

NOMBRE DEL ALUMNO/A: Jose Rodríguez Torrejón

---

MÓDULO: ENTORNOS DE DESARROLLO

NOMBRE DEL PROFESOR/A: ALBERTO BERNARDEZ JIMENEZ

EVALUACIÓN Y FECHA: – Prueba Practica Tema 3 (30/11/2023)

**CRITERIOS DE EVALUACIÓN:** La prueba práctica es un 70% del total de la prueba. La puntuación de cada ejercicio se indica en el mismo.

---

### Ejercicio 1 (3 puntos)

Cree el repositorio en GITHUB: P3A\_NombreAlumno. Con TortoiseGIT, debe replicar este repositorio en un repositorio local.

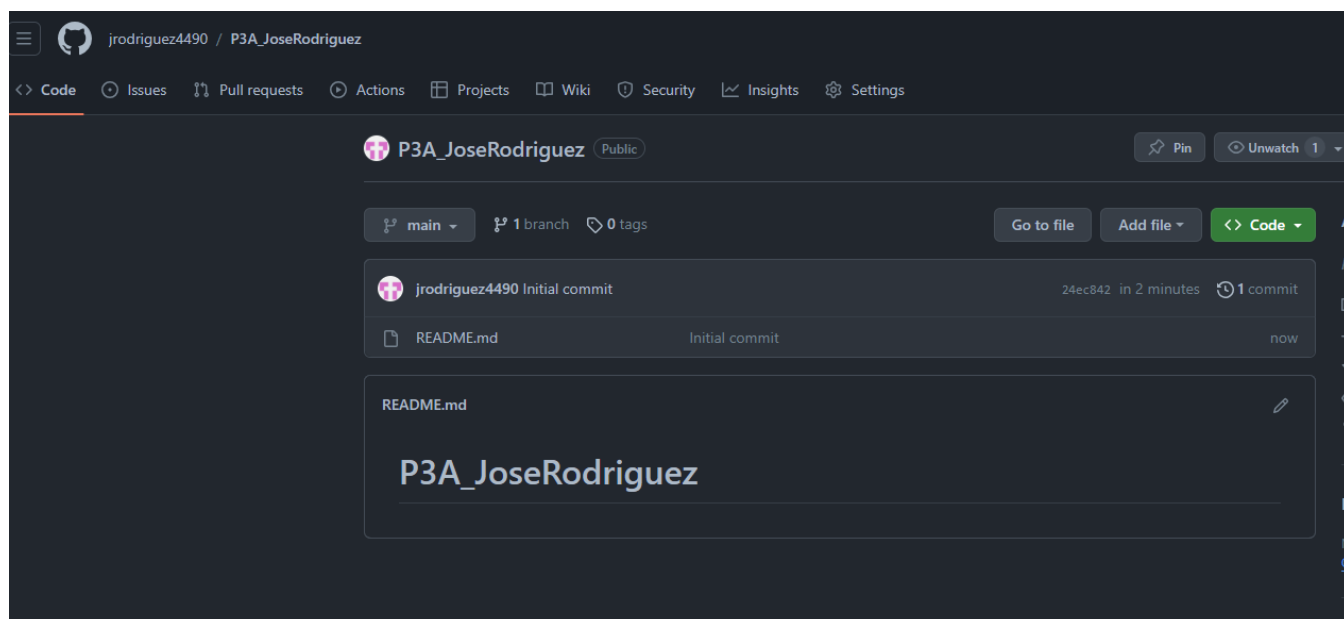
En este repositorio local debe incluir un documento (pdf, por ejemplo) que contenga su nombre [Indique como hacer que el documento pase a su repositorio local y verifique que está en local pero no en remoto (GITHUB)]

En un siguiente paso, haga que se actualice el proyecto de Github y se incluya el documento antes generado.

Nota: Para la correcta evaluación del ejercicio debe presentar:

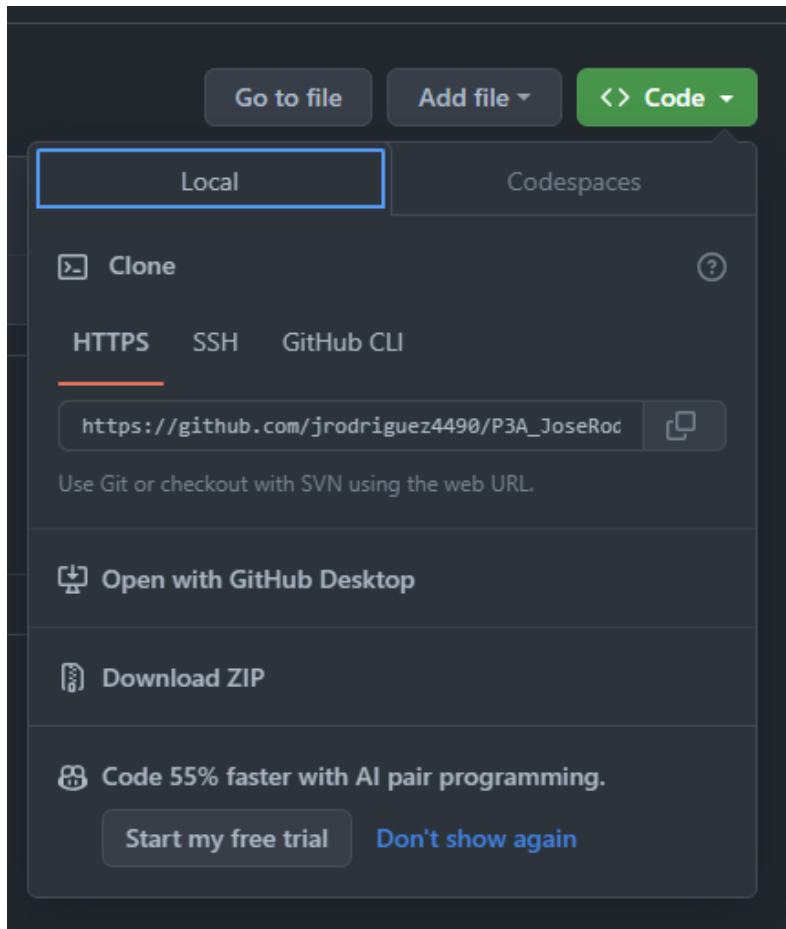
- Capturas de cada acción realizada, acompañadas de una breve explicación
- URL de su repositorio GITHUB P3A\_NombreAlumno, donde se revisará el historial de versiones

-Para realizar el primer paso, debemos de crear en primer lugar el repositorio en Github con nuestro usuario.

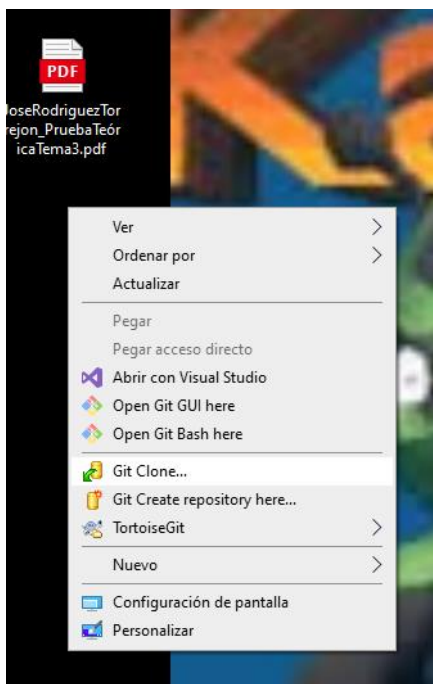


En segundo lugar, pinchamos en la opción "code", la cual se puede ver en la captura anterior, y copiamos

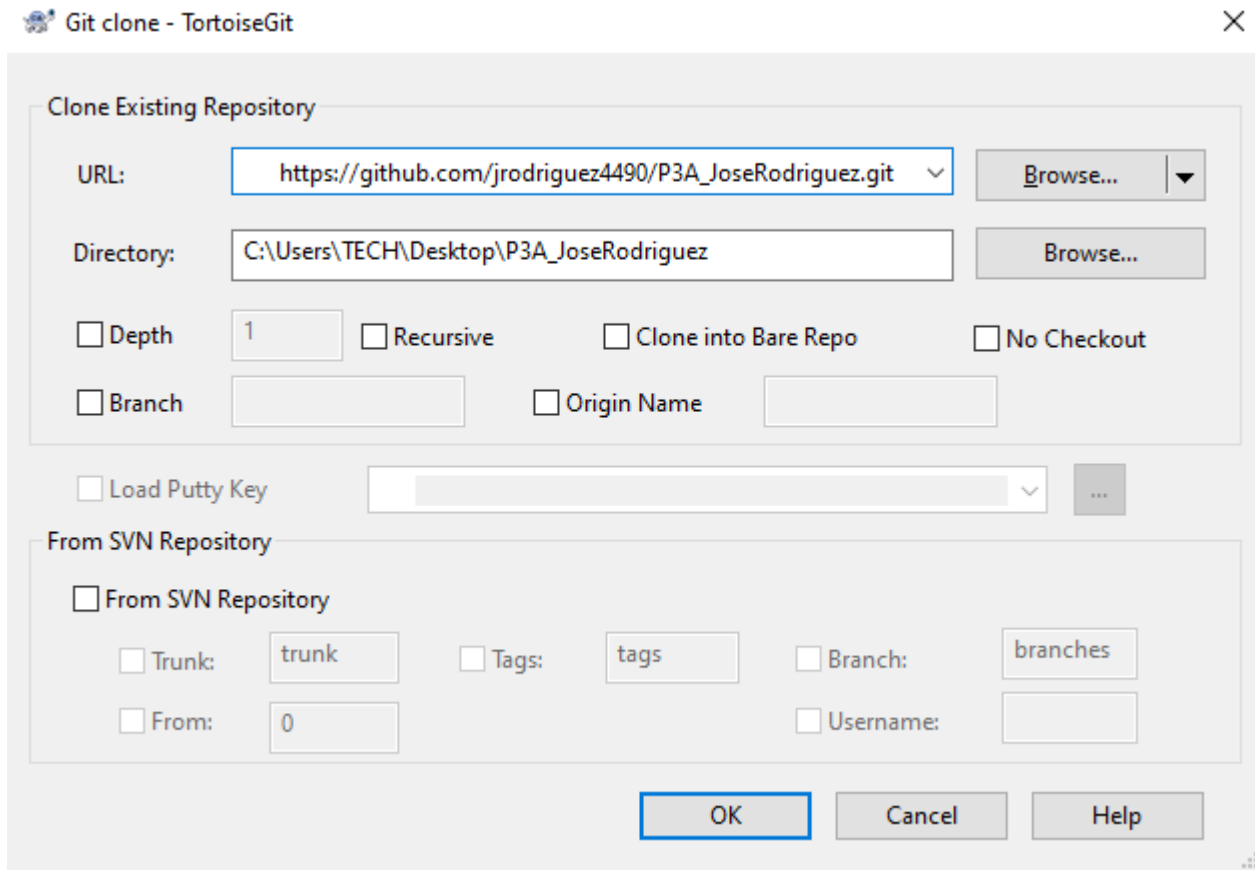
el respectivo "Https".



