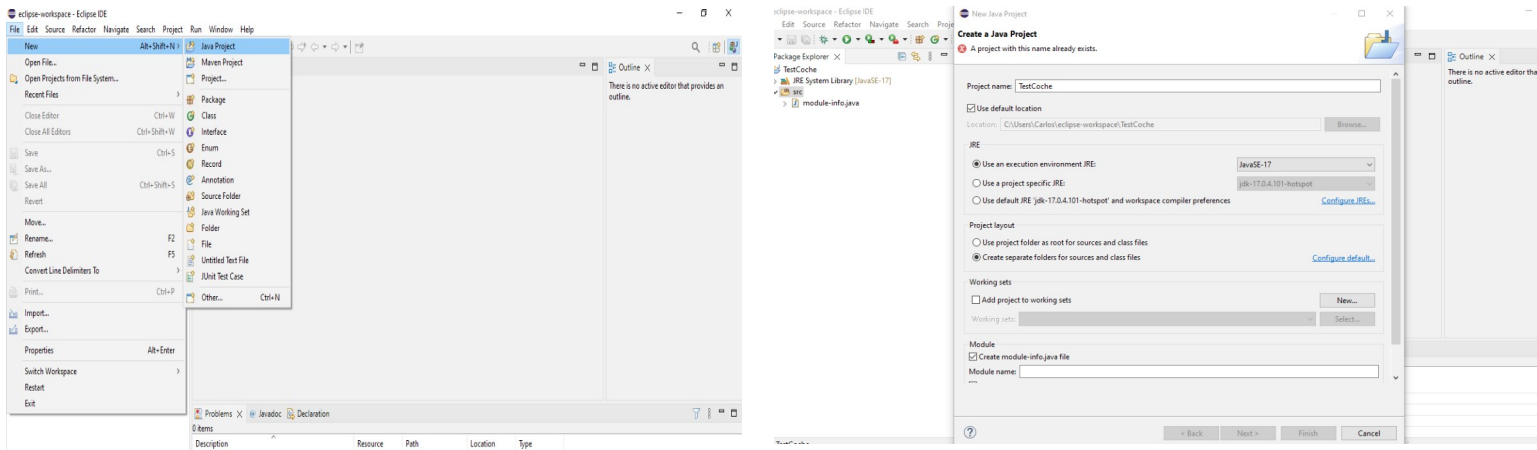
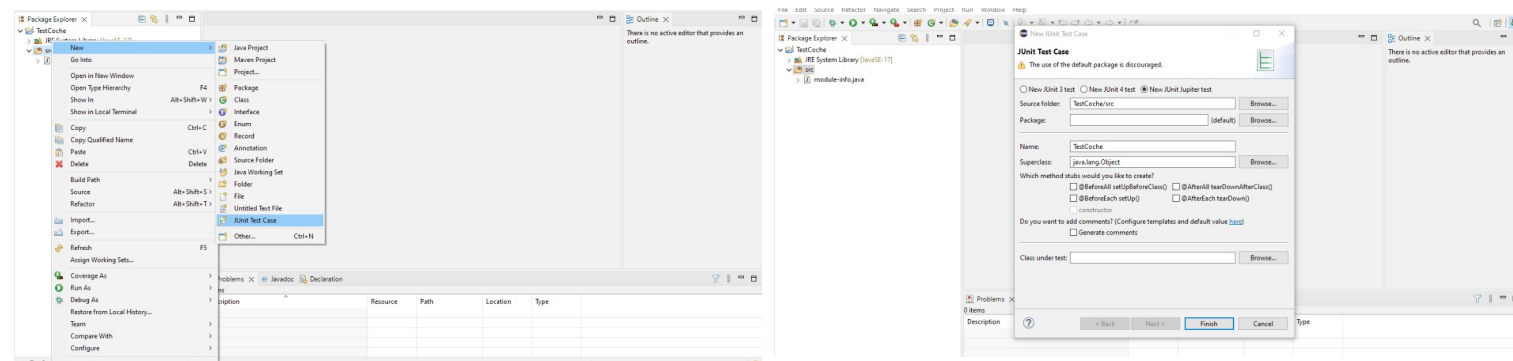


