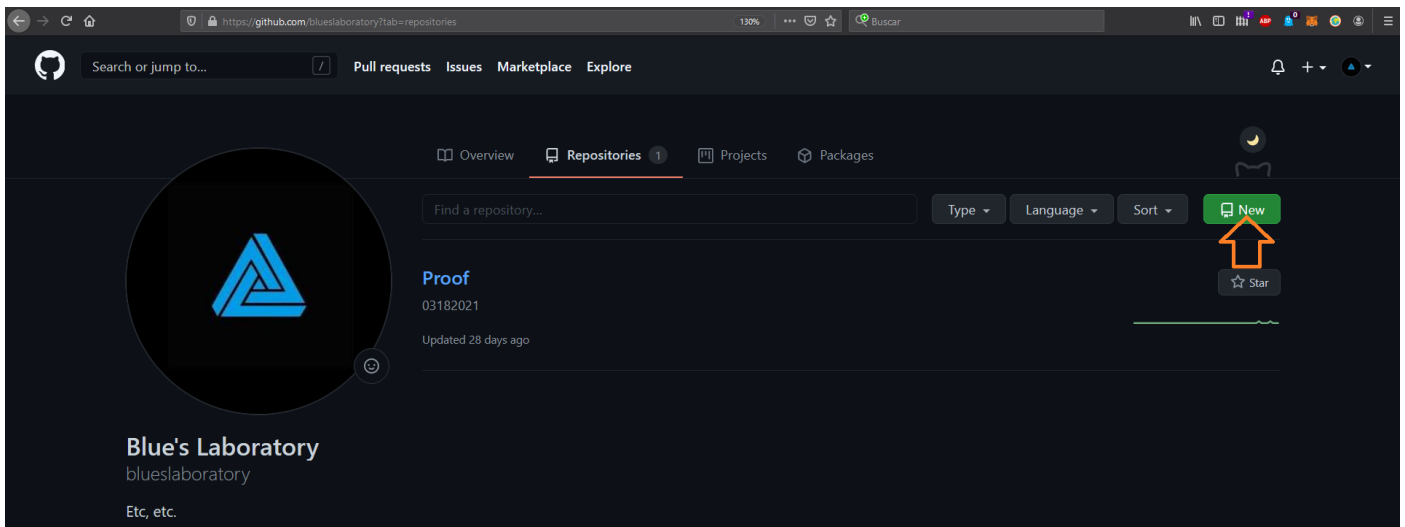


## Actividad 4.1

Alumno: Alejandro Díez – DA1D1E

Crea un repositorio remoto público en GitHub con tunombre\_actividad41.



### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*

blueslaboratory

Repository name \*

Alejandro\_actividad41

Great repository names are short and memorable. Need inspiration? How about [bookish-doodle?](#)

Description (optional)

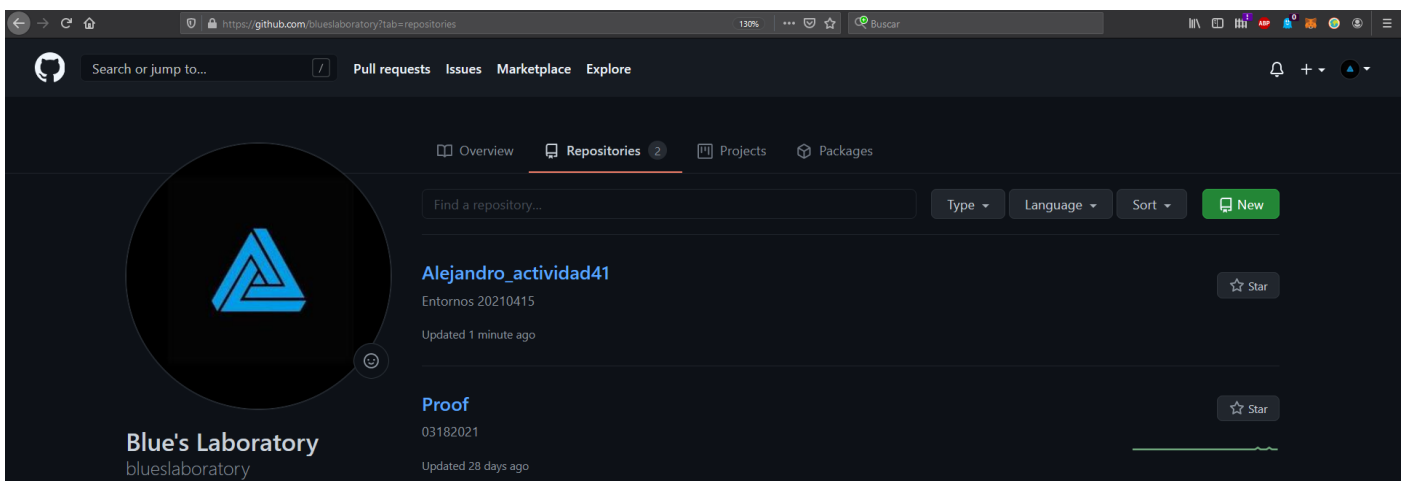
Entornos 20210415

☒ Public

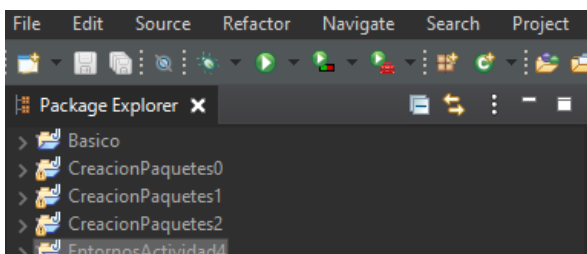
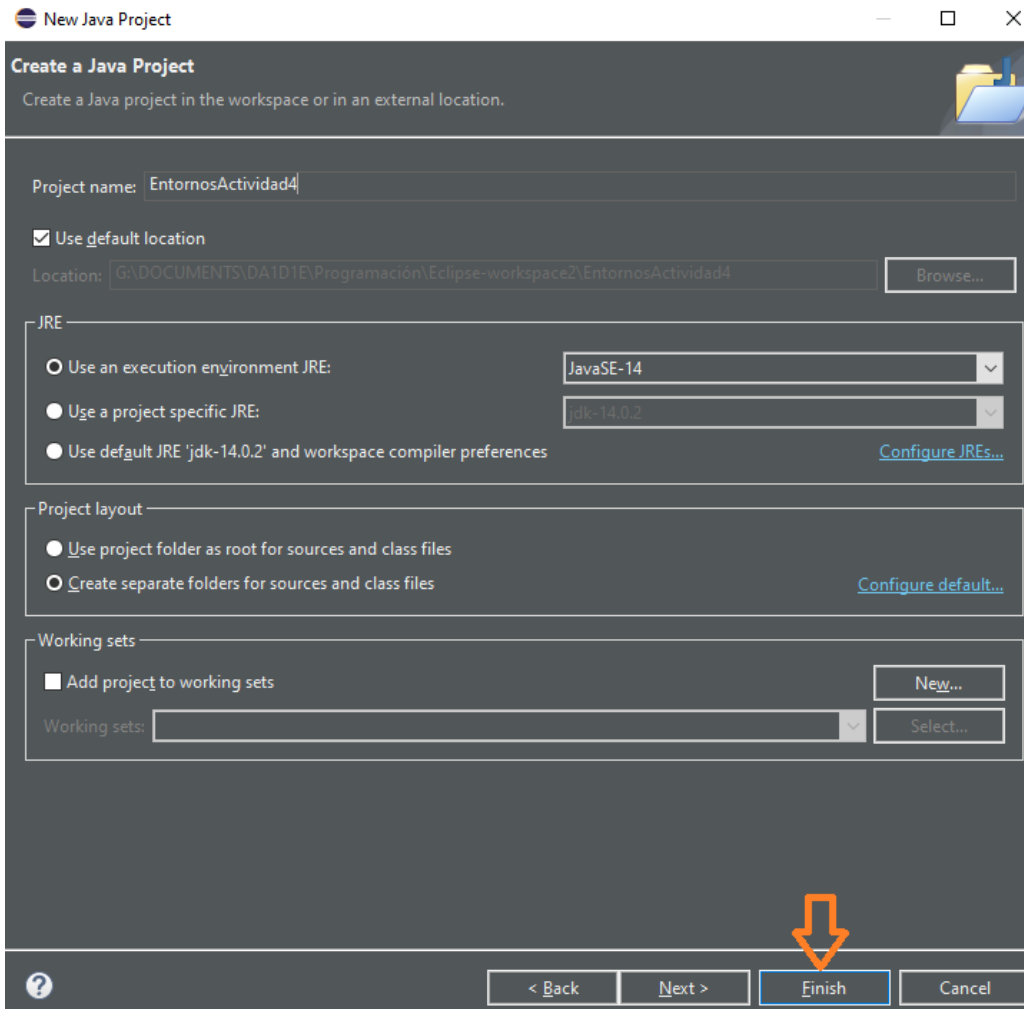
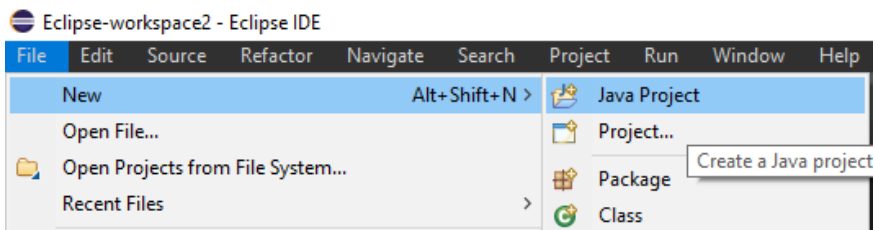
Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

