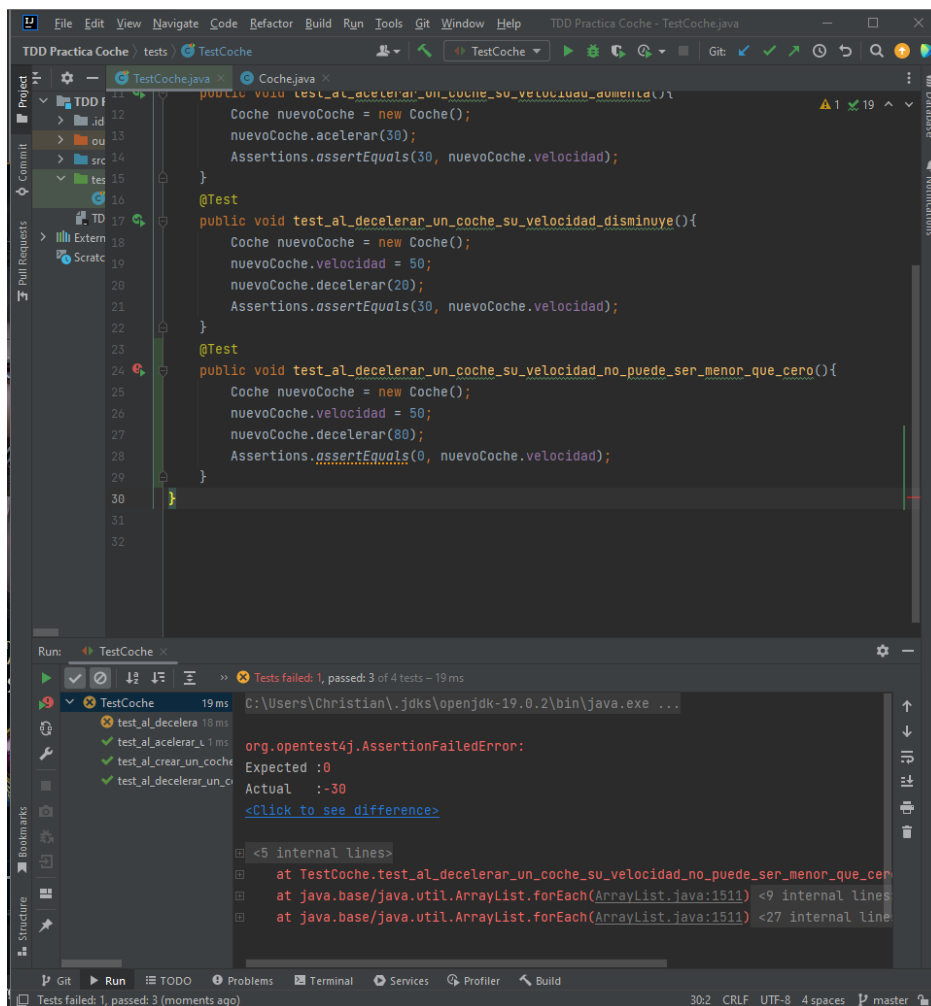
Ahora hacemos click en “Run” y vemos que pasa los tests sin problemas.



Creamos un nuevo test para que al decelerar su velocidad no pueda ser menor de 0.

```
23      @Test
24      public void test_al_decelerar_un_coche_su_velocidad_no_puede_ser_menor_que_cero(){
25          Coche nuevoCoche = new Coche();
26          nuevoCoche.velocidad = 50;
27          nuevoCoche.decelerar(80);
28          Assertions.assertEquals(0, nuevoCoche.velocidad);
29      }
```