Mi primer TDD V2.0

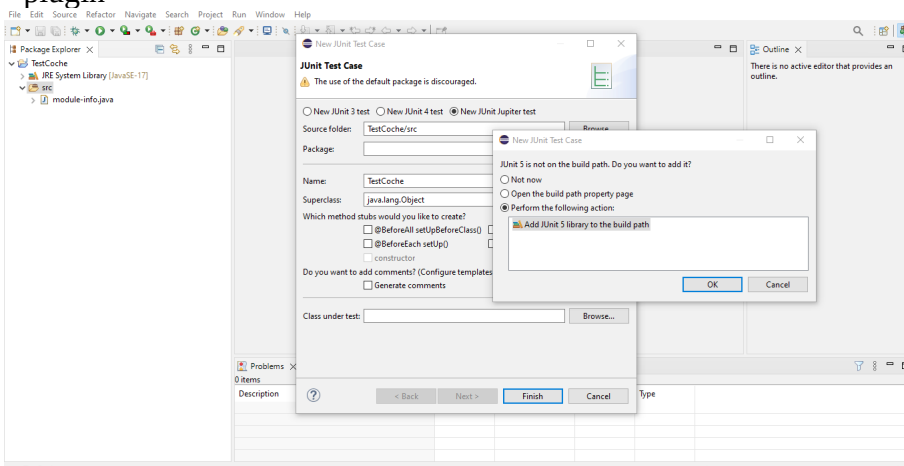
Abriremos Java y crearemos un nuevo proyecto, en mi caso será llamada “TestCoche”



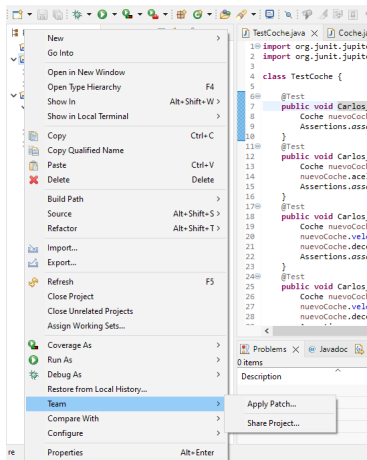
En mi caso no lo crearé ya que ya lo tengo creado. Crearemos un Junit test case, lo llamaremos TestCoche también



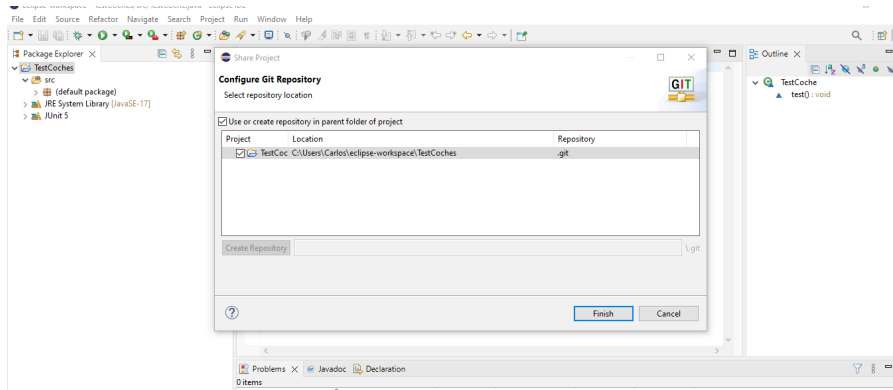
En mi caso JUnit5 se instalará solo, a diferencia que en intelliJ el cual lo tienes que instalar con un plugin