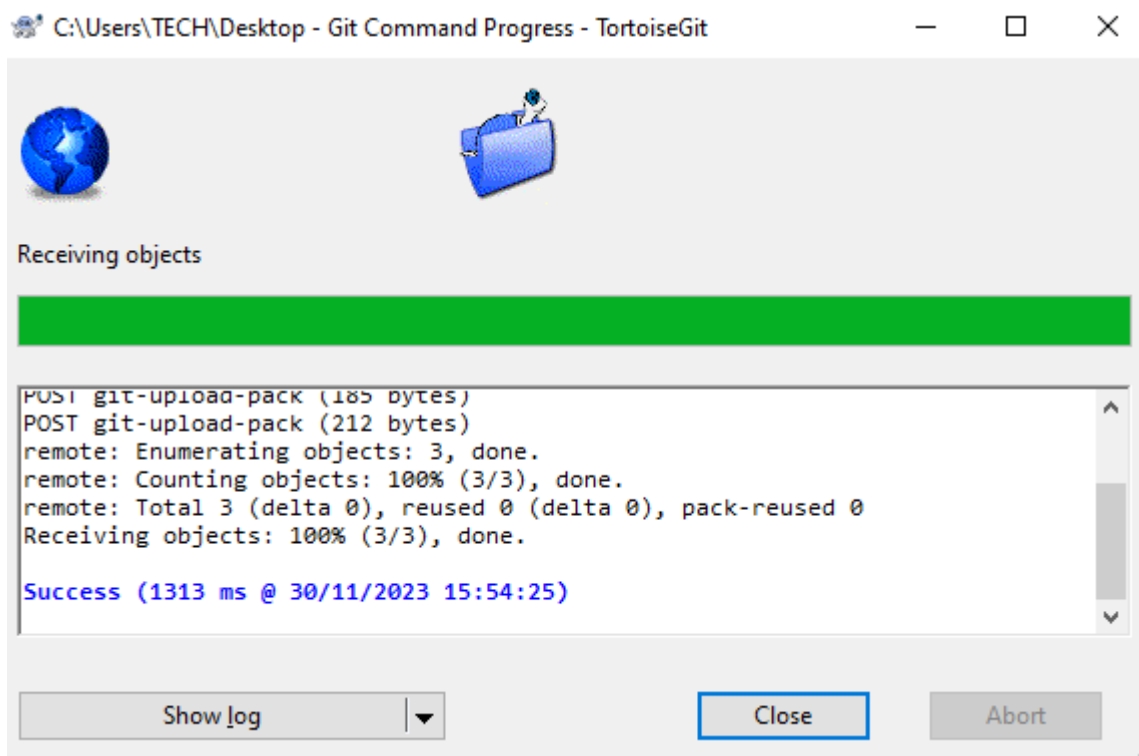
Acto seguido, damos click derecho en nuestro escritorio, y seleccionamos la opción "Gitclone..." para traer el repositorio remoto a un repositorio local.



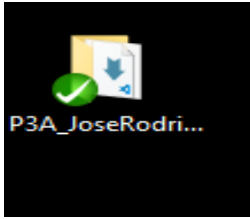
Nos aparecerá una ventana donde debemos incluir el enlace "https" que hemos copiado de github.



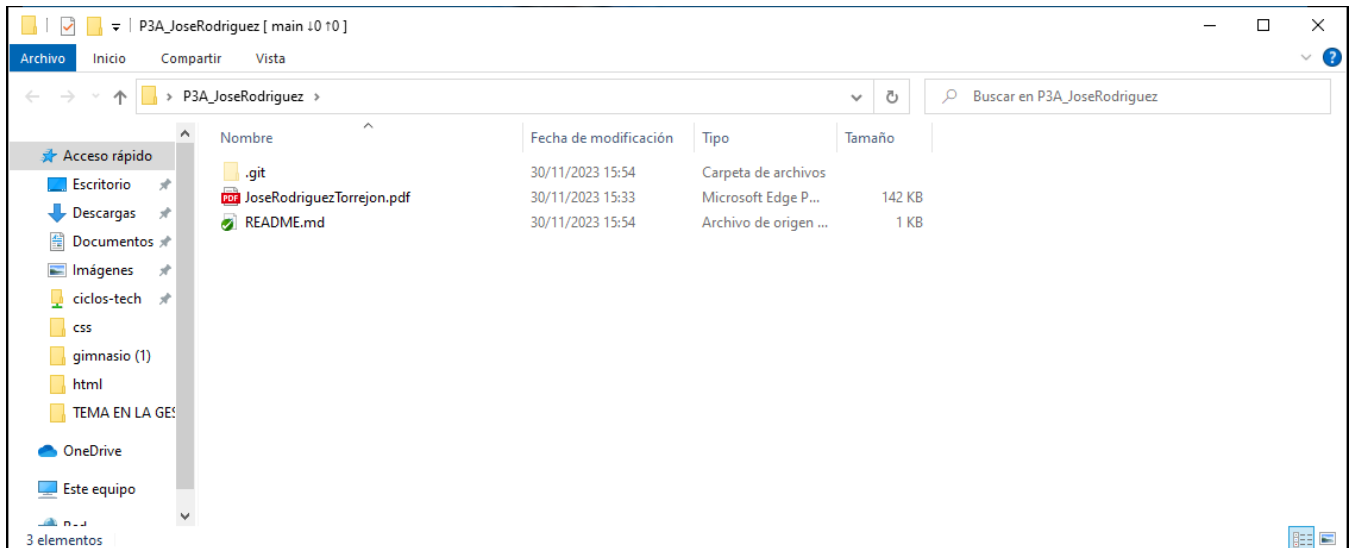
Una vez hecho, se nos mostrará el proceso.



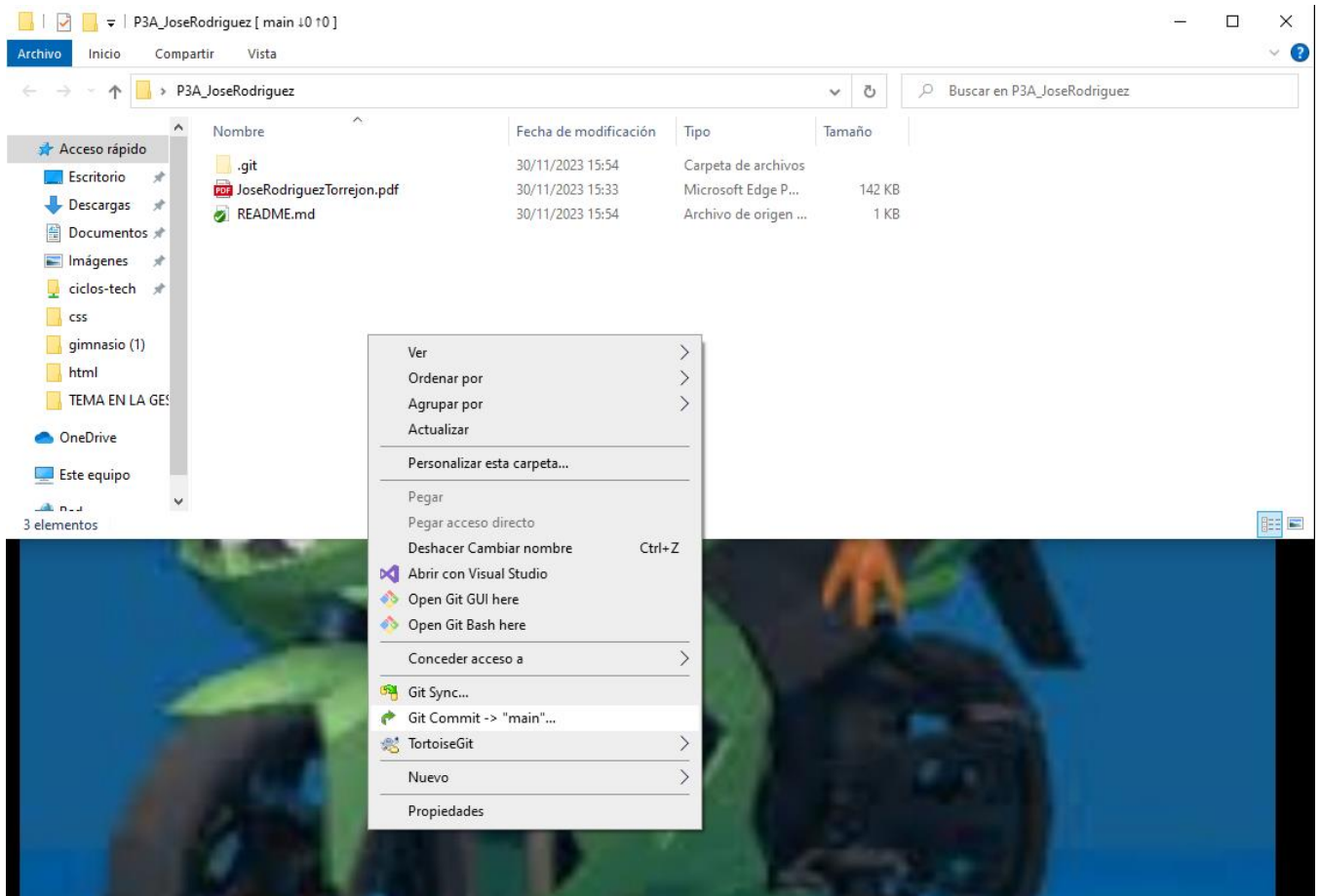
Una vez terminado, ya hemos traído el repositorio de Github a un repositorio local.



