

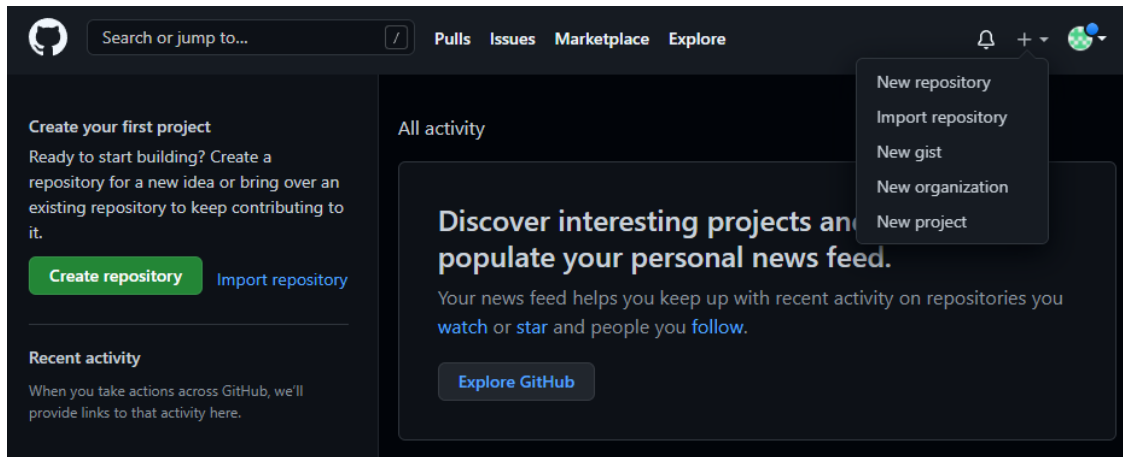
Enunciado: Práctica 5 Con un compañero subir un proyecto realizar modificaciones por separado y fusionarlas en el repositorio. Con los siguiente pasos haz capturas de esta práctica y explica los pasos para entregarla:

1. Javier: crea proyecto y empuja cambios a un repositorio remoto
2. Robert: clona proyecto
3. Javier: realiza cambios en el proyecto y los empuja al repositorio remoto (Javier debe compartir el repositorio y agregar los usuarios con los que desea compartir ese repositorio remoto)
4. Robert: trae cambios del repositorio remoto
5. Robert: modifica archivo y empuja cambios al repositorio remoto compartido
6. Javier: trae cambios del repositorio remoto

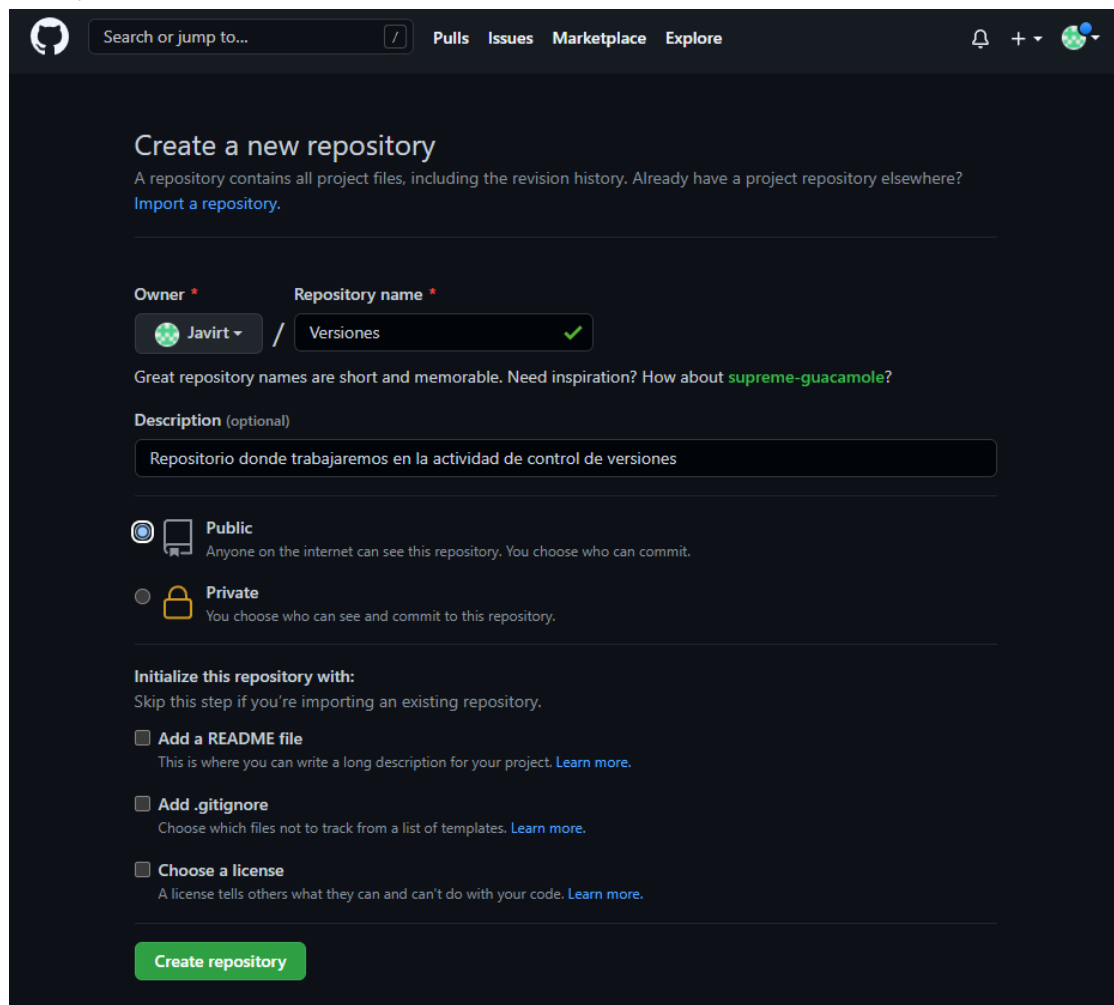
Nota: puedes añadir distintos tipos de archivos al proyecto no solo código.

Paso 1 : Javier: crea proyecto y empuja cambios a un repositorio remoto

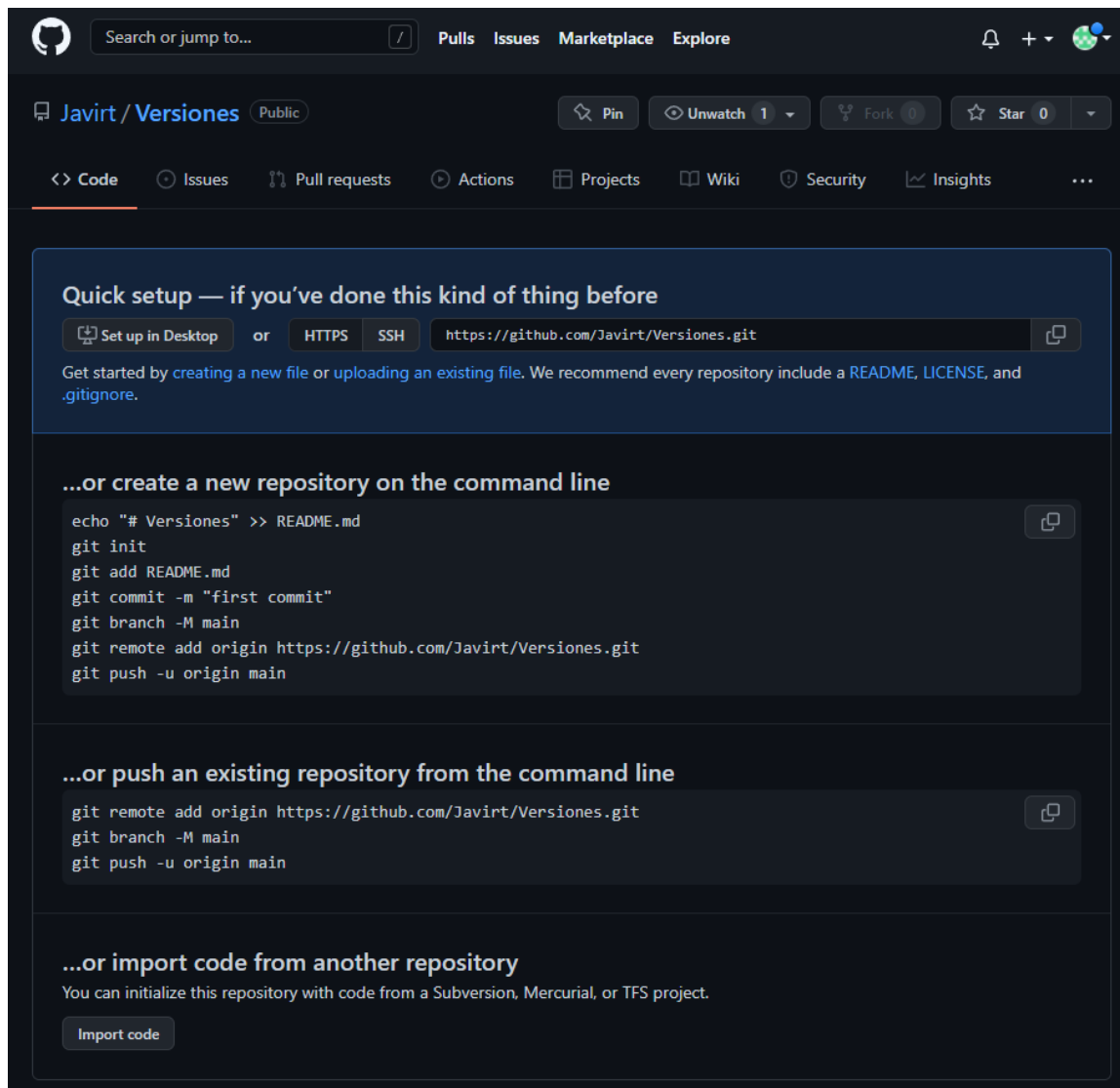
Primero crearemos un repositorio. Para ello, usaremos GitHub, daremos clic en "New Repository".



Luego, le asignamos un nombre sencillo, una descripción y si se trata de un proyecto público o privado. Además podemos añadir un archivo README, seleccionar qué archivos no queremos permitir y establecer una licencia.

