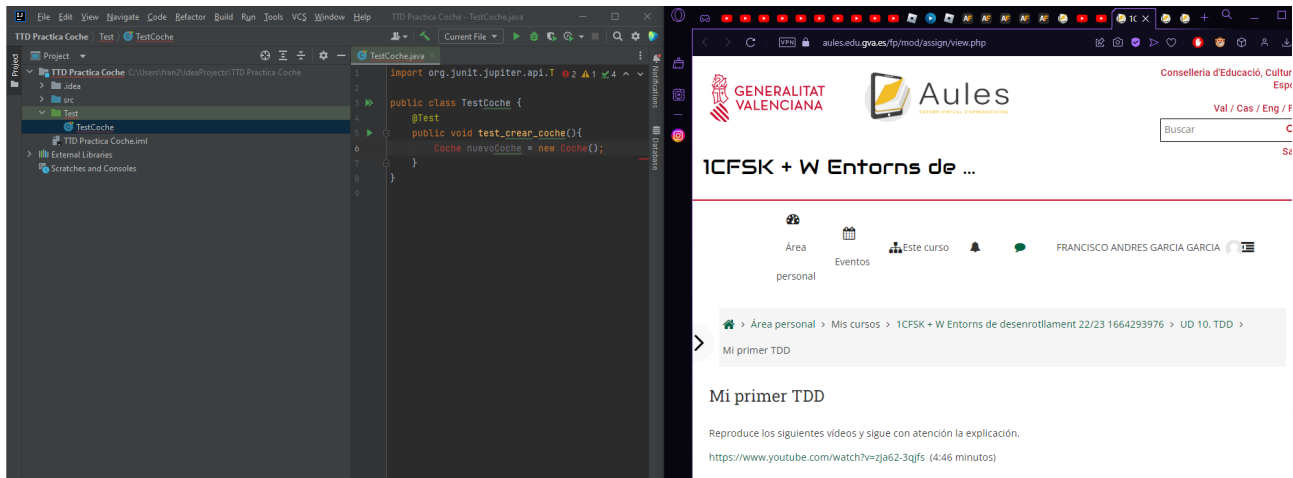
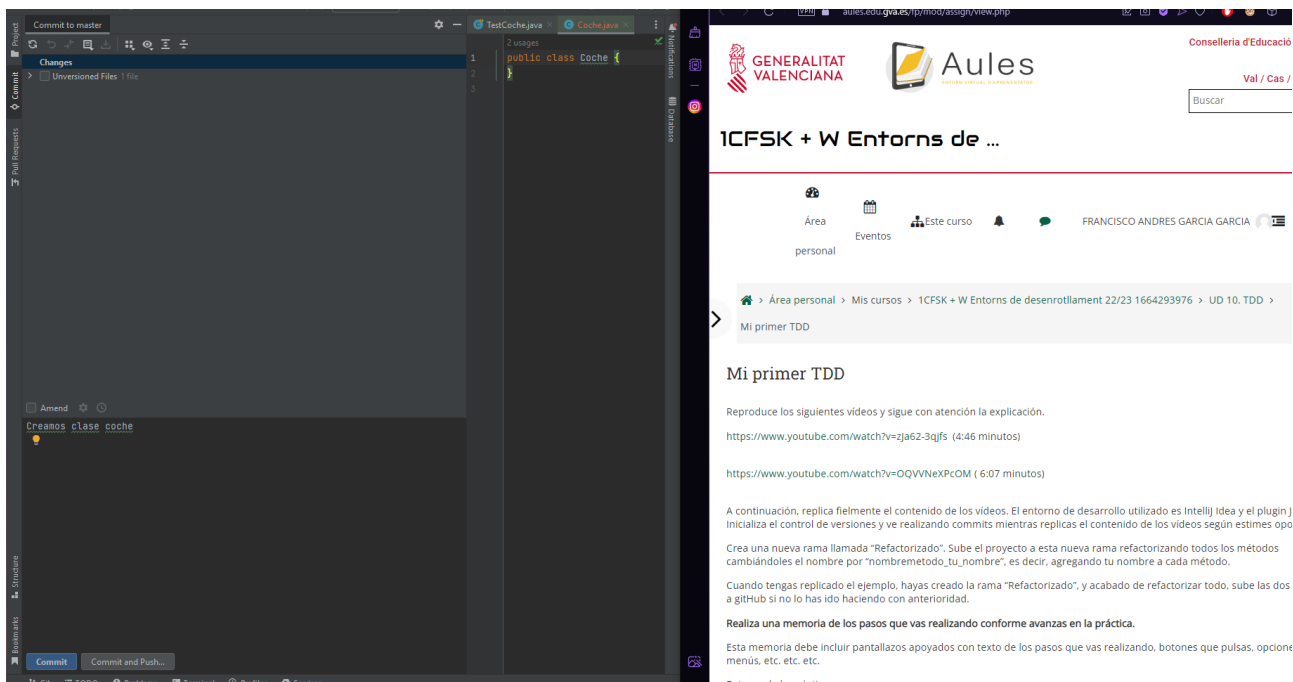


