

JUnit Concesionario

En este trabajo he podido hacer 5 test:



- ColorTest
- ModeloTest
- MarcaTest
- CocheTest
- ConcesionarioTest

ColorTest:

He comprobado que existan los colores:

```
public void testProbarColor() {  
    //Existen estos colores  
    assertNotNull(Color.PLATA);  
    assertNotNull(Color.AZUL);  
    assertNotNull(Color.ROJO);  
}
```

finished after 0,065 seconds

Runs: 1/1  Errors: 0  Failures: 0

  concesionarioCoches.ColorTest [Runner: JUn
  testProbarColor (0,000 s)

ModeloTest:

Primero he comprobado que existan los modelos:

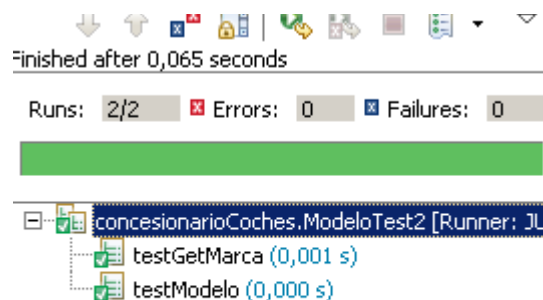
```
//Existen estos modelos  
assertNotNull(Modelo.CORDOBA);  
assertNotNull(Modelo.IBIZA);  
assertNotNull(Modelo.SERIE1);  
assertNotNull(Modelo.SERIE2);  
assertNotNull(Modelo.SERIE3);  
assertNotNull(Modelo.SERIE5);  
assertNotNull(Modelo.TOLEDO);
```

Luego he comprobado que los modelos tuvieran bien puesta su marca:

```
//Las marcas pertenecen a los modelos  
assertEquals(Modelo.CORDOBA.getMarca(), Marca.SEAT);  
assertEquals(Modelo.TOLEDO.getMarca(), Marca.SEAT);  
assertEquals(Modelo.IBIZA.getMarca(), Marca.SEAT);  
assertEquals(Modelo.SERIE1.getMarca(), Marca.BMW);  
assertEquals(Modelo.SERIE2.getMarca(), Marca.BMW);  
assertEquals(Modelo.SERIE3.getMarca(), Marca.BMW);  
assertEquals(Modelo.SERIE5.getMarca(), Marca.BMW);
```

Y para terminar he comprobado que no se le pudiera asignar la marca que no es la suya:

```
//Las marcas no pertenecen a los modelos
assertNotEquals(Modelo.CORDOBA.getMarca(), Marca.BMW);
assertNotEquals(Modelo.TOLEDO.getMarca(), Marca.BMW);
assertNotEquals(Modelo.IBIZA.getMarca(), Marca.BMW);
assertNotEquals(Modelo.SERIE1.getMarca(), Marca.SEAT);
assertNotEquals(Modelo.SERIE2.getMarca(), Marca.SEAT);
assertNotEquals(Modelo.SERIE3.getMarca(), Marca.SEAT);
assertNotEquals(Modelo.SERIE5.getMarca(), Marca.SEAT);
```



MarcaTest:

He comprobado que existan las marcas:

```
//Existen estas marcas
assertNotNull(Marca.BMW);
assertNotNull(Marca.SEAT);
```



CocheTest:

Primero he comprobado que se podían instanciar coches correctos:

```
//Coches correctamente creados.
assertNotNull(Coche.instanciarCoche("1234BBC", Color.AZUL, Modelo.CORDOBA));
assertNotNull(Coche.instanciarCoche("1234 BBC", Color.AZUL, Modelo.CORDOBA));
assertNotNull(Coche.instanciarCoche("1234-BBC", Color.AZUL, Modelo.CORDOBA));
```

Luego he comprobado que no se puedan instanciar coches incorrectos:

```
//Coches incorrectamente creados.
assertNull(Coche.instanciarCoche("1234ABC", Color.AZUL, Modelo.CORDOBA));
assertNull(Coche.instanciarCoche("1234AB", Color.AZUL, Modelo.CORDOBA));
assertNull(Coche.instanciarCoche("123BBC", Color.AZUL, Modelo.CORDOBA));
assertNull(Coche.instanciarCoche("1234_BBC", Color.AZUL, Modelo.CORDOBA));
assertNull(Coche.instanciarCoche("1234ABCD", Color.AZUL, Modelo.CORDOBA));
assertNull(Coche.instanciarCoche("1234&BC", Color.AZUL, Modelo.CORDOBA));
assertNull(Coche.instanciarCoche("BBC1234", Color.AZUL, Modelo.CORDOBA));
assertNull(Coche.instanciarCoche("1234ABC", null, null));
```