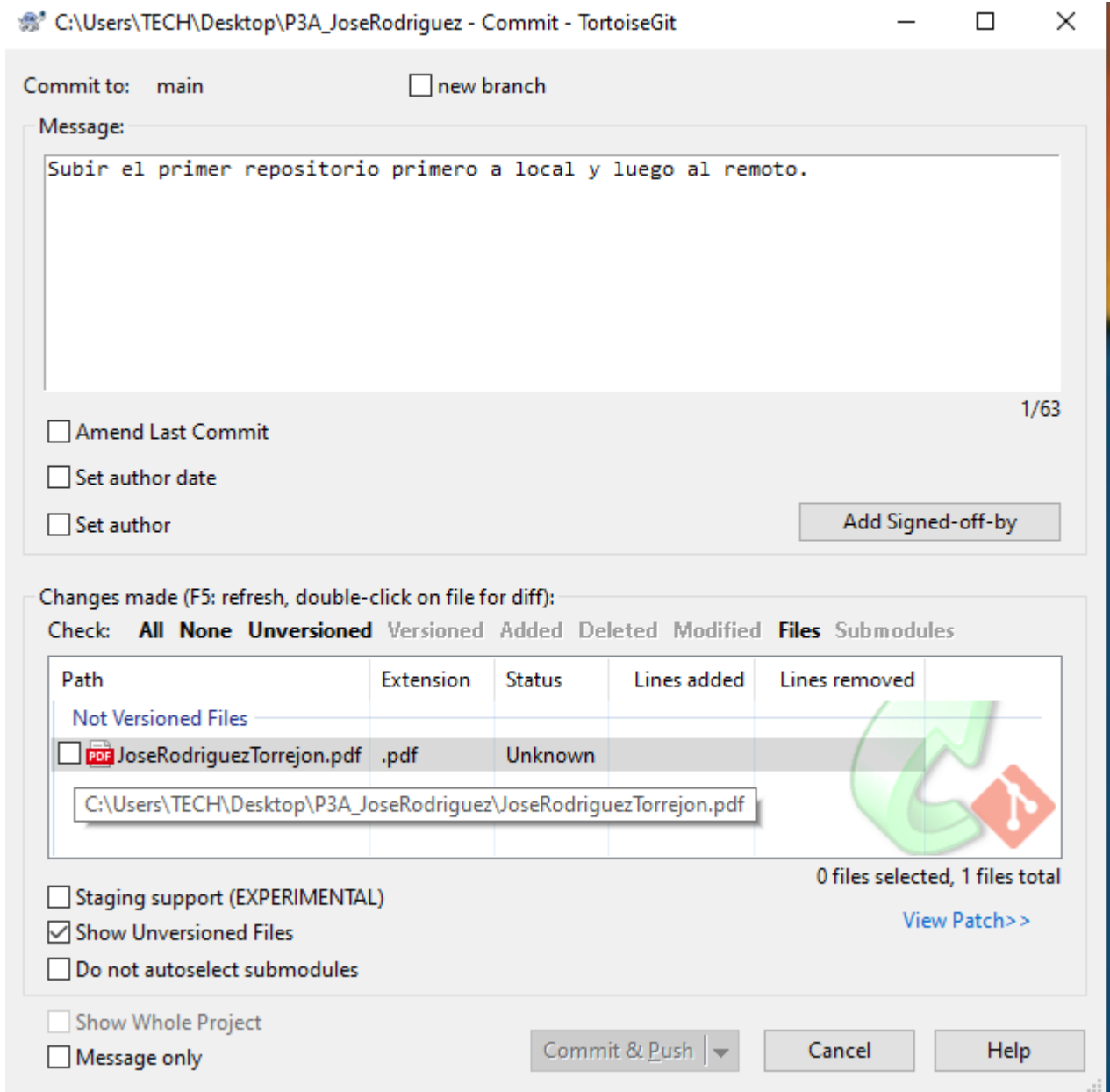
Para realizar el segundo paso, es decir, crear un pdf con nuestro nombre en el repositorio local y guardarlo para subirlo al repositorio remoto, es decir, github, tenemos que abrir en primer lugar la carpeta que he sa clonado y en segundo lugar, crear el archivo pdf.



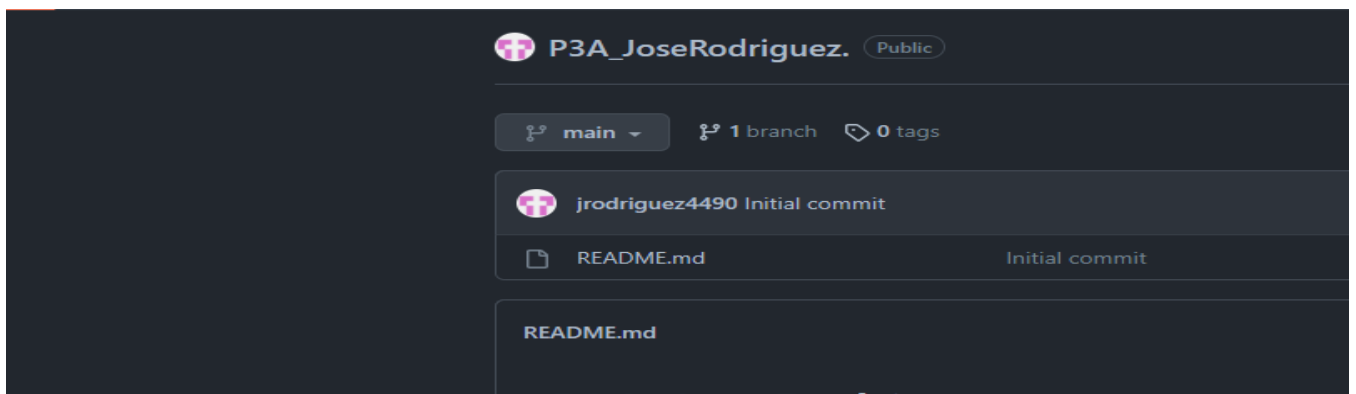
Quedaría de este modo. Una vez creado el pdf, todavía no se ha guardado en el repositorio local, para ello debemos de hacer un “commit”. Tenemos que hacer click derecho en la carpeta y seleccionar la opción “commit”.

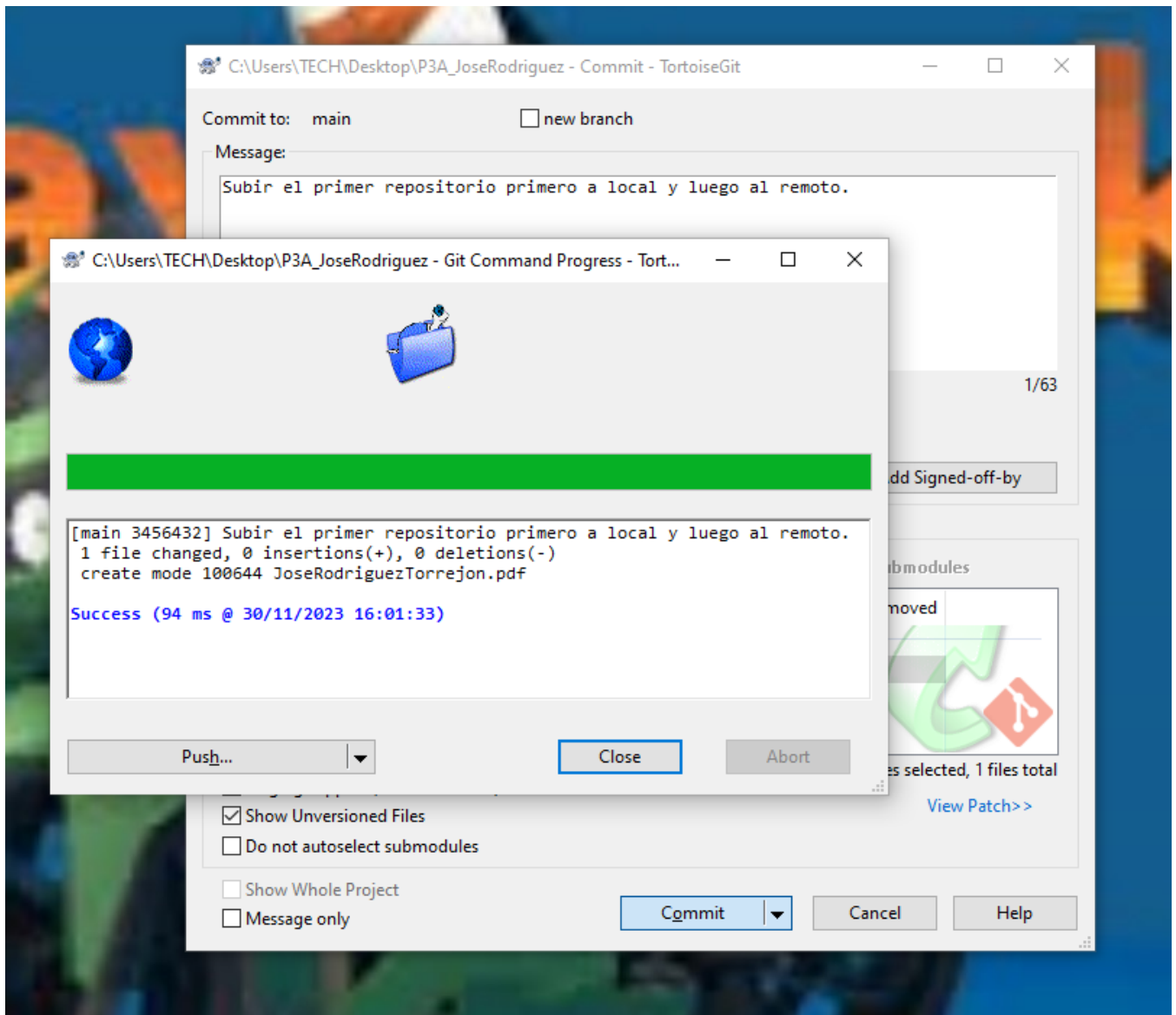


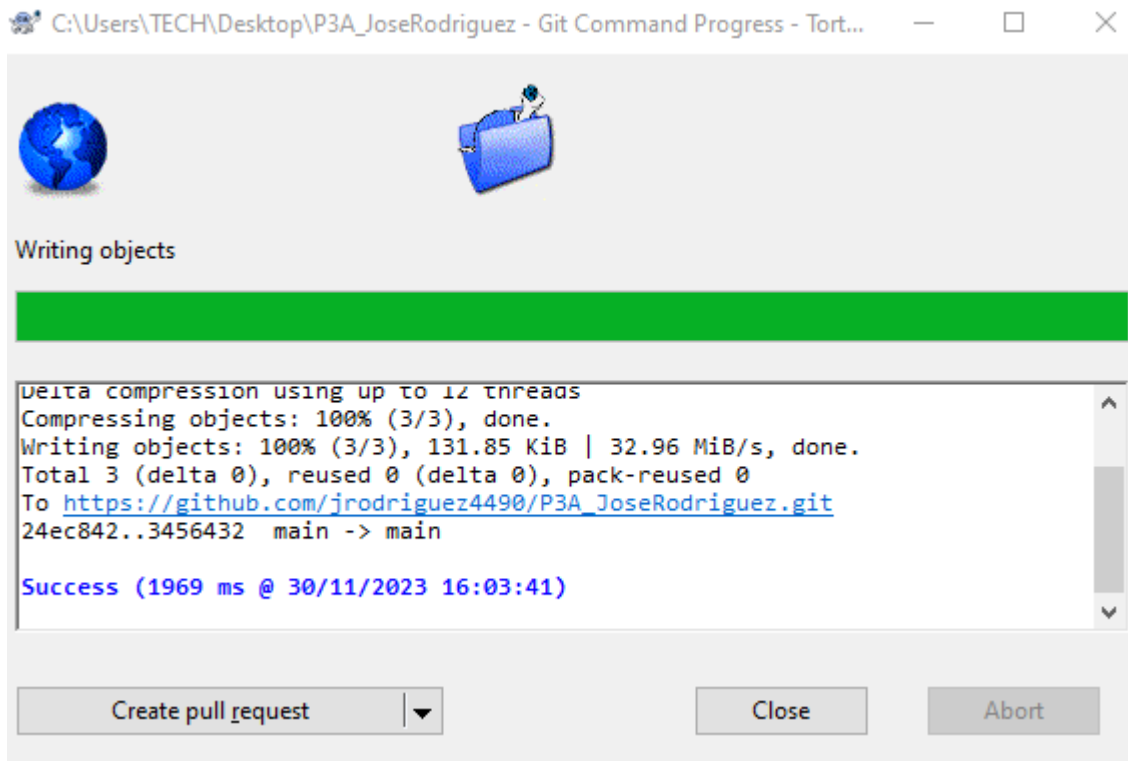
Se nos abrirá esta pestaña, en la que debemos de escribir primero un mensaje y luego “commit” para guardarlo en el repositorio local y “commit and push” para guardarlo en el local y subirlo directamente.