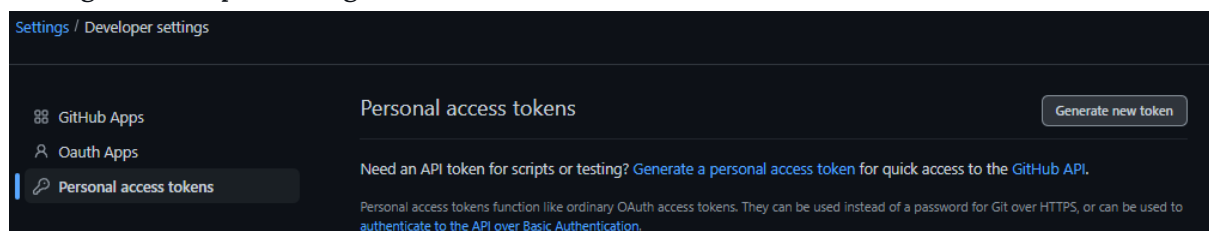


Al crear el repositorio, nos aparecerá una pantalla como esta donde debemos copiar el enlace que aparece en la sección “Quick setup”.



Crearemos unos tokens como alternativa al uso de contraseñas para la autenticación en GitHub cuando usemos NetBeans y seguiremos los siguientes pasos:

Settings > Developer settings > Personal access tokens > Generate new token



Le asignamos al token un nombre descriptivo, seleccionamos la fecha de expiración y seleccionamos los permisos que le otorgamos a ese token.

Nuevo token de acceso personal

Los tokens de acceso personal funcionan como tokens de acceso OAuth ordinarios. Se pueden usar en lugar de una contraseña para Git a través de HTTPS, o se pueden usar para autenticarse en la API a través de la autenticación básica.

Nota

TokeBeans

¿Para qué es esta ficha?

Vencimiento *

30 días

El token caducará el sábado 2 de abril de 2022

Seleccionar ámbitos

Los ámbitos definen el acceso para tokens personales. [Obtenga más información sobre los ámbitos de OAuth.](#)

- | | |
|--|--|
| <input checked="" type="checkbox"/> repositorio | Control total de repositorios privados |
| <input checked="" type="checkbox"/> repositorio: estado | Acceder al estado de confirmación |
| <input checked="" type="checkbox"/> repo_deployment | Acceder al estado de implementación |
| <input checked="" type="checkbox"/> public_repo | Acceder a repositorios públicos |
| <input checked="" type="checkbox"/> repositorio: invitar | Acceder a las invitaciones del repositorio |
| <input checked="" type="checkbox"/> eventos_de_seguridad | Leer y escribir eventos de seguridad |

Generar token

Cancelar

Settings / Developer settings

GitHub Apps

OAuth Apps

Personal access tokens

Personal access tokens

Generate new token

Revoke all

Tokens you have generated that can be used to access the GitHub API.

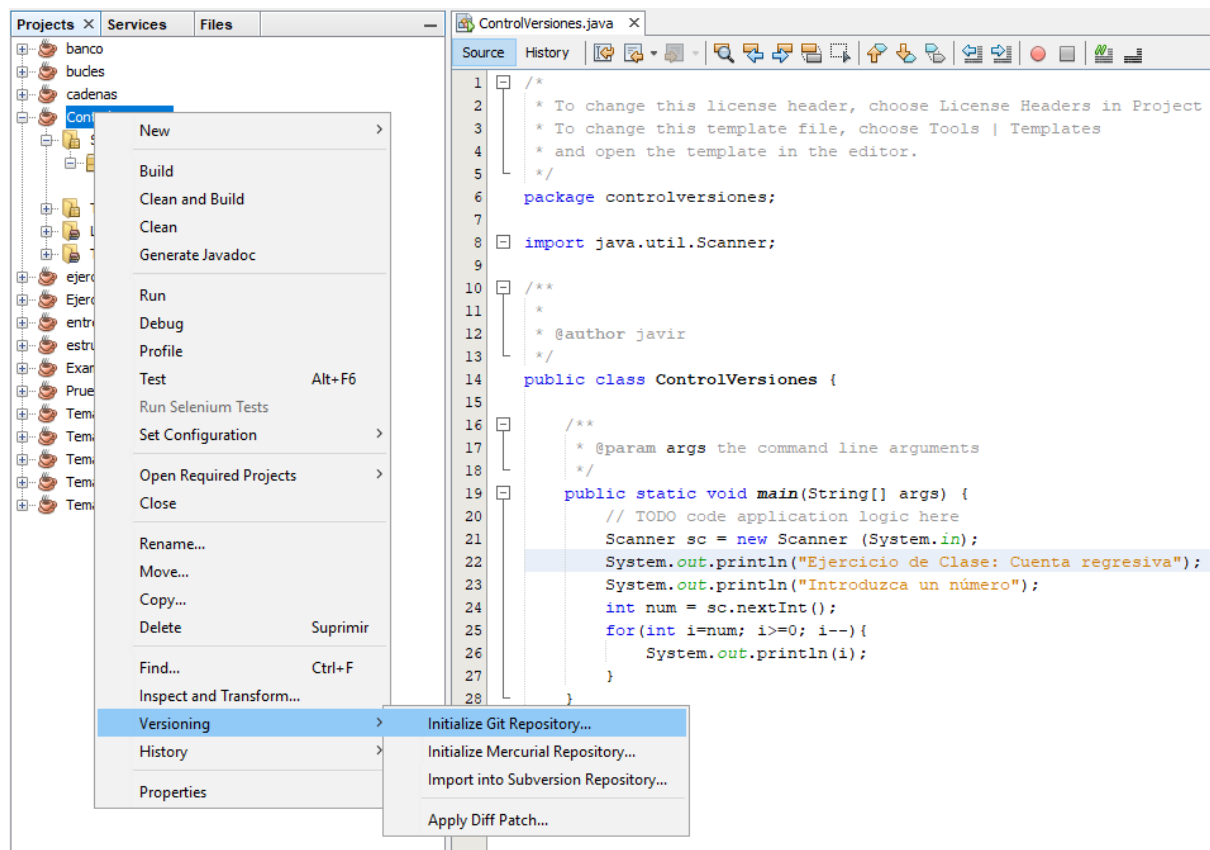
Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp_bCmqXwStMA1eNrvjcnTmHvmjbcR5yc1Y08nY

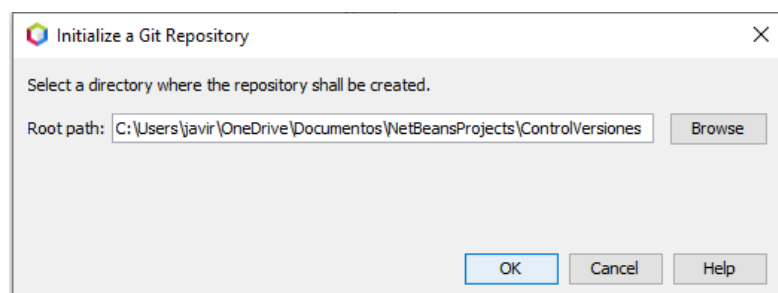
Delete

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

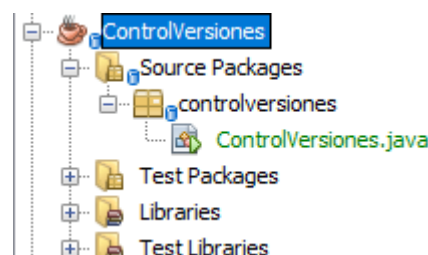
Crearemos un proyecto en NetBeans, le daremos clic derecho, abrimos el menú contextual y selecciona la opción Versioning > Initialize Git Repository



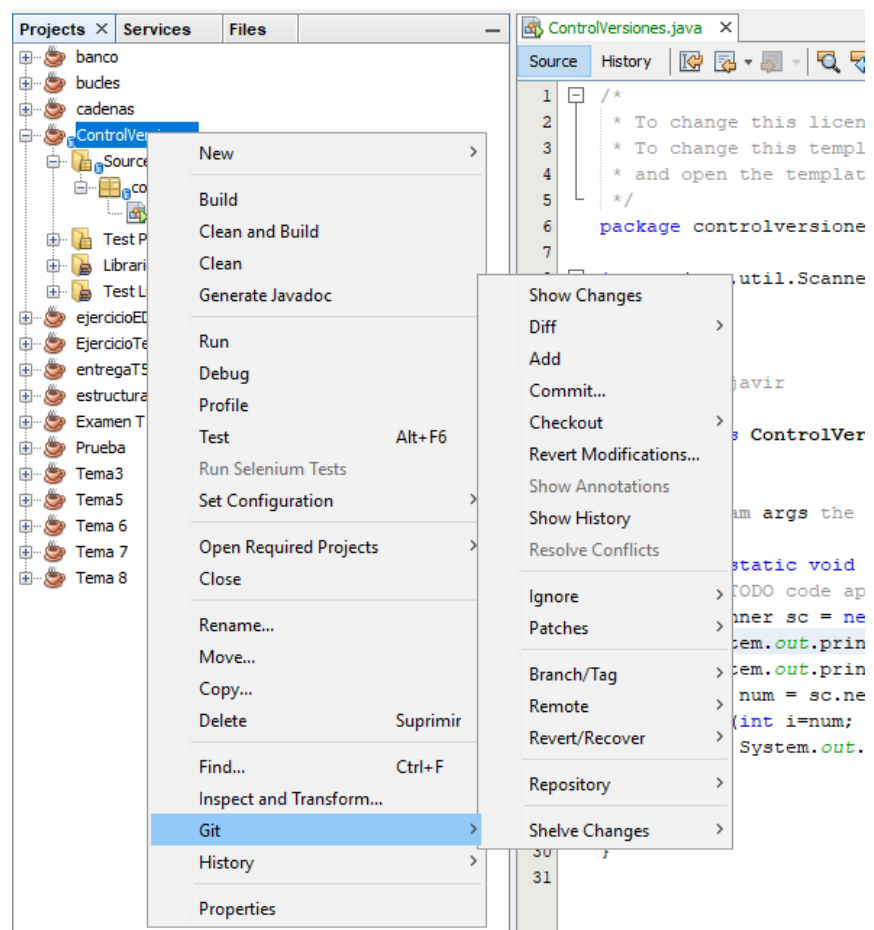
Para crear el repositorio local Git, se indicará en qué carpeta vamos a crear el repositorio. Lo crearemos en la misma carpeta en la que se encuentra el proyecto. En esta carpeta se creará automáticamente una carpeta llamada `.git` con la información correspondiente al repositorio local Git del proyecto.