# MEMORIA

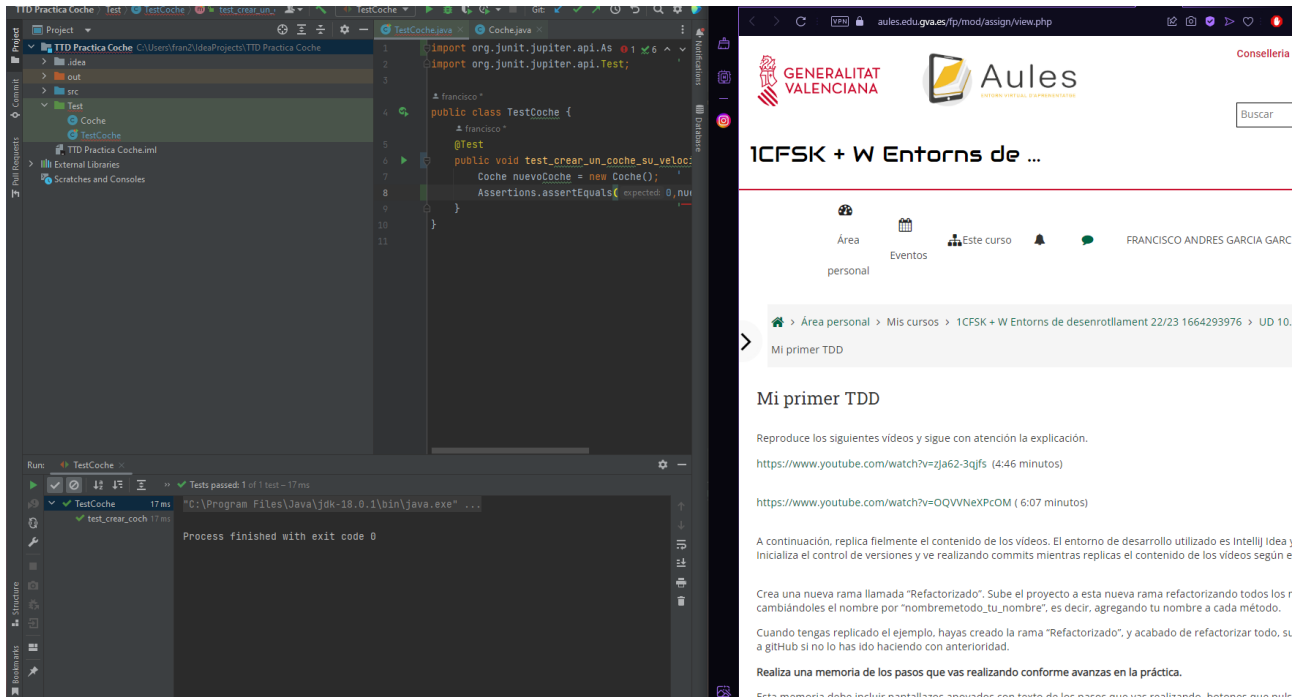
## Creando el test



## Creamos clase coche



## Modifico el test



The screenshot shows two side-by-side windows. On the left is the IntelliJ IDEA IDE, and on the right is the Aules LMS website.