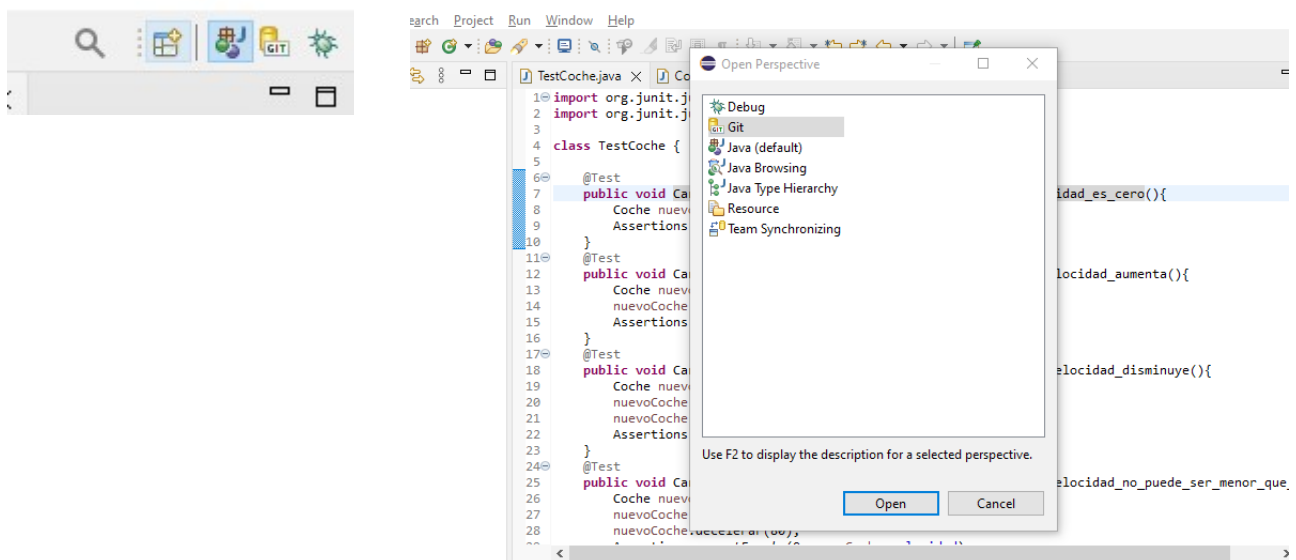
Despues de esto crearemos el repositorio GIT en Eclipse, para esto haremos click derecho sobre nuestro proyecto → Team → Share Project



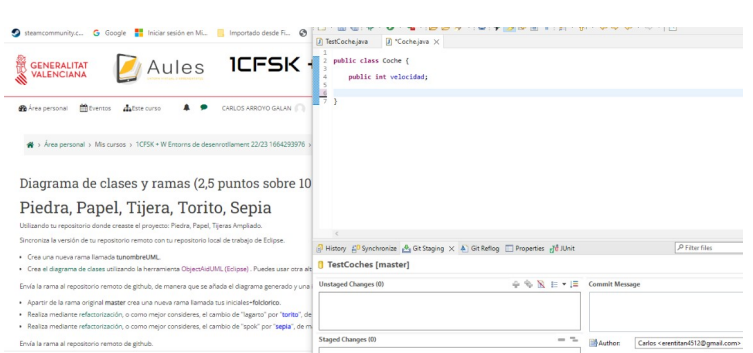
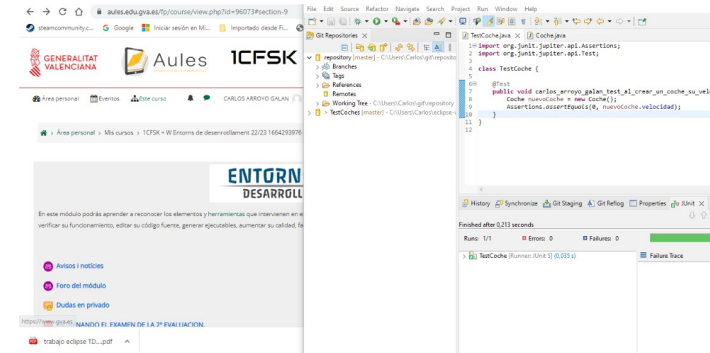
Pulsamos en use or create repository → pulsamos en nuestro proyecto → create repository → Finish