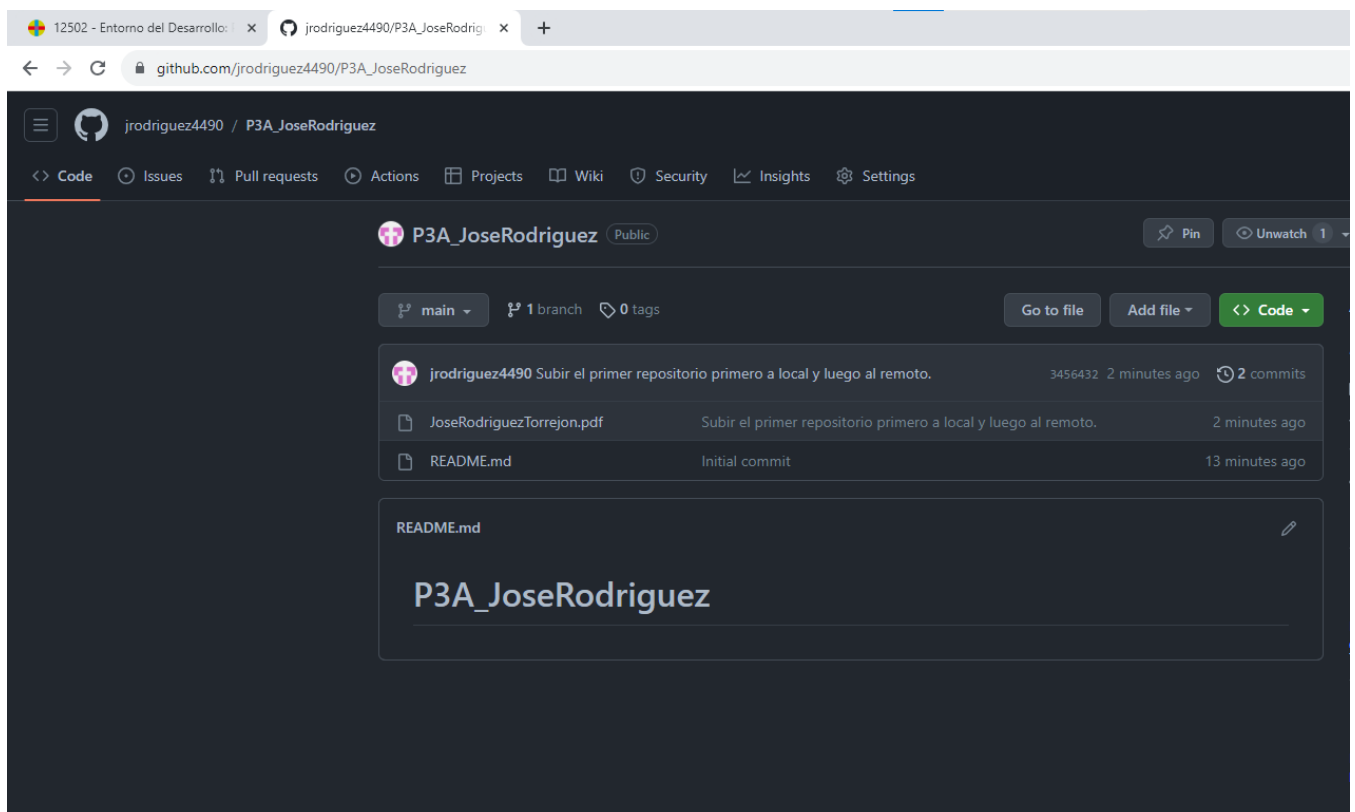
Una vez seleccionado el archivo y habiéndole dado a “commit” ya se nos habrá guardado en el repositorio local. Ahora no saldrá el proceso y una vez terminado, elegimos la opción “push” para subirlo al repositorio remoto. Hasta que no seleccionemos la opción “push”, el contenido no se subirá a github.







Una vez realizamos esto, revisamos que se haya subido a Github debidamente.



[https://github.com/jrodriguez4490/P3A\\_JoseRodriguez](https://github.com/jrodriguez4490/P3A_JoseRodriguez)



## Ejercicio 2 (7 puntos)

Cree el repositorio en GITHUB: P3B\_NombreAlumno.

Desde Eclipse, clone el repositorio

Cree el proyecto en Eclipse Proyecto3B e incluya una clase que imprima por pantalla su nombre, incluya dicho proyecto en su repositorio local de eclipse.

Envíe el proyecto Proyecto3B a GITHUB, confirme que en GITHUB tenemos la clase creada.

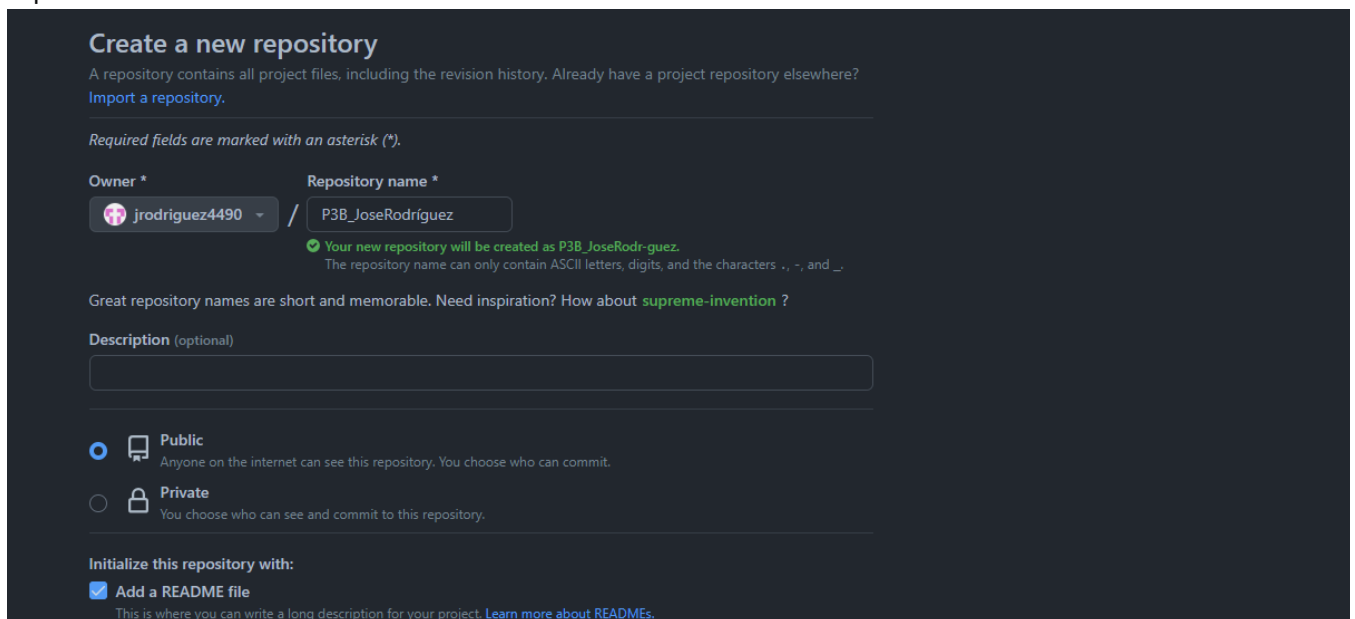
Modifique en GITHUB el contenido de la clase para que imprima también por pantalla: "Estoy llegando al final".

Realice las acciones necesarias en eclipse para traer el cambio de GITHUB a nuestro repositorio local. Muestre en Eclipse el contenido de clase modificado.

Nota: Para la correcta evaluación del ejercicio debe presentar:

- Capturas de cada acción realizada, acompañadas de una breve explicación
- URL de su repositorio GITHUB P3B\_NombreAlumno, donde se revisará el historial de versiones

Para realizar el primer paso, y al igual que en la práctica anterior, debemos ir a Github y crear dicho repositorio:




**Create a new repository**

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

*Required fields are marked with an asterisk (\*).*

**Owner \*** **Repository name \***

 jrodriguez4490 / P3B\_JoseRodriguez

✓ Your new repository will be created as P3B\_JoseRodriguez.  
The repository name can only contain ASCII letters, digits, and the characters -, ., and \_.

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

**Description** (optional)

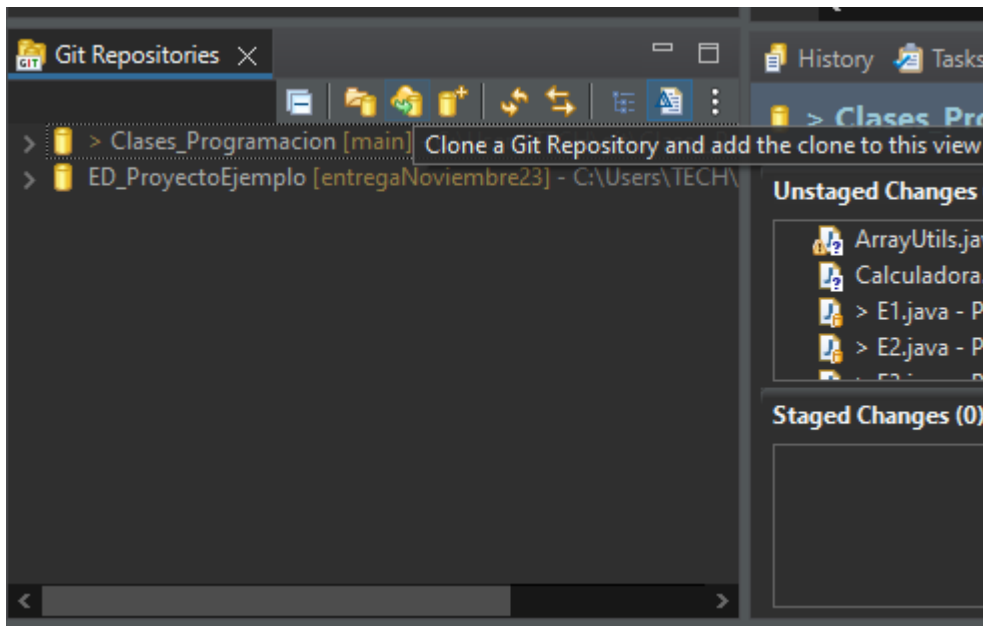
☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

