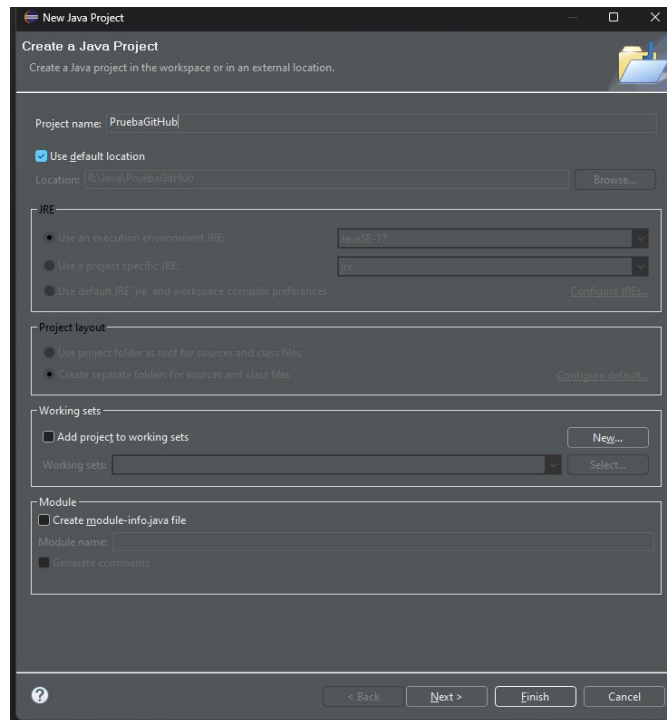
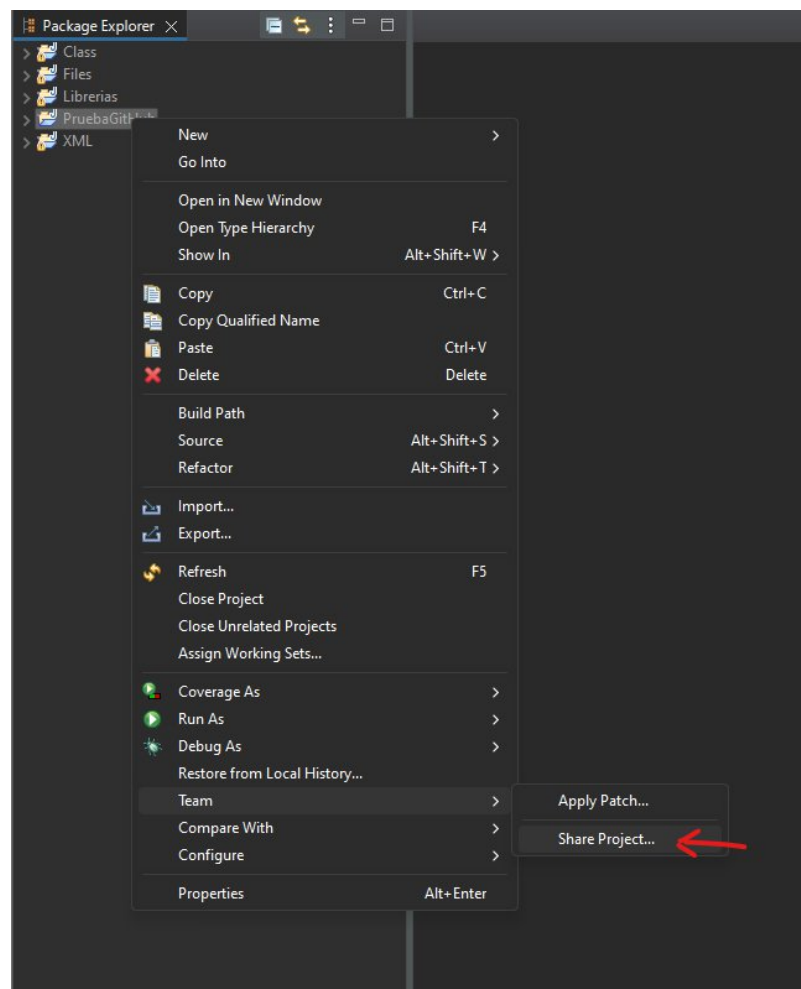


Para poder conectar Github con un repositorio de Eclipse hay que hacer los siguientes pasos.

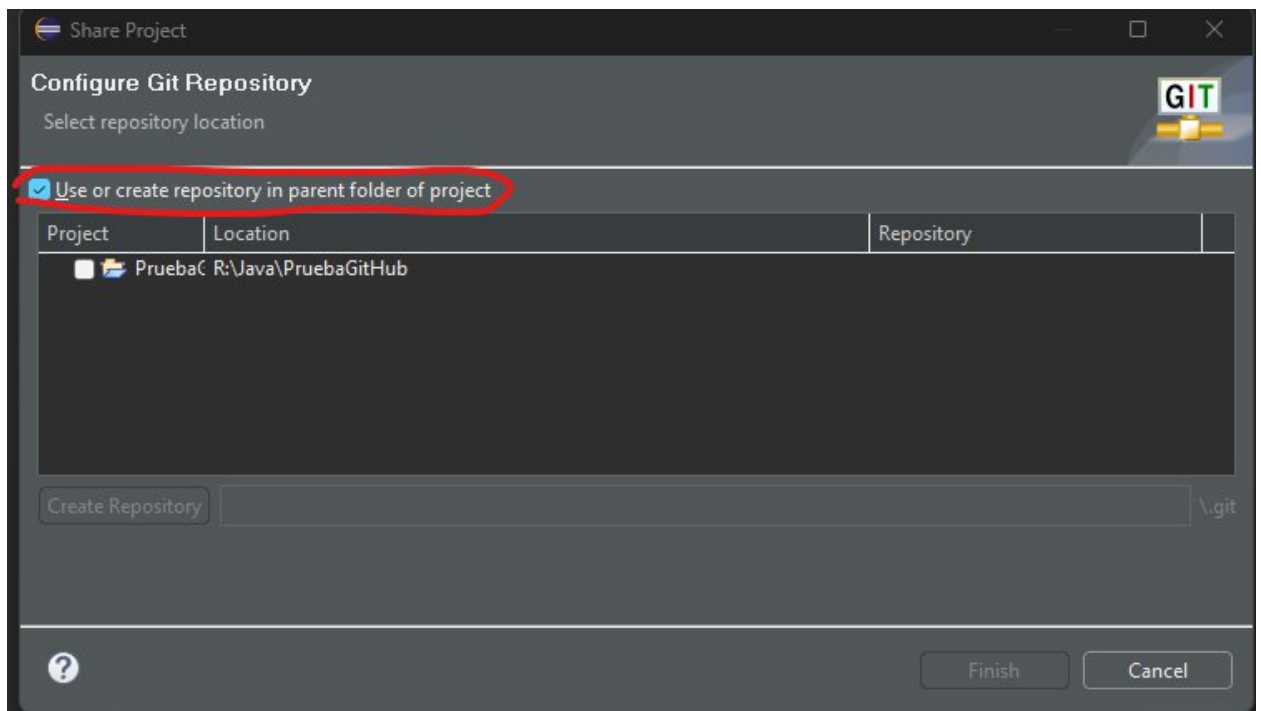
Primero creamos un repositorio de prueba para realizar la conexión



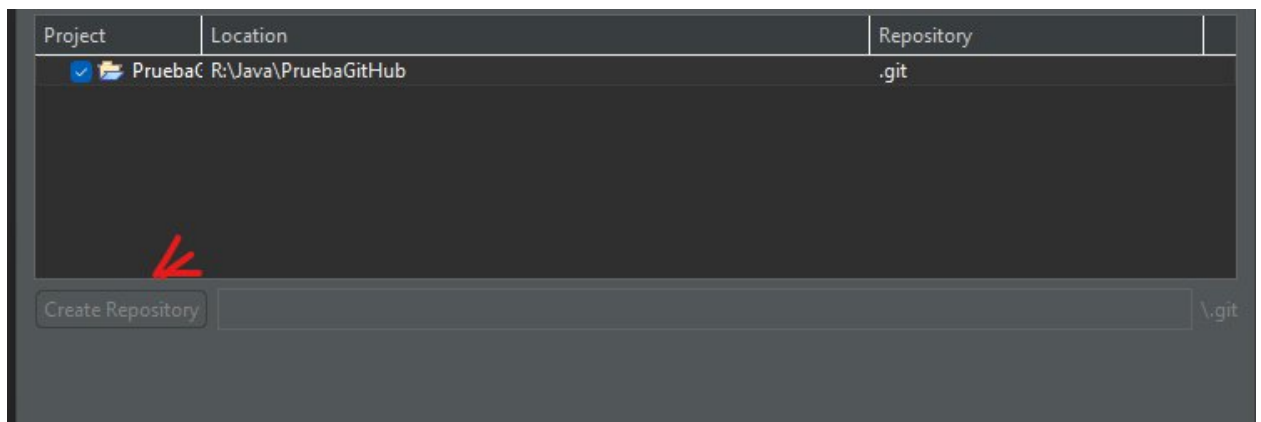
Después de crear el repositorio seleccionamos la opcion TEAM / share Project



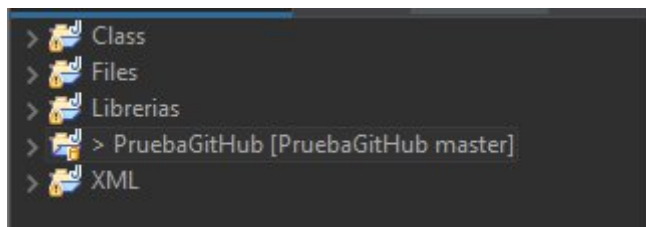
Despues de dar a opción tenemos que dar al check use or create repository para que nos salga la siguiente ventana



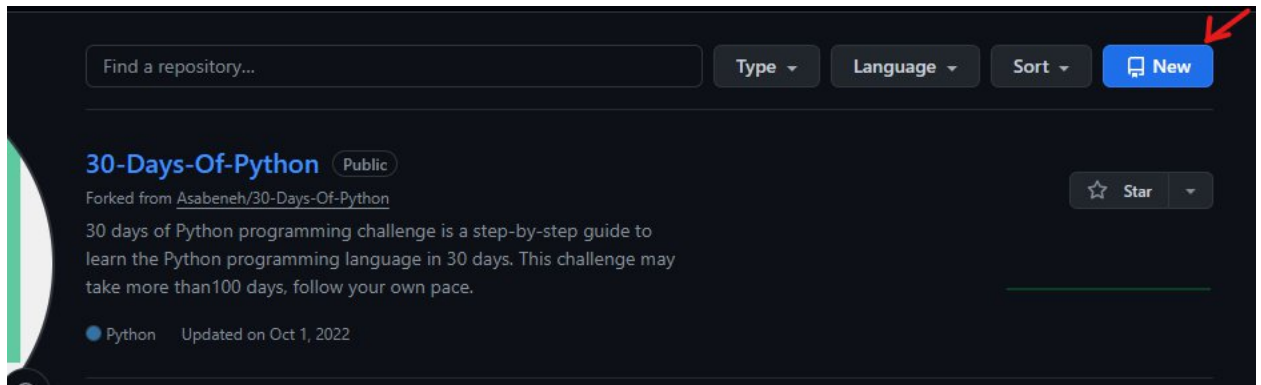
Para poder seleccionar la carpeta que nos sale en pantalla la seccionamos y damos a la opción Create repository



Y le damos al boton de finish. Creandonos el repositorio para el proyecto.



Ahora nos dirigimos a GitHub. Iniciamos sesión y creamos un 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 * ivanBasCub / **Repository name *** PruebaEclipse
PruebaEclipse is available.

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

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.](#)

Add .gitignore
.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

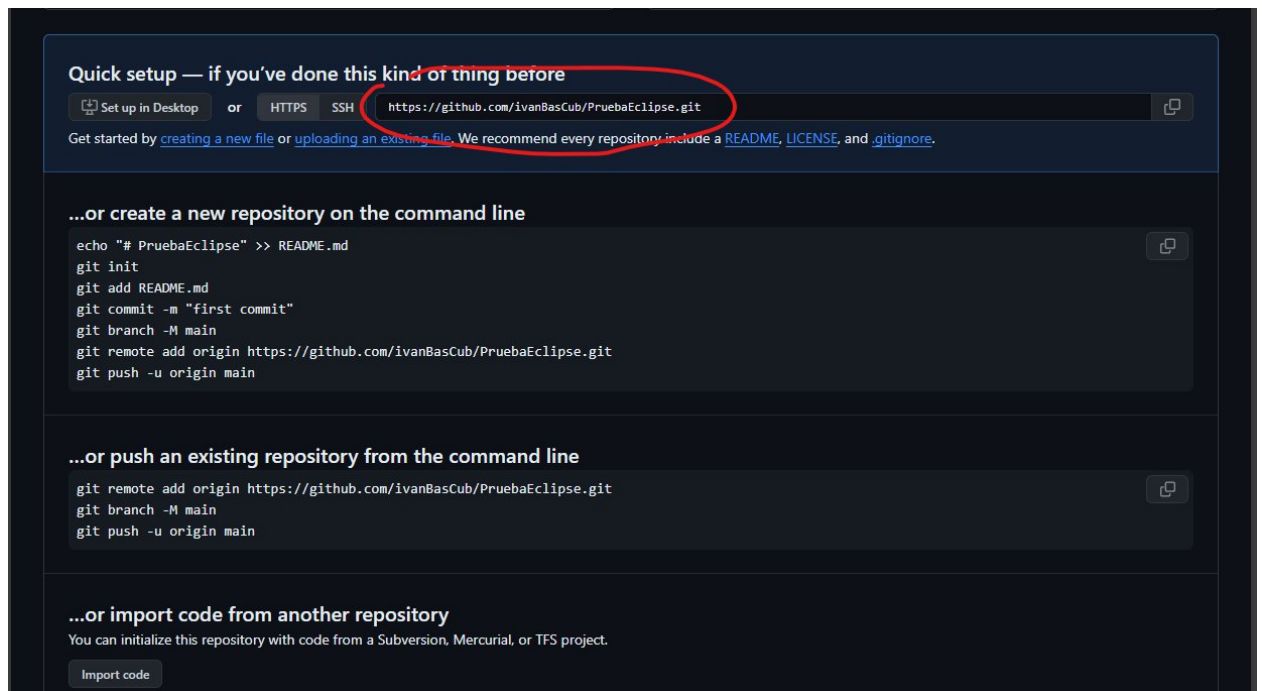
Choose a license
License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

ⓘ You are creating a private repository in your personal account.

Create repository

Cuando se crea copiamos la url que nos aparece en pantalla



Quick setup — if you've done this kind of thing before

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/ivanBasCub/PruebaEclipse.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# PruebaEclipse" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ivanBasCub/PruebaEclipse.git
git push -u origin main
```

...or push an existing repository from the command line

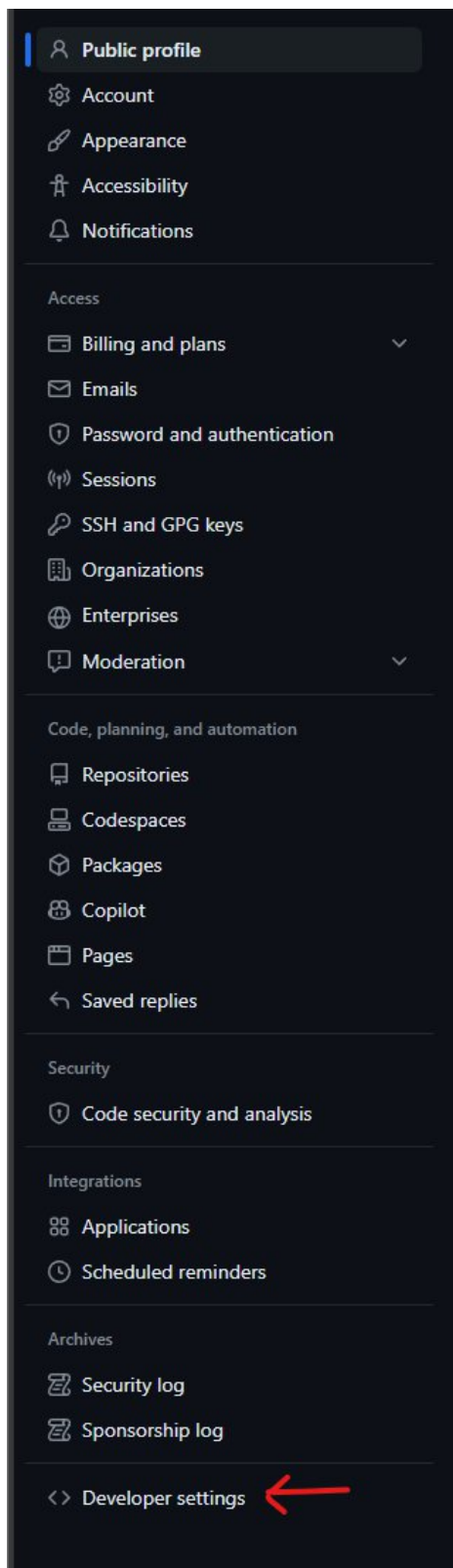
```
git remote add origin https://github.com/ivanBasCub/PruebaEclipse.git
git branch -M main
git push -u origin main
```

...or import code from another repository

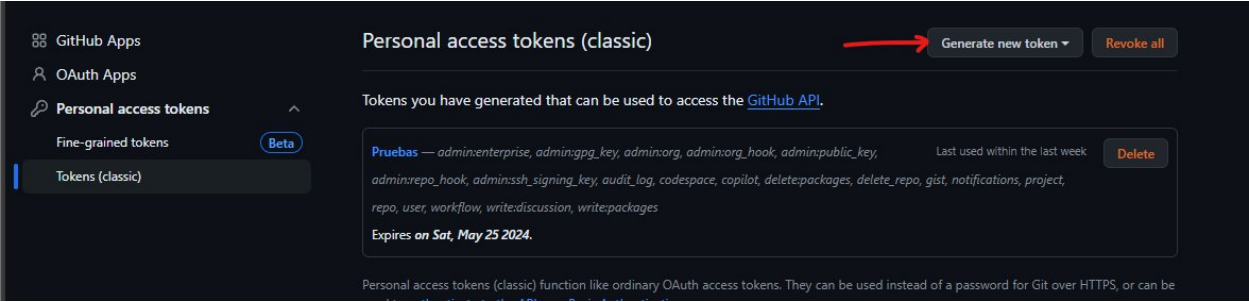
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

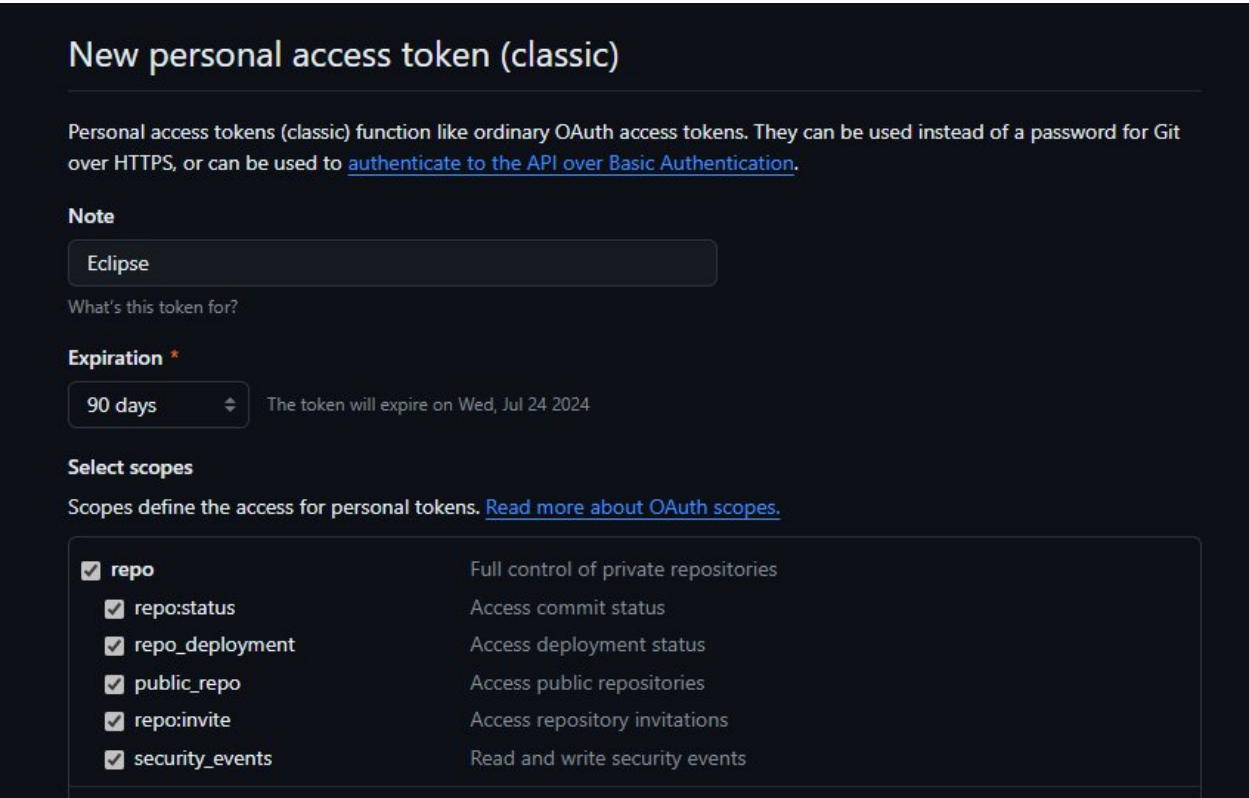
Continuando con Eclipse hay que generar un token para poder conectar con Eclipse. Para poder crearlo tienes que ir a Ajustes / Developer Settings



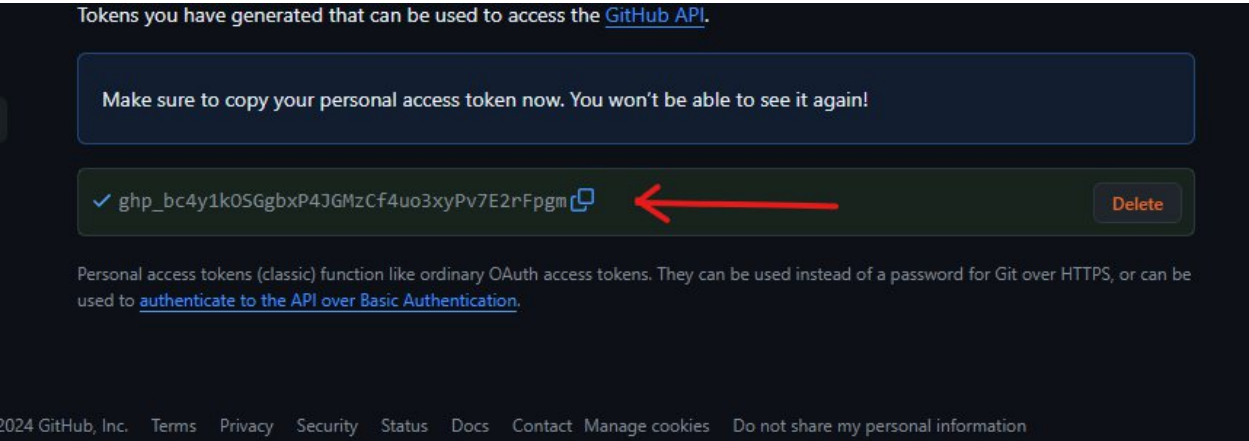
Continuamos en tokens (classic) generamos un nuevo token



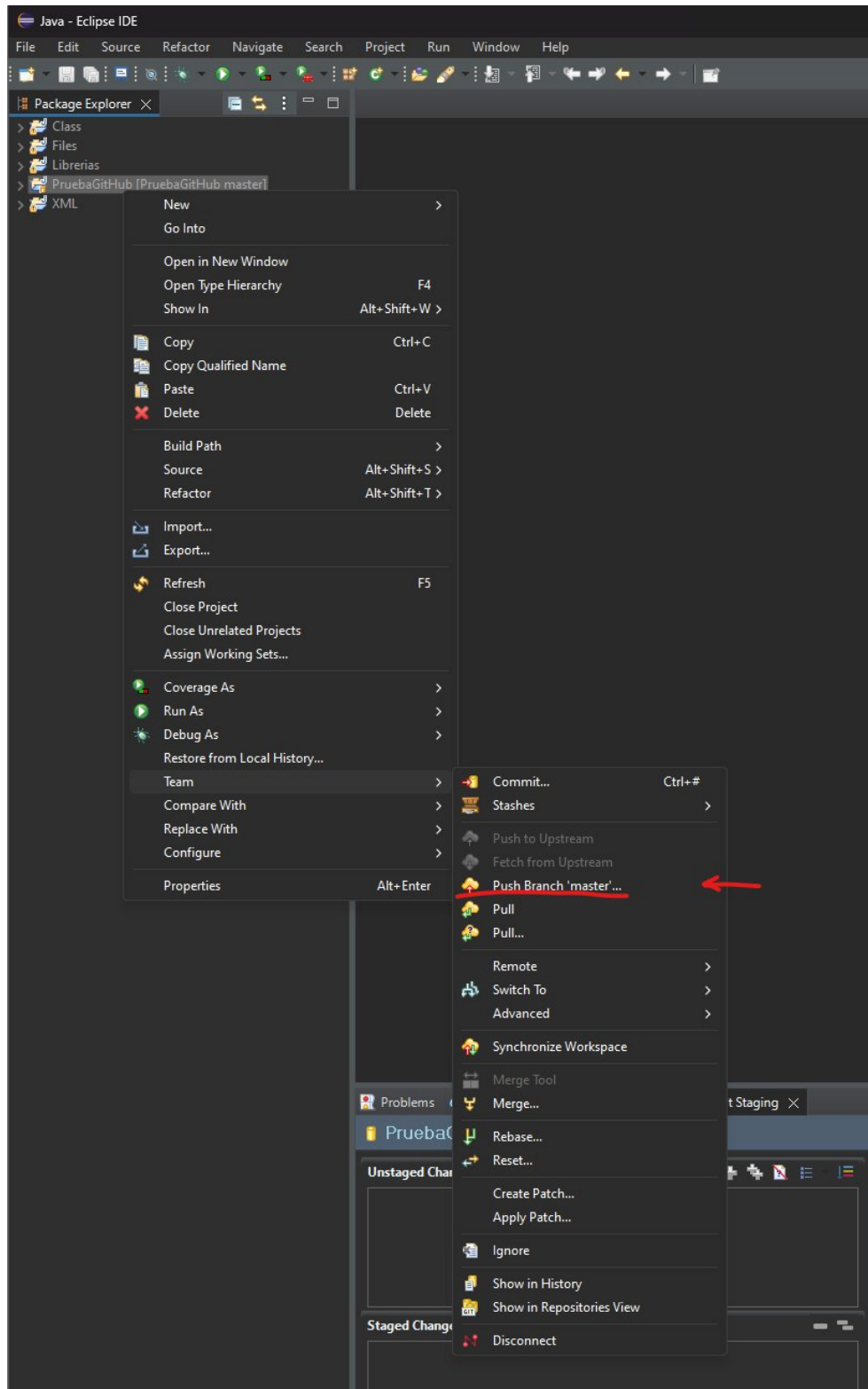
Le das un nombre y todos los permisos



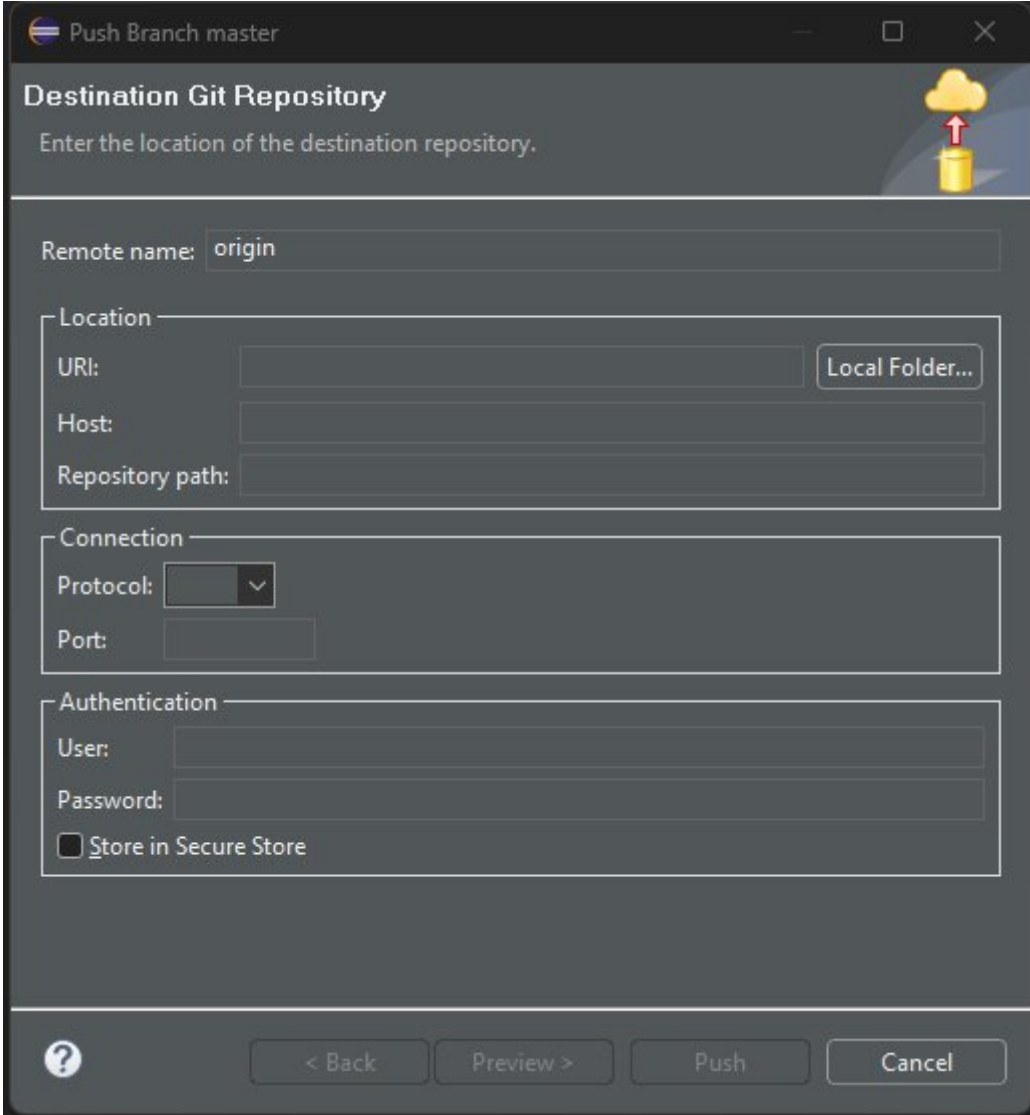
Copiamos la id que nos sale en pantalla



Nos volveremos al Eclipse y seleccionamos la siguiente opcion



Apareciendo la siguiente pantalla



The screenshot shows a dialog box titled "Push Branch master" with standard window controls (minimize, maximize, close). The main heading is "Destination Git Repository" with a subtitle "Enter the location of the destination repository." and a cloud upload icon. The form is organized into sections: "Remote name:" with a text field containing "origin"; "Location:" with fields for "URI:", "Host:", and "Repository path:", plus a "Local Folder..." button; "Connection:" with a "Protocol:" dropdown and a "Port:" field; and "Authentication:" with "User:" and "Password:" fields, and a checkbox for "Store in Secure Store". At the bottom, there is a help icon, and buttons for "< Back", "Preview >", "Push", and "Cancel".

Push Branch master

Destination Git Repository

Enter the location of the destination repository.

Remote name:

Location

URI: Local Folder...

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