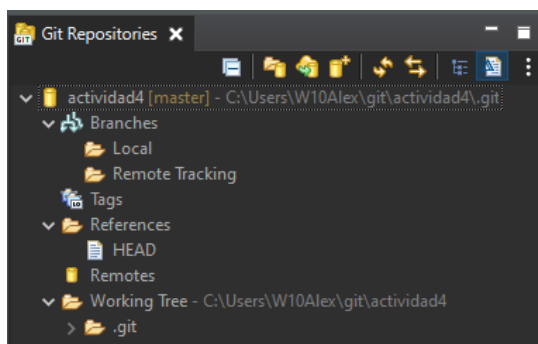
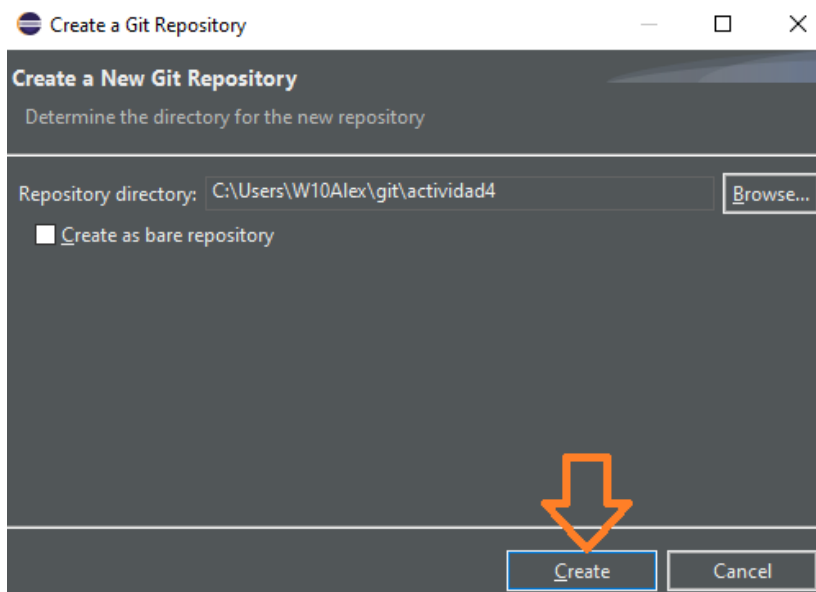
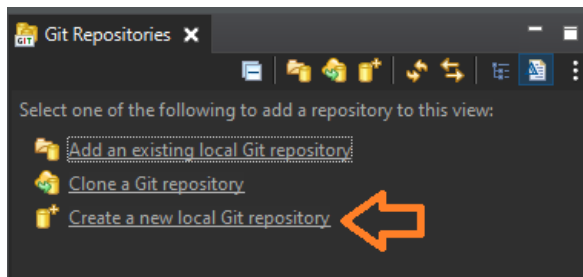
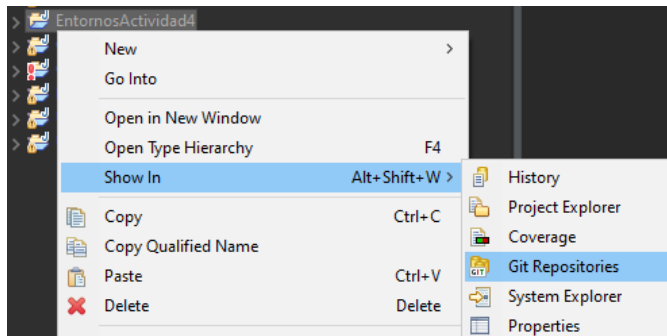


En local, crea el proyecto actividad4.

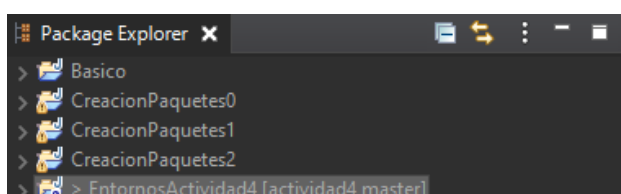
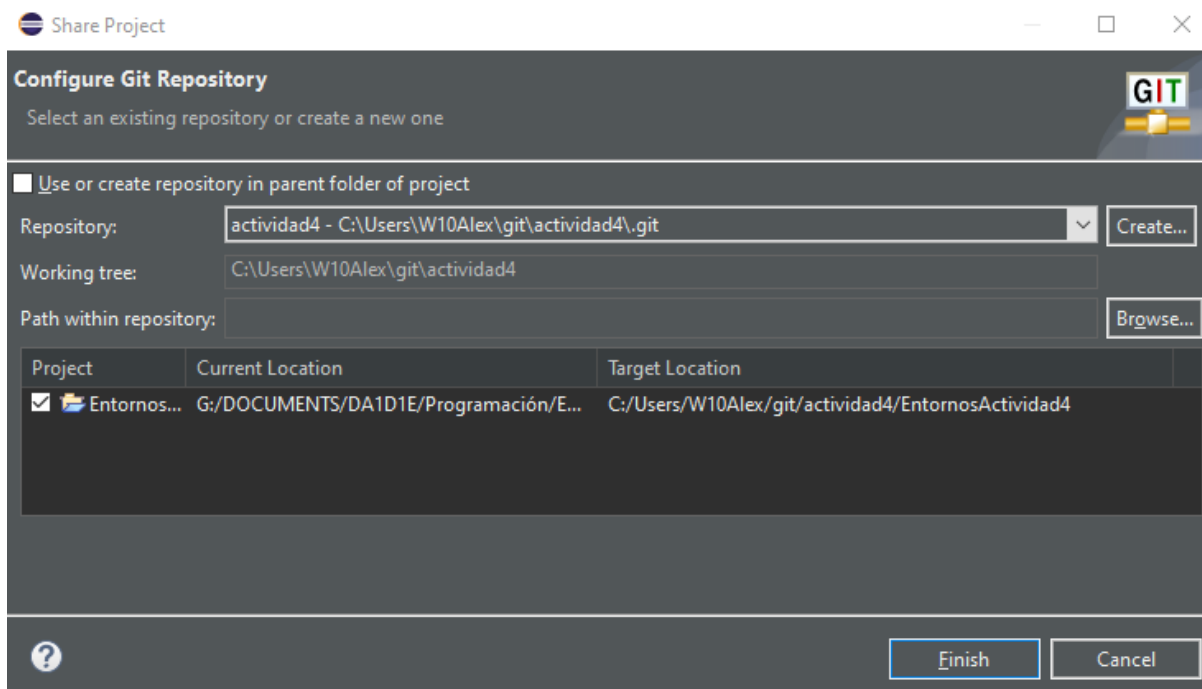
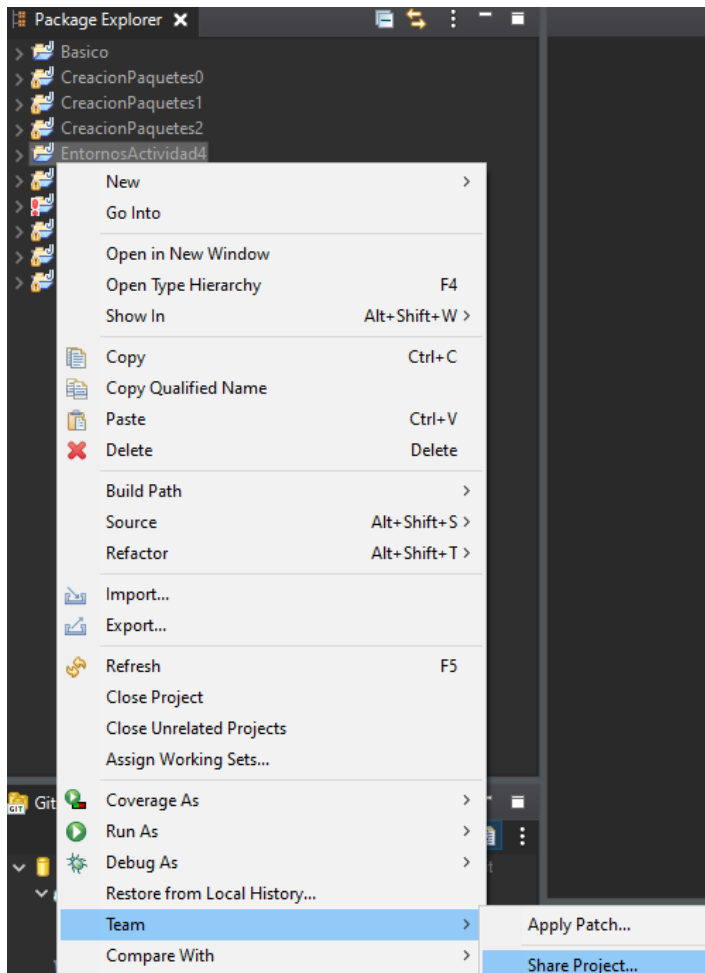


Luego crea el repositorio Git. Lo llamaremos actividad4. Y realizamos las siguientes operaciones:

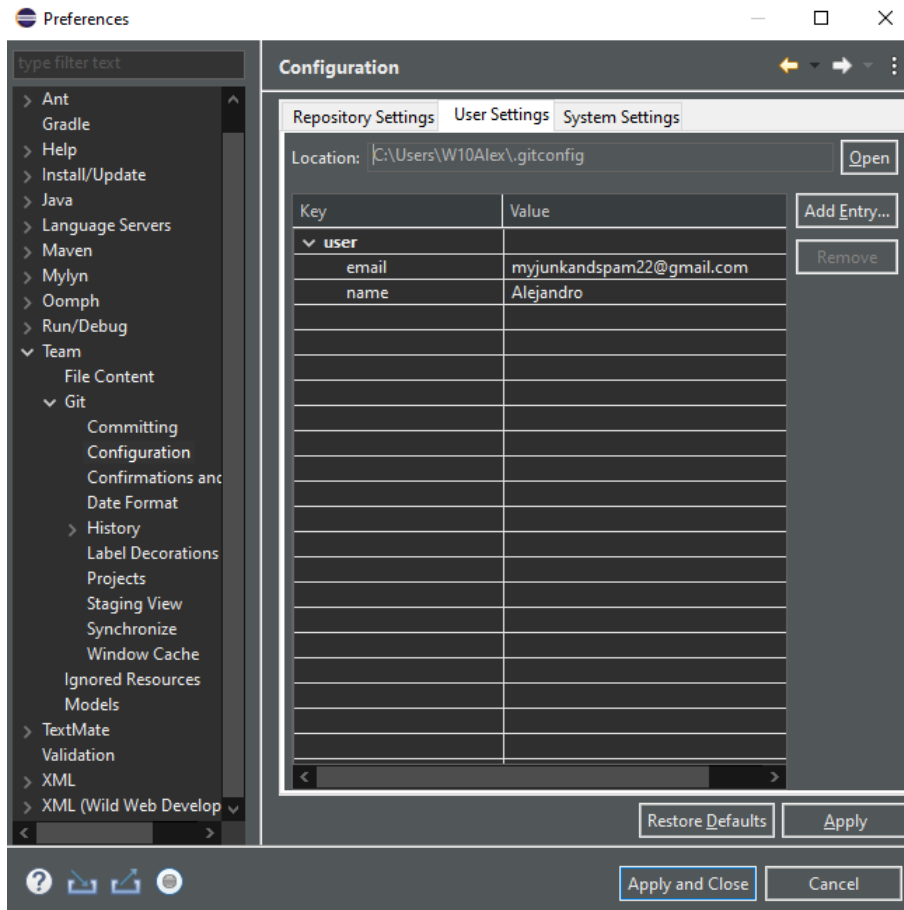
Para crear el repositorio Git, botón derecho sobre package explorer – show in – Git repositories



- Conecta el proyecto al repositorio local y valida el proyecto. Haz lo mismo en remoto.
- Añade una línea de visualización en cada método, pero valida una a una. Debe haber 3 commits. Y sincroniza en remoto.