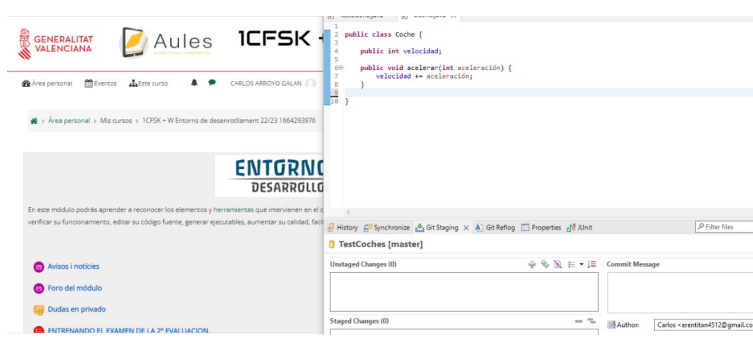
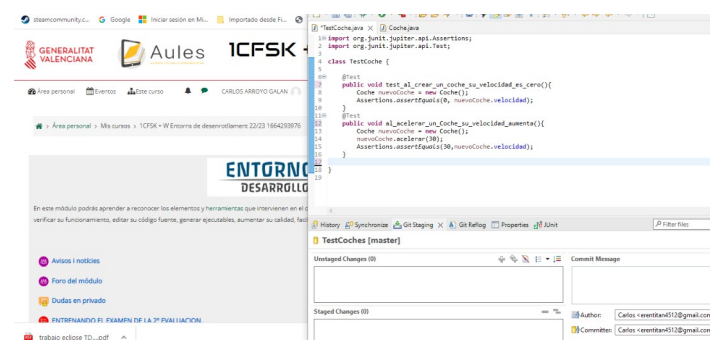
Seguido de esto cambiaremos a la interfaz de git, iremos a open perspective → Git, una diferencia que he podido ver respecto a intelliJ es que en eclipse el entorno para usar git es mucho mas completo a mi parecer que el de intelliJ



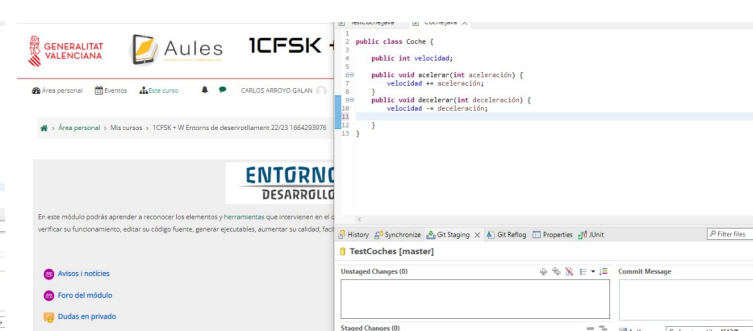
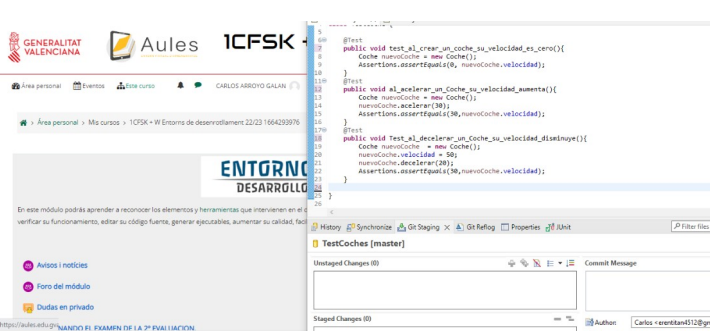
Primer Test