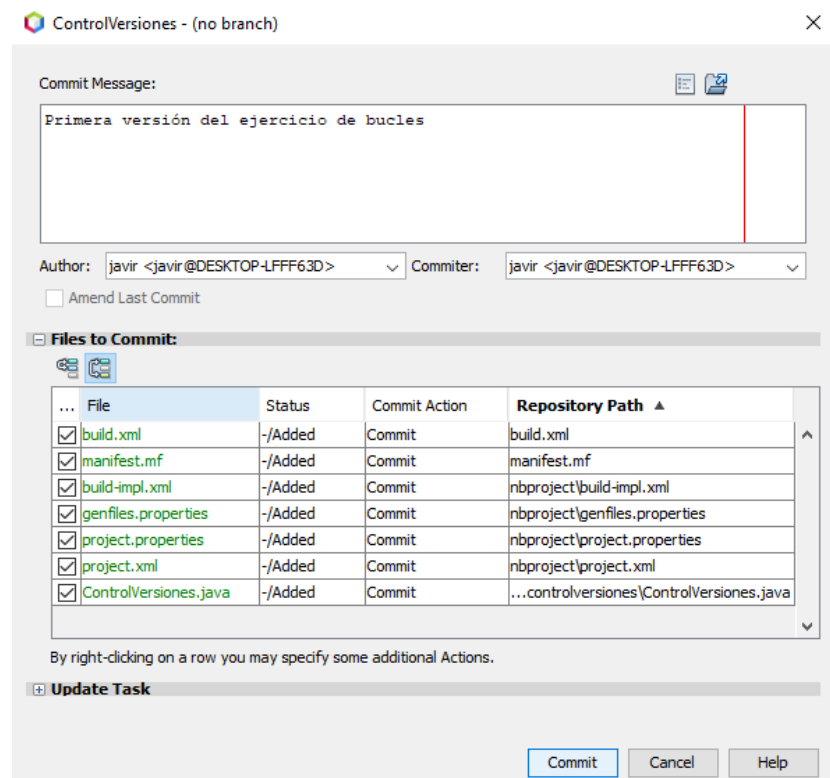
Una vez creado el repositorio local Git para el proyecto, aparecerán una serie de pequeños iconos azules en el proyecto y paquetes y los archivos que contenga el proyecto aparecerán en verde, indicando que son archivos que aún no se han añadido al nuevo repositorio local Git.



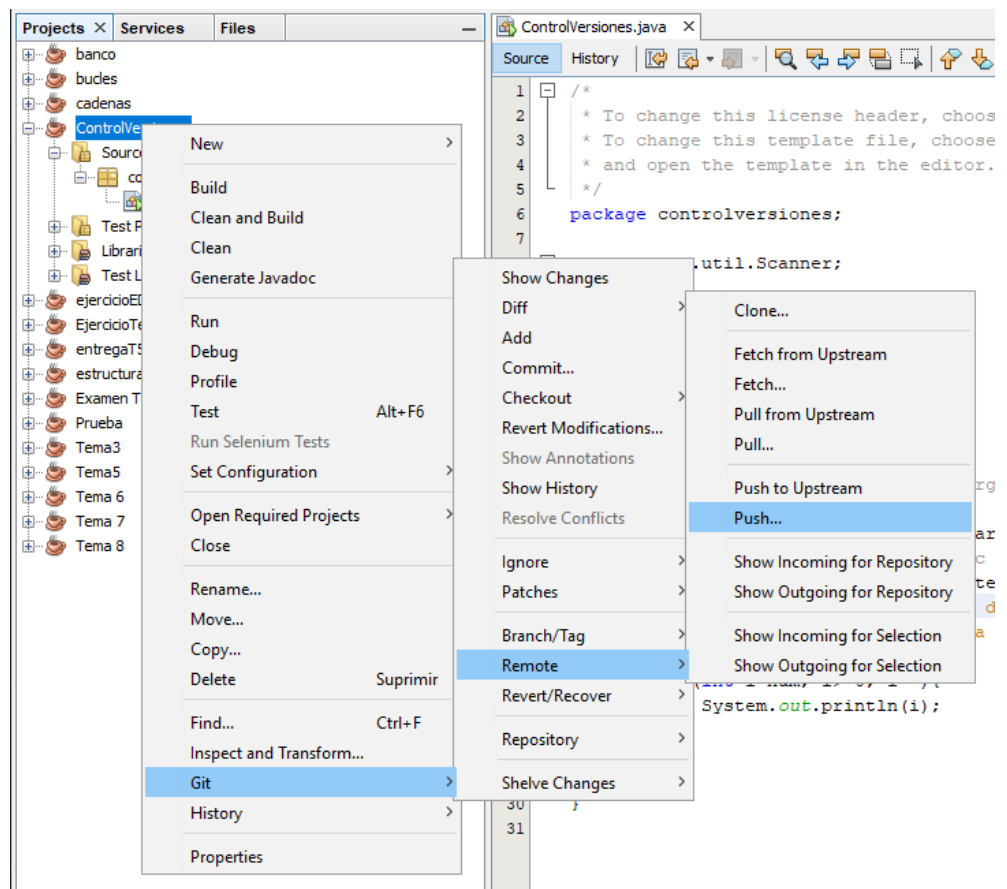
Para crear una nueva versión del proyecto se deberá hacer un volcado de los archivos (commit) del proyecto al repositorio local Git, pulsando desde el menú contextual del proyecto la opción Git > Commit.



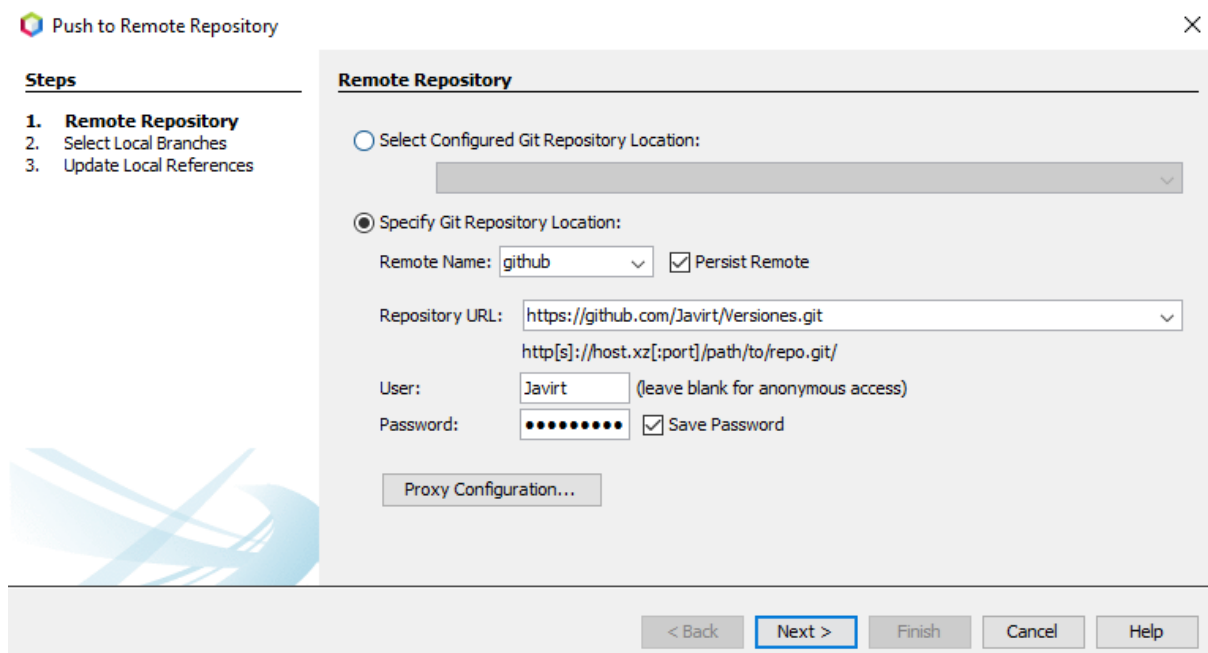
Durante el commit se indicará con un mensaje la versión que se va a almacenar en el repositorio local, así como los archivos que se van a almacenar en él.



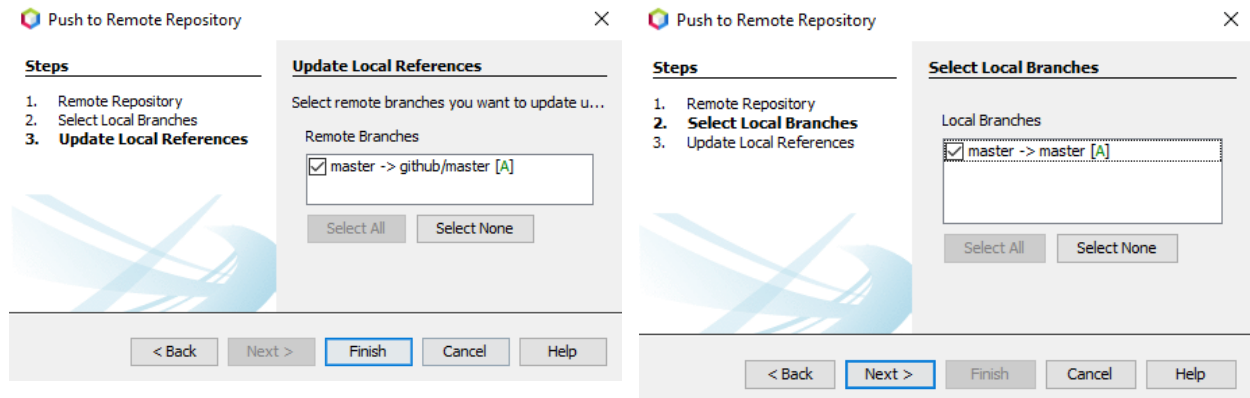
Una vez realizado el commit seleccionamos en el menú contextual la opción Git > Remote > Push.