Window -> Preferences -> Team -> Git -> Configuration:

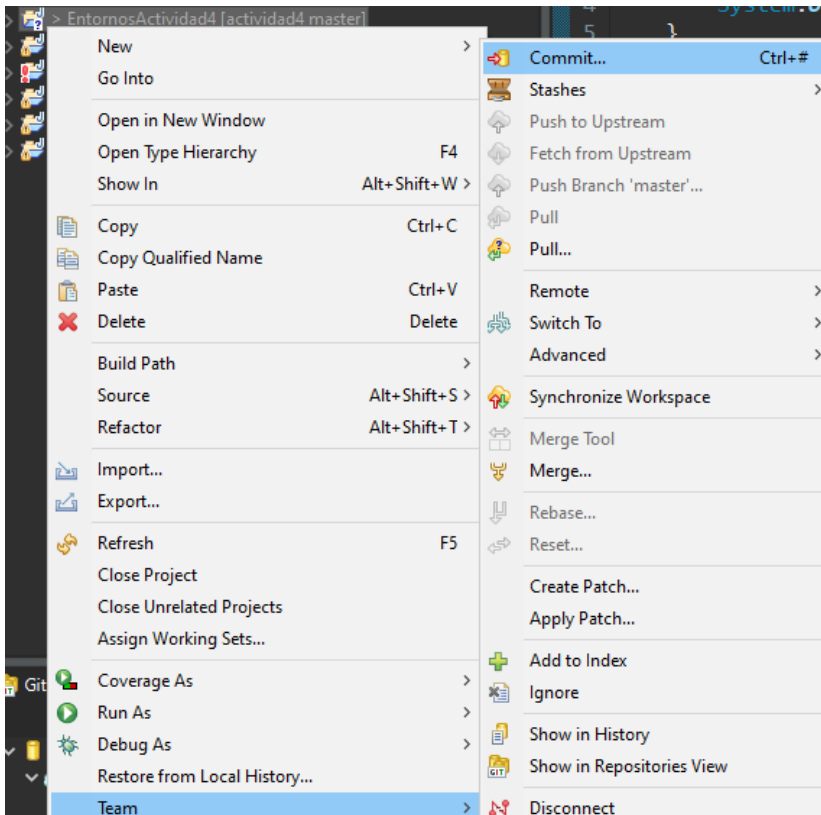


Validar los cambios en el repositorio:

Hemos creado la clase Hi.java con un main:

```
Hi.java x
1 public class Hi {
2     public static void main(String[] args) {
3         // TODO Auto-generated method stub
4         System.out.println("Hi Java.git");
5     }
6 }
```

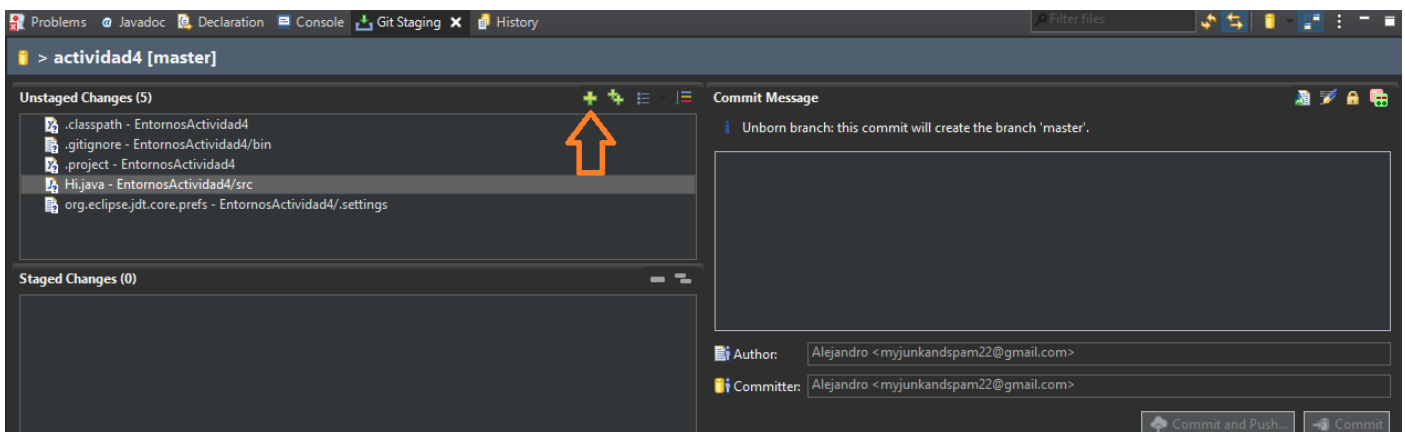
Para preparar y validar los cambios, ejecutamos la acción Commit que abrirá la pestaña de Git Staging.



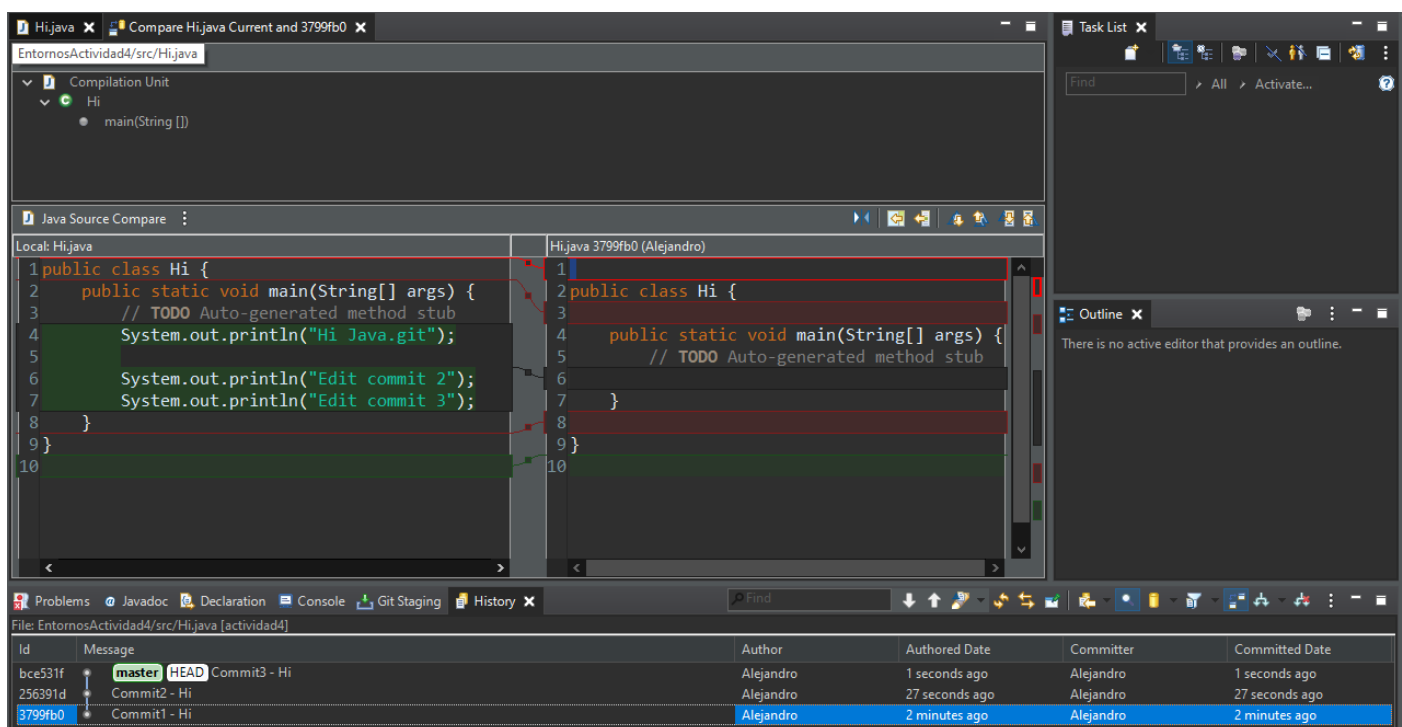
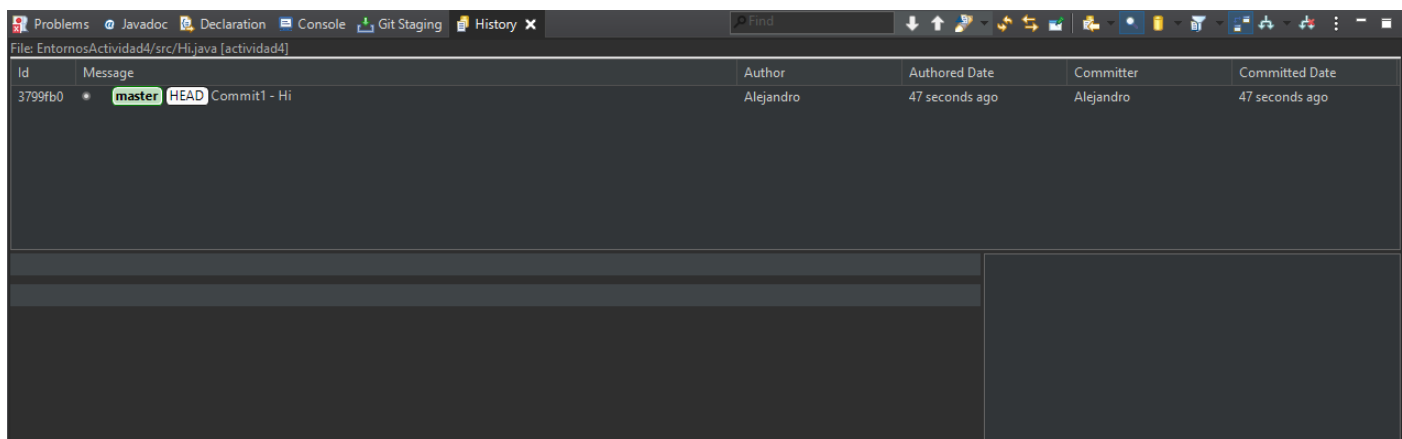
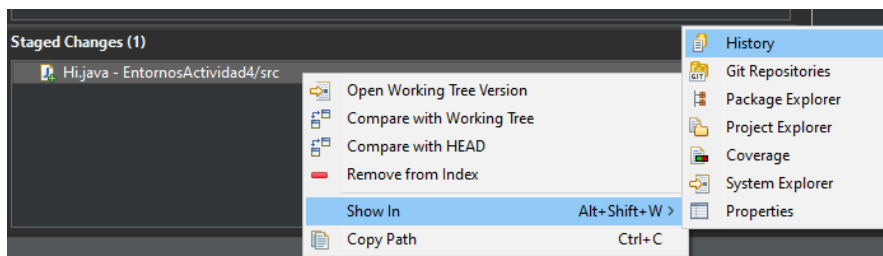
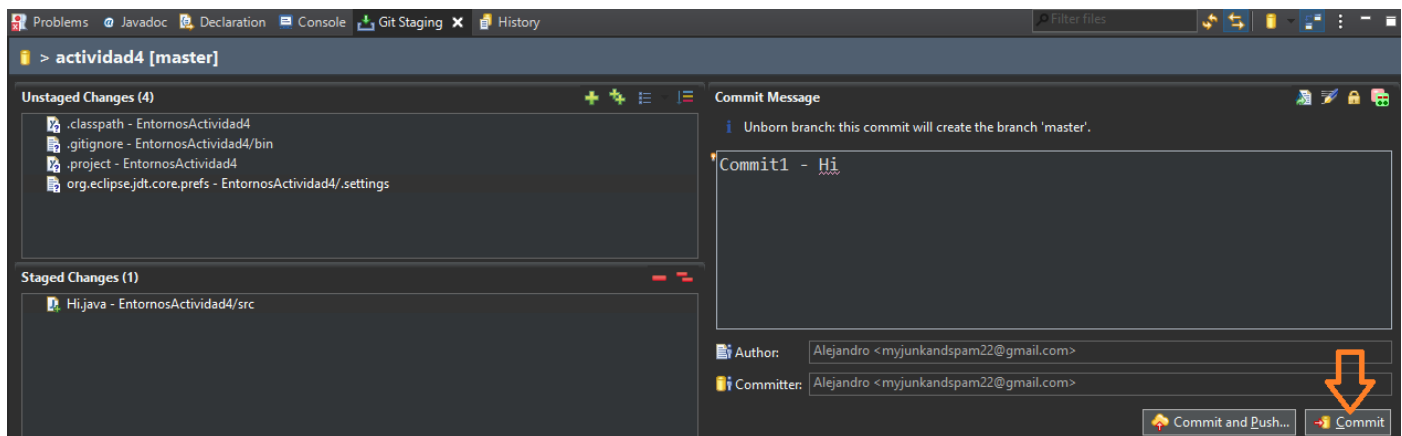
En Git Staging se muestran los cambios que se han detectado y que no se han validado en el repositorio.