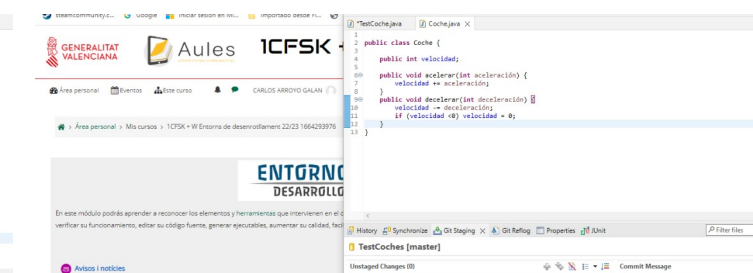
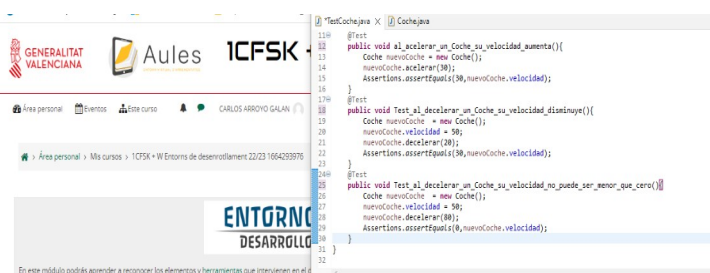
Segundo Test