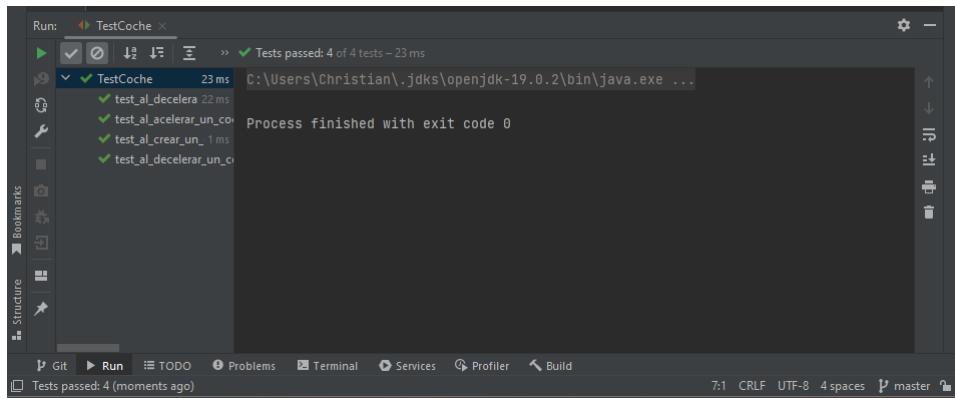
Hacemos click en “Run” para comprobar si pasamos los tests y, como podemos ver, nos encontramos con que este último falla, ya que se esperaba que el resultado fuese 0 pero ha dado -30.



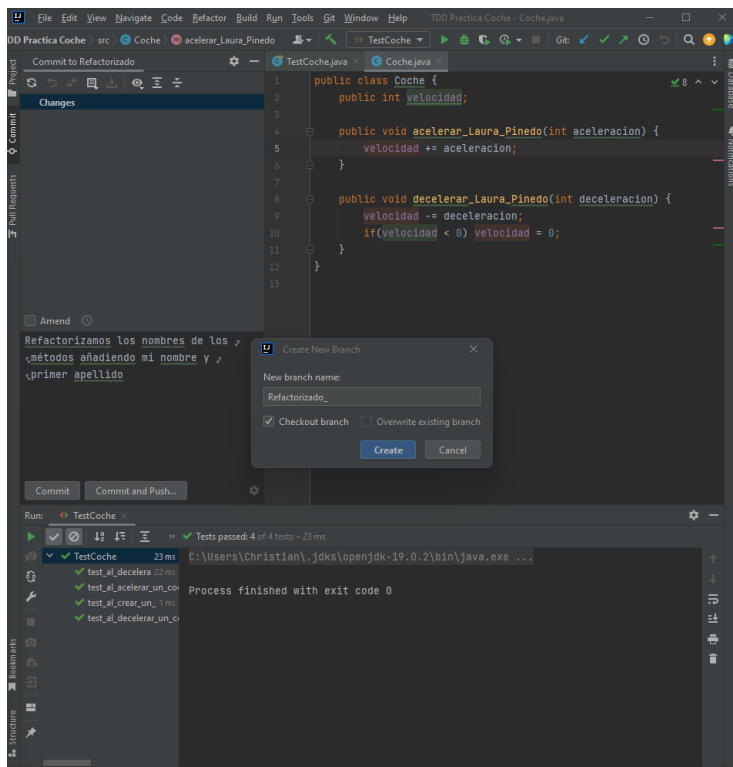
Para evitar este error, nos vamos a la clase “Coche” y especificamos que, si la velocidad es menor que 0, entonces el valor de velocidad será igual a 0.

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

Hecho esto, volvemos a pulsar “Run” y vemos que todo se ejecuta sin problemas.



Por último, pulsamos en Git -> Create new Branch y creamos una nueva rama llamada “Refactorizado”



Una vez creada, hacemos click derecho en el nombre del método, pinchamos en “Refactor” -> “Rename”, cambiamos al nombre deseado, en este caso: nombremetodo_Laura_Pinedo y pulsamos Enter.

```
public void decelerar_Laura_Pinedo(int deceleracion) {
```

Y así, habremos acabado la práctica.