Área Unstaged Changes = archivos en “working”

Área Staged Changes = archivos en staged



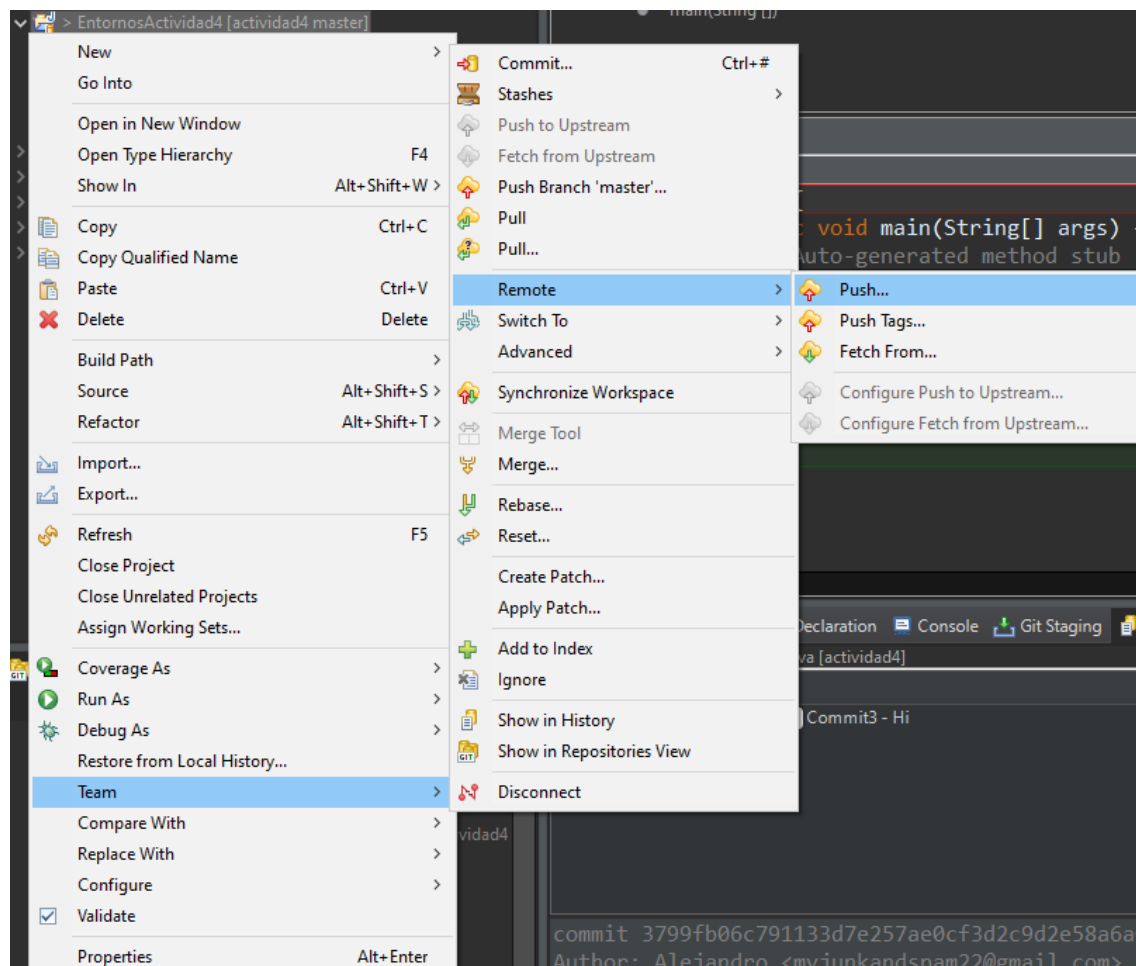
Podemos añadir todos los archivos a staged a la vez, será como hacer un add; pero tal y como se nos pide, vamos a validar solo la clase Hi.java, hacemos el commit, la editamos, hacemos otro commit y así hasta tener 3 commits.



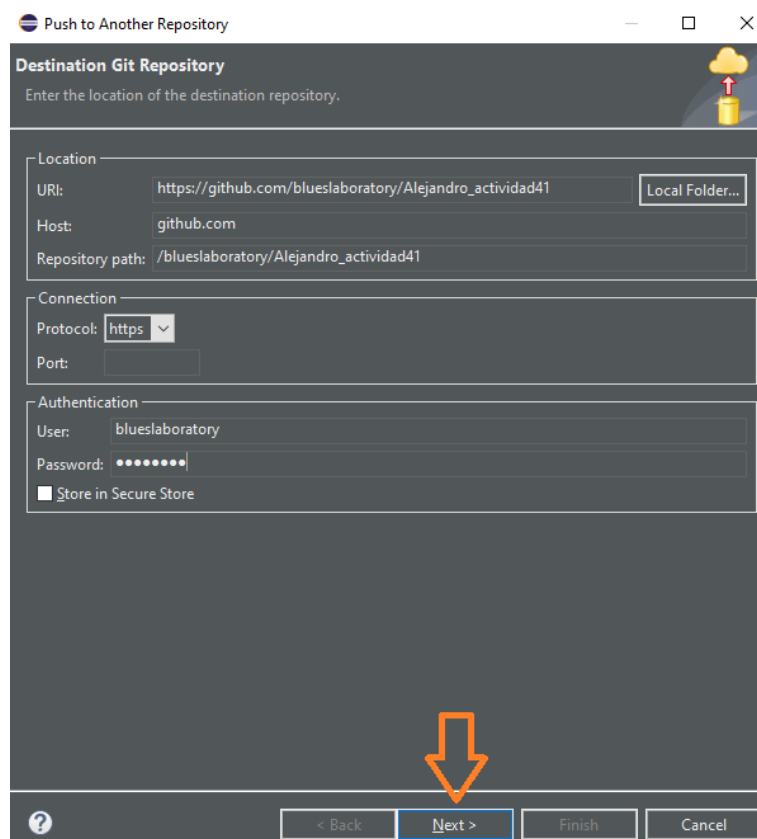
Aquí vemos la comparación de los cambios ocurridos en ese momento de validación. A la izquierda la versión actual y a la derecha el commit seleccionado: commit1 – Hi.

Ahora vamos a conectarnos al repositorio remoto, el que creamos al principio: Alejandro\_actividad41

Para conectarnos a él, sobre el proyecto local seleccionamos Team – remote – push:



Escribimos la URL del repositorio y los datos de autenticación:





Push to: [https://github.com/blueslaboratory/Alejandro\\_actividad41](https://github.com/blueslaboratory/Alejandro_actividad41)

### Push Ref Specifications

Select refs to push.

Add create/update specification

Source ref:  Destination ref:

Specifications for push

Mode	Source Ref	Destination Ref	Force Update	Remove
<input checked="" type="checkbox"/> Update	refs/heads/master	refs/heads/master	<input type="checkbox"/>	<input type="button" value="Remove"/>

Después pulsamos finish y nos vamos al github donde podemos ver los commits que hemos hecho:

master 1 branch 0 tags Go to file Add file Code

**blueslaboratory** Commit3 - Hi bce531f 21 minutes ago **3 commits**

EntornosActividad4/src Commit3 - Hi 21 minutes ago

Help people interested in this repository understand your project by adding a README.

master

Commits on Apr 15, 2021

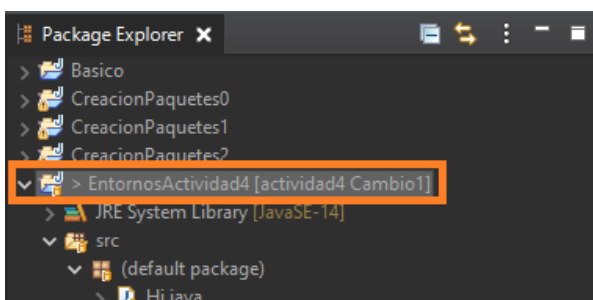
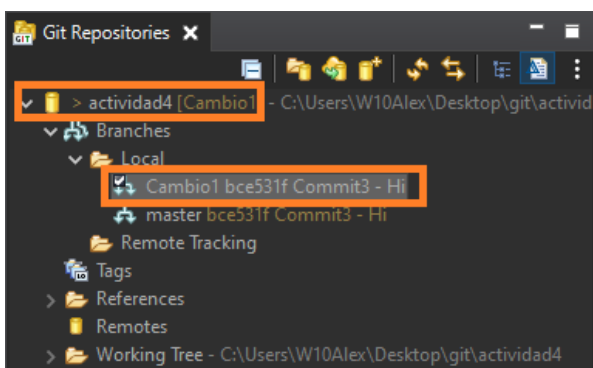
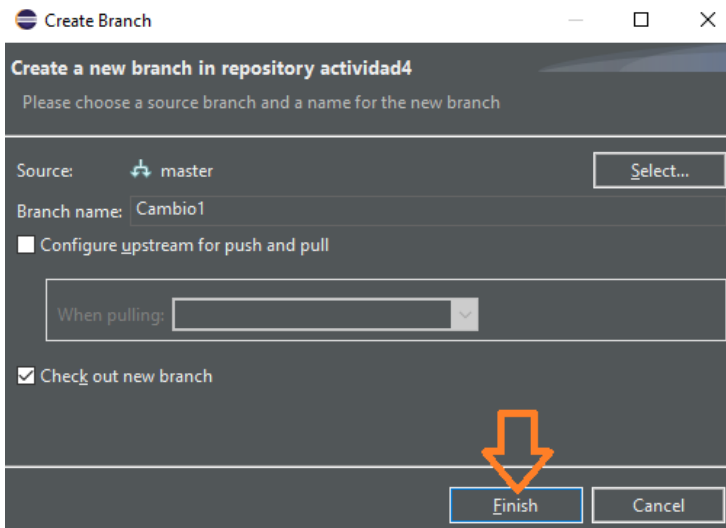
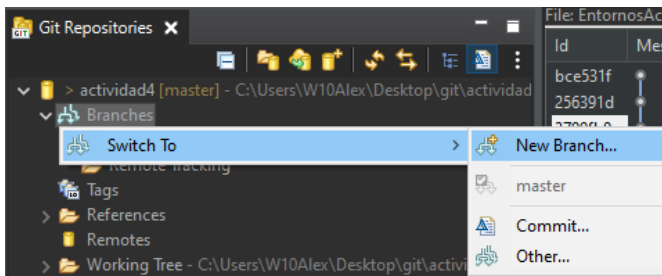
**Commit3 - Hi**  
blueslaboratory committed 21 minutes ago

**Commit2 - Hi**  
blueslaboratory committed 21 minutes ago

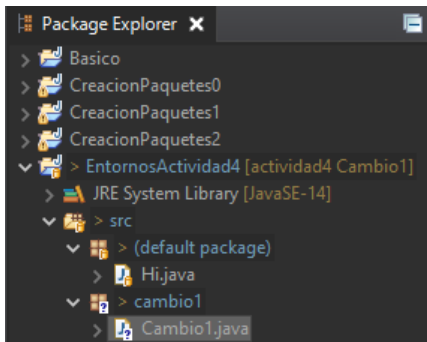
**Commit1 - Hi**  
blueslaboratory committed 23 minutes ago

- Crea la rama cambios. Muévete a ella, y añade una clase de java nueva con una línea. Y valida en local. Añade otra clase nueva y valida en local. Debe hacer dos commits más en la rama.

Para crear la rama, en la vista “Git Repositories”, abrimos el menú contextual del nodo Branches y seleccionamos Switch to → new Branch:

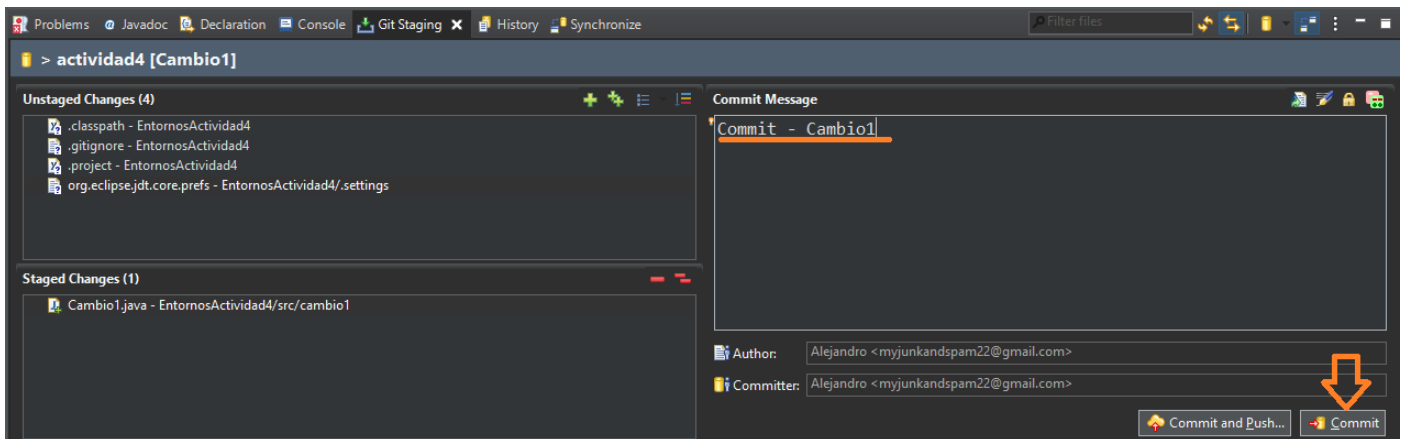
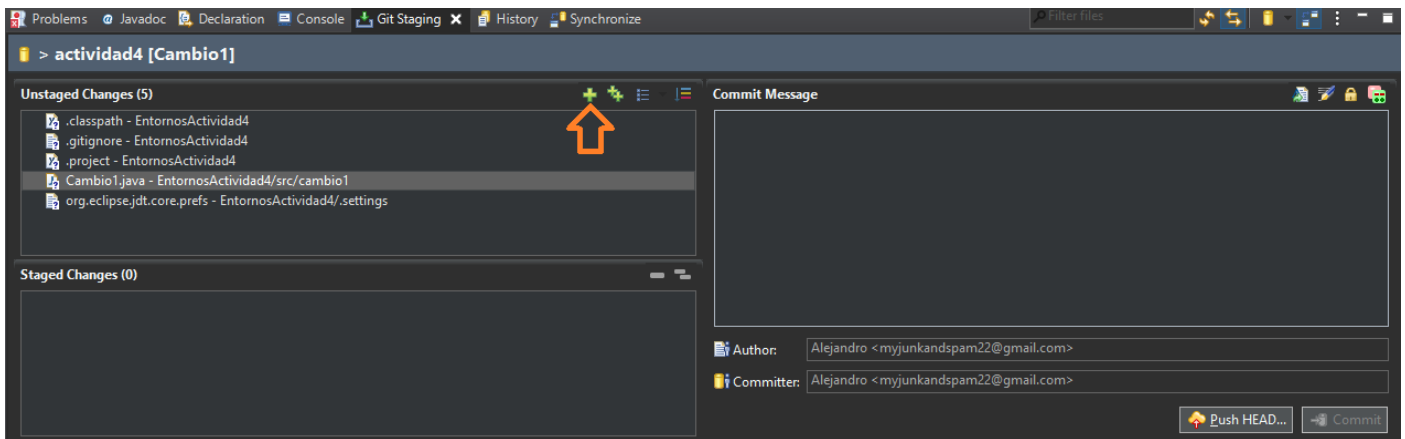


Ahora el proyecto apunta a la rama Cambio1. Vamos a crear en el proyecto el paquete “cambio1” y la clase Cambio1:

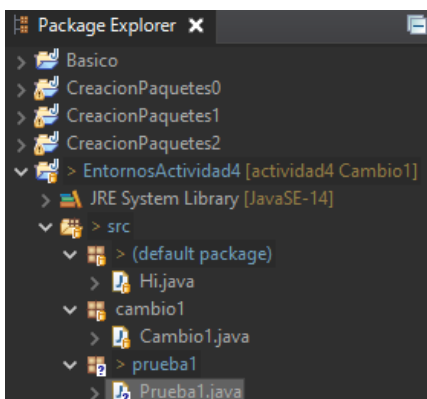


```
1 package cambio1;
2
3 public class Cambio1 {
4     public static void main(String[] args) {
5         System.out.println("Pruebas Git. Clase cambio1");
6     }
7 }
```

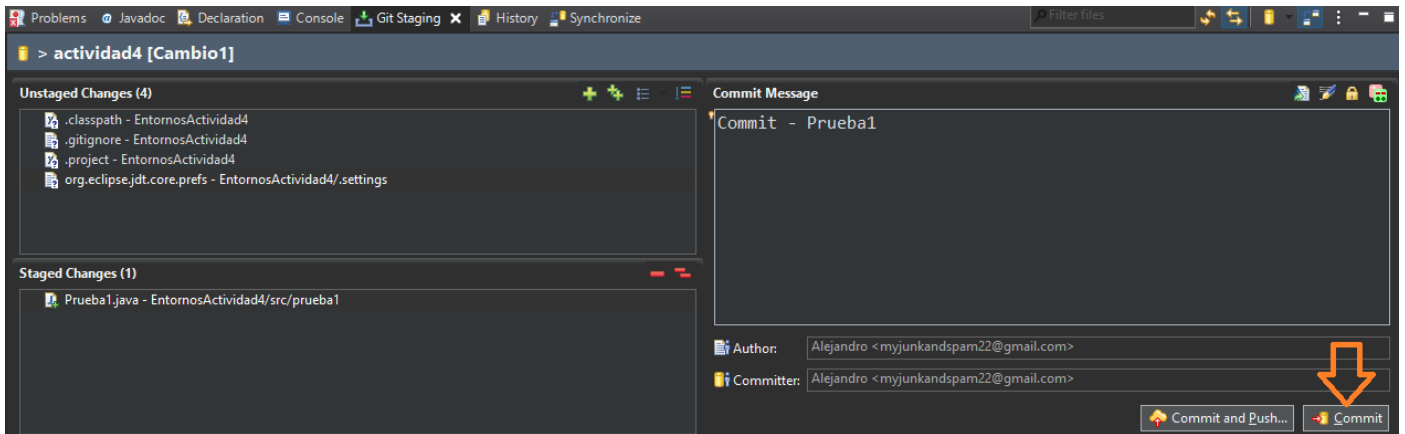
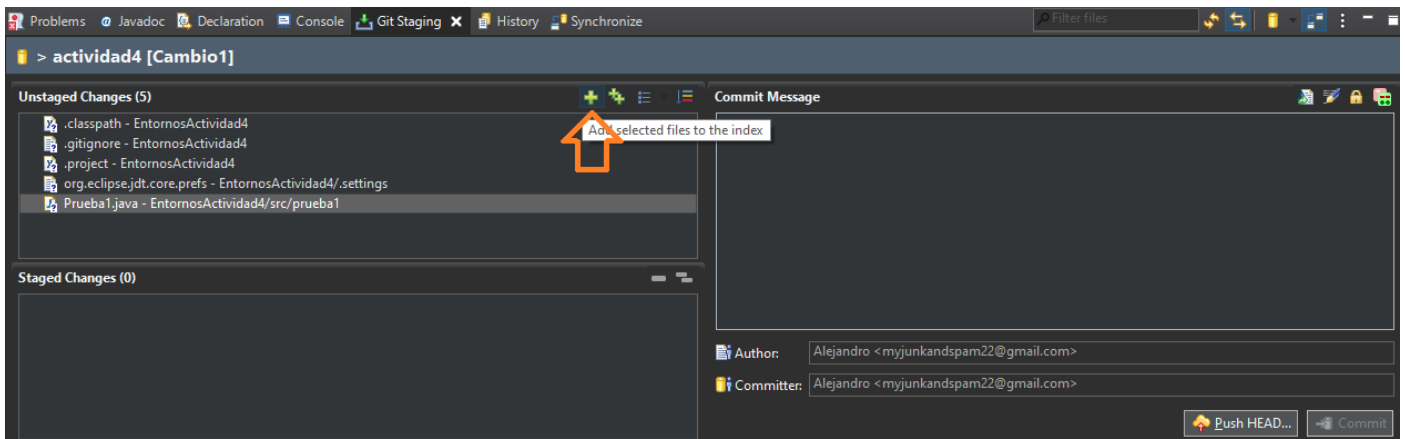
Validamos en local:



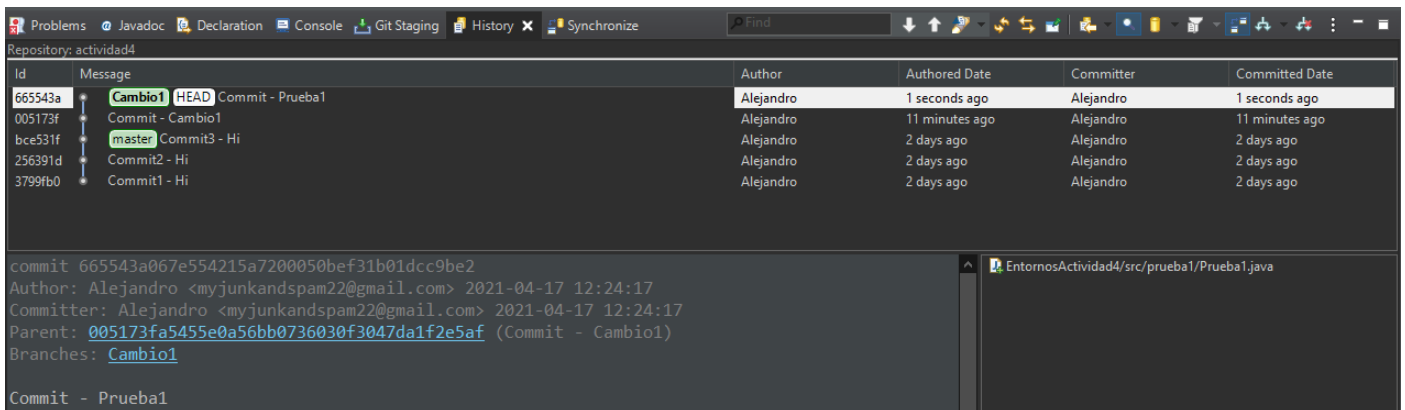
Creamos otro paquete en el proyecto, “prueba1”, añadimos otra clase, “Prueba1”, al paquete prueba1 y validamos en local otra vez:



```
1 package prueba1;
2
3 public class Prueba1 {
4     public static void main(String[] args) {
5         System.out.println("Pruebas Git. Clase Prueba1");
6     }
7 }
```

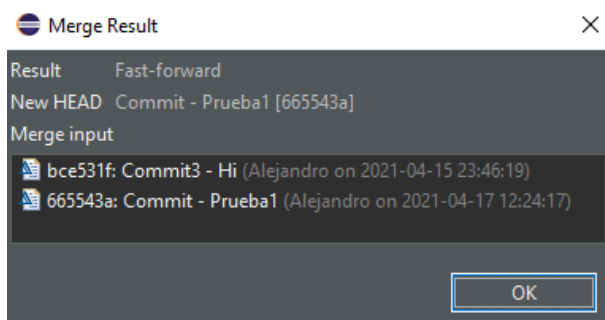
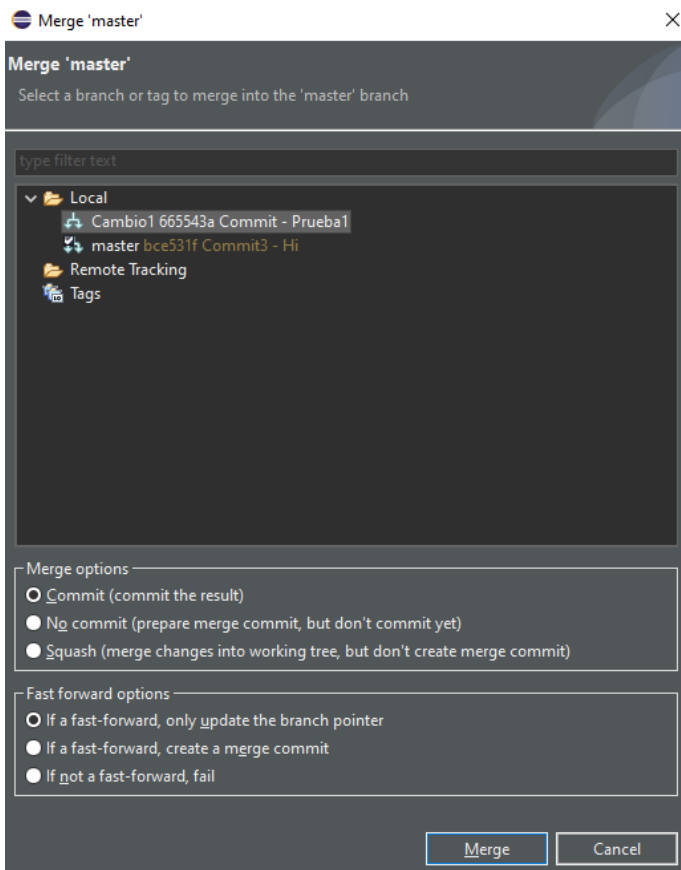
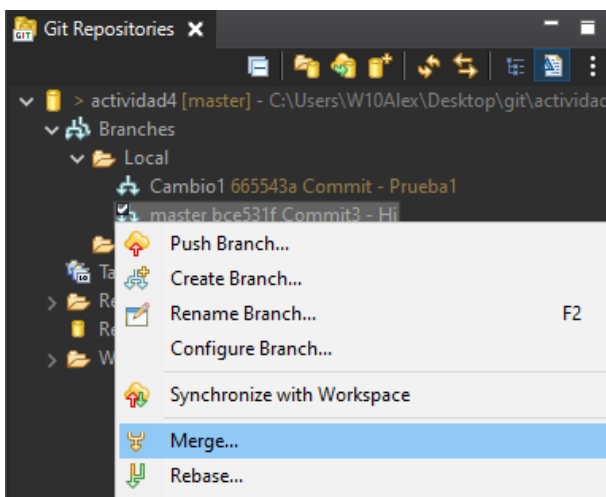
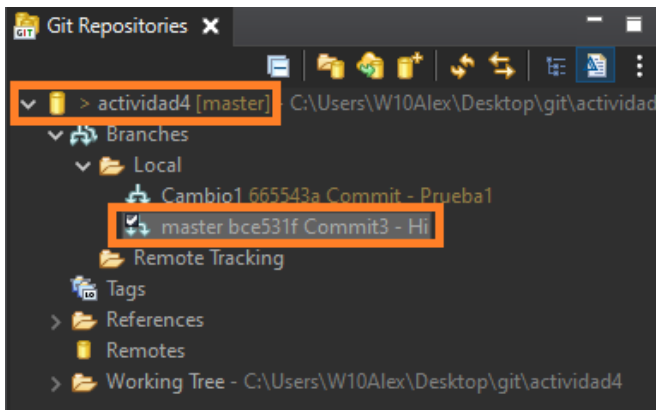


Como podemos ver a continuación en la pestaña de “History” hay 2 commits más bajo la rama Cambio1 que parte de la rama principal master; la rama cambio1 aparece con el apuntador HEAD, que indica que es la versión más reciente:

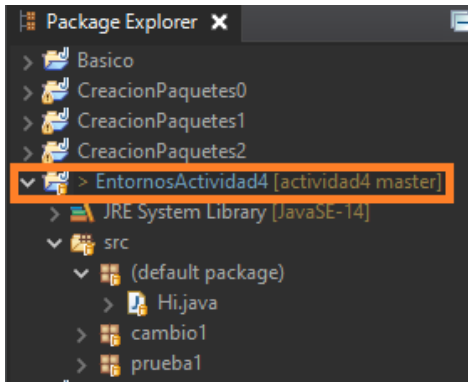


- Fusiona la rama creada al master local e indica lo que ha ocurrido.

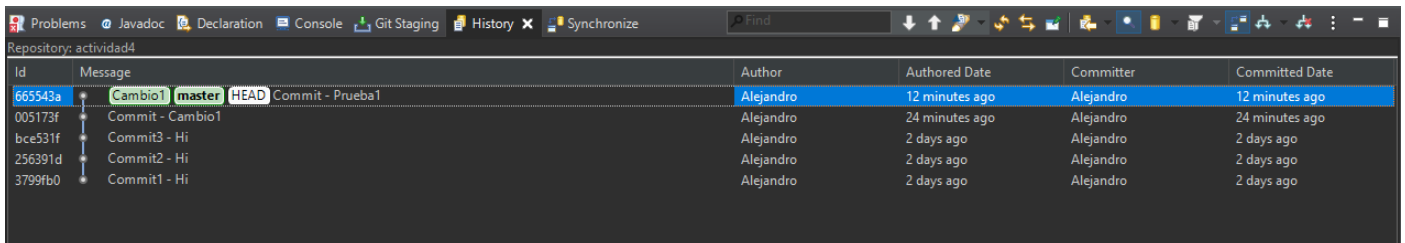
Desde “Git Repositories” nos movemos a la rama master y hacemos doble clic sobre ella, lo que provoca el Check out ‘master’ (que se seleccione), luego abrimos el menú contextual de master y seleccionamos “merge”. En la ventana que aparece se selecciona la rama a fusionar, hacemos doble clic sobre “Cambio1” o pulsamos el botón merge:



Comprobamos que ahora vemos los cambios hechos en la rama:



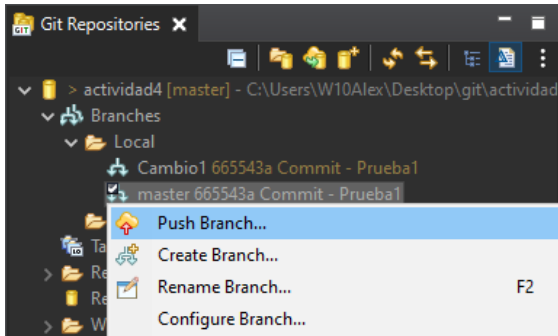
Lo que ha ocurrido es que la rama que habíamos hecho que partía de master, cambio1, se ha “fusionado (merge)” con la rama master” y ahora master contiene esa rama:



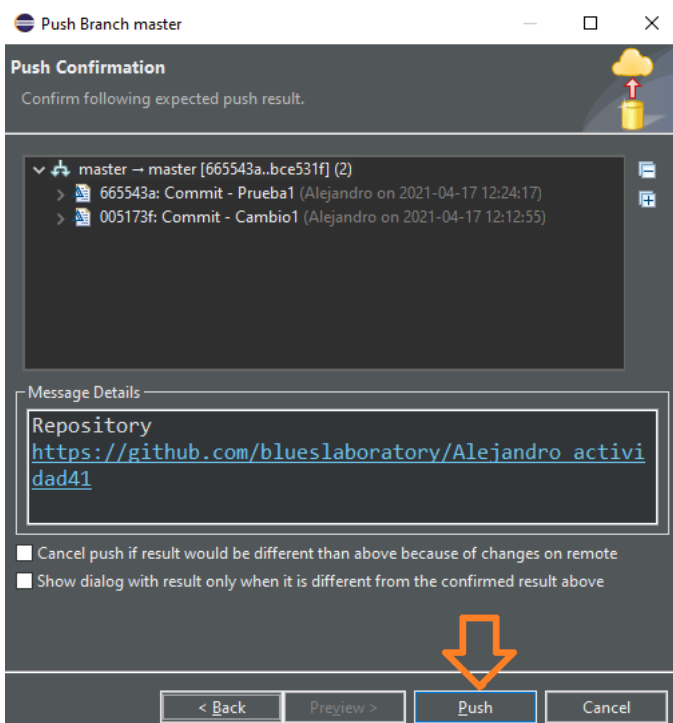
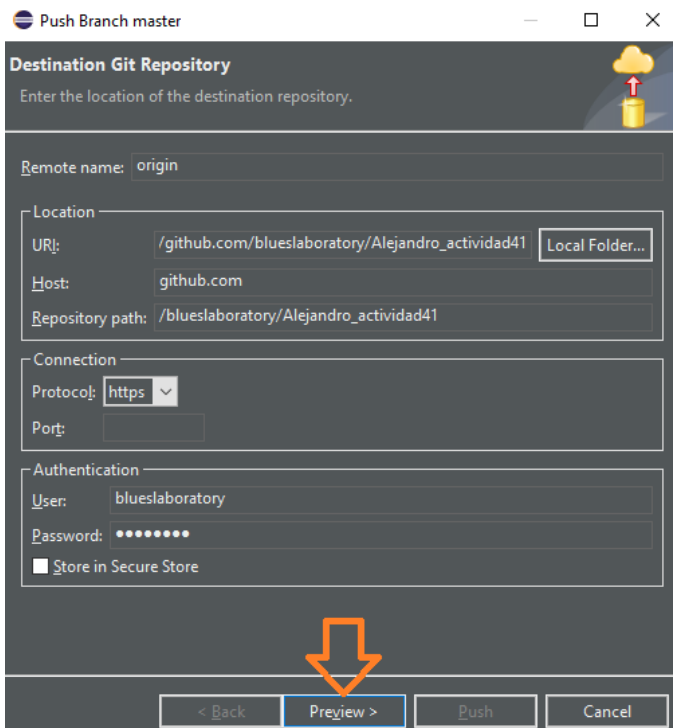
Id	Message	Author	Authored Date	Committer	Committed Date
665543a	Cambio1 <b>master</b> HEAD Commit - Prueba1	Alejandro	12 minutes ago	Alejandro	12 minutes ago
005173f	Commit - Cambio1	Alejandro	24 minutes ago	Alejandro	24 minutes ago
bce531f	Commit3 - Hi	Alejandro	2 days ago	Alejandro	2 days ago
256391d	Commit2 - Hi	Alejandro	2 days ago	Alejandro	2 days ago
3799fb0	Commit1 - Hi	Alejandro	2 days ago	Alejandro	2 days ago

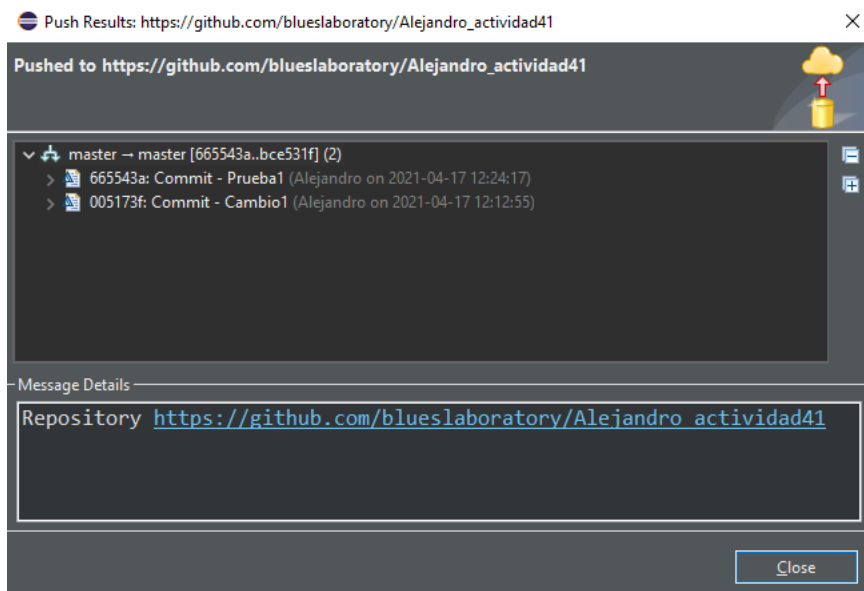
## - Sincroniza con el remoto.

Vamos a validar la rama master en el repositorio remoto.

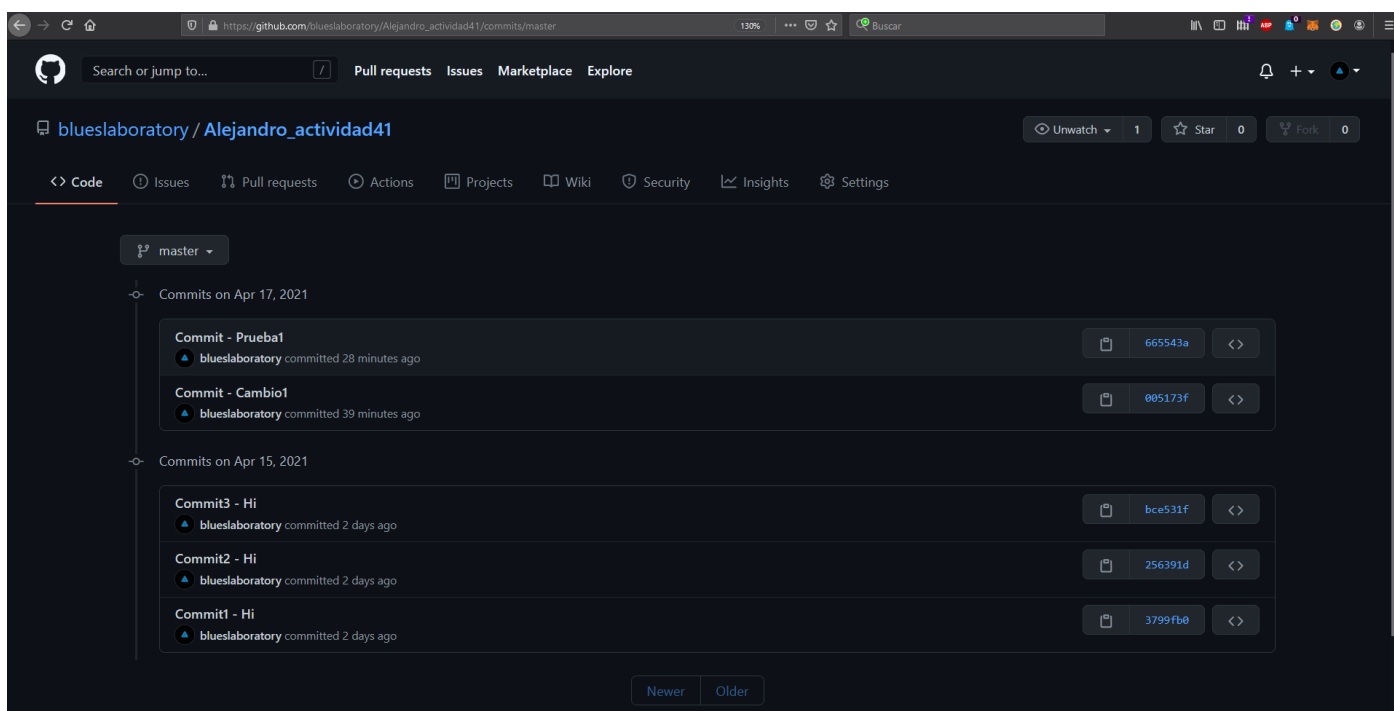


Seleccionamos Preview hasta que nos aparezca Push y entonces seleccionamos Push:





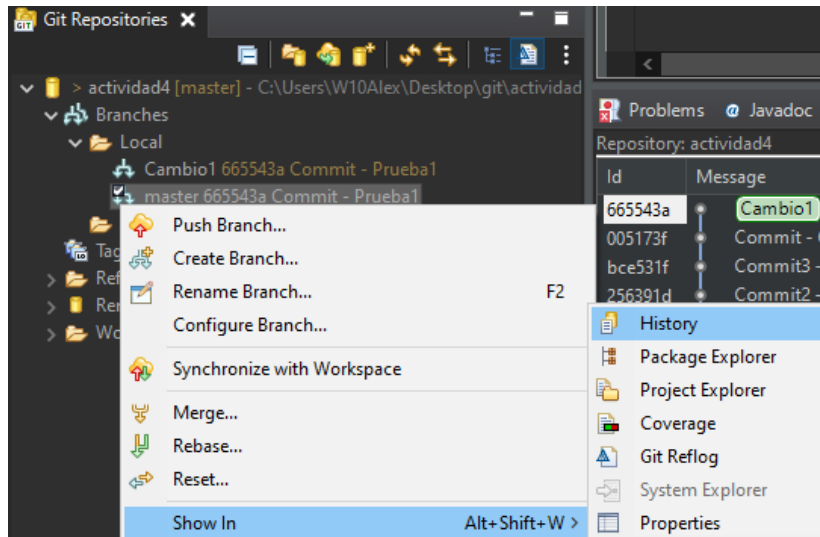
Parece que el push, ha sido un éxito, lo comprobamos en el github:



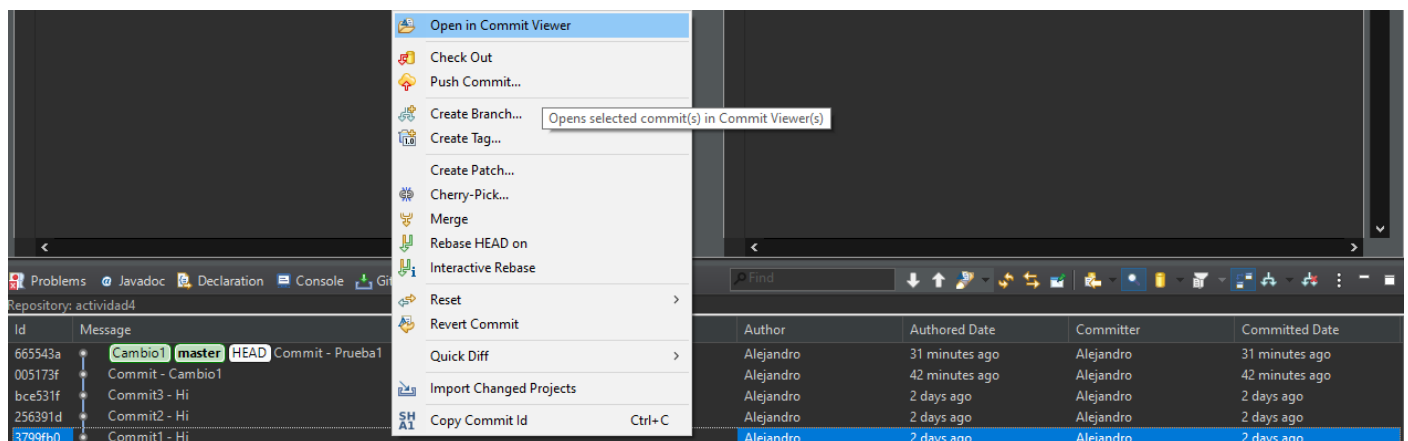


- Compara los cambios realizados en la clase principal en los distintos momentos de validación

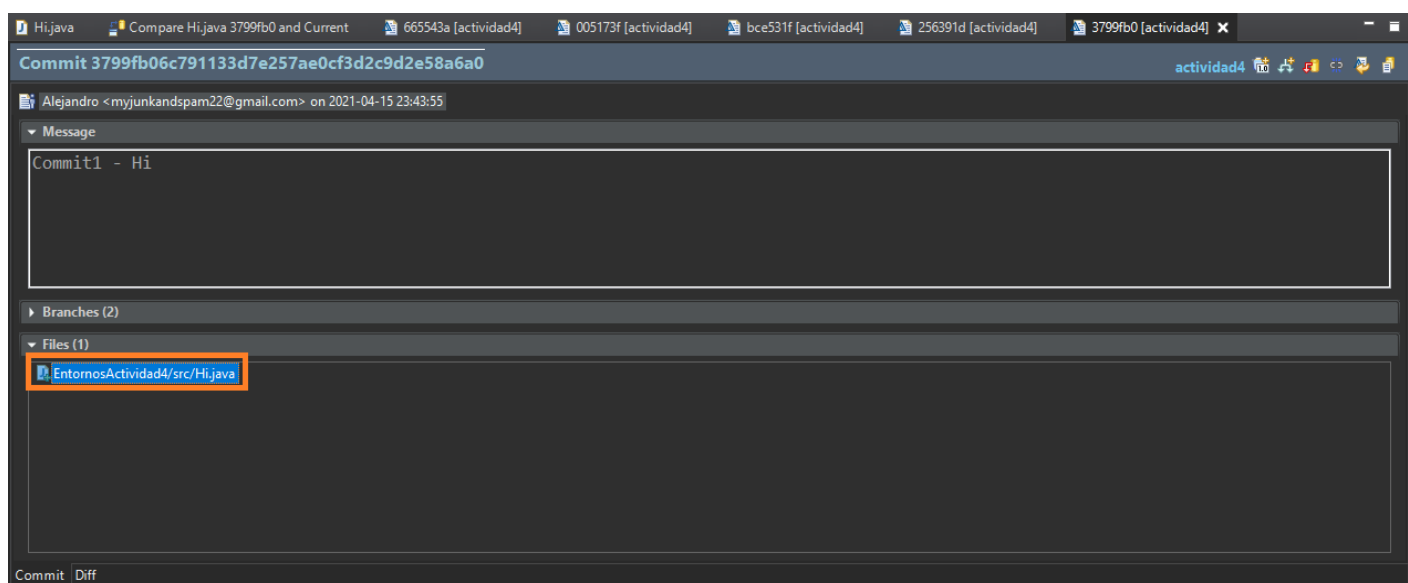
Abrimos el menú contextual de la rama master y seleccionamos “Show in” -> “History”.



Ya en la pestaña de “History” abrimos el menú contextual de los distintos commits y seleccionamos “Open in Commit Viewer”:



Doble clic en el archivo de Files



Y repetimos esto para cada uno de los commits, obteniendo las comparaciones de los cambios:

## Commit1:

```
1 public class Hi {
2
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6
7     }
8
9 }
```

## Commit2:

```
1 public class Hi {
2     public static void main(String[] args) {
3         // TODO Auto-generated method stub
4         System.out.println("Hi Java.git");
5         System.out.println("Edit commit 2");
6     }
7 }
8
9
```

## Commit3:

```
1 public class Hi {
2     public static void main(String[] args) {
3         // TODO Auto-generated method stub
4         System.out.println("Hi Java.git");
5
6         System.out.println("Edit commit 2");
7         System.out.println("Edit commit 3");
8     }
9 }
```

## Commit – Cambio1:

```
1 package cambio1;
2
3 public class Cambio1 {
4     public static void main(String[] args) {
5         System.out.println("Pruebas Git. Clase cambio1");
6     }
7 }
```

## Commit – Prueba1:

```
1 package prueba1;
2
3 public class Prueba1 {
4     public static void main(String[] args) {
5         System.out.println("Pruebas Git. Clase Prueba1");
6     }
7 }
```