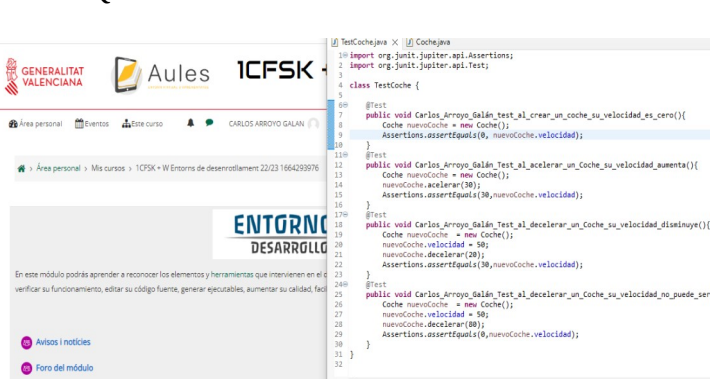
Tercer Test



Cuarto Test



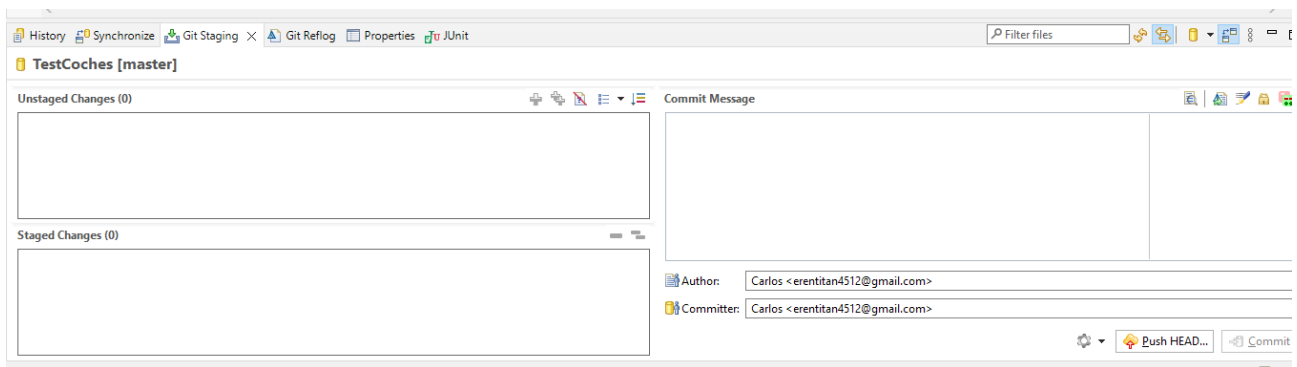
Quinto Test



Usamos el test para comprobar que todos los funcionen correctamente

```
10 import org.junit.jupiter.api.Assertions;
11 import org.junit.jupiter.api.Test;
12
13 class TestCoche {
14
15     @Test
16     public void Carlos_Arroyo_Galán_test_al_crear_un_coche_su_velocidad_es_cero(){
17         Coche nuevoCoche = new Coche();
18         Assertions.assertEquals(0, nuevoCoche.velocidad);
19     }
20
21     @Test
22     public void Carlos_Arroyo_Galán_Test_al_acelerar_un_Coche_su_velocidad_aumenta(){
23         Coche nuevoCoche = new Coche();
24         nuevoCoche.acelerar(30);
25         Assertions.assertEquals(30,nuevoCoche.velocidad);
26     }
27
28     @Test
29     public void Carlos_Arroyo_Galán_Test_al_decelerar_un_Coche_su_velocidad_disminuye(){
30         Coche nuevoCoche = new Coche();
31         nuevoCoche.velocidad = 50;
32         nuevoCoche.decelerar(20);
33         Assertions.assertEquals(30,nuevoCoche.velocidad);
34     }
35
36     @Test
37     public void Carlos_Arroyo_Galán_Test_al_decelerar_un_Coche_su_velocidad_es_mayor_que_cero(){
38         Coche nuevoCoche = new Coche();
39         nuevoCoche.velocidad = 50;
40         nuevoCoche.decelerar(20);
41         Assertions.assertEquals(30,nuevoCoche.velocidad);
42     }
43 }
```

Por cada test que iba haciendo les hacia un commit, esto en el Git staging subiendo los unstaged changes a staged changes y les hacia un push commit



En esta captura podemos observar que estaba haciendo los commits de forma correcta

Git Repositories

repository [master] - C:\Users\Carlos\git\repository\git

Branches

Local

Remote Tracking

Tags

References

Remotes

Working Tree - C:\Users\Carlos\git\repository

TestCoches [master] - C:\Users\Carlos\eclipse-workspace\TestCoches

Branches

Local

Remote Tracking

Tags

References

Remotes

TestCoche.java

```
1 import org.junit.jupiter.api.Assertions;
2 import org.junit.jupiter.api.Test;
3
4 class TestCoche {
5
6     @Test
7     public void Carlos_Arroyo_Galán_test_al_crear_un_coche_su_velocidad_es_cero(){
8         Coche nuevoCoche = new Coche();
9         Assertions.assertEquals(0, nuevoCoche.velocidad);
10    }
11
12    @Test
13    public void Carlos_Arroyo_Galán_Test_al_acelerar_un_Coche_su_velocidad_aumenta(){
14        Coche nuevoCoche = new Coche();
15        nuevoCoche.acelerar(30);
16        Assertions.assertEquals(30,nuevoCoche.velocidad);
17    }
18
19    @Test
20    public void Carlos_Arroyo_Galán_Test_al_decelerar_un_Coche_su_velocidad_disminuye(){
21        Coche nuevoCoche = new Coche();
22        nuevoCoche.velocidad = 50;
23        nuevoCoche.decelerar(20);
24        Assertions.assertEquals(30,nuevoCoche.velocidad);
25    }
26
27    @Test
28    public void Carlos_Arroyo_Galán_Test_al_decelerar_un_Coche_su_velocidad_es_mayor_que_cero(){
29        Coche nuevoCoche = new Coche();
30        nuevoCoche.velocidad = 50;
31        nuevoCoche.decelerar(20);
32        Assertions.assertEquals(30,nuevoCoche.velocidad);
33    }
34 }
```