**IntelliJ IDEA:** The main editor shows the `TestCoche.java` file. The code is as follows:

```
1 import org.junit.jupiter.api.Assertions;
2 import org.junit.jupiter.api.Test;
3
4 public class TestCoche {
5     @Test
6     public void test_crear_un_coche_su_velocidad() {
7         Coche nuevoCoche = new Coche();
8         Assertions.assertEquals(0, nuevoCoche.getVelocidad());
9     }
10 }
11
```

The Run window at the bottom shows the test results: "Tests passed: 1 of 1 test - 17 ms". The console output shows "Process finished with exit code 0".

**Aules LMS:** The page is titled "1CFSK + W Entorns de ...". It shows a navigation bar with "Área personal", "Eventos", "Este curso", and "Francisco Andres Garcia Garcia". The main content area is titled "Mi primer TDD" and contains the following text:

Reproduce los siguientes vídeos y sigue con atención la explicación.

<https://www.youtube.com/watch?v=zja62-3qjfs> (4:46 minutos)

<https://www.youtube.com/watch?v=OQVVNeXPCOM> (6:07 minutos)

A continuación, replica fielmente el contenido de los vídeos. El entorno de desarrollo utilizado es IntelliJ Idea y inicializa el control de versiones y ve realizando commits mientras replicas el contenido de los vídeos según vayas avanzando.

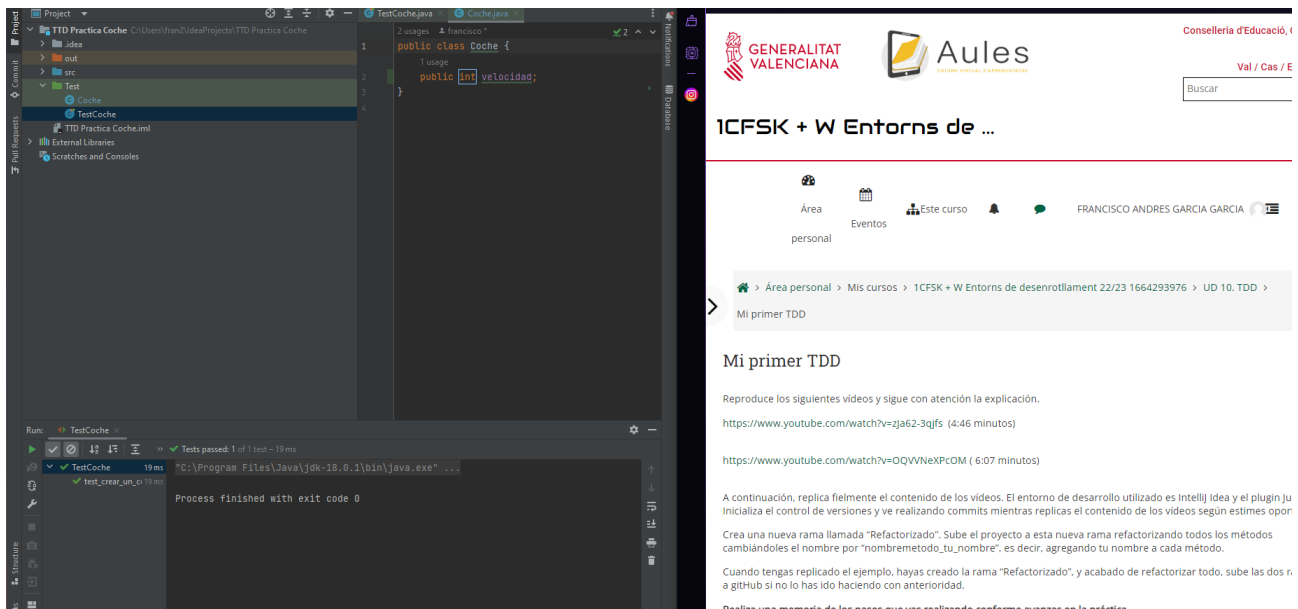
Crea una nueva rama llamada "Refactorizado". Sube el proyecto a esta nueva rama refactorizando todos los métodos cambiando el nombre por "nombremetodo\_tu\_nombre", es decir, agregando tu nombre a cada método.

Cuando tengas replicado el ejemplo, has creado la rama "Refactorizado", y acabado de refactorizar todo, sube las dos ramas a GitHub si no lo has ido haciendo con anterioridad.

**Realiza una memoria de los pasos que vas realizando conforme avanzas en la práctica.**

Esta memoria debe incluir pantallazos anotados con texto de los pasos que vas realizando, botones que pulsas, etc.

