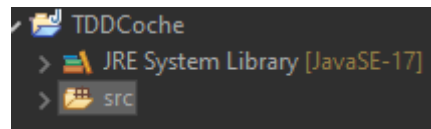


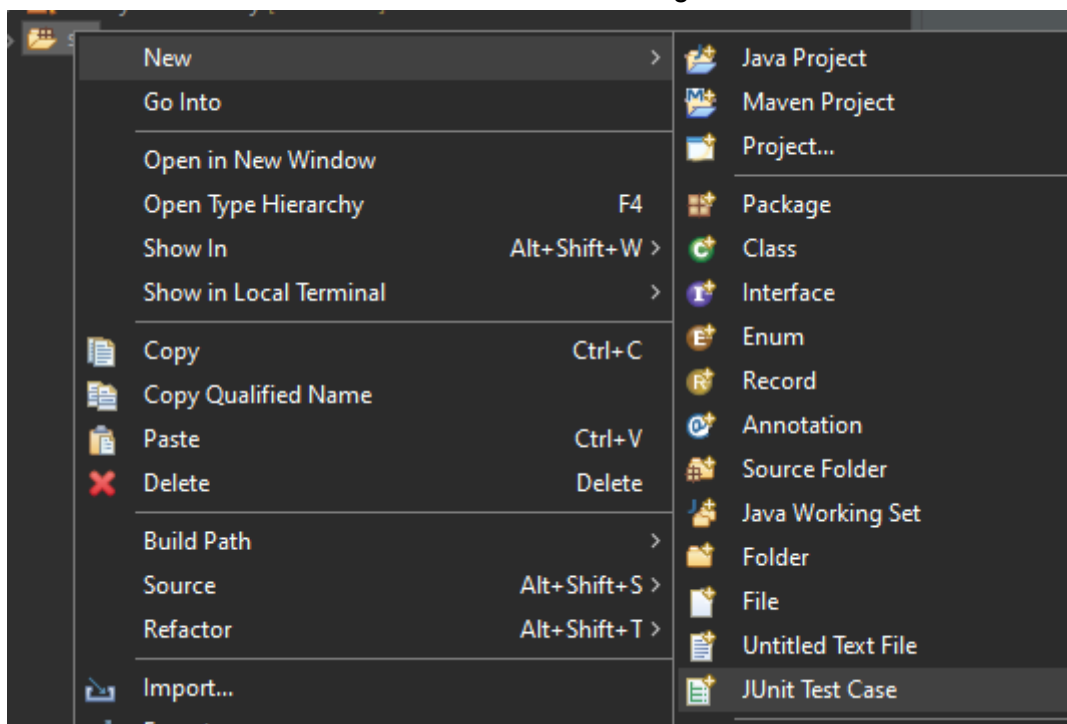
Entornos de desarrollo

Mi primer TDD V2.0

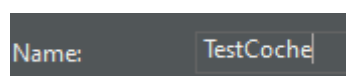
En esta práctica voy a replicar lo realizado en la práctica anterior pero en Eclipse. Para comenzar creamos un proyecto nuevo:



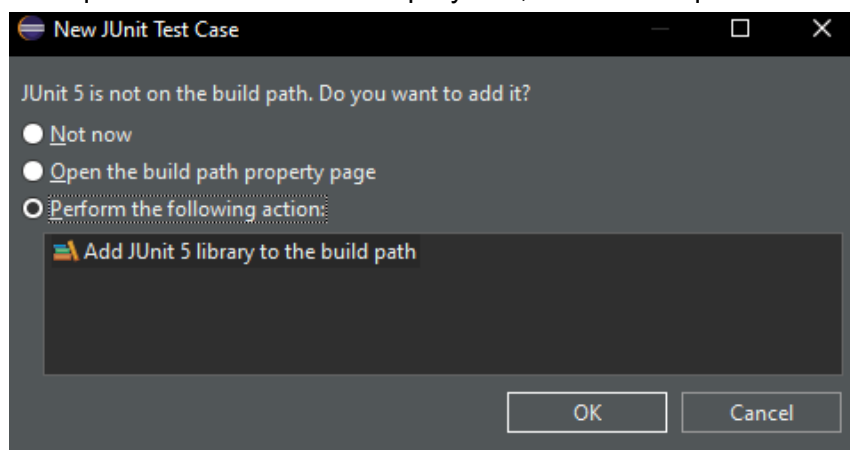
Ahora vamos a crear una clase de tests de JUnit de la siguiente forma:



La vamos a llamar TestCoche:



Nos preguntará si queremos añadir JUnit al proyecto, le decimos que si:



Creamos un test para crear un coche igual que antes y resolvemos el error por medio del IDE para crear la clase coche:

```
@Test
void test_crear_coche() {
    Coche nuevoCoche = new Coche();
}
```

```
@Test
void test_crear_coche() {
    Coche nuevoCoche = new Coche();
}
```

Coche cannot be resolved to a type

7 quick fixes available:

- Create class 'Coche'

```
1 public class Coche {
2
3 }
```

Ejecutamos el test y comprobamos que funciona:

Runs: 1/1 ✖ Errors: 0 ✖ Failures: 0

> TestCoche [Runner: JUnit 5] (0,017 s)

Modificamos el test y creamos uno que compruebe que la velocidad al crearse un coche es 0, usando un assert. Usaremos este test para crear el atributo velocidad en la clase coche:

```
@Test
void test_al_crear_un_coche_su_velocidad_es_0() {
    Coche nuevoCoche = new Coche();
    Assertions.assertEquals(0, nuevoCoche.velocidad);
}
```

velocidad cannot be resolved or is not a field

2 quick fixes available:

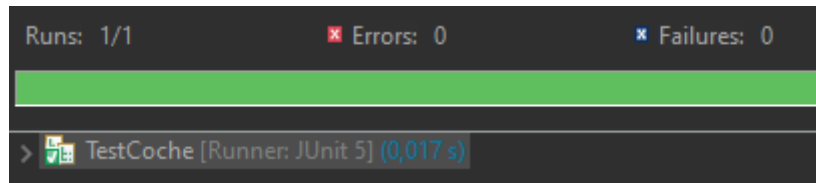
- Create field 'velocidad' in type 'Coche'
- Create constant 'velocidad' in type 'Coche'

```
public class Coche {

    public int velocidad;

}
```

Ejecutamos otra vez el test y comprobamos que funciona:



Vamos a crear 3 nuevos tests para terminar nuestra clase Coche:

```
@Test
void test_al_acelerar_un_coche_su_velocidad_aumenta() {

}

@Test
void test_al_decelerar_un_coche_su_velocidad_disminuye() {

}

@Test
void test_al_decelerar_un_coche_su_velocidad_no_puede_ser_inferior_a_0() {

}

}
```

Usaremos el primer test para crear el método acelerar:

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

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

@Test
void test_al_de
Coche nuevo
```

The method `acelerar(int)` is undefined for the type `Coche`

2 quick fixes available:

- Create method '`acelerar(int)`' in type '`Coche`'

```
public void acelerar(int aceleracion) {
    velocidad = velocidad + aceleracion;
}
```

El segundo test servirá para el método desacelerar:

```
@Test
void test_al_decelerar_un_coche_su_velocidad_disminuye() {
    Coche nuevoCoche = new Coche();
    nuevoCoche.acelerar(30);
    nuevoCoche.decelerar(20);
    Assertions.
}