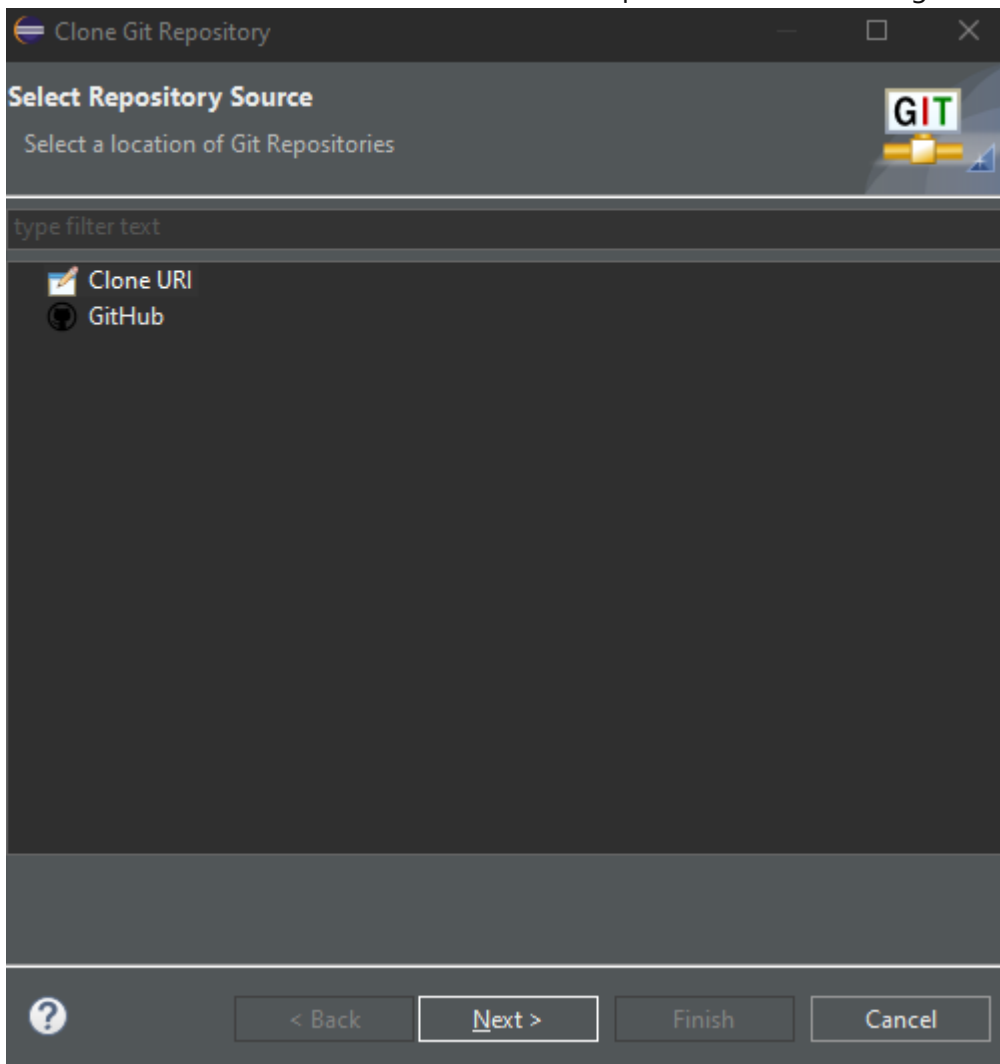
**Initialize this repository with:**

☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

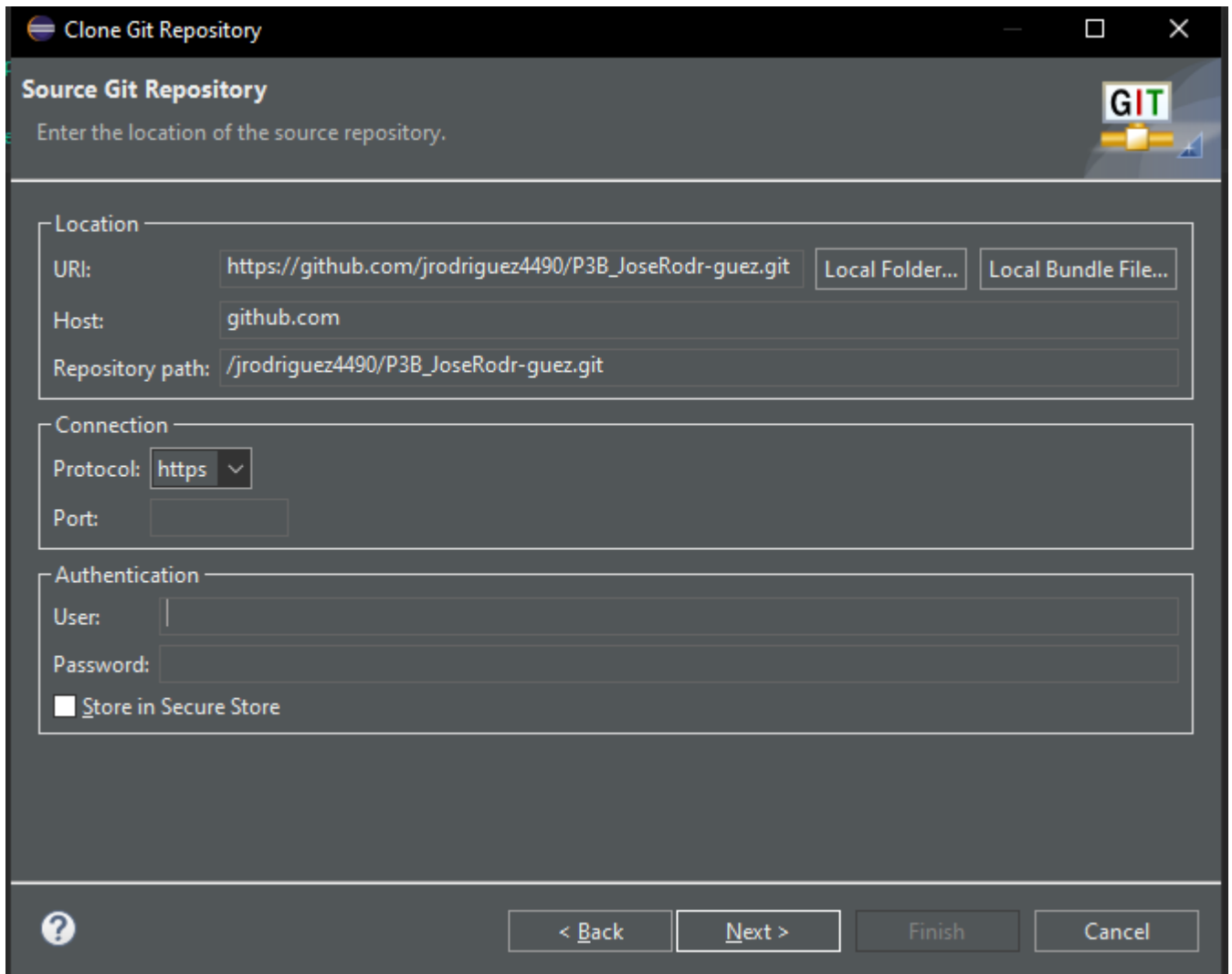
Una vez creado, copiamos el archivo "https" en la pestaña de code tal y como hicimos en la práctica anterior. Una vez realizado este paso, abrimos eclipse y en la ventana "Git Repositories" seleccionamos la opción "Clone a Git Repository and add the clone to this view".



Se nos abrirá una ventana en la que debemos escoger la opción "Clone URI"



Se nos abrirá otra ventana en la cual debemos de pegar el https que hemos copiado de nuestro repositorio remoto:



The image shows a 'Clone Git Repository' dialog box with a dark theme. It has a title bar with standard window controls. The main area is divided into three sections: 'Location', 'Connection', and 'Authentication'. The 'Location' section contains fields for 'URI' (with a button to open a file explorer), 'Host', and 'Repository path'. The 'Connection' section has a 'Protocol' dropdown set to 'https' and a 'Port' field. The 'Authentication' section has 'User' and 'Password' fields, and a checkbox for 'Store in Secure Store'. At the bottom, there are four buttons: a help icon, '< Back', 'Next >', 'Finish', and 'Cancel'.

**Clone Git Repository**

**Source Git Repository**  
Enter the location of the source repository.

**Location**

URI:

Host:

Repository path:

**Connection**

Protocol:

Port:

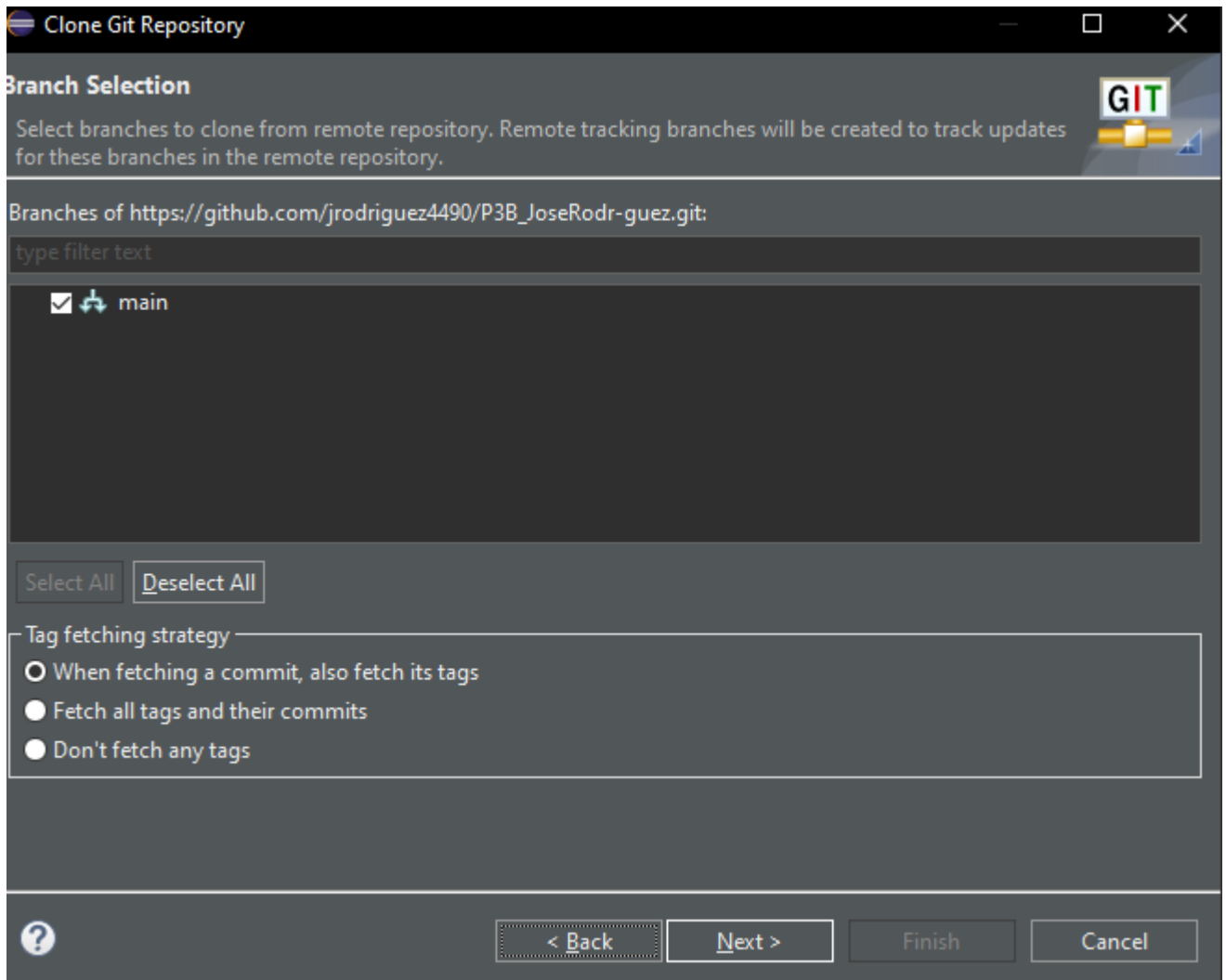
**Authentication**

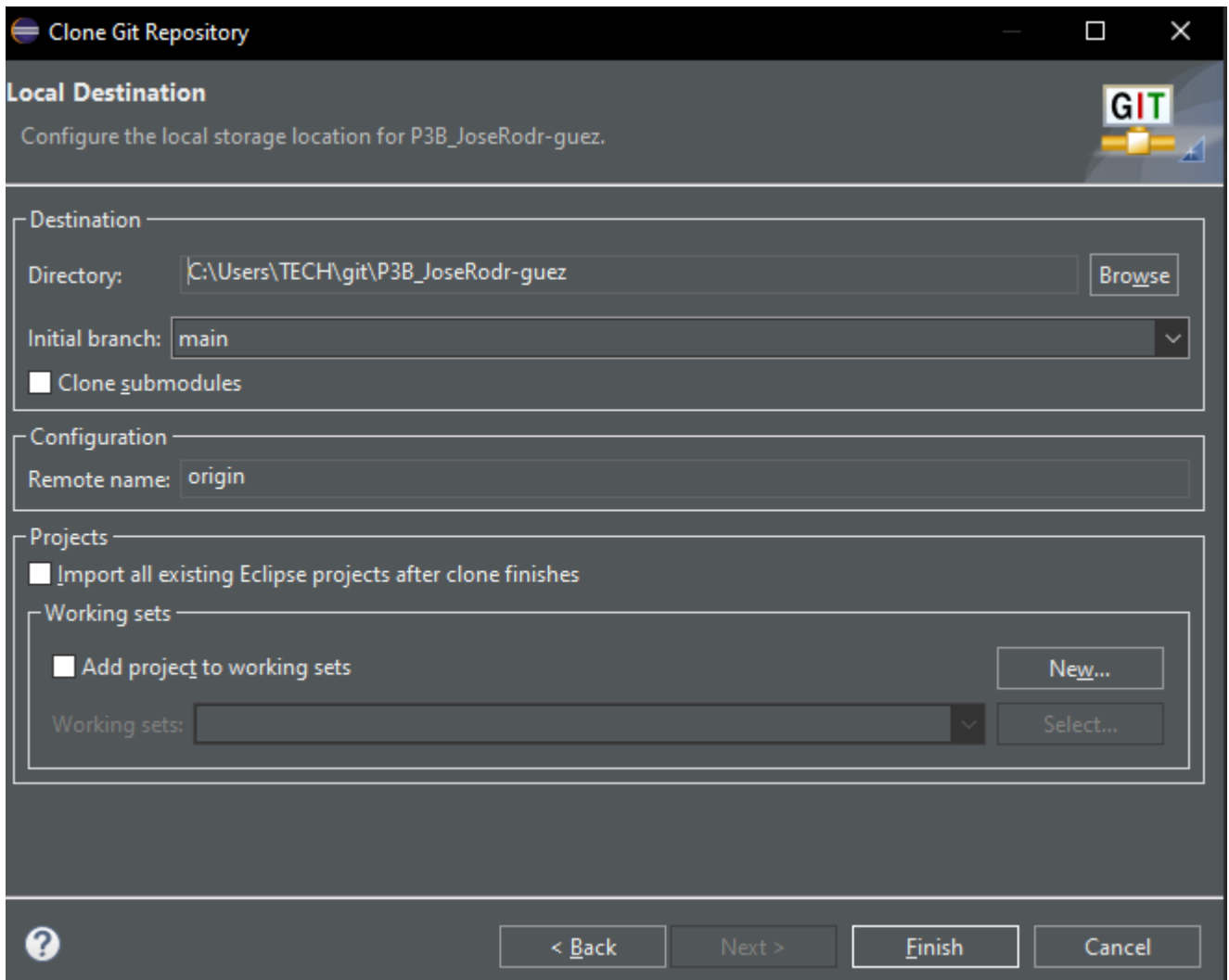
User:

Password:

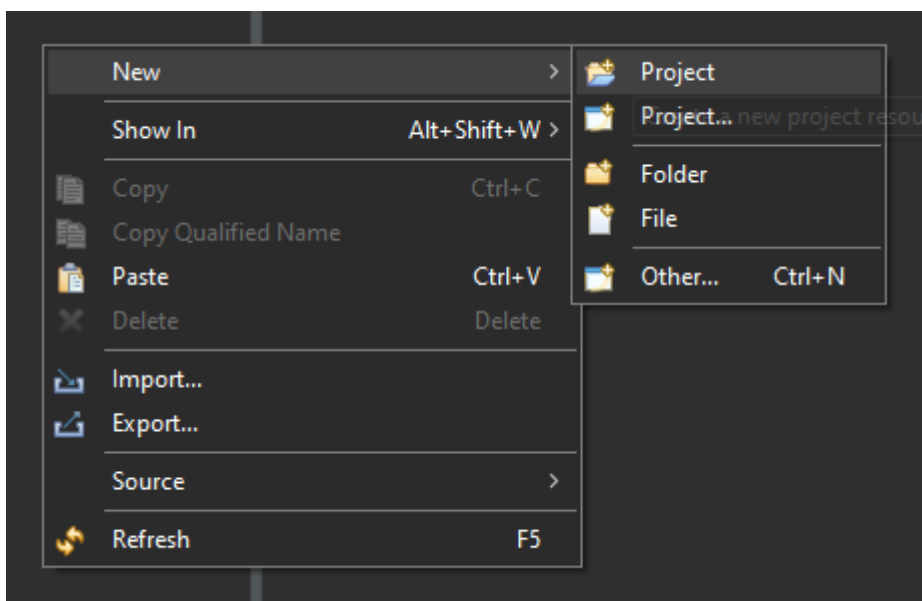
☐ Store in Secure Store

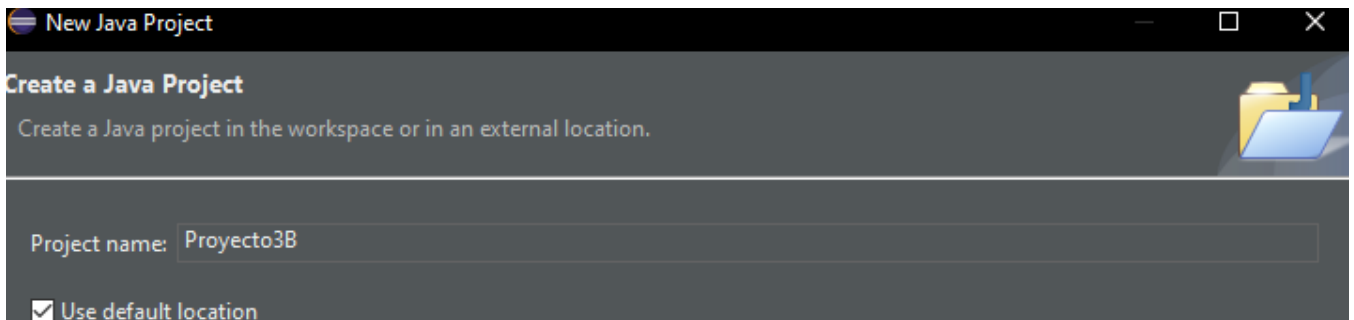
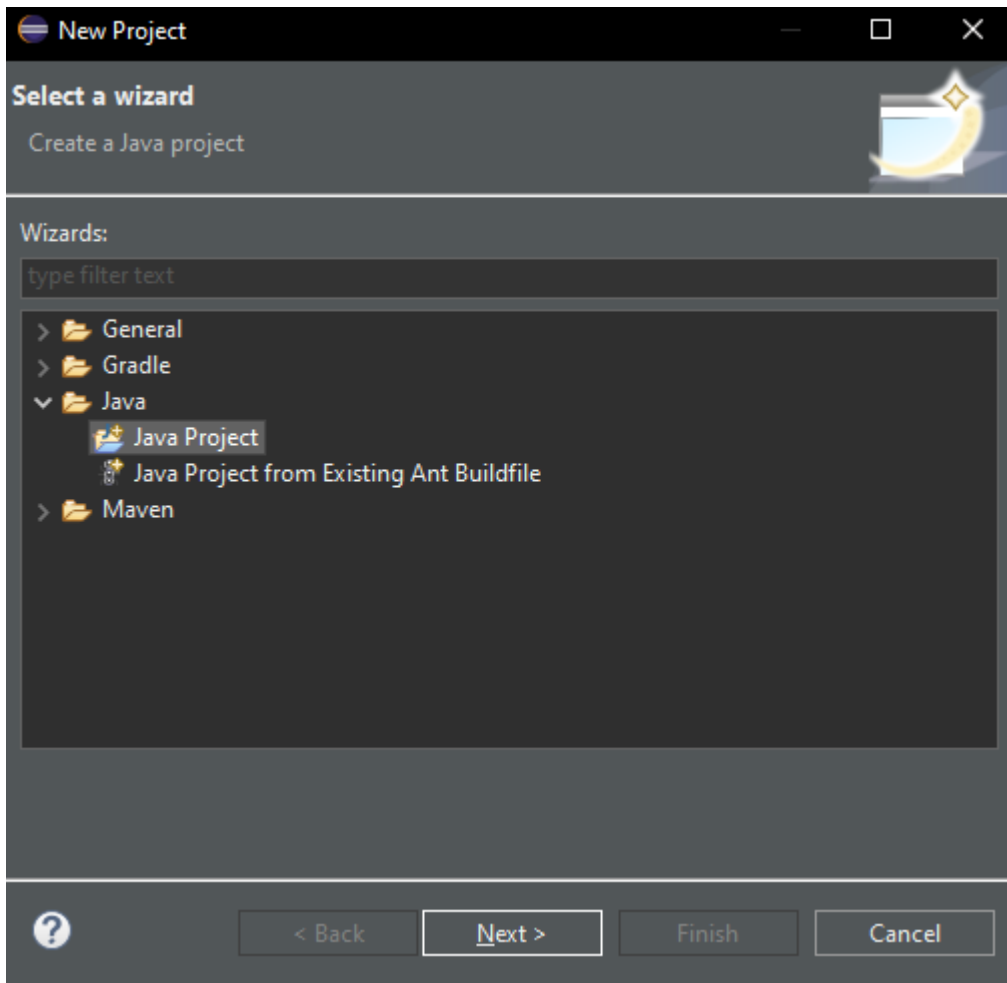
Le damos a "next" y se nos abrirá una nueva ventana en la cual elegimos la rama del proyecto. En este caso, sólo tenemos la principal, así que daremos sobre el "next".