## Se añade atributo velocidad en clase coche



The screenshot shows two side-by-side windows. On the left is the IntelliJ IDEA IDE, and on the right is the Aules LMS website.

**IntelliJ IDEA:** The main editor shows the `Coche.java` file. The code is as follows:

```
1 public class Coche {
2     public int velocidad;
3 }
4
```

The Run window at the bottom shows the test results: "Tests passed: 1 of 1 test - 19 ms". The console output shows "Process finished with exit code 0".

**Aules LMS:** The page is titled "1CFSK + W Entorns de ...". It shows a navigation bar with "Área personal", "Eventos", "Este curso", and "Francisco Andres Garcia Garcia". The main content area is titled "Mi primer TDD" and contains the following text:

Reproduce los siguientes vídeos y sigue con atención la explicación.

<https://www.youtube.com/watch?v=zja62-3qjfs> (4:46 minutos)

<https://www.youtube.com/watch?v=OQVVNeXPCOM> (6:07 minutos)

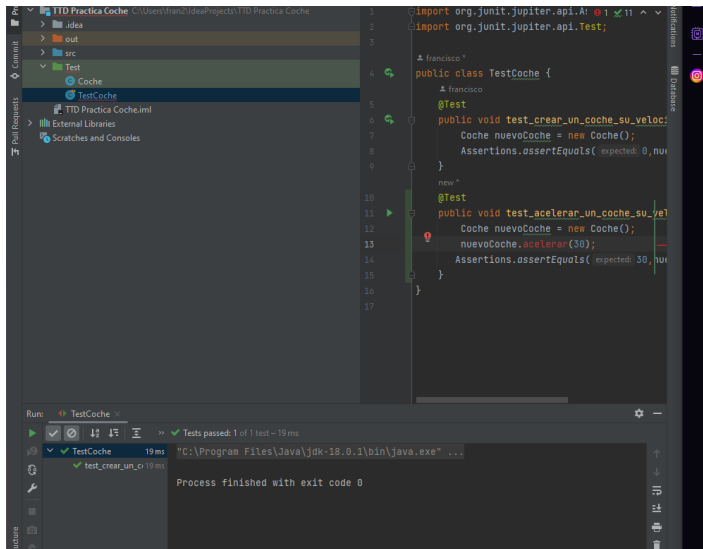
A continuación, replica fielmente el contenido de los vídeos. El entorno de desarrollo utilizado es IntelliJ Idea y el plugin JUnit. Inicializa el control de versiones y ve realizando commits mientras replicas el contenido de los vídeos según vayas avanzando.

Crea una nueva rama llamada "Refactorizado". Sube el proyecto a esta nueva rama refactorizando todos los métodos cambiando el nombre por "nombremetodo\_tu\_nombre", es decir, agregando tu nombre a cada método.

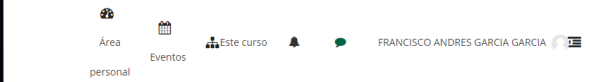
Cuando tengas replicado el ejemplo, has creado la rama "Refactorizado", y acabado de refactorizar todo, sube las dos ramas a GitHub si no lo has ido haciendo con anterioridad.

**Realiza una memoria de los pasos que vas realizando conforme avanzas en la práctica.**

## Creo nuevo test que acelera el coche



### 1CFSK + W Entorns de ...



#### Mi primer TDD

Reproduce los siguientes vídeos y sigue con atención la explicación.

<https://www.youtube.com/watch?v=zja62-3qjfs> (4:46 minutos)

<https://www.youtube.com/watch?v=QVvNeXPcOM> ( 6:07 minutos)

A continuación, replica fielmente el contenido de los vídeos. El entorno de desarrollo utilizado es IntelliJ IDEA y el plugin Inicializa el control de versiones y ve realizando commits mientras replicas el contenido de los vídeos según estimes oportuno.

