## **MEMORIA**

## MI PRIMER TDD

- 1. Abrir Intellij, crear un proyecto nuevo con el nombre "TDD Practica Coche". Una vez creado, encima del proyecto, click derecho + new + Directory. Se crea un directorio nuevo llamado tests, después presionamos click derecho sobre tests y le damos a "Mark directory as" + "Test Resources Root", el icono de la carpeta se vuelve verde.
- 2. Una vez creado el directorio, se crea una clase llamada TestCoche y se añade @Test, se importa jUnit5 con las anotaciones activadas.
- 3. Se crea el metodo

```
public void test_crear_coche_su_velocidad_es_cero() {
Coche nuevoCoche = new Coche();
Assertions.assertEquals(0,nuevoCoche.velocidad);
```

```
### File Edit View Navigate Code Belactor Build Run Tools Git Window Help TDD Practica Coche (Extentorio/DAW IES SEVERO/15 Entormor de Desarrollo (EDE/UD10/TDD Practica Coche) - TestCoche, Java > 10 TestCoche, Java > 10
```

4. Se crea la clase Coche con public int velocidad.

```
Elle Edit View Navigate Code Refactor Build Run Iools Git Window Help TDD Practica Coche [E\Escritorio\DAW IES SEVERO\15 Entomos de Descritorio\DAW IES SEVERO\1
```

- 5. Se le da a Run TestCoche para verificar si todo va correcto.
- 6. Crear otro método:

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

```
Window Help TDD Practica Coche [E\Escritorio\DAW IES SEVERO\1S Entornos de Desarrollo (EDE\UD10\TDD Practica Coche] - TestCoche.java

d_es_cero

Day TestCoche → Day Coche

Day TestCoche

Day TestCoche

Day TestCoche

Day TestCoche

Day TestCoche

Day TestCoche

Day Coche

Day Coche
```

7. Se crea el metodo acelerar en la clase Coche

8. Ejecutar Run Test Coche para comprobar que todo esta correcto.

```
9.Una vez hecho, crear otro método:
```

```
@Test
public void test_decelerar_un_coche_su_velocidad_disminuye() {
    Coche nuevoCoche = new Coche();
    nuevoCoche.velocidad = 50;
    nuevoCoche.decelerar(20);
    Assertions.assertEquals(30, nuevoCoche.velocidad);
```

```
new*
     @Test
public void test_decelerar_un_coche_su_velocidad_disminuye() {
          Coche nuevoCoche = new Coche();
          nuevoCoche.velocidad = 50;
          nuevoCoche.decelerar( deceleracion: 20);
          Assertions.assertEquals( expected: 30, nuevoCoche.velocidad);
     }
}
Adoptium\jdk-17.0.4.101-hotspot\bin\java.exe" ...
```

10. En la clase Coche implementar metodo decelerar.

```
1 usage new*

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

- 11. Ejecutar de nuevo Run Test Coche
- 12. Por último, crear otro método:

```
@Test
public void
test_decelerar_un_coche_su_velocidad_no_puede_ser_menor_que_cero() {
    Coche nuevoCoche = new Coche();
    nuevoCoche.velocidad = 50;
    nuevoCoche.decelerar(80);
    Assertions.assertEquals(0,nuevoCoche.velocidad);
}
```

- 13. Se ejecuta Run Test Coche y se observa que hay un AssertionFailedError. Se va a la clase coche y se debe hacer una condicion if en el metodo anterior: if (velocidad < 0) velocidad =0;
- 14.Le das por última vez a ejecutar Run Test Coche y salen correctamente.