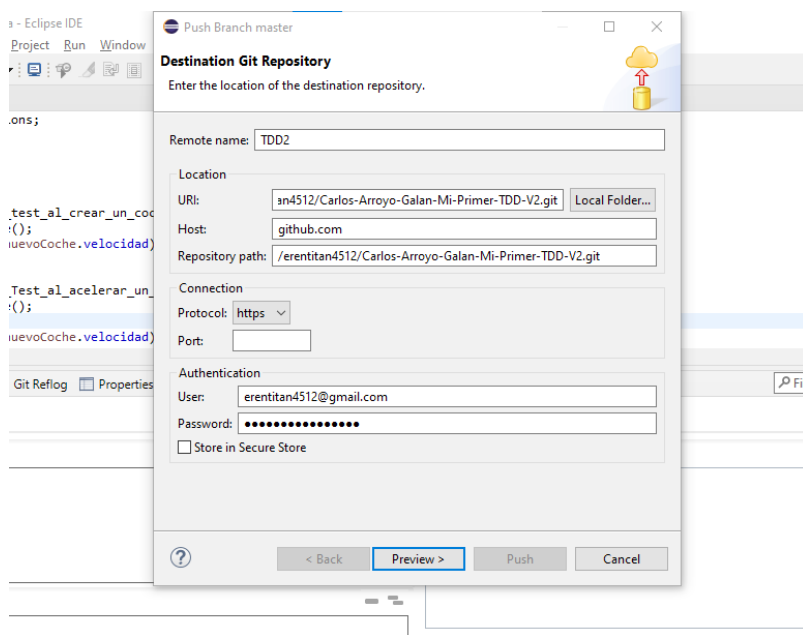
TestCoches

HEAD

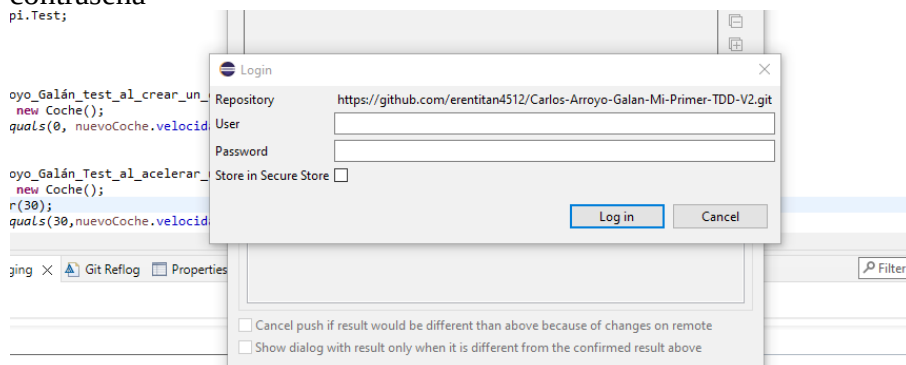
type filter text

Commit	Commit Message	Date	Reflog Message
35f5d58	Fifth Commit	2023-06-11 18:00:15	commit: Fifth Commit
fd7b129	Fourth Commit	2023-06-11 17:59:31	commit: Fourth Commit
07a0632	Third Commit	2023-06-11 17:58:18	commit: Third Commit
328d0d3	First Commit	2023-06-11 17:45:52	commit (initial): First Commit

Al intentar enlazar el proyecto en “Push HEAD” con mi repositorio de github todo estaba funcionando correctamente, hasta que me ha empezado a dar problemas con el usuario y contraseña,



Cuando pulso preview solo me aparece esta pestaña para que introduzca el usuario (gmail) y la contraseña



Habiendo comprobado que tanto el usuario como la contraseña son correctas y que me sigue dando el mismo error me he visto en la obligación de entregarlo de esta forma, si es necesario grabaré un video demostrando este error.

Carlos Arroyo Galán DAM1K