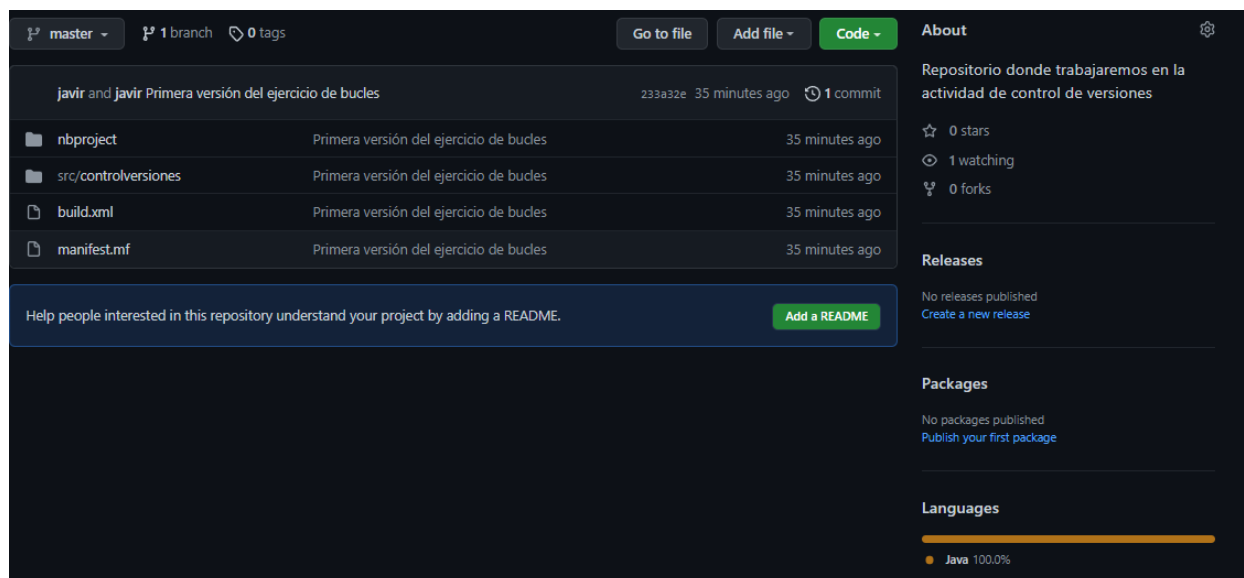
A continuación pegaremos en el cuadro de texto “Repository URL” el enlace que proporcionó GitHub. Insertamos el nombre de usuario y el token de acceso a GitHub.



En esta pantallas indicamos la rama que vamos enviar al repositorio remoto. Si no hemos creado nuevas ramas en el proyecto, seleccionamos las opciones ofrecidas (master).

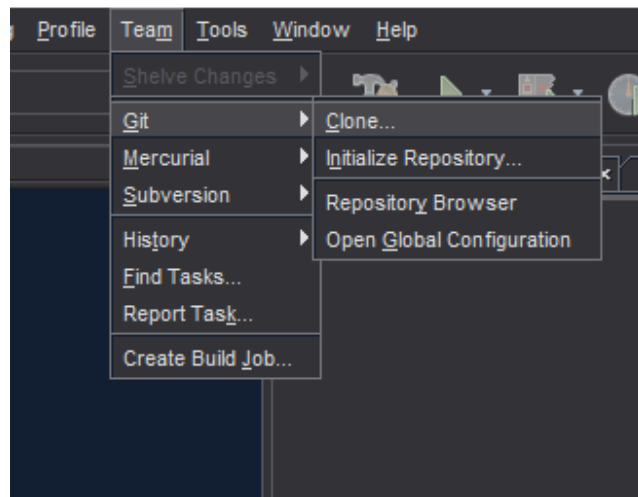


Una vez finalizado estos pasos encontraremos en la web de GitHub el código fuente de la versión del proyecto enviado.

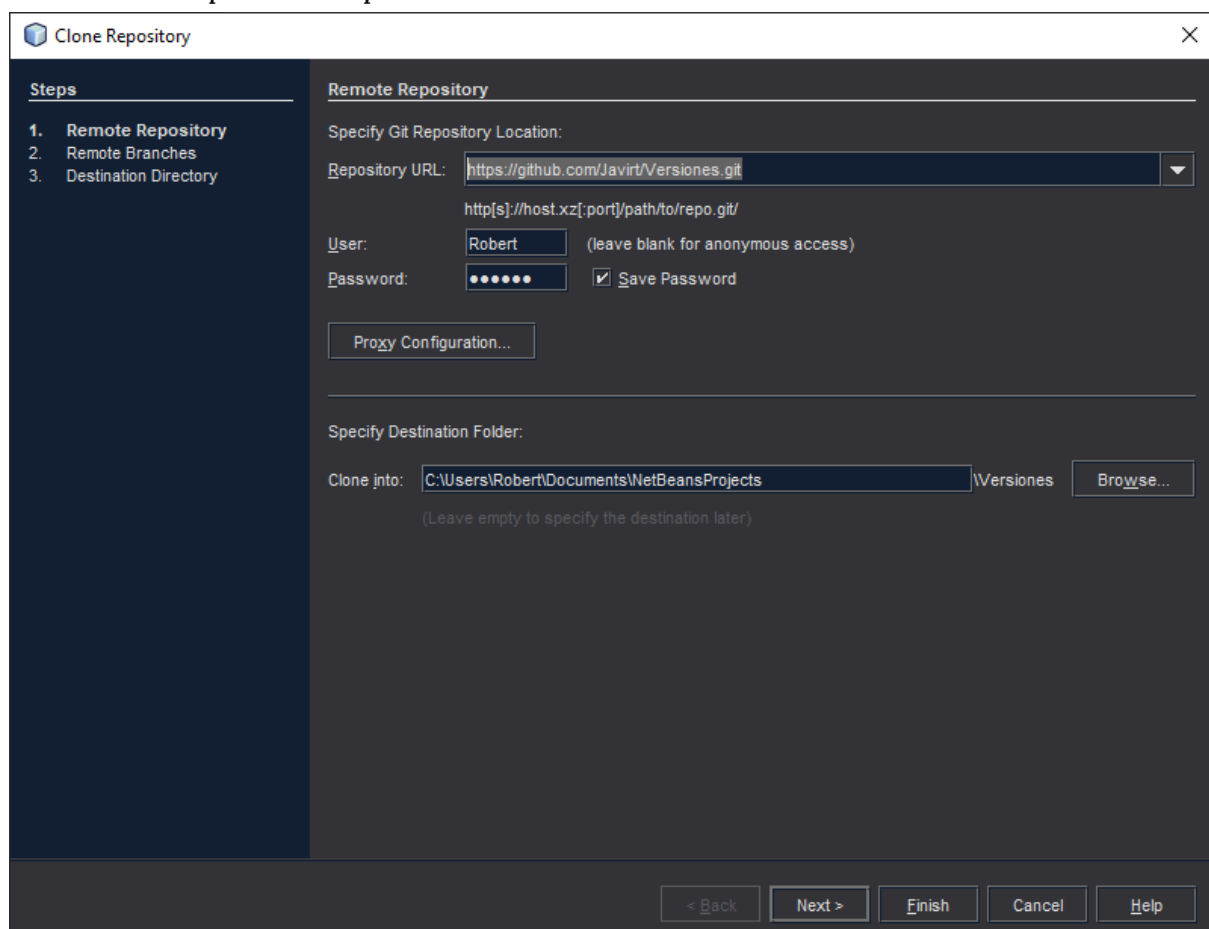


Paso 2 : Robert: clona proyecto

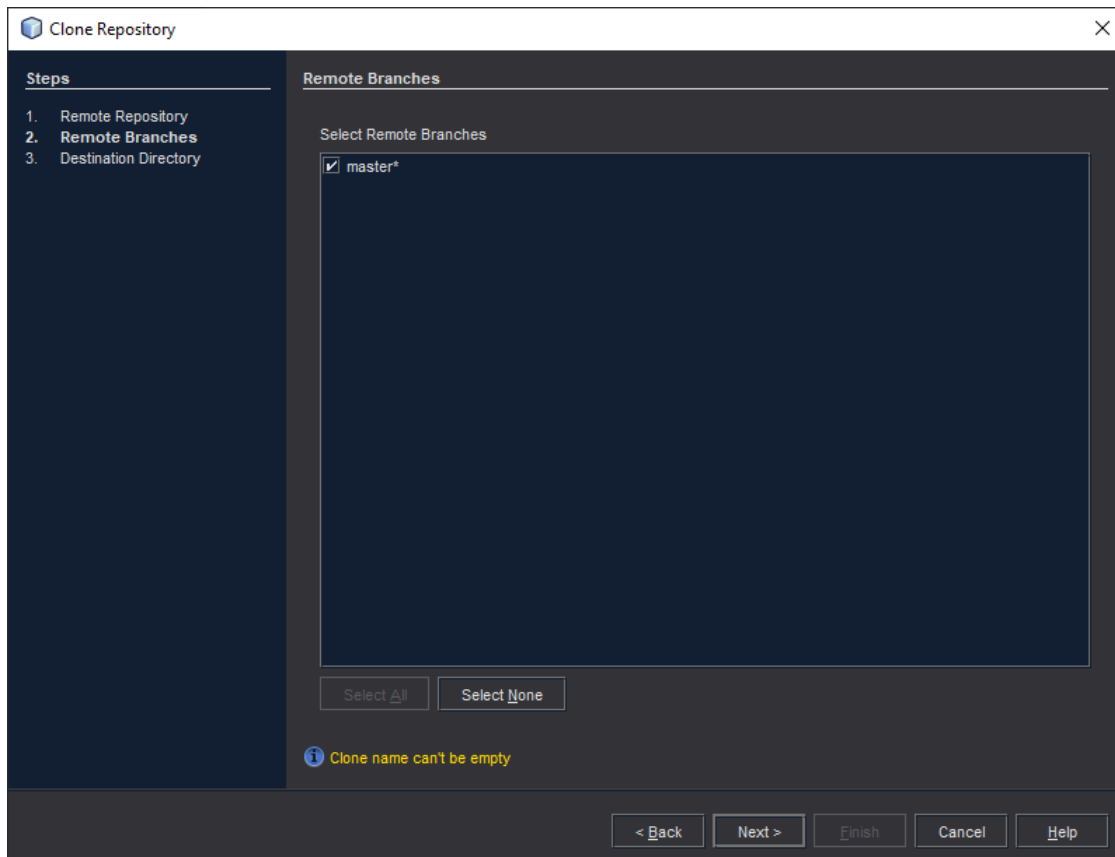
Abrimos Netbeans y nos dirigimos a la pestaña “Team” y seleccionamos “Git” y “Clone” para clonar nuestro proyecto.