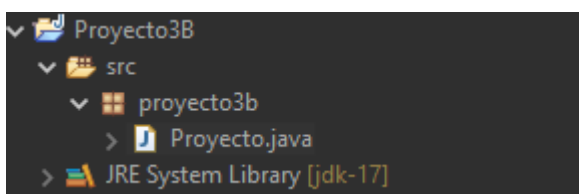


Y en el último paso, le damos a "finalizar" y quedaría completado. Con esto listo, ya hemos finalizado el proceso de clonar el repositorio. Ahora comenzamos con el proceso de crear el proyecto de java con su clase correspondiente. Para ello creamos el proyecto:





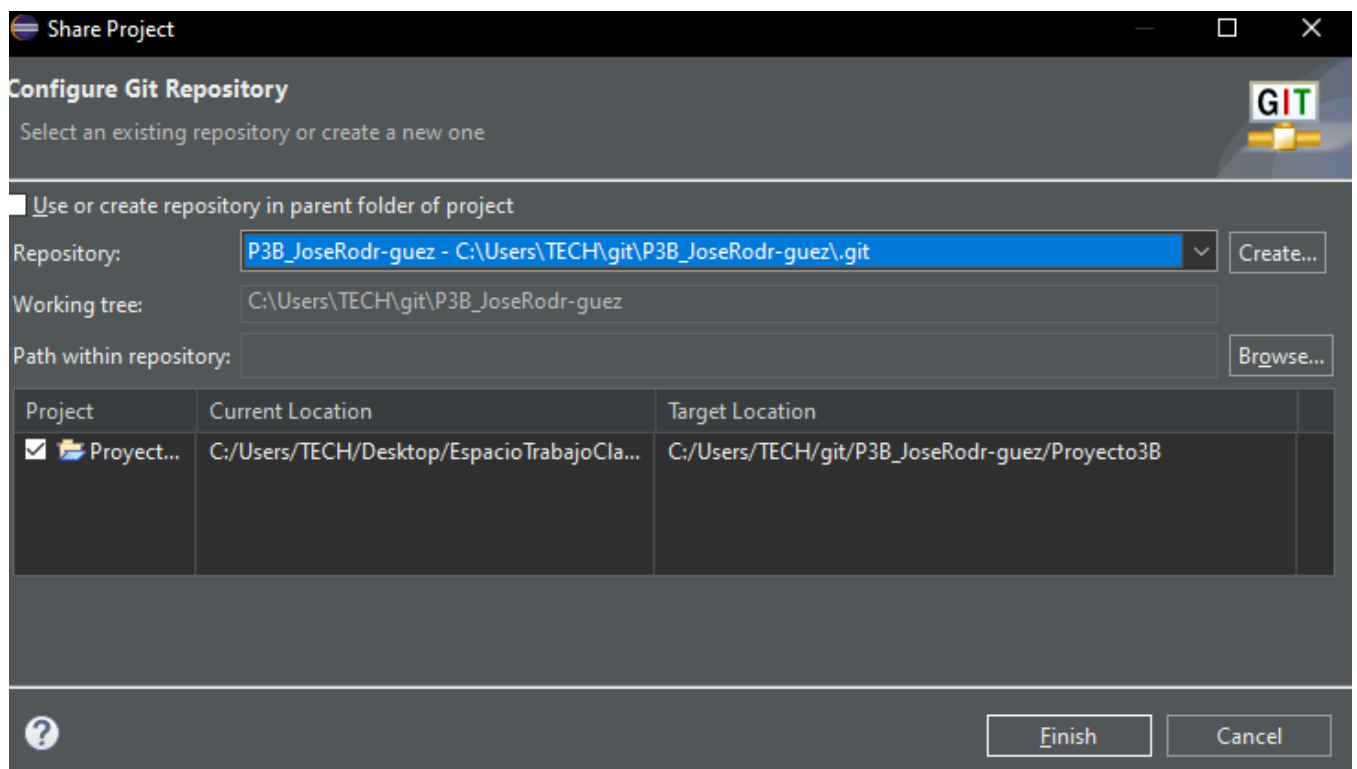
Una vez terminado, creamos una clase



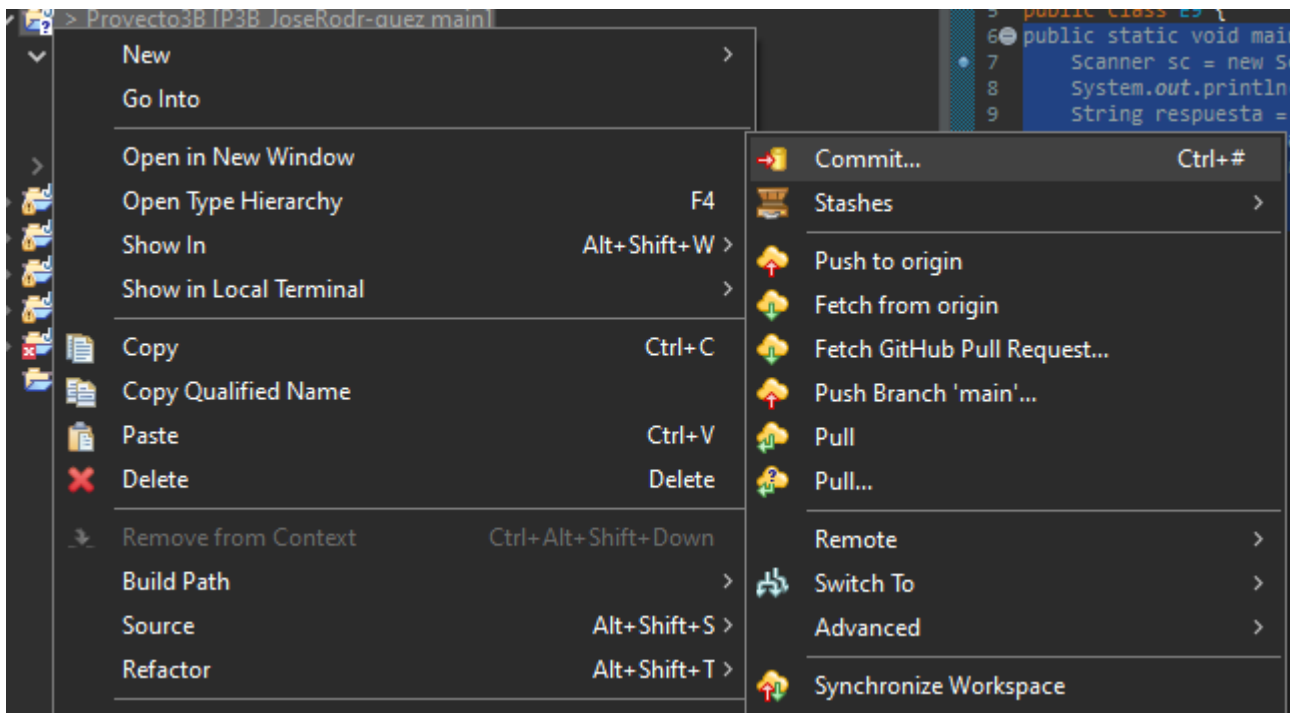
```
1 package proyecto3b;  
2  
3 public class Proyecto {  
4  
5     public static void main(String[] args) {  
6         System.out.println("Jose Rodríguez Torrejón");  
7     }  
8  
9 }  
10
```

Una vez hecho el proyecto, para subirlo al repositorio local debemos darle click derecho al proyecto y en las opciones de "team" darle a "commit" para guardar todos los cambios realizados en el repositorio local. Pero antes de eso, tenemos que vincular el proyecto de esta manera:

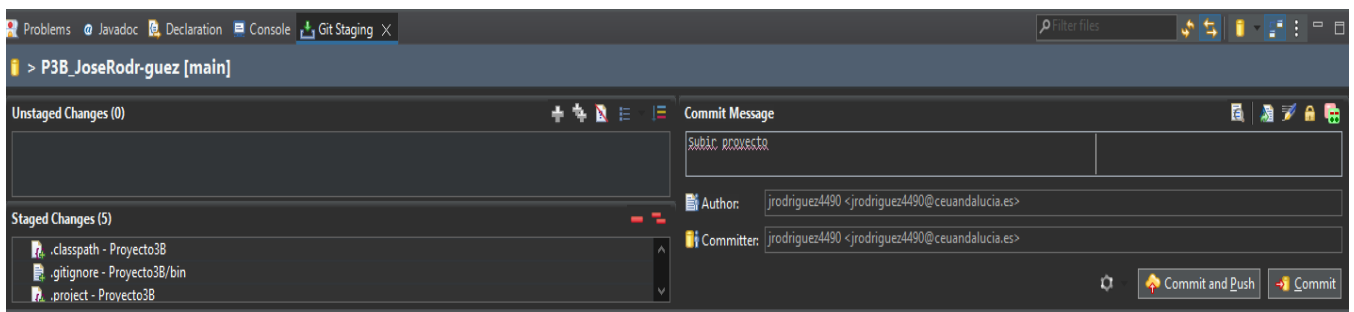
En "team", darle a "share project".



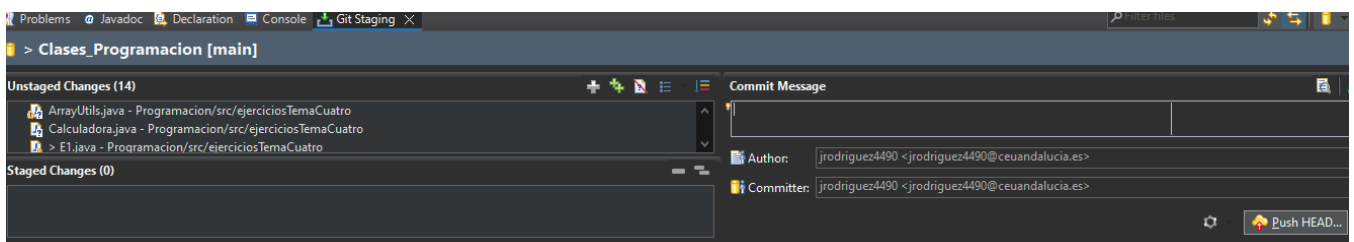
Buscar el repositorio local correspondiente para vincularlo a él. Una vez hecho esto, tenemos los archivos vinculados, pero no guardados en el repositorio local, para ello debemos de hacer un "commit" en primer lugar.



En la ventana "Git Staging" le damos a las dos cruces verdes para añadir el proyecto al guardado del repositorio local y escribir un comentario. Una vez hagamos esto, darle a "commit".

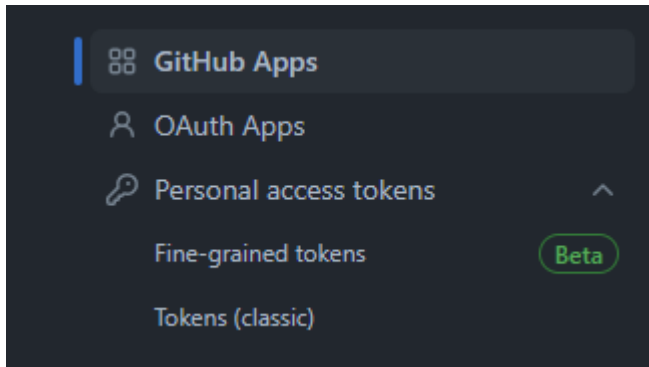


De este modo, lo hemos guardado en el repositorio local, pero no en el remoto. Para terminar, lo subiremos al remoto dándole a "push head".