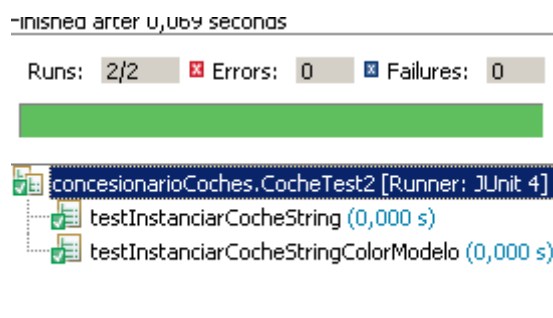
Después me he asegurado de que se podían instanciar matriculas correctas:

```
//Instanciar matricula correctamente
assertNotNull(Coche.instanciarCoche("1234BBC"));
assertNotNull(Coche.instanciarCoche("1234 BBC"));
assertNotNull(Coche.instanciarCoche("1234-BBC"));
```

Y por último que no se podían instanciar matriculas incorrectas:

```
//Instanciar matricula incorrectamente
assertNull(Coche.instanciarCoche("1234ABC"));
assertNull(Coche.instanciarCoche("1234AB"));
assertNull(Coche.instanciarCoche("123BBC"));
assertNull(Coche.instanciarCoche("1234_BBC"));
assertNull(Coche.instanciarCoche("1234ABCD"));
assertNull(Coche.instanciarCoche("1234&BC"));
assertNull(Coche.instanciarCoche("BBC1234"));
```

Y aquí los resultados de las comprobaciones.



ConcesionarioTest:

Primero he comprobado que se podía añadir coches correctos:

```
//Añadir coche correctamente
assertTrue(concesionario.annadir("1234BBC", Color.AZUL, Modelo.CORDOBA));
assertTrue(concesionario.annadir("1234CCX", Color.AZUL, Modelo.CORDOBA));
```

Luego que no se podían añadir coches incorrectos:

```
//añadir coches incorrectamente
assertFalse(concesionario.annadir("1234BBC", null, null));
assertFalse(concesionario.annadir("1234ABC", Color.AZUL, Modelo.CORDOBA));
assertFalse(concesionario.annadir("1234AB", Color.AZUL, Modelo.CORDOBA));
assertFalse(concesionario.annadir("123BBC", Color.AZUL, Modelo.CORDOBA));
assertFalse(concesionario.annadir("1234_BBC", Color.AZUL, Modelo.CORDOBA));
assertFalse(concesionario.annadir("1234ABCD", Color.AZUL, Modelo.CORDOBA));
assertFalse(concesionario.annadir("1234&BC", Color.AZUL, Modelo.CORDOBA));
assertFalse(concesionario.annadir("BBC1234", Color.AZUL, Modelo.CORDOBA));
//Repetir coche
```

Después he comprobado que no se podía añadir un coche repetido:

```
//Repetir coche
assertFalse(concesionario.annadir("1234BBC", Color.AZUL, Modelo.CORDOBA));
```

Y si se podía eliminar un coche crado anteriormente:

```
//Eliminar coche
assertTrue(concesionario.eliminar("1234BBC"));
```

Después he comprobado que se podía ver el tamaño del almacen en este caso sería 3 y que no se podía ver una cantidad incorrecta:

```
@Test
public void testSize() {
    //Comprobar almacen del coche, ahora mismo esta creado "1234CCX"
    assertEquals(concesionario.size(), 3);
    assertNotEquals(concesionario.size(), 0);
}
```

También he comprobado que no se podía eliminar un coche inexistente:

```
//Eliminar coche inexistente
assertFalse(concesionario.eliminar("1234VVX"));
//...
}
```

Después he pasado a la comprobación del método mostrar, si se puede mostrar un coche:

```
//Mostrar coche
assertNotNull(concesionario.get("1234CCX"));
//...
```

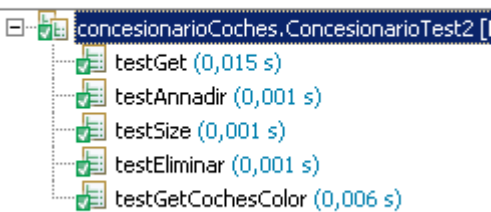
Que no se pueda mostrar un coche inexistente:

```
//Mostrar coche inexistente
assertNull(concesionario.get("1234VVX"));
```

Y por último el método `getCochesColor`:

```
@Test
public void testGetCochesColor() {
    assertNotNull(concesionario.getCochesColor(Color.AZUL));
    assertNotNull(concesionario.getCochesColor(Color.ROJO));
    assertNotNull(concesionario.getCochesColor(Color.PLATA));
}
```

Runs: 5/5 ❌ Errors: 0 ❌ Failures: 0



```
concesionarioCoches.ConcesionarioTest2 [
  testGet (0,015 s)
  testAnnadir (0,001 s)
  testSize (0,001 s)
  testEliminar (0,001 s)
  testGetCochesColor (0,006 s)
```