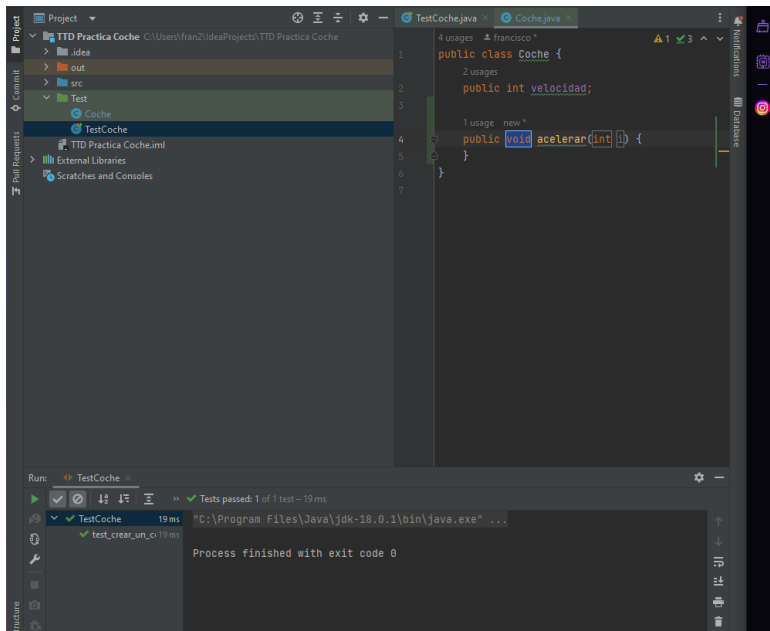
Crea una nueva rama llamada "Refactorizado". Sube el proyecto a esta nueva rama refactorizando todos los métodos cambiando el nombre por "nombremetodo\_tu\_nombre", es decir, agregando tu nombre a cada método.

Cuando tengas replicado el ejemplo, has creado la rama "Refactorizado", y acabado de refactorizar todo, sube las dos a GitHub si no lo has ido haciendo con anterioridad.

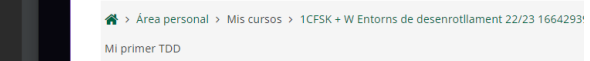
**Realiza una memoria de los pasos que vas realizando conforme avanzas en la práctica.**

Esta memoria debe incluir pantallazos anotados con texto de los pasos que vas realizando, botones que pulcas, oración

## Creo metodo acelerar



### 1CFSK + W Entorns de ...



#### Mi primer TDD

Reproduce los siguientes vídeos y sigue con atención la explicación.

<https://www.youtube.com/watch?v=zja62-3qjfs> (4:46 minutos)

<https://www.youtube.com/watch?v=QVvNeXPcOM> ( 6:07 minutos)

A continuación, replica fielmente el contenido de los vídeos. El entorno de desarrollo utilizado es IntelliJ IDEA y el plugin Inicializa el control de versiones y ve realizando commits mientras replicas el contenido de los vídeos según estimes oportuno.

Crea una nueva rama llamada "Refactorizado". Sube el proyecto a esta nueva rama refactorizando todos los métodos cambiando el nombre por "nombremetodo\_tu\_nombre", es decir, agregando tu nombre a cada método.

Cuando tengas replicado el ejemplo, has creado la rama "Refactorizado", y acabado de refactorizar todo, sube las dos a GitHub si no lo has ido haciendo con anterioridad.

**Realiza una memoria de los pasos que vas realizando conforme avanzas en la práctica.**

Esta memoria debe incluir pantallazos apoyados con texto de los pasos que vas realizando, botones que pulcas, oración