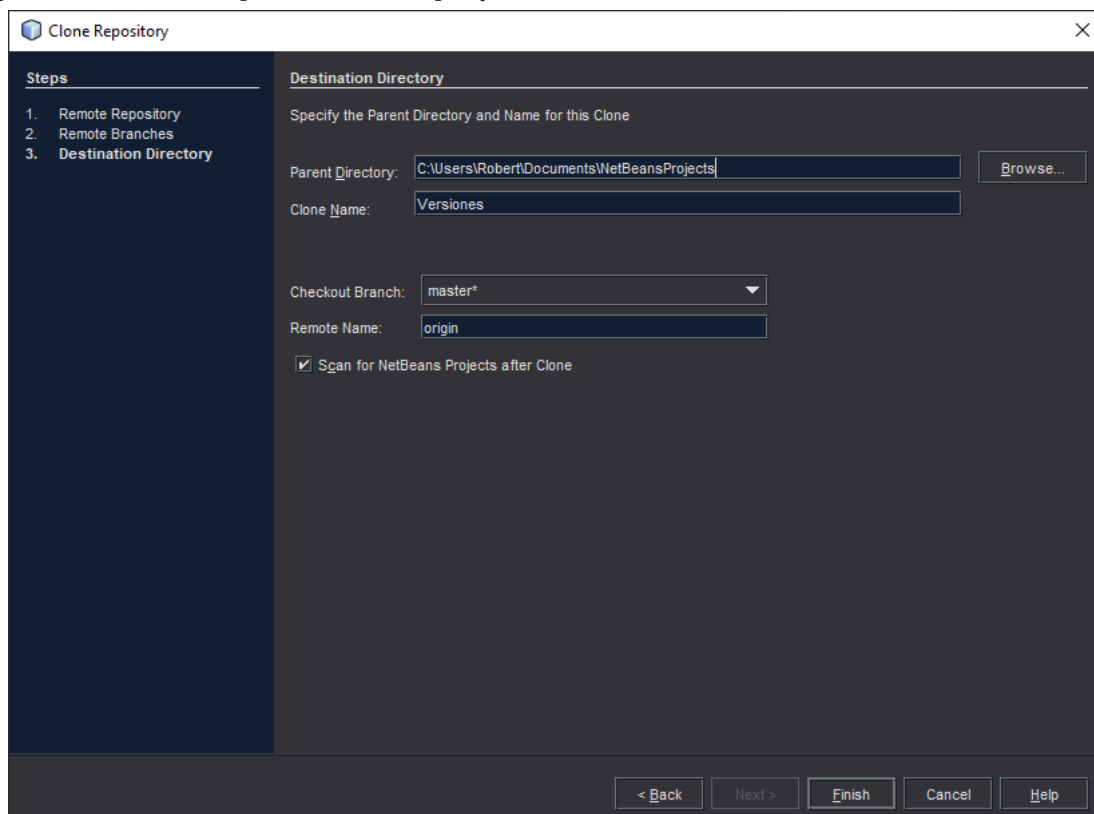
Introducimos la URL de nuestro proyecto, un nombre de usuario y una contraseña. Por último indicamos la carpeta donde queremos clonar la información.



Seleccionamos las ramas que queremos importar. En este caso sólo disponemos de una.



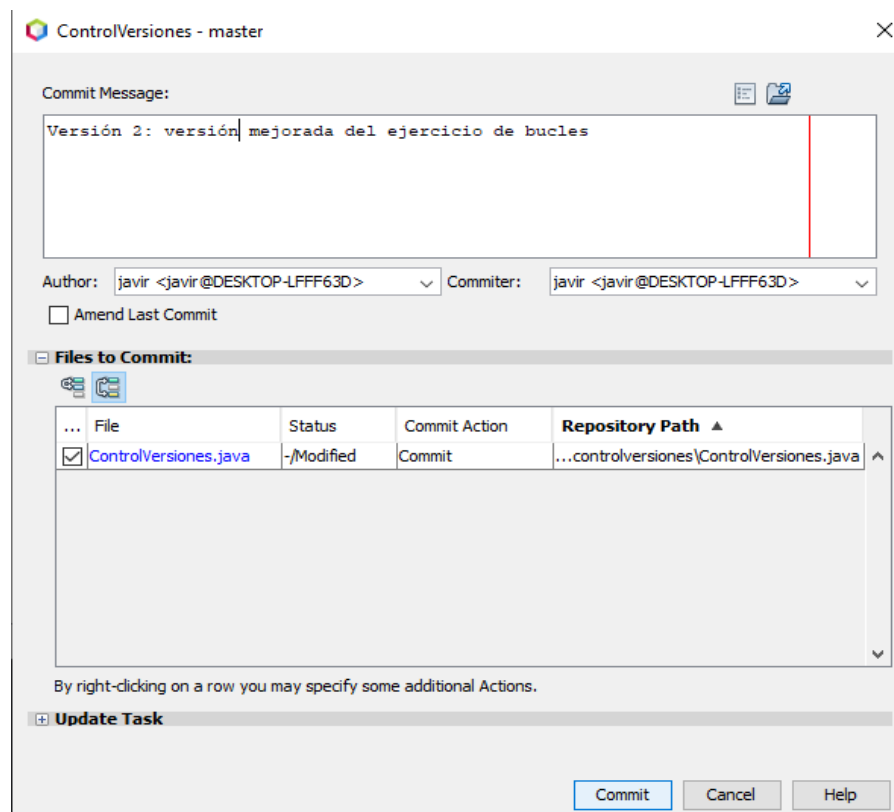
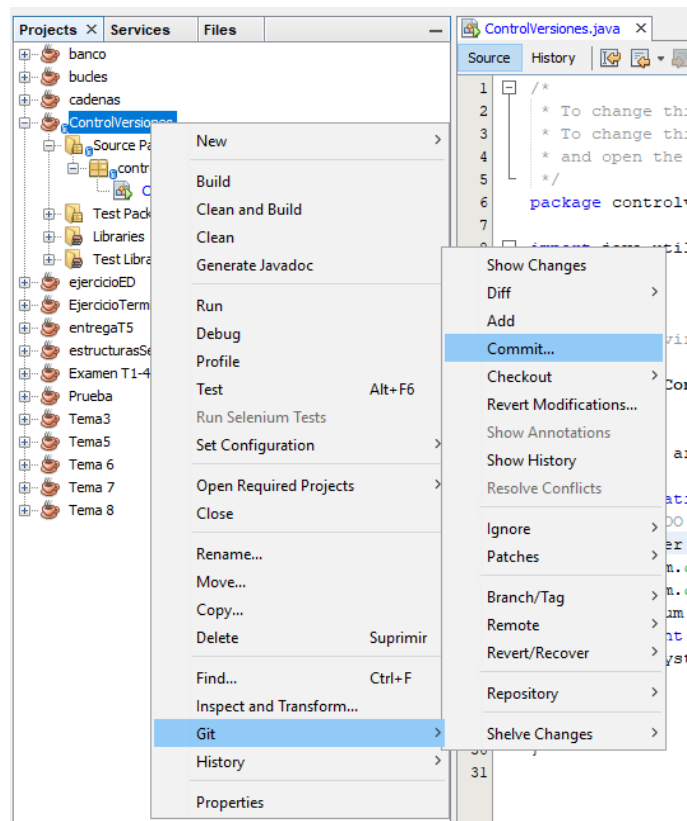
Asignamos un nombre para nuestra copia y le damos a finalizar.



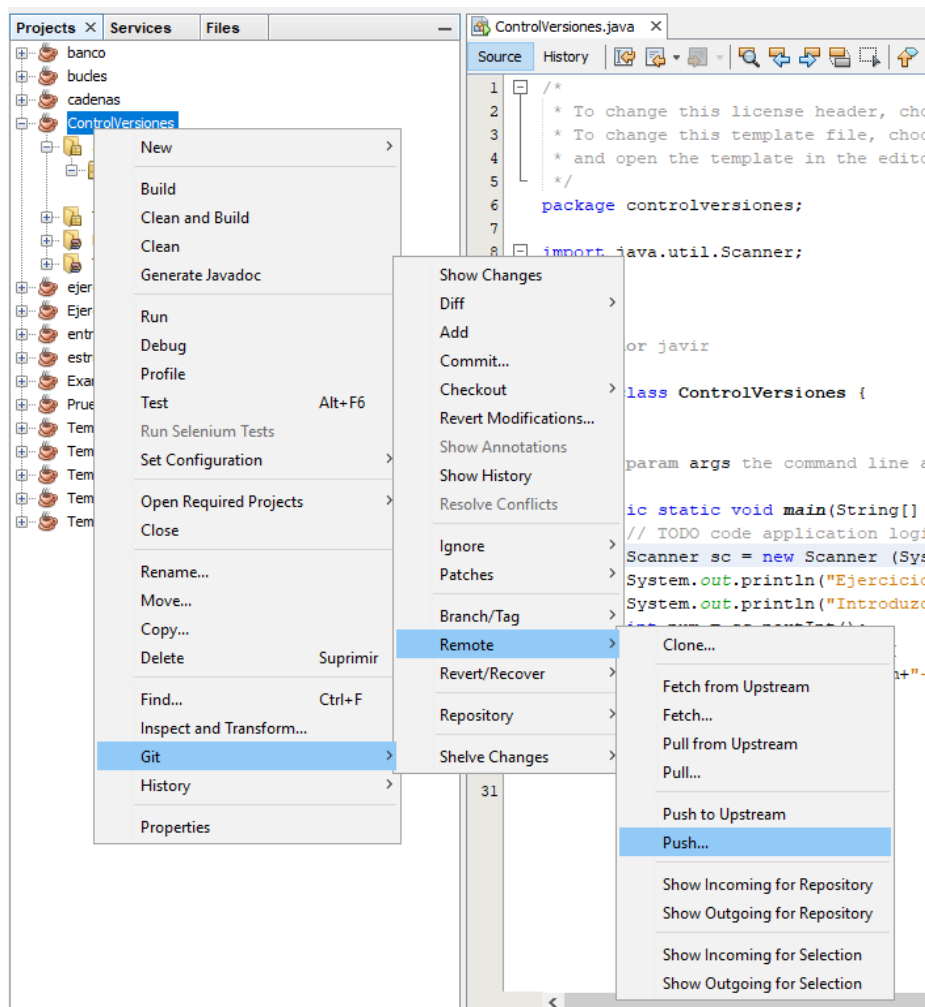
Paso 3 : Javier: realiza cambios en el proyecto y los empuja al repositorio remoto (Javier debe compartir el repositorio y agregar los usuarios con los que desea compartir ese repositorio remoto)

Realizaremos cambios en el código y haremos un commit y escribiremos una descripción para la versión subida.

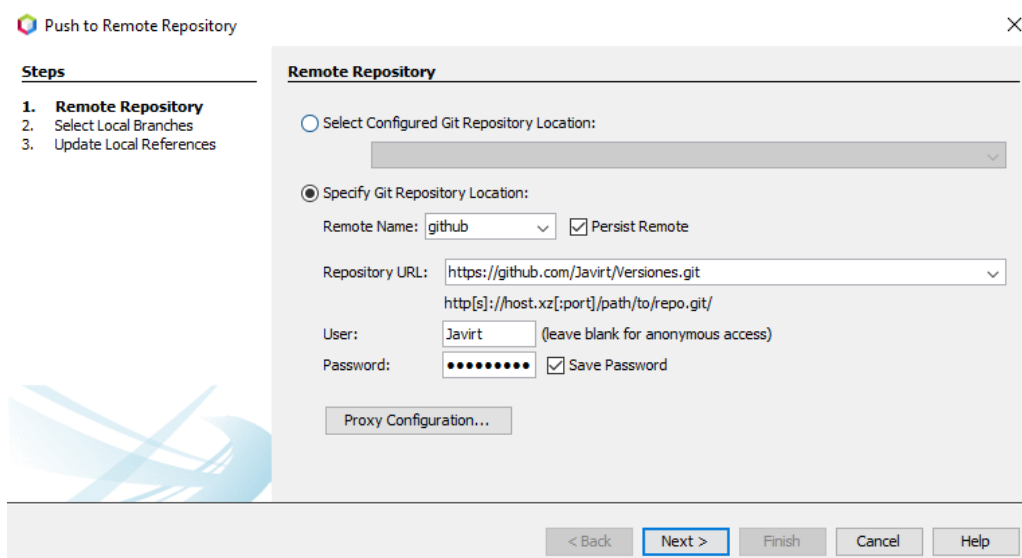
En el commit pondremos en mensaje de modo que sepamos que es otra versión.



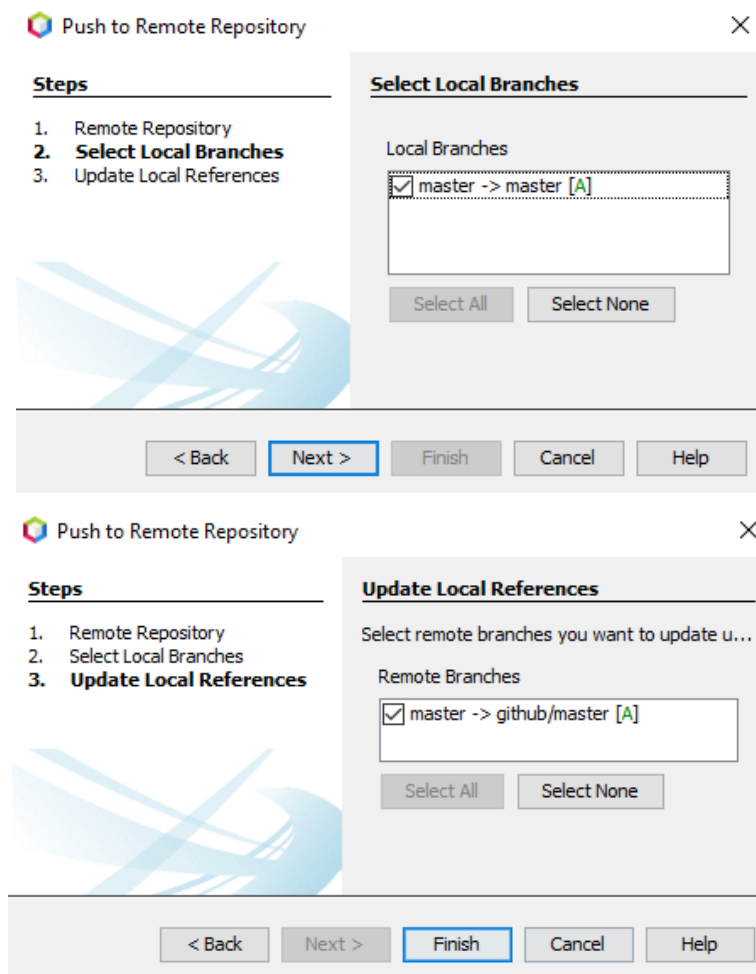
Luego lo enviaremos al repositorio remoto pulsando en Git > Remote > Push.



A continuación pegaremos en el cuadro de texto “Repository URL” el enlace que proporcionó GitHub. Insertamos el nombre de usuario y el token de acceso a GitHub.



En esta pantallas indicamos la rama que vamos enviar al repositorio remoto. Si no hemos creado nuevas ramas en el proyecto, seleccionamos las opciones ofrecidas (master).



En el repositorio remoto de GitHub veremos que se producen los cambios realizados.

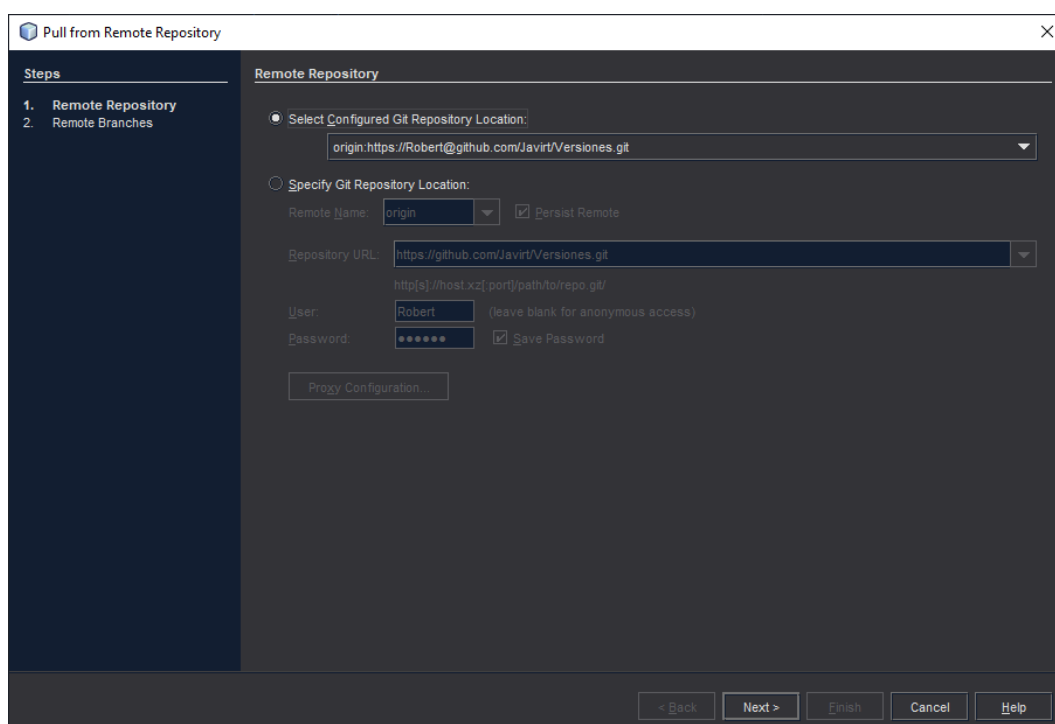
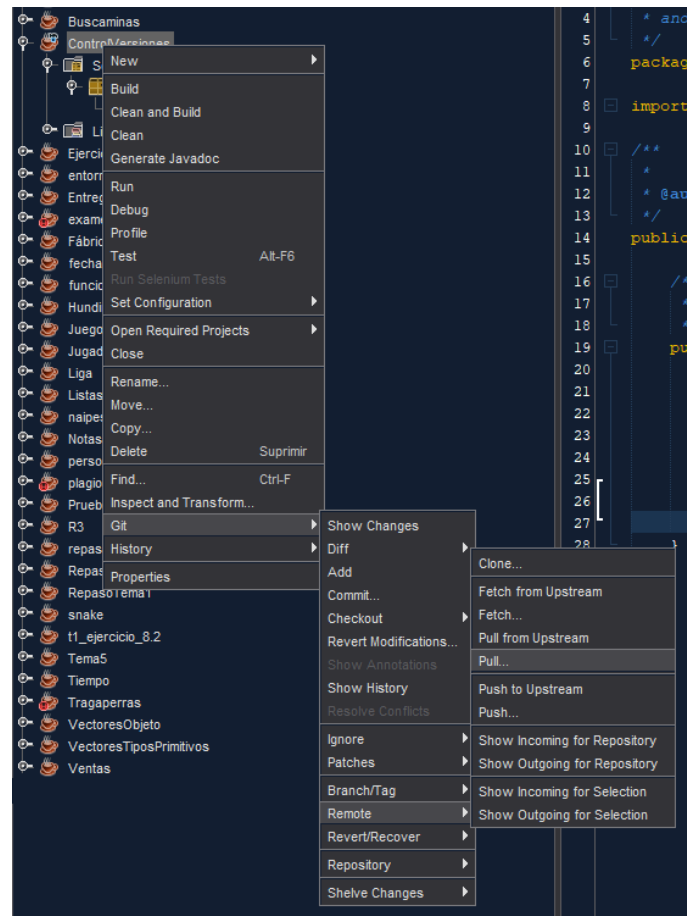
javir and javir Versión 2: versión mejorada del ejercicio de bucles		9d0bb9b 35 minutes ago	🕒 2 commits
nbproject	Primera versión del ejercicio de bucles		2 hours ago
src/controlversiones	Versión 2: versión mejorada del ejercicio de bucles		35 minutes ago
build.xml	Primera versión del ejercicio de bucles		2 hours ago
manifest.mf	Primera versión del ejercicio de bucles		2 hours ago

Paso 4 : Robert: trae cambios del repositorio remoto

Hacemos clic derecho en nuestro proyecto y abrimos la pestaña Git> Remote> Pull.

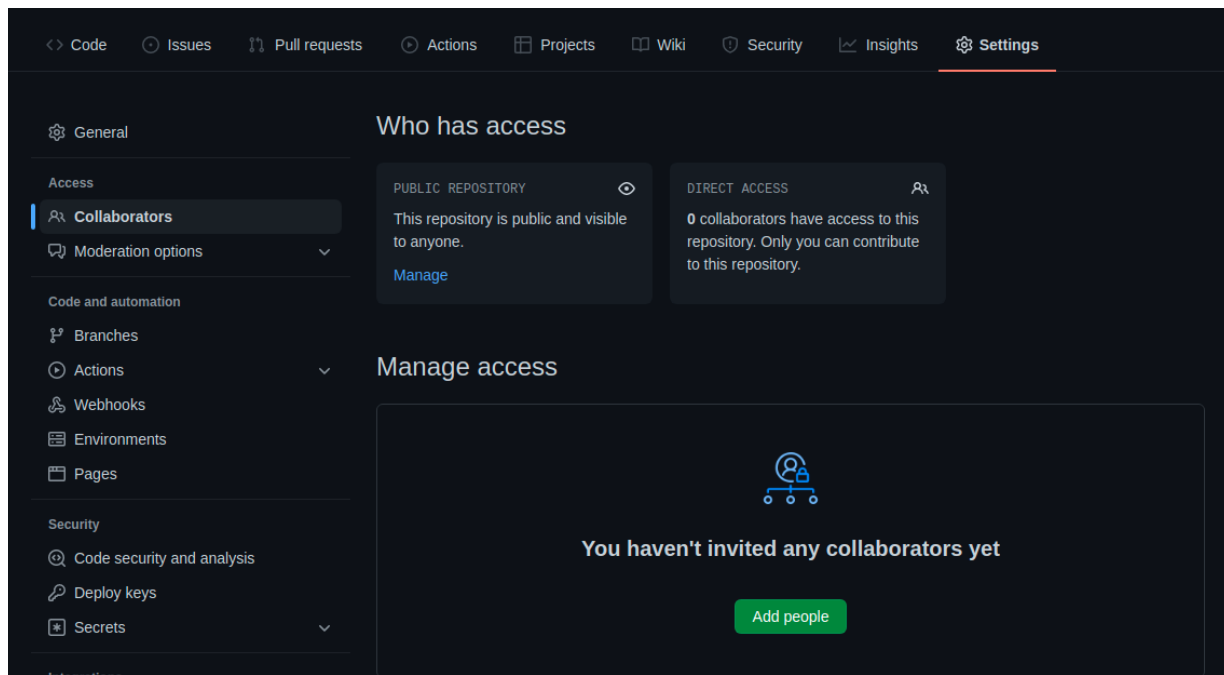
Seleccionamos la localización del repositorio configurado que aparecerá por defecto y hacemos clic en continuar.

Por último nos preguntará por las ramas que queremos actualizar. Seleccionamos la única rama que tenemos y finalizamos la actualización de nuestro repositorio local.

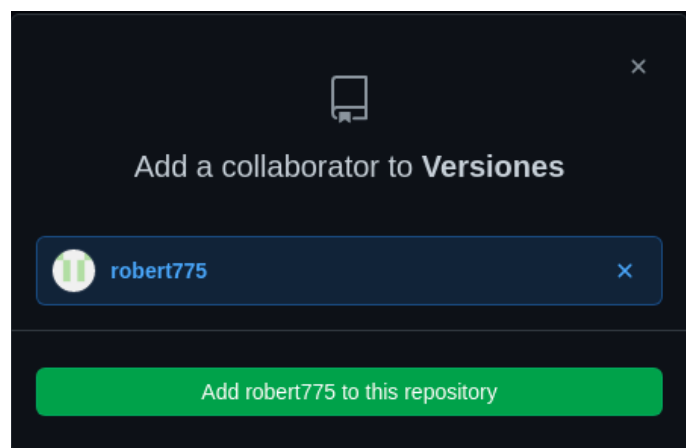


Paso 5 : Robert: modifica archivo y empuja cambios al repositorio remoto compartido

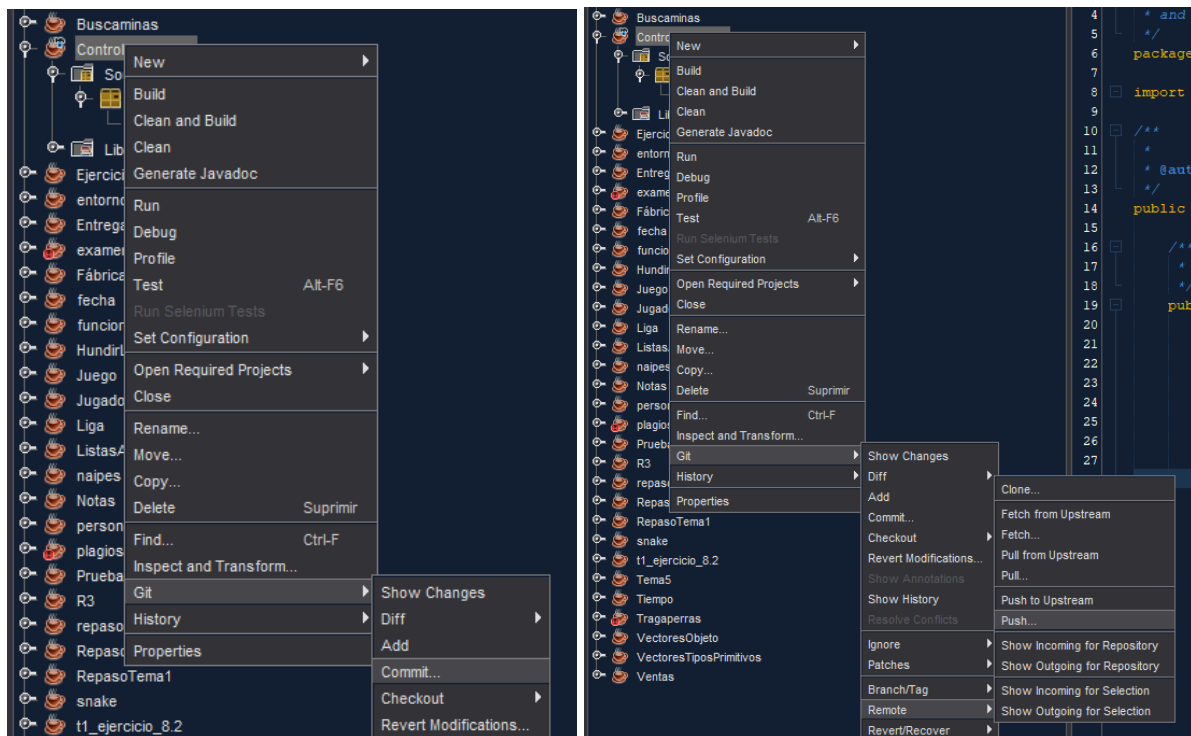
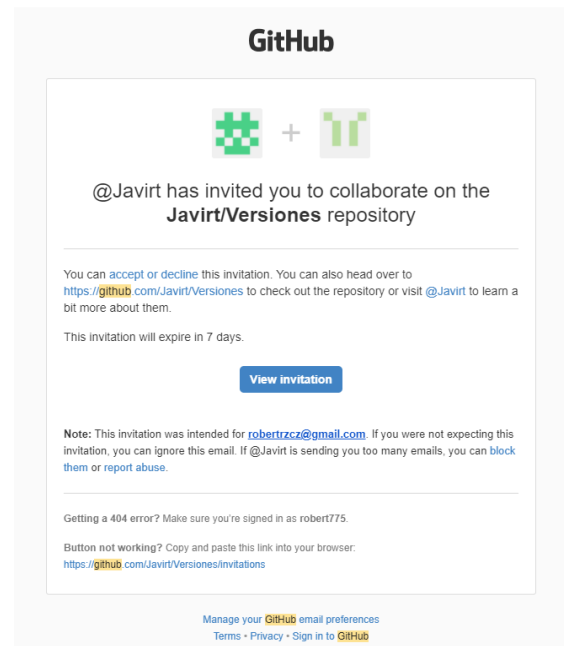
Para poder empujar los cambios, el poseedor del repositorio remoto le debe dar permisos. Para ello iremos en la cuenta del repositorio Setting > Access > Collaborators , y en la parte de “Manage access” pulsaremos el botón “Add people”.



Aquí añadiremos al usuario que vaya a modificar el repositorio para darle permisos.



A continuación seleccionamos la opción push dentro de git > remote para subir nuestras modificaciones.



Especificamos la localización de nuestro repositorio, introducimos nuestro nombre de usuario y la contraseña de nuestro token.

The screenshot shows the 'Push to Remote Repository' dialog box with the 'Remote Repository' step selected. The 'Steps' list on the left includes '1. Remote Repository', '2. Select Local Branches', and '3. Update Local References'. The main area is titled 'Remote Repository' and contains the following options:

- ☐ Select Configured Git Repository Location: A dropdown menu showing 'github:https://robert775@github.com/Javirt/Versiones.git'.
- ☒ Specify Git Repository Location:
 - Remote Name: A dropdown menu showing 'github'.
 - ☒ Persist Remote
 - Repository URL: A text field showing 'https://github.com/Javirt/Versiones.git'.
 - User: A text field showing 'robert775' (leave blank for anonymous access).
 - Password: A text field with masked characters (dots).
 - ☒ Save Password
 - Proxy Configuration... button.

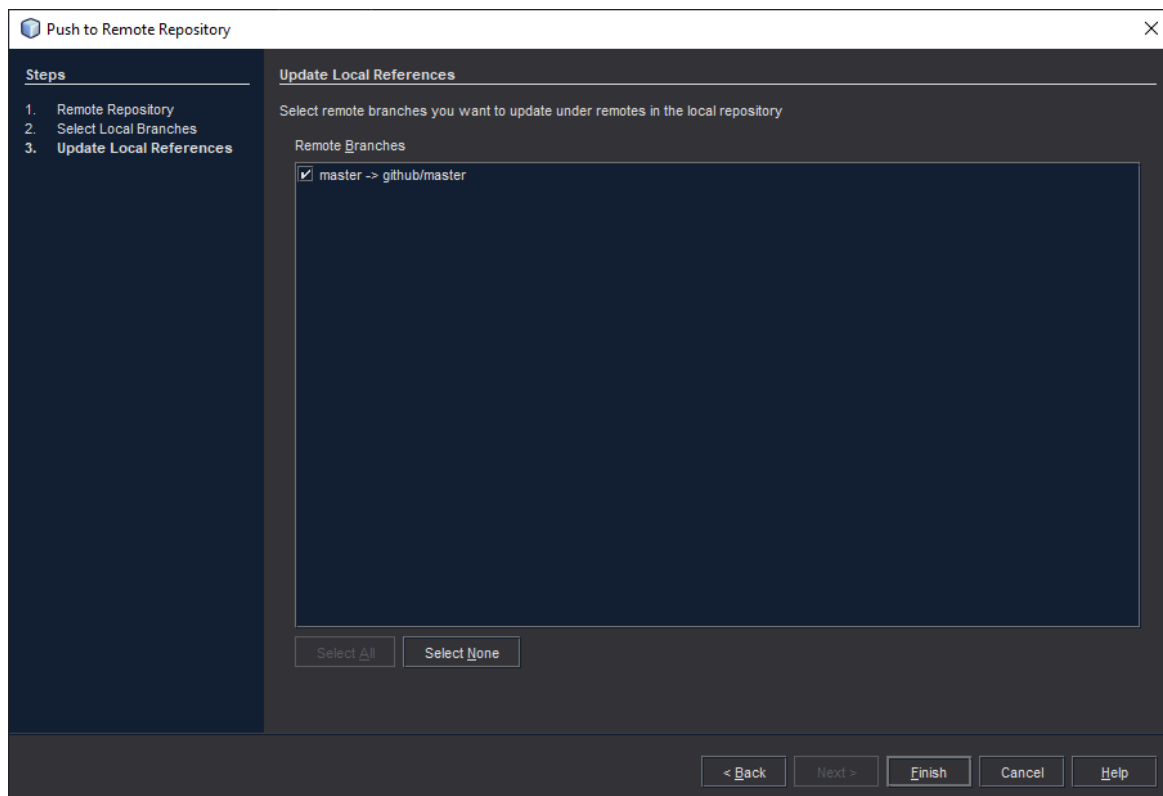
At the bottom right, there are buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Seleccionamos la rama que hemos modificado y la relacionamos con su referencia en el proyecto.

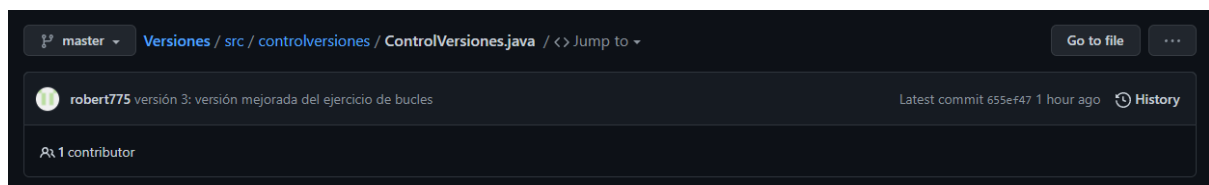
The screenshot shows the 'Push to Remote Repository' dialog box with the 'Select Local Branches' step selected. The 'Steps' list on the left includes '1. Remote Repository', '2. Select Local Branches', and '3. Update Local References'. The main area is titled 'Select Local Branches' and contains the following options:

- Local Branches: A list box showing 'master -> master' with a checked checkbox.
- Select All button.
- Select None button.

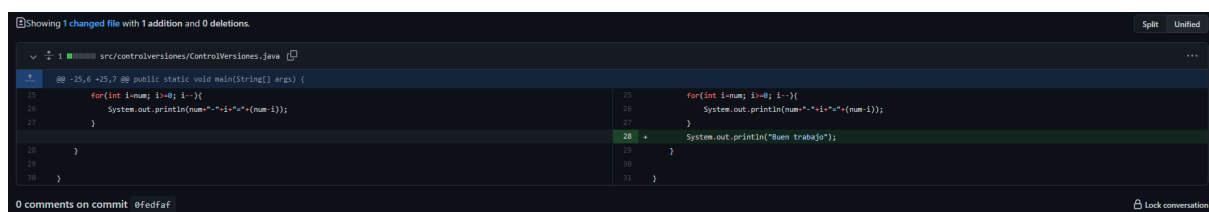
At the bottom right, there are buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.



Una vez le demos a finalizar cargaremos el contenido de nuestro repositorio local al repositorio remoto.

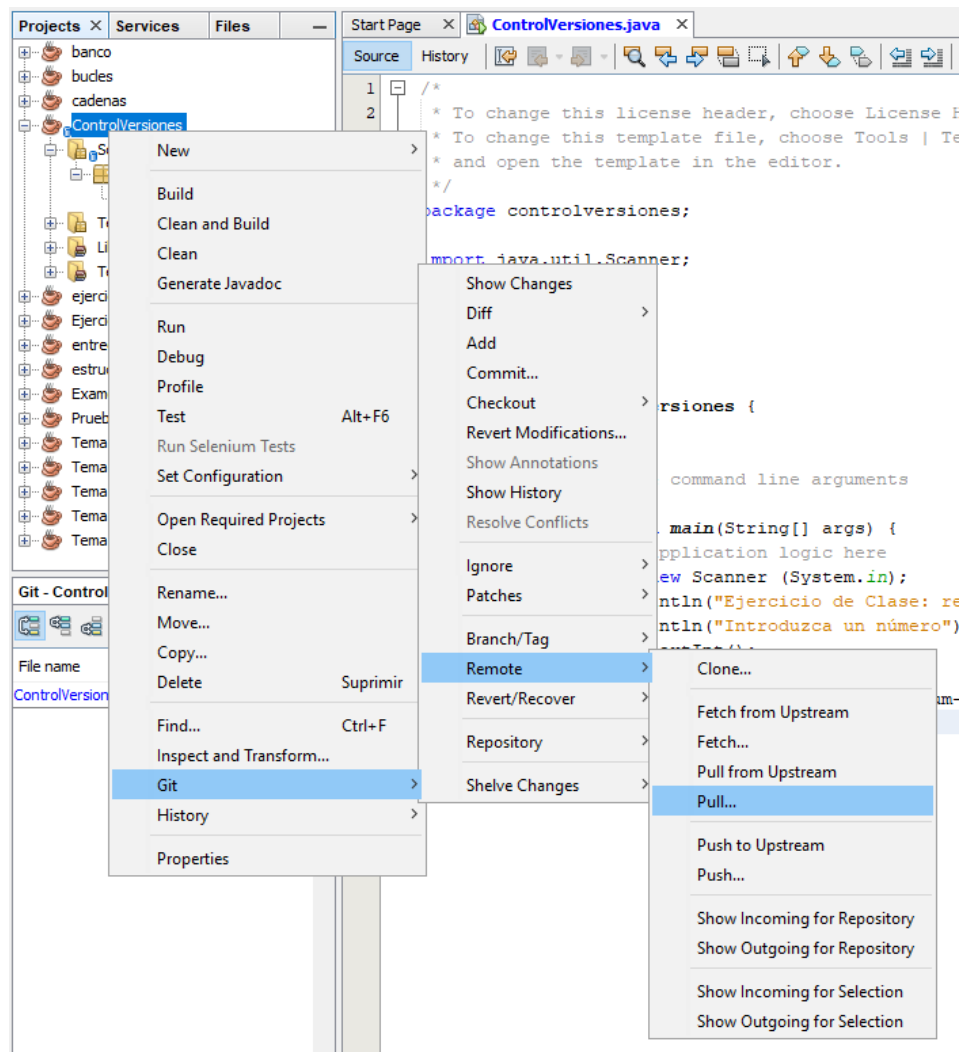


Podemos observar los cambios realizados sobre la rama que queramos seleccionando la rama y haciendo clic en el commit.



Paso 6 : Javier: trae cambios del repositorio remoto

Hacemos clic derecho en nuestro proyecto y abrimos la pestaña Git> Remote> Pull.



Seleccionamos la localización del repositorio configurado que aparecerá por defecto y hacemos clic en continuar.

The screenshot shows the 'Pull from Remote Repository' dialog box with the 'Remote Repository' step selected. The 'Specify Git Repository Location' option is chosen. The 'Remote Name' is 'github' and 'Persist Remote' is checked. The 'Repository URL' is 'https://github.com/Javirt/Versiones.git'. The 'User' is 'Javirt' and 'Save Password' is checked. The 'Next >' button is highlighted.

Pull from Remote Repository

Steps

1. Remote Repository
2. Remote Branches

Remote Repository

☐ Select Configured Git Repository Location:
github:https://Javirt@github.com/Javirt/Versiones.git

☒ Specify Git Repository Location:

Remote Name: ☒ Persist Remote

Repository URL:
http[s]://host.xz[:port]/path/to/repo.git/

User: (leave blank for anonymous access)

Password: ☒ Save Password

< Back Next > Finish Cancel Help

Por último nos preguntará por las ramas que queremos actualizar. Seleccionamos la única rama que tenemos y finalizamos la actualización de nuestro repositorio local. Tras terminar este paso ya habremos traído los cambios del repositorio remoto.

The screenshot shows the 'Pull from Remote Repository' dialog box with the 'Remote Branches' step selected. The 'master' branch is selected for pulling. The 'Finish' button is highlighted.

Pull from Remote Repository

Steps

1. Remote Repository
2. Remote Branches

Remote Branches

Remote Branches

☒ master -> github/master [U]

Select All Select None

Branch github/master will be merged into the current branch master.

< Back Next > Finish Cancel Help