@Test
void test_al_de
Coche nuevo
```

The method `decelerar(int)` is undefined for the type `Coche`

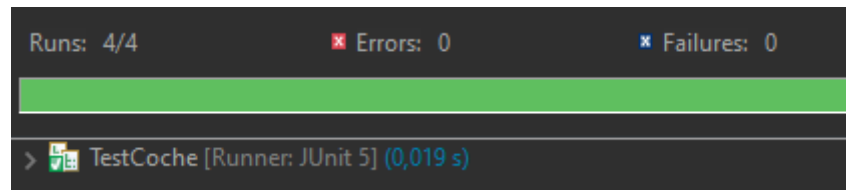
3 quick fixes available:

- Change to '`acelerar(...)`'

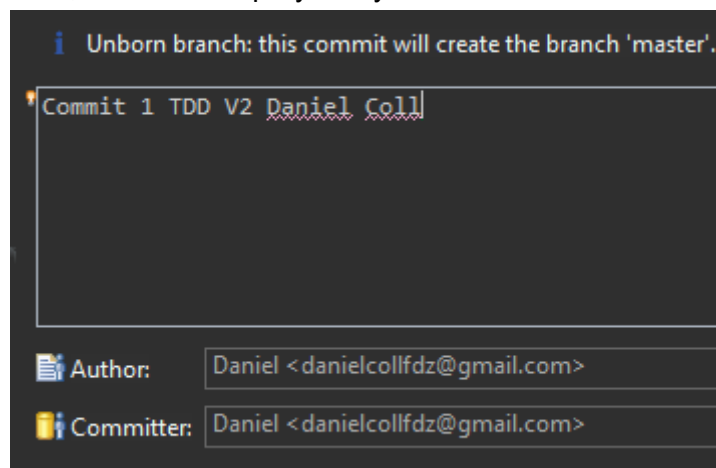
Y con el último test vamos a modificar el método decelerar para que la velocidad no pueda ser inferior a 0:

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

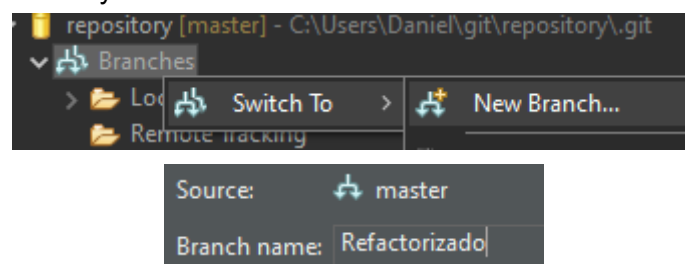
Ejecutamos los 4 tests y comprobamos que funcionen:



Creamos un repositorio, añadimos el proyecto y hacemos un commit con todo esto:



Creamos una nueva rama y nos movemos a ella. La vamos a llamar refactorizado:



Mediante refactorización, añado mi nombre a todos los tests:

```
@Test
void test_al_crear_un_coche_su_velocidad_es_0DaniColl() {
    Coche nuevoCoche = new Coche();
    Assertions.assertEquals(0, nuevoCoche.velocidad);
}
@Test
void test_al_acelerar_un_coche_su_velocidad_aumentaDaniColl() {
    Coche nuevoCoche = new Coche();
    nuevoCoche.acelerar(30);
    Assertions.assertEquals(30, nuevoCoche.velocidad);
}
@Test
void test_al_decelerar_un_coche_su_velocidad_disminuyeDaniColl() {
    Coche nuevoCoche = new Coche();
    nuevoCoche.acelerar(30);
    nuevoCoche.decelerar(20);
    Assertions.assertEquals(10, nuevoCoche.velocidad);
}
@Test
void test_al_decelerar_un_coche_su_velocidad_no_puede_ser_inferior_a_0DaniColl() {
    Coche nuevoCoche = new Coche();
    nuevoCoche.acelerar(30);
    nuevoCoche.decelerar(50);
    Assertions.assertEquals(0, nuevoCoche.velocidad);
}
```


Hago un nuevo commit con estos cambios y de paso subo el repositorio a github:


Commit 2 TDD V2 Daniel Coll

Author: Daniel <danielcollfdz@gmail.com>

Committer: Daniel <danielcollfdz@gmail.com>

Refactorizado 1 branch 0 tags Go to file Add file <> Code

 **Daniel Coll** Commit 3 TDD V2 Daniel Coll e00cb77 7 minutes ago 3 commits

 TDDCoche Commit 3 TDD V2 Daniel Coll 7 minutes ago

Y ahora voy a crear la rama memoria y subir este documento pdf:

