



PRÁCTICA 2

GITHUB

MARIO JIMÉNEZ MARSET

ÍNDICE

1. ENUNCIADO - OBJETIVOS	3
2. DESARROLLO – PROCEDIMIENTOS.....	3
3. WEBGRAFÍA	12

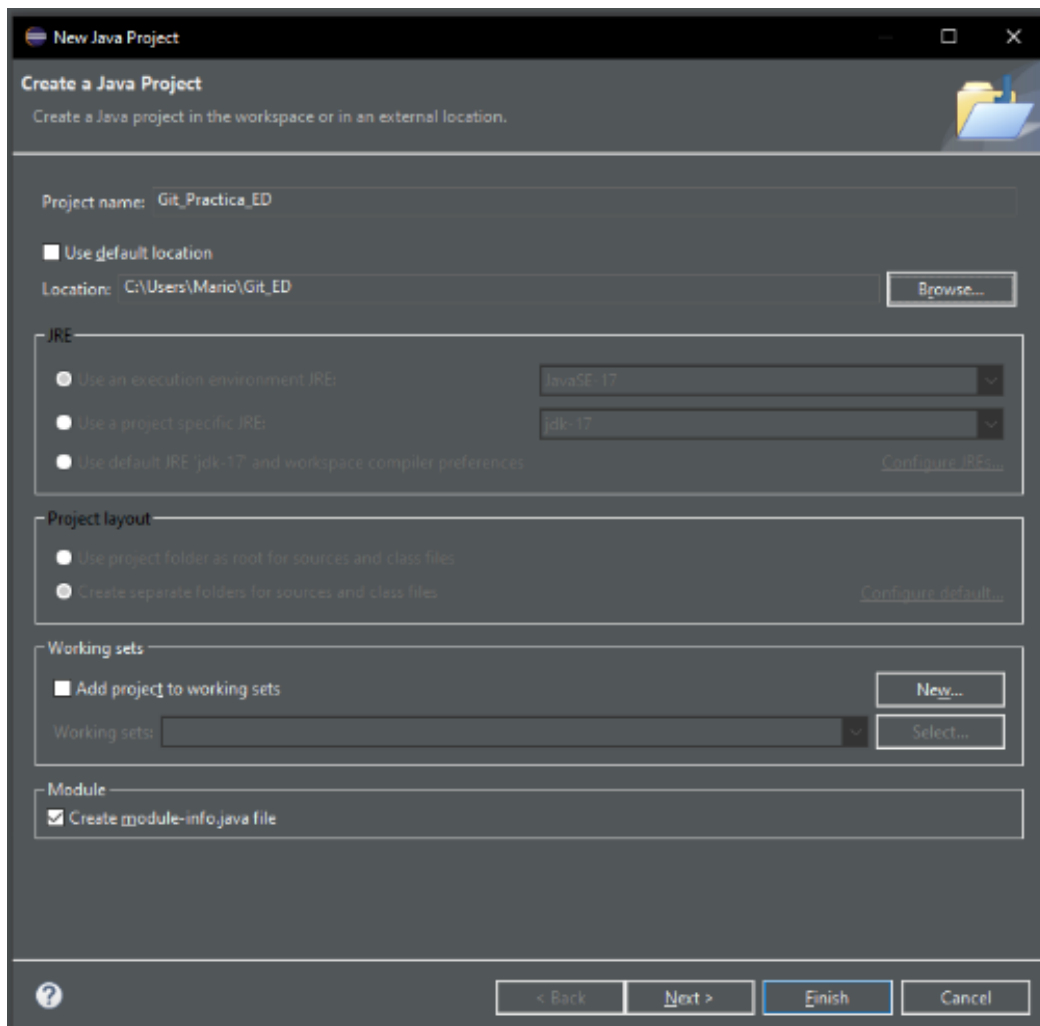
1. ENUNCIADO - OBJETIVOS

En esta práctica se pedía mostrar a través de capturas de pantalla el proceso de creación (desde Eclipse) para el control de versiones con el programa Git Hub.

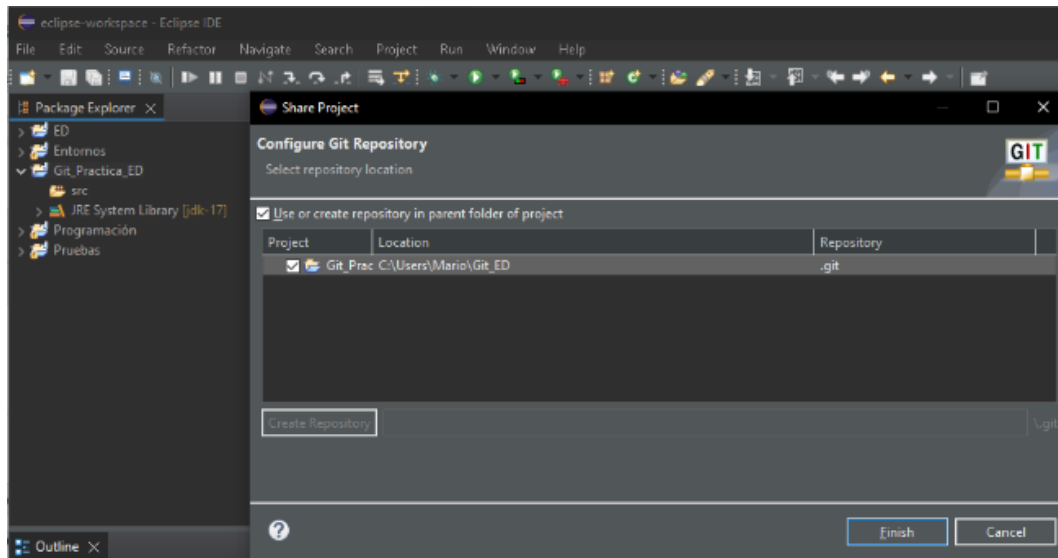
Se requerían los siguientes pasos:

- 1) Crear un proyecto Java y configurarlo para que esté bajo control de versiones.
- 2) Incluir un paquete y varias clases (archivos *.java) y realizar tres commits distintos.
- 3) Mostrar el histórico de los commits.
- 4) Hacer una restauración del proyecto en el commit 2.
- 5) Hacer un push del proyecto (en ese commit 2).
- 6) Mostrar en GitHub el estado del repositorio que debe incluir el proyecto en ese commit 2.

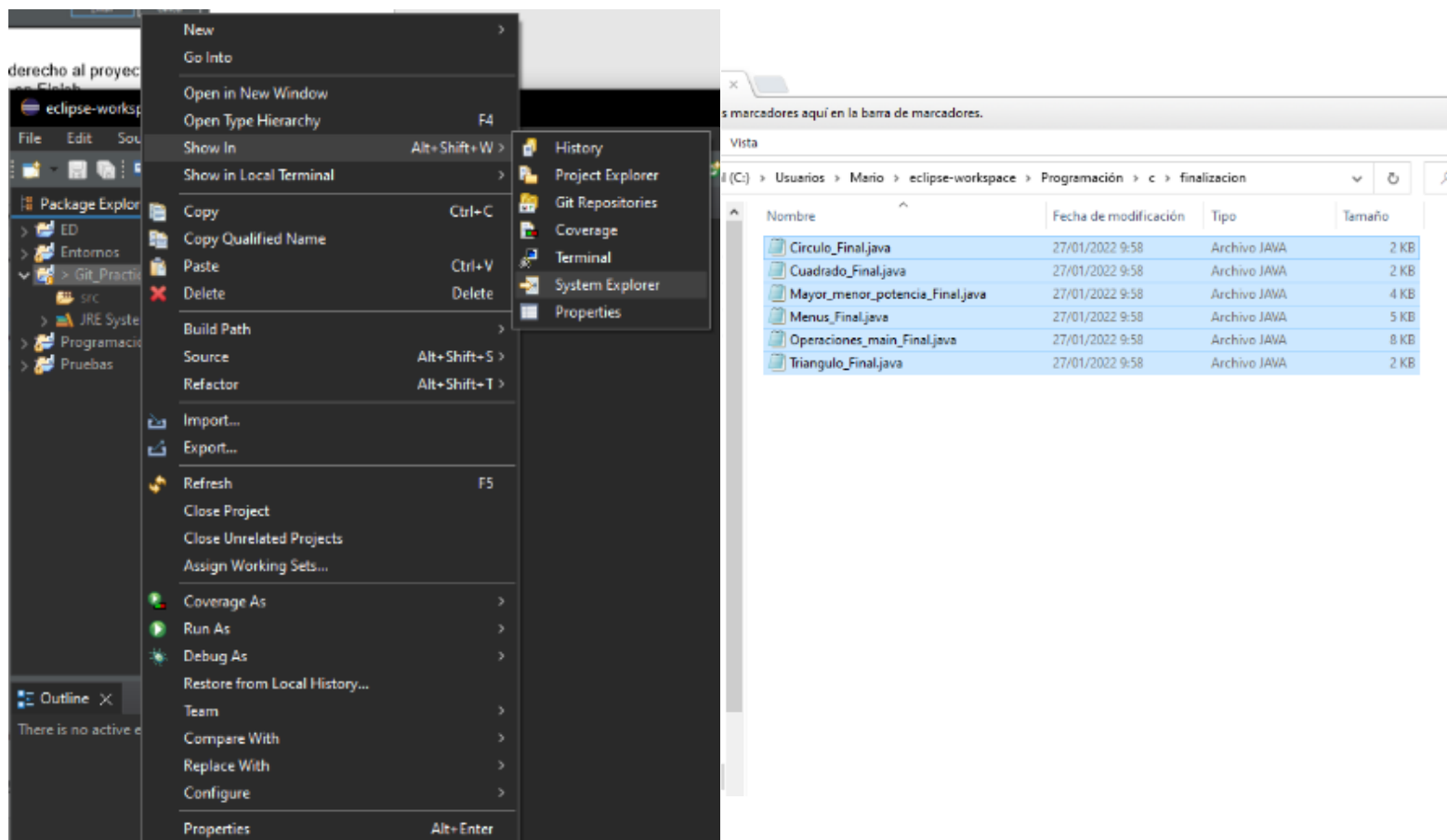
2. DESARROLLO – PROCEDIMIENTOS



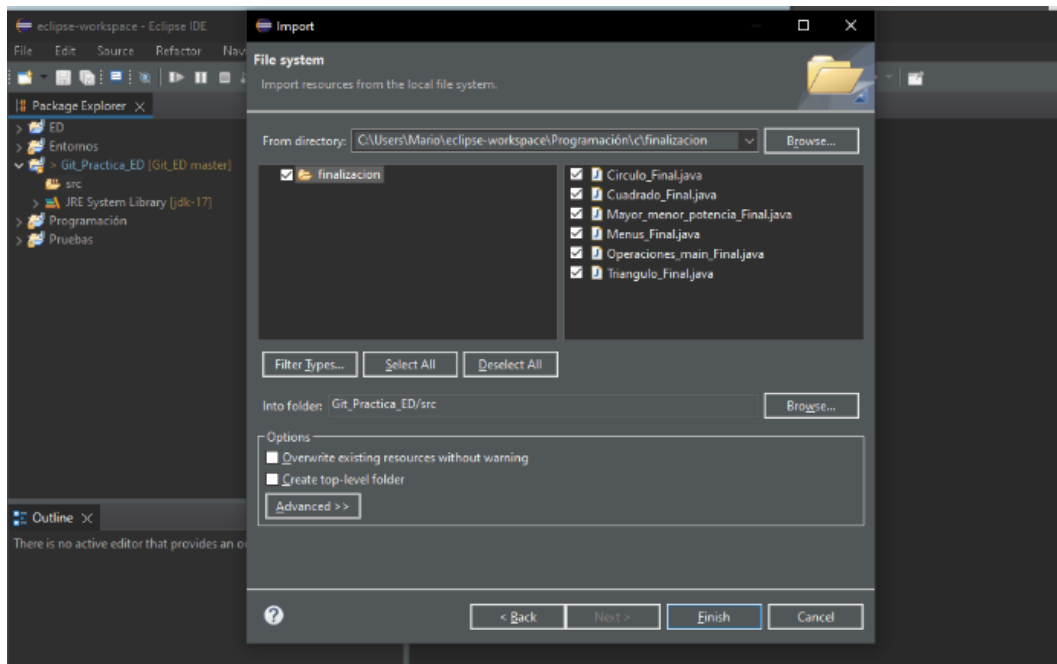
En este primer paso se crea el proyecto Java dentro de Eclipse (en una carpeta diferente al eclipse-workspace que viene por defecto). Este va a ser el directorio que se va a utilizar para esta práctica de Git.



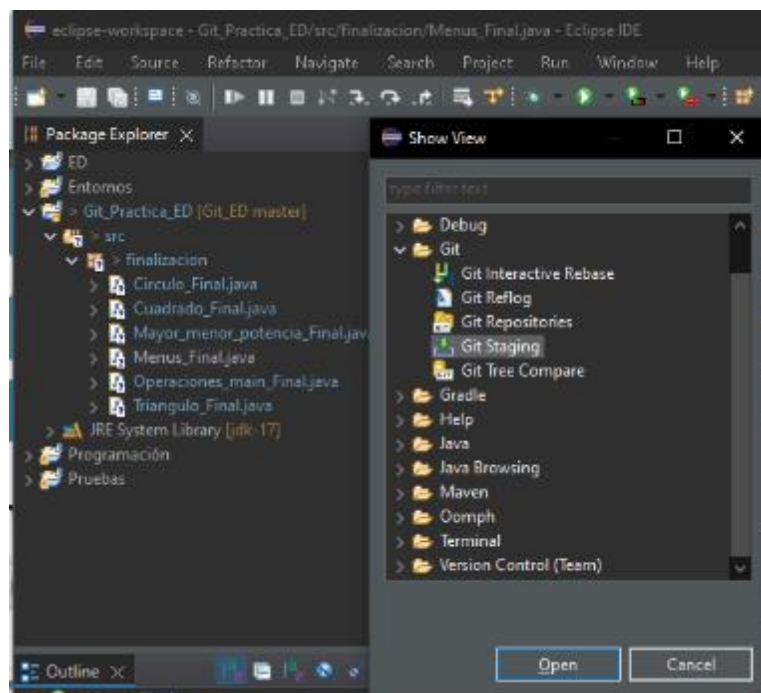
Después, para configurar el Git (GIT INIT), se accede con clic derecho al proyecto, a Team y a Share Project. Dentro de este, se crea un repositorio y se clicka en Finish.



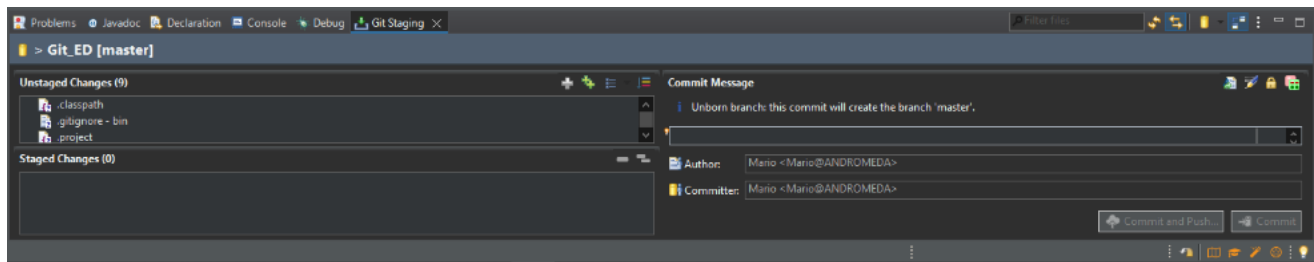
En estas dos capturas, se muestra cómo visualizar el proyecto, a través de Show In y Navigator. Al clicar este último, salta al Windows Explorer, eligiendo las clases .java que se quieren introducir dentro del proyecto.



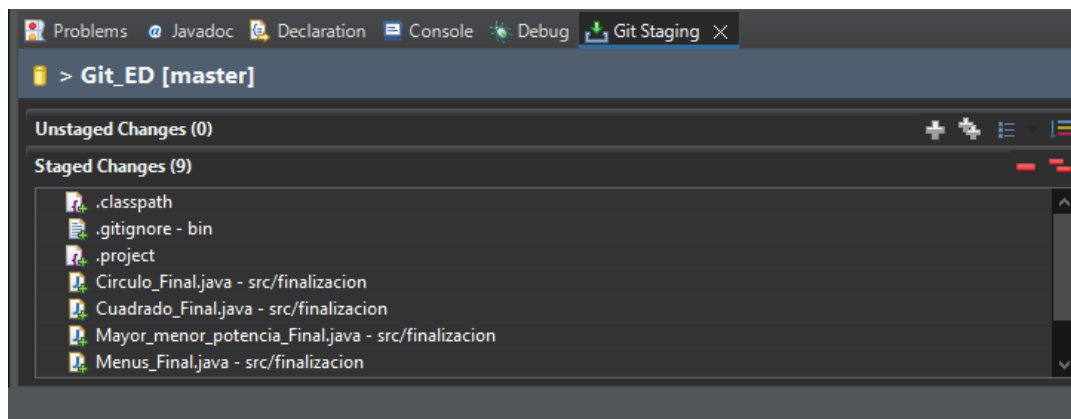
Se introduce el código dentro del proyecto, en un package (son clases de otro proyecto).



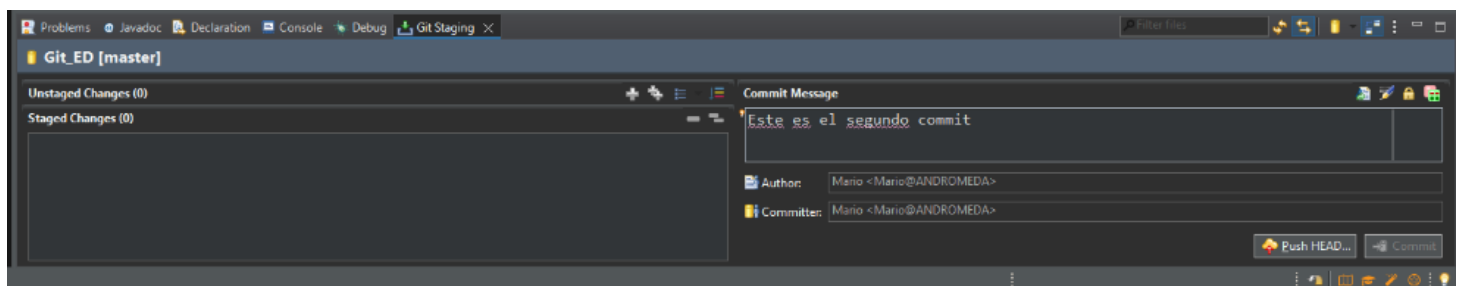
Se abre una nueva perspectiva de clase: se clicca en Window, Show View, en la pestaña Other (para encontrar Git) y clicar en Git Staging.



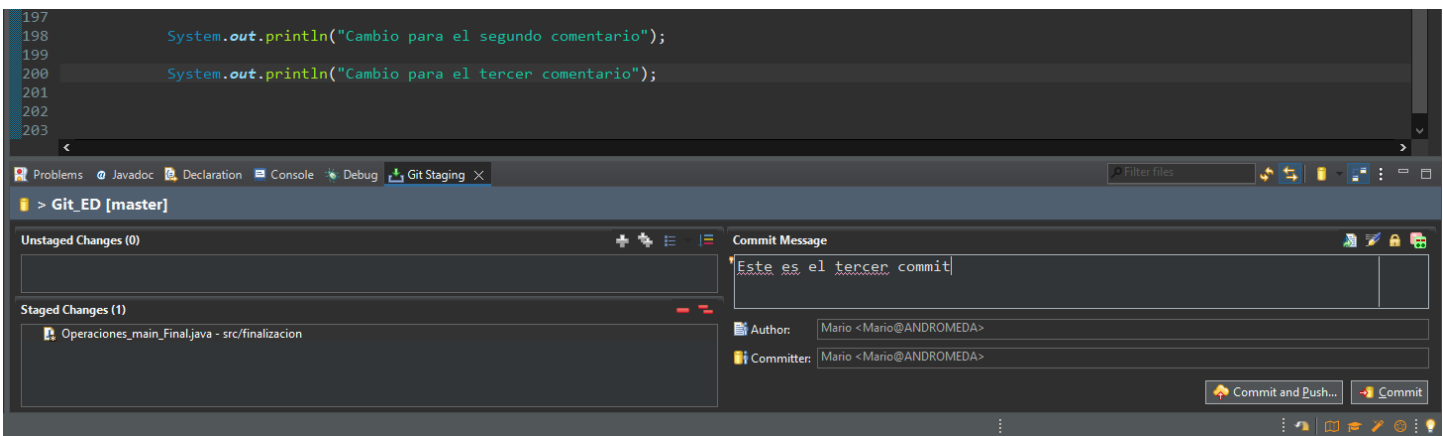
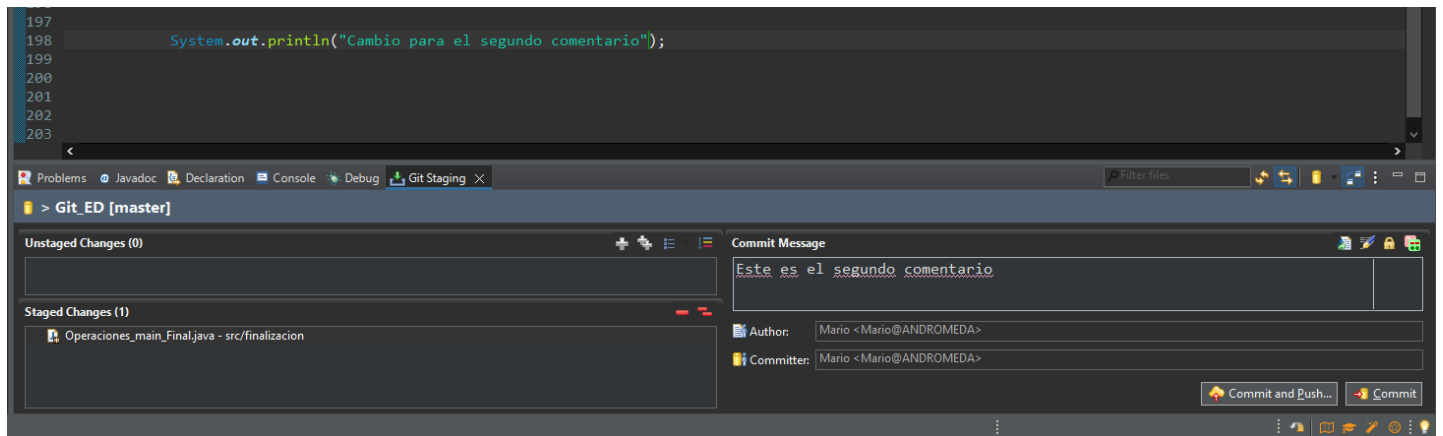
Este es el Área Temporal o Staging, donde se realizan los commits.



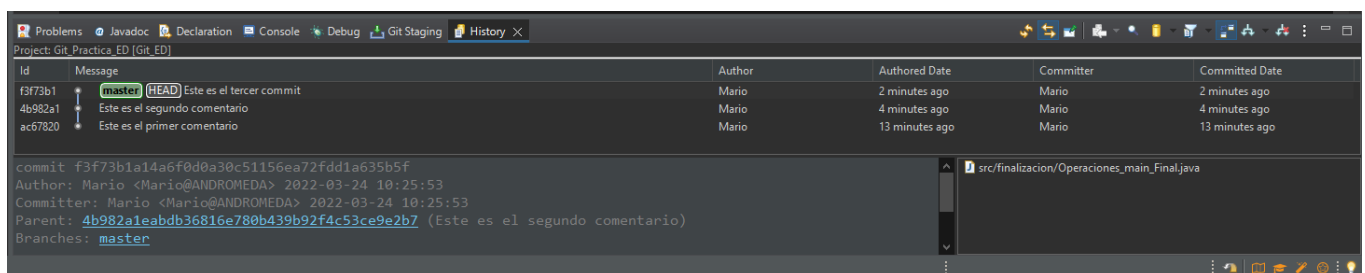
Con Git Add, se pasa el código Unstaged a Staged Changes, seleccionando todo el código con el botón ++.



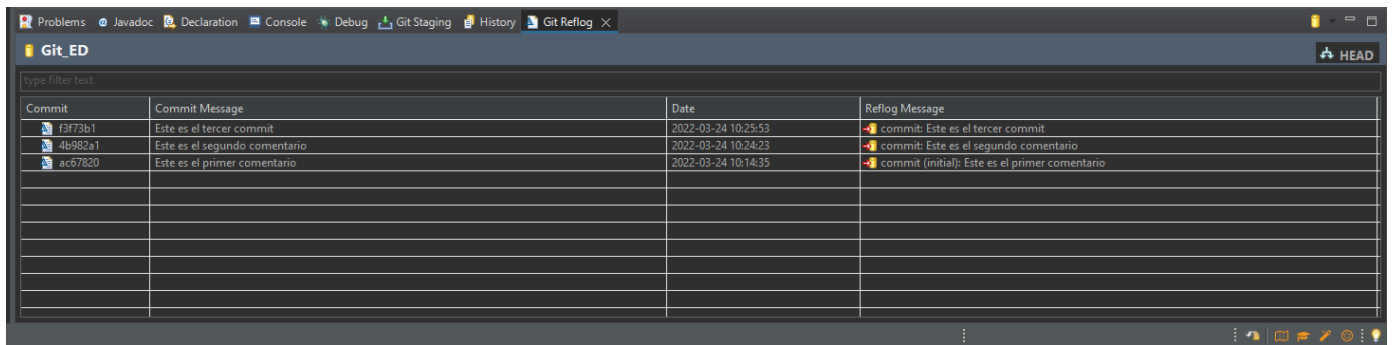
Al poner un commit y pulsar en el botón “Commit”, desaparece lo que se tiene en el Área Staging (desaparecen las clases que antes se habían añadido). Se realizan tres commits.



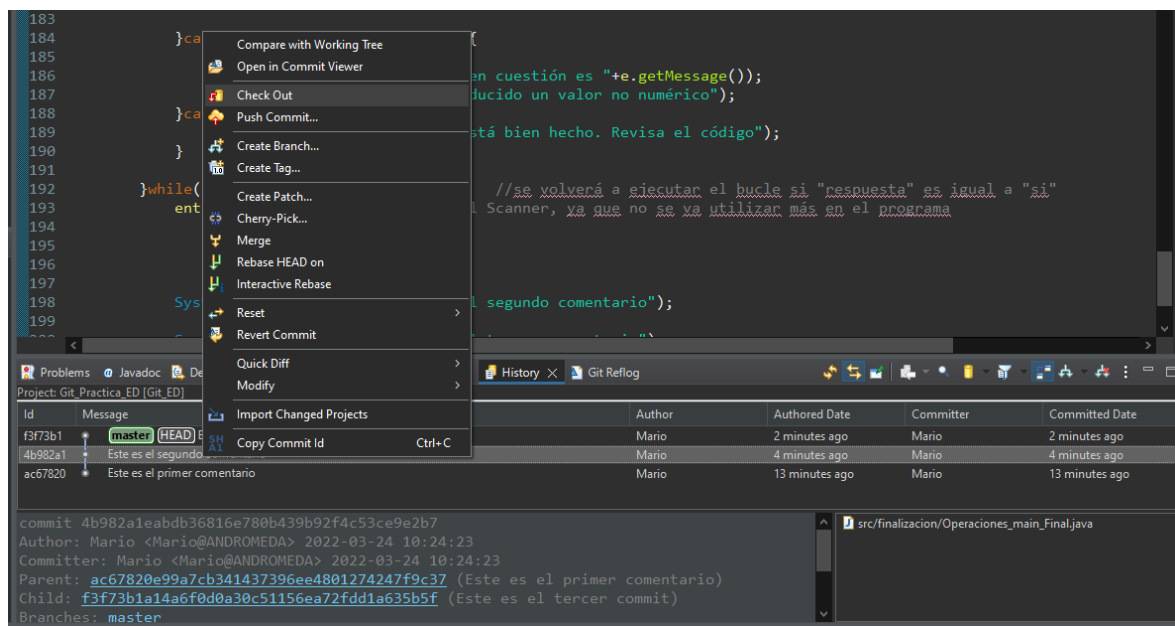
Para que se realicen los dos últimos commits, se debe hacer algún cambio en alguna de las clases. Por ejemplo, en este caso se han introducido para cada uno de los 2 commits una impresión en pantalla diferente.



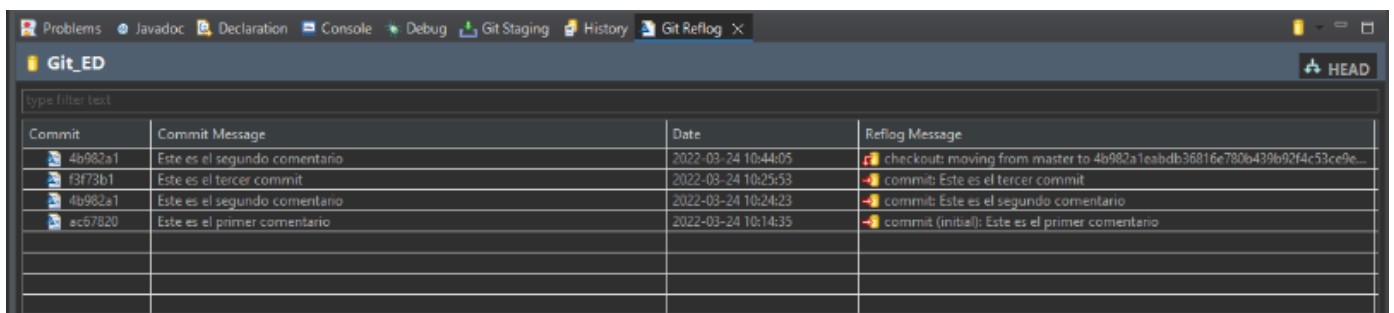
Para ver los cambios realizados (los commits) se abre dentro del proyecto la pestaña Team and Show in History. Los cambios se visualizan en orden cronológico.



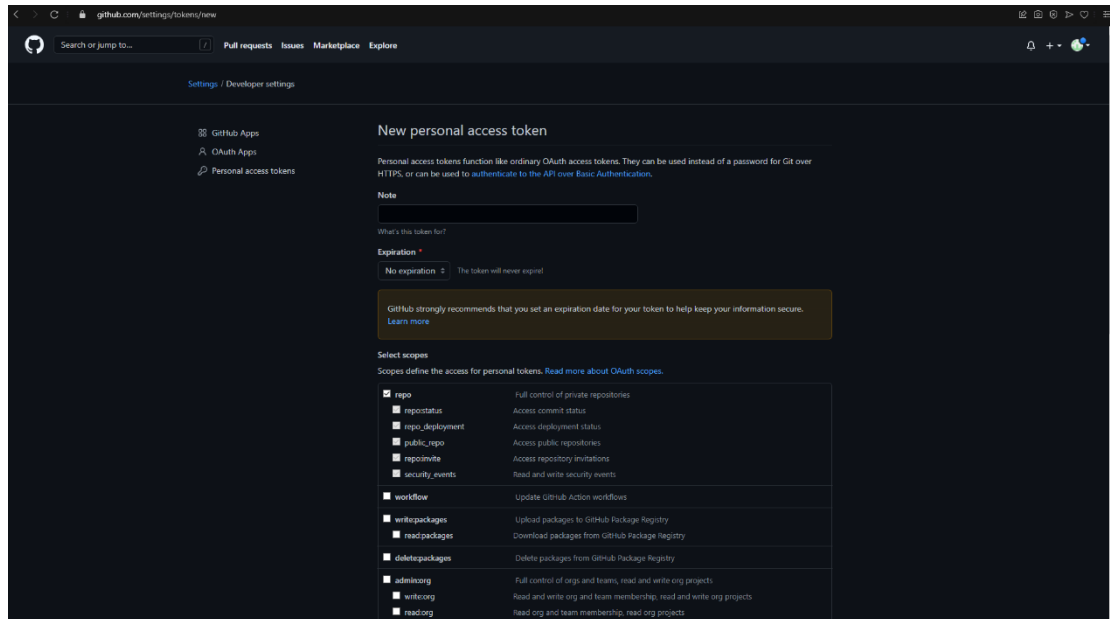
Con la pestaña Gitt Reflog y Git Diff se pueden comprobar diferencias en el código, según el commit que sea.



Para restaurar el código del segundo commit, se hace clic derecho en este en la pestaña CheckOut.

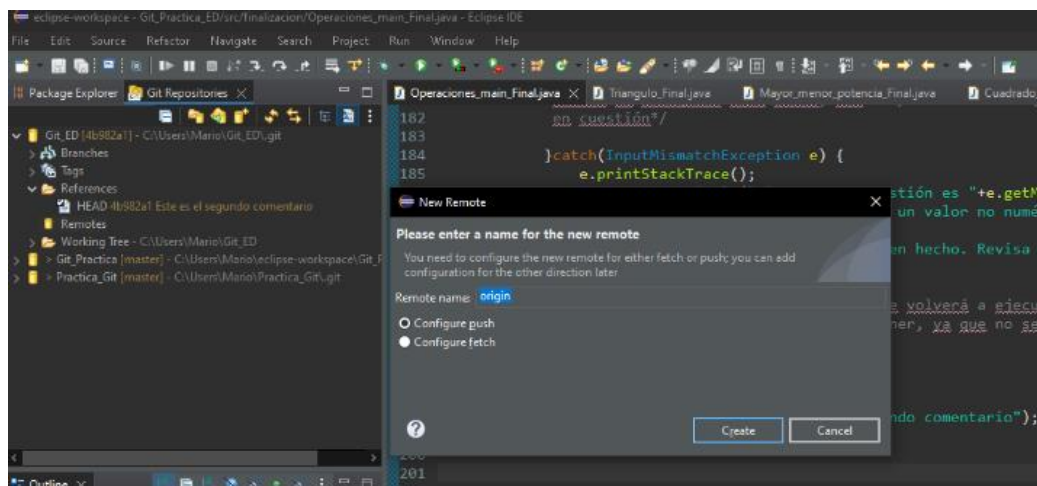


Muestra un mensaje de aviso de que no existe copia de respaldo en otro repositorio. Se muestra en el Git Reflog que el segundo commit ha sido movido por el Checkout.

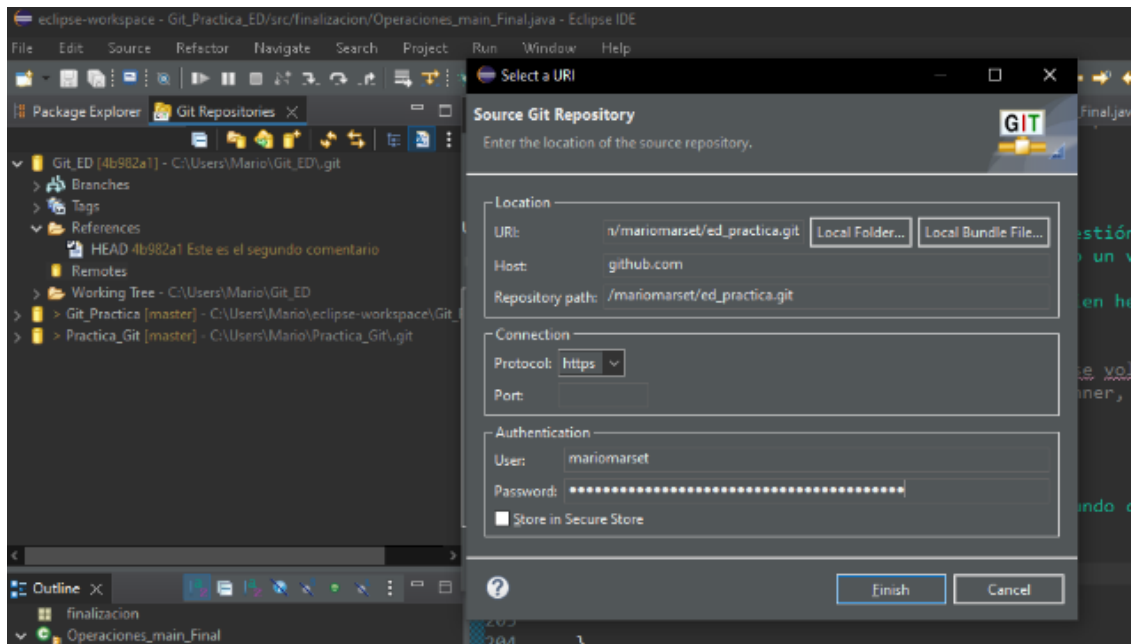


Dentro de la aplicación web Git Hub (habiendo entrado con el perfil) se accede a través de Settings, Developer Settings, Personal Access tokens hasta Generate new token. Se genera un nuevo token, que no tenga límite de expiración, y seleccionando la opción “repo”.

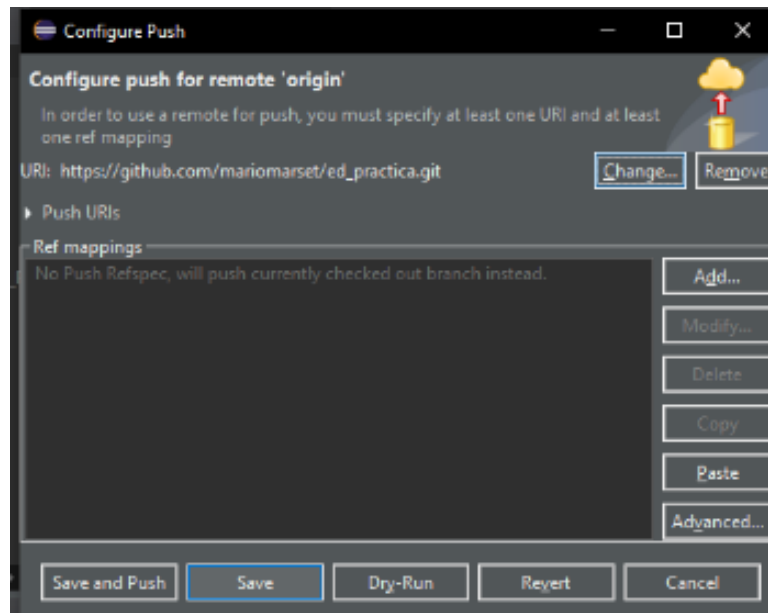
El repositorio creado es “ed_practica”.



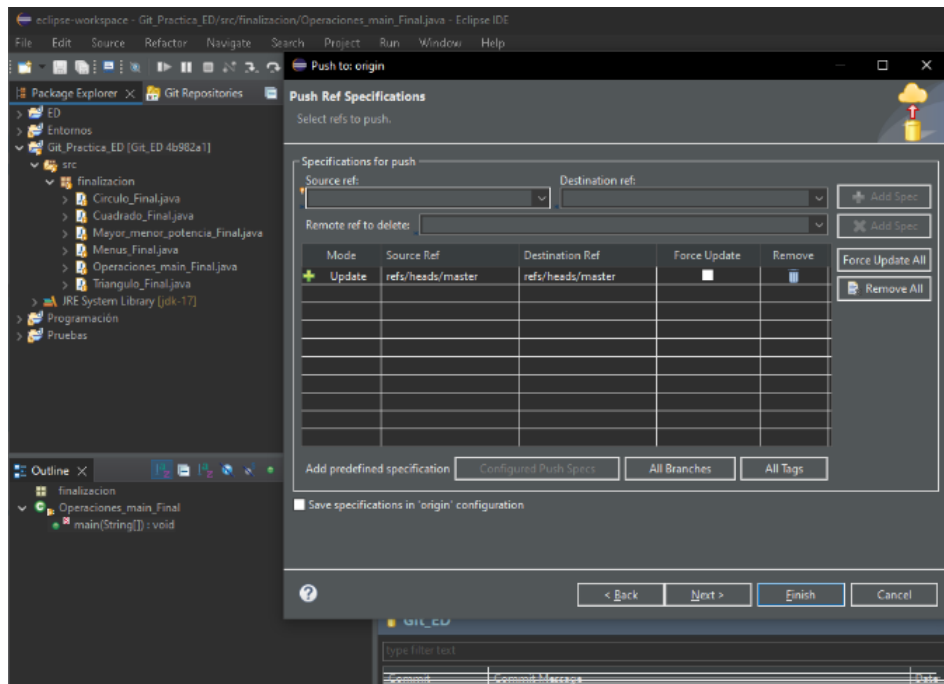
Dentro de la pestaña Windows, en Show View (Other), se clic en Git Repositories. Dentro de este, se clic en Remote (Create Remote).



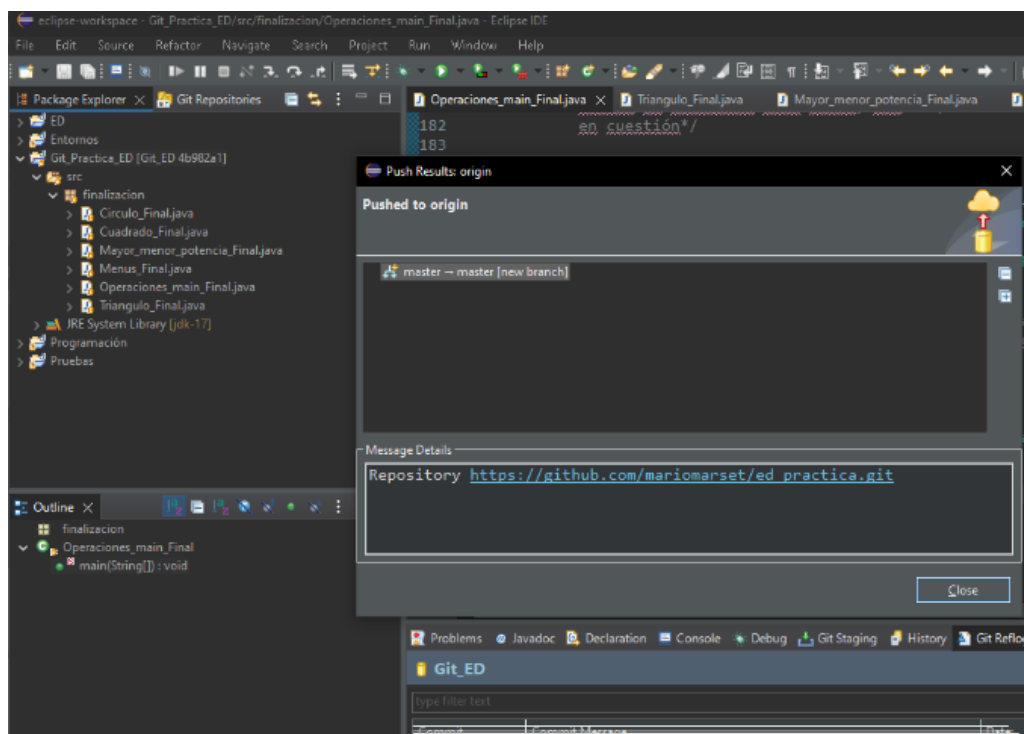
Se cambia la URI por la del repositorio del Git-Hub, se pone el usuario de este mismo y en la contraseña el token creado anteriormente.



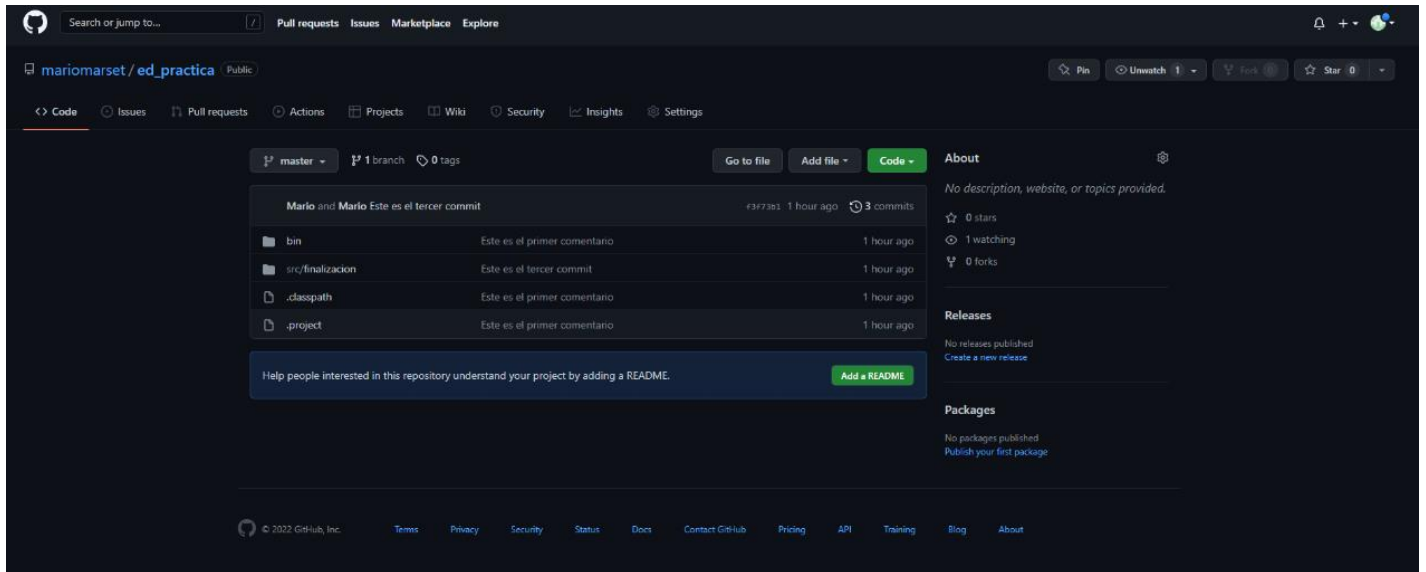
Se pulsa en "Save".



Haciendo clic derecho en el proyecto, en Team, se clic en Push Head. Dentro de él, se clic en un sitio que pone “Show advanced push dialog”. Dentro de él, se añade un tag. Se clic en “Finish”.



Al clicar en “Finish” pide el user y la password de Git-Hub, mostrando a continuación el repositorio en que se encuentra el tag añadido.



Finalmente, se muestra dentro de Git-Hub el estado del repositorio.

3. WEBGRAFÍA

- <https://github.com>