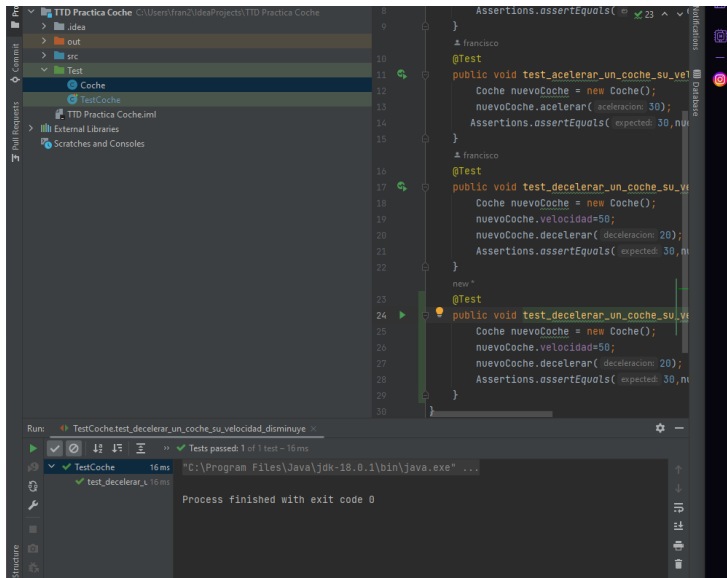
## Creo nuevo test decelerar

The screenshot shows two windows. On the left, the IntelliJ IDEA IDE displays the `TestCoche.java` file. The code defines a `Coche` class with a `velocidad` attribute and an `acelerar` method. A new test method `test_decelerar_un_coche_su_velocidad` is being added, which creates a `Coche` object, sets its velocity to 50, and calls `decelerar(30)`. The test asserts that the velocity is 20. The right window shows a web browser with the 'Aules' portal of the 'GENERALITAT VALENCIANA'. The page title is 'ICFSK + W Entorns de ...'. The user is logged in as 'FRANCISCO ANDRES GARCIA GARCIA'. The page content includes a navigation bar with 'Área personal', 'Eventos', 'Este curso', and a search bar. Below the navigation bar, there is a section titled 'Mi primer TDD' with instructions to watch two YouTube videos: <https://www.youtube.com/watch?v=zja62-3qjfs> (4:46 minutos) and <https://www.youtube.com/watch?v=QVVNeXPcOM> (6:07 minutos). The instructions state: 'A continuación, replica fielmente el contenido de los videos. El entorno de desarrollo utilizado es IntelliJ Idea y el plugin junit 4. Inicializa el control de versiones y ve realizando commits mientras replicas el contenido de los videos según estimes oportuno.'

## Creo metodo decelerar

```
TestCoche.java x Coche.java x
6 usages francisco *
1 public class Coche {
2     6 usages
3     public int velocidad;
4
5     1 usage francisco
6     public void acelerar(int aceleracion) {
7         velocidad += aceleracion;
8     }
9
10    1 usage new *
11    public void decelerar(int deceleracion) {
12        velocidad -= deceleracion;
13    }
14 }
```

Creo nuevo test no puede ser menor que cero



```
1 package TTD;
2
3 public class Coche {
4     private int velocidad;
5
6     public Coche() {
7         velocidad = 0;
8     }
9
10    public void acelerar(int aceleracion) {
11        velocidad += aceleracion;
12    }
13
14    public void decelerar(int deceleracion) {
15        velocidad -= deceleracion;
16        if (velocidad < 0) {
17            velocidad = 0;
18        }
19    }
20 }
21
22 @Test
23 public void test_acelerar_un_coche_su_velocidad_aumenta() {
24     Coche nuevoCoche = new Coche();
25     nuevoCoche.acelerar(30);
26     Assertions.assertEquals(30, nuevoCoche.velocidad);
27 }
28
29 @Test
30 public void test_decelerar_un_coche_su_velocidad_disminuye() {
31     Coche nuevoCoche = new Coche();
32     nuevoCoche.acelerar(50);
33     nuevoCoche.decelerar(20);
34     Assertions.assertEquals(30, nuevoCoche.velocidad);
35 }
36
37 @Test
38 public void test_decelerar_un_coche_su_velocidad_no_puede_ser_negativa() {
39     Coche nuevoCoche = new Coche();
40     nuevoCoche.acelerar(50);
41     nuevoCoche.decelerar(60);
42     Assertions.assertEquals(0, nuevoCoche.velocidad);
43 }
```



GENERALITAT VALENCIANA

Aules

Conselleria c

1CFSK + W Entorns de ...

FRANCISCO ANDRES GARCIA GARCIA

Área personal

Eventos

Este curso

Francisco Andres Garcia Garcia

Área personal > Mis cursos > 1CFSK + W Entorns de desenvolupament 22/23 1664293976 > UD 10.1

Mi primer TDD

Reproduce los siguientes videos y sigue con atención la explicación.