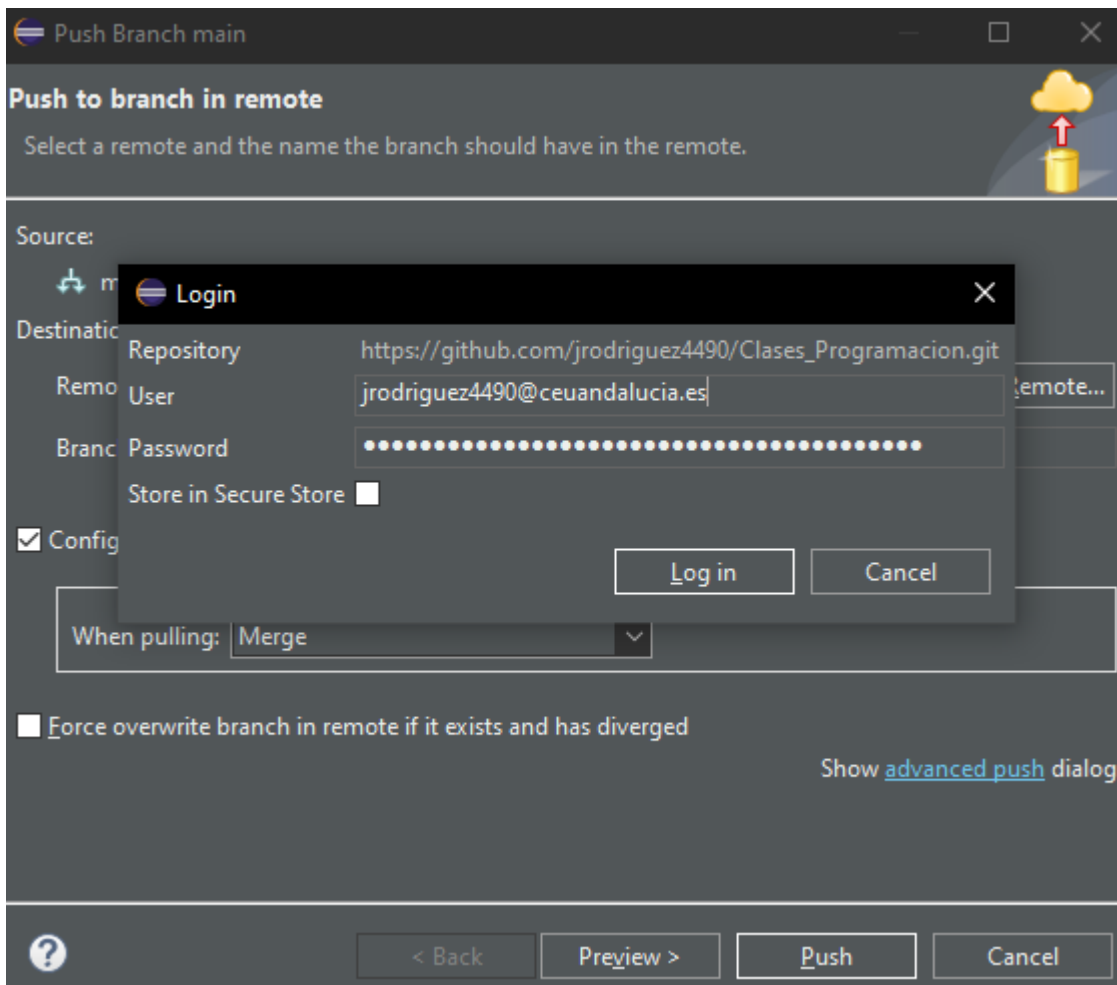


Ahora nos pedirá un usuario y una contraseña. La contraseña está configurada por un Token. El token es una clave que te genera Github, en "settings" -> Developer settings -> Personal Access tokens.





Generamos así un token para poder usarlo a modo de contraseña, y en el usuario ponemos nuestro correo de github.





Terminamos y le damos a "push".

Push Branch main

**Push to branch in remote**

Select a remote and the name the branch should have in the remote.

Source:

 main  8921e6f Dos tonterías más

Destination:

Remote:

Branch:

☒ Configure upstream for push and pull

When pulling:


☐ Force overwrite branch in remote if it exists and has diverged

[Show advanced push dialog](#)

Nos debería salir:

Push Results: https://github.com/jrodriguez4490/Clases\_Programacion.git

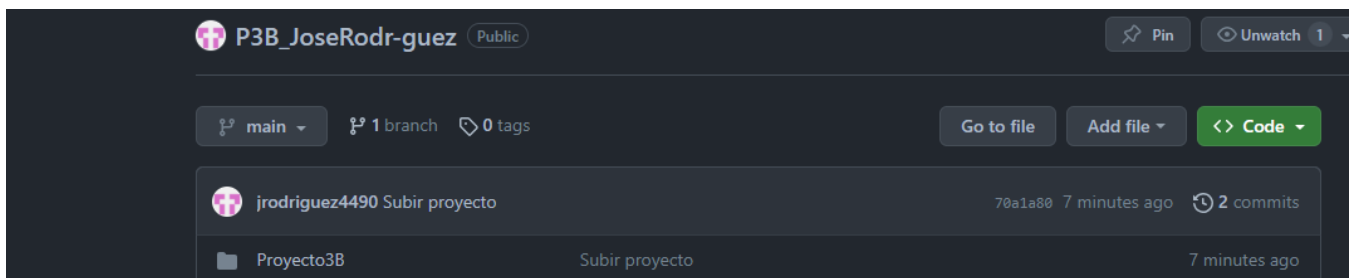
**Pushed to https://github.com/jrodriguez4490/Clases\_Programacion.git**

 main → main [up to date]

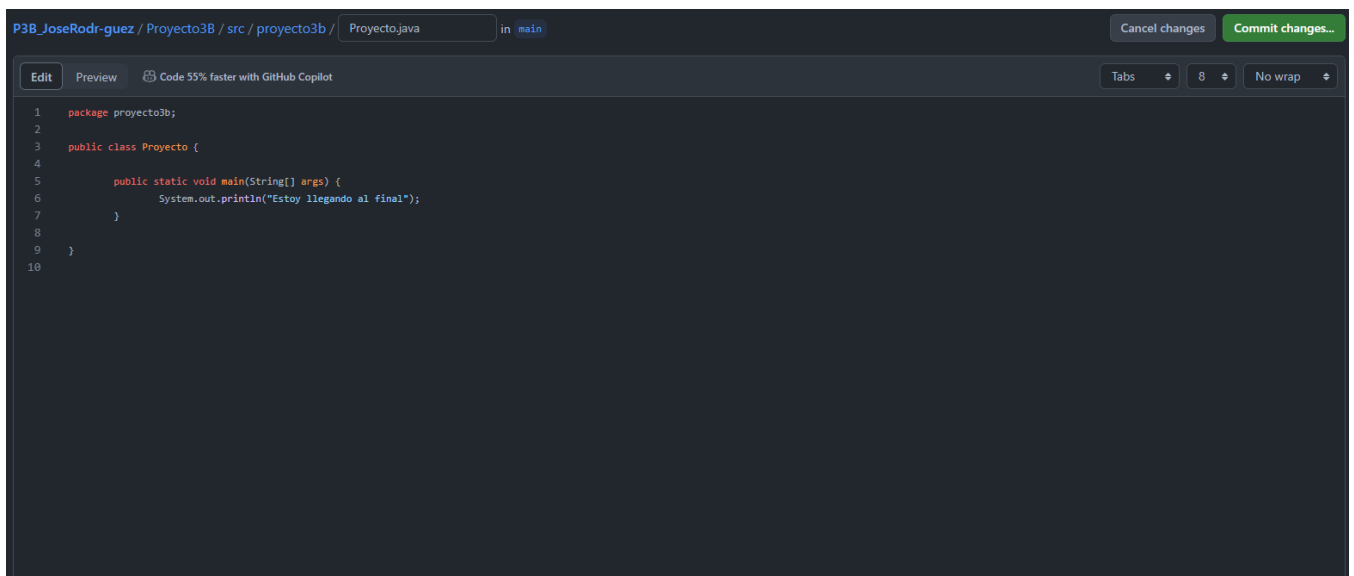
Message Details


Repository [https://github.com/jrodriguez4490/Clases\\_Programacion.git](https://github.com/jrodriguez4490/Clases_Programacion.git)

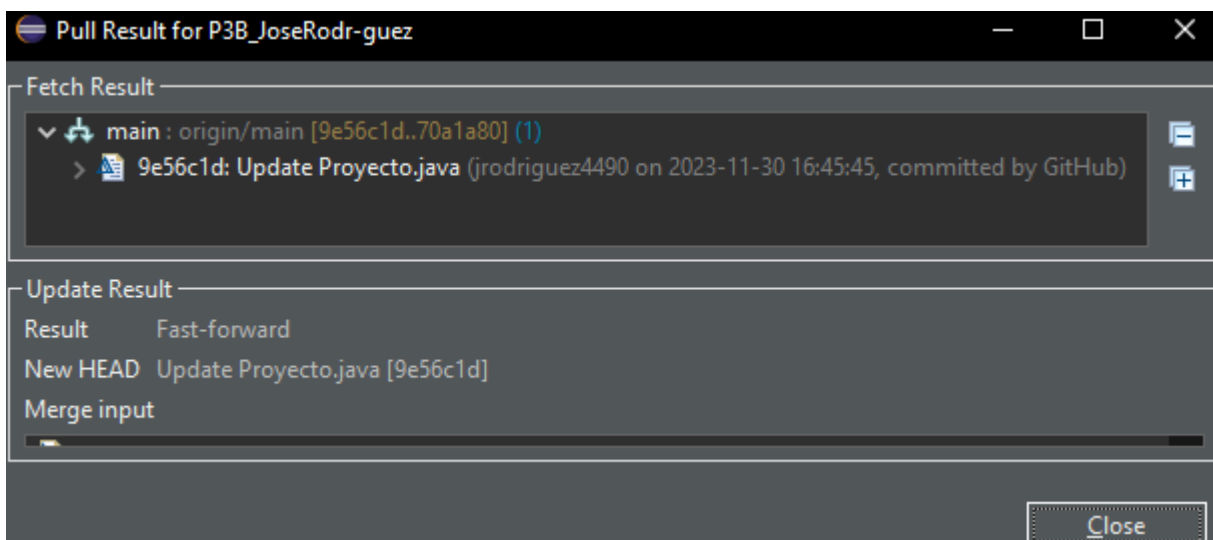
Comprobamos que se haya subido a Github:



Una vez en github, modificamos la clase para que imprima:



Seleccionamos "Commit changes y volvemos a Eclipse para hacer un "pull". Hacemos click derecho sobre nuestro proyecto, en la opción Team seleccionamos  Pull



Comprobamos que se nos haya actualizado la clase.

```
1 package proyecto3b;  
2  
3 public class Proyecto {  
4  
5     public static void main(String[] args) {  
6         System.out.println("Estoy llegando al final");  
7     }  
8  
9 }  
10
```

Y ya hemos terminado. [https://github.com/jrodriguez4490/P3B\\_JoseRodr-guez/tree/main](https://github.com/jrodriguez4490/P3B_JoseRodr-guez/tree/main)