<https://www.youtube.com/watch?v=zja62-3qjfs> (4:46 minutos)

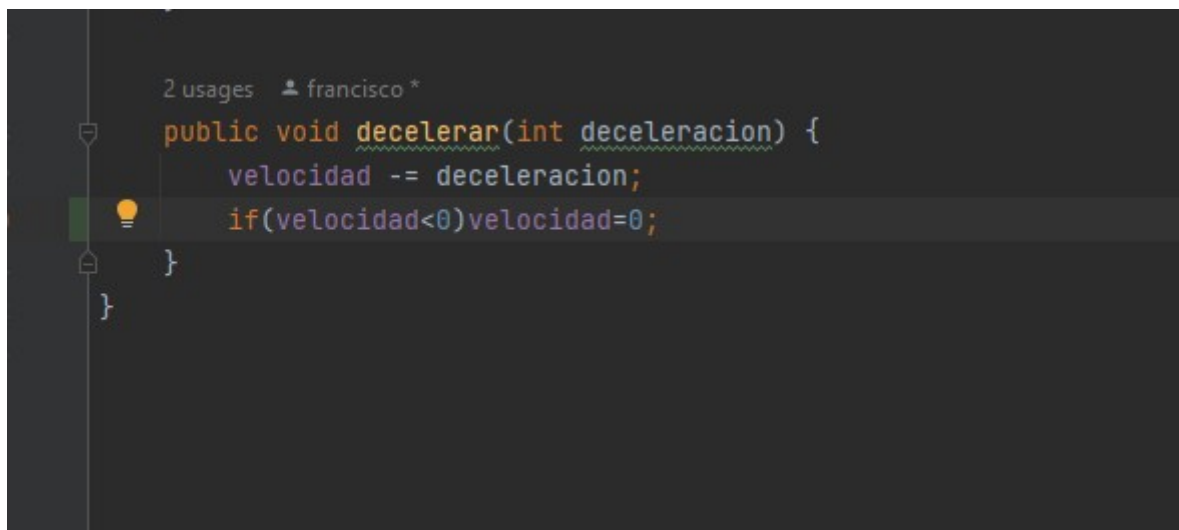
<https://www.youtube.com/watch?v=QVWNeXPcOM> (6:07 minutos)

A continuación, replica fielmente el contenido de los videos. El entorno de desarrollo utilizado es IntelliJ Idea y Inicializa el control de versiones y ve realizando commits mientras replicas el contenido de los videos según es

Crea una nueva rama llamada "Refactorizado". Sube el proyecto a esta nueva rama refactorizando todos los m cambiándoles el nombre por "nombremetodo\_tu\_nombre", es decir, agregando tu nombre a cada método.

Cuando tengas replicado el ejemplo, hayas creado la rama "Refactorizado", y acabado de refactorizar todo, sut

Modificacion clase coche



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

## Refactorizo metodos

```
2      public int velocidad;  
3  
4      1 usage  🧑 francisco *  
5      public void acelerar_Francisco_andres_garcia_garcia(int aceleracion) {  
6          velocidad += aceleracion;  
7      }  
8  
9      2 usages  🧑 francisco *  
10     public void decelerar_Francisco_andres_garcia_garcia(int deceleracion) {  
11         velocidad -= deceleracion;  
12         if(velocidad<0) velocidad=0;  
13     }  
14 }
```

```
@Test  
public void test_crear_un_coche_su_velocidad_es_cero() {  
    Coche nuevoCoche = new Coche();  
    Assertions.assertEquals( expected: 0, nuevoCoche.velocidad);  
}  
  
🧑 francisco *  
@Test  
public void test_acelerar_un_coche_su_velocidad_aumenta() {  
    Coche nuevoCoche = new Coche();  
    nuevoCoche.acelerar_Francisco_andres_garcia_garcia( aceleracion: 30);  
    Assertions.assertEquals( expected: 30, nuevoCoche.velocidad);  
}  
  
🧑 francisco *  
@Test  
public void test_decelerar_un_coche_su_velocidad_disminuye() {  
    Coche nuevoCoche = new Coche();  
    nuevoCoche.velocidad=50;  
    nuevoCoche.decelerar_Francisco_andres_garcia_garcia( deceleracion: 20);  
    Assertions.assertEquals( expected: 30, nuevoCoche.velocidad);  
}  
  
🧑 francisco *  
@Test  
public void test_decelerar_un_coche_su_velocidad_no_puede_ser_menos_que_cero() {  
    Coche nuevoCoche = new Coche();  
    nuevoCoche.velocidad=50;  
    nuevoCoche.decelerar_Francisco_andres_garcia_garcia( deceleracion: 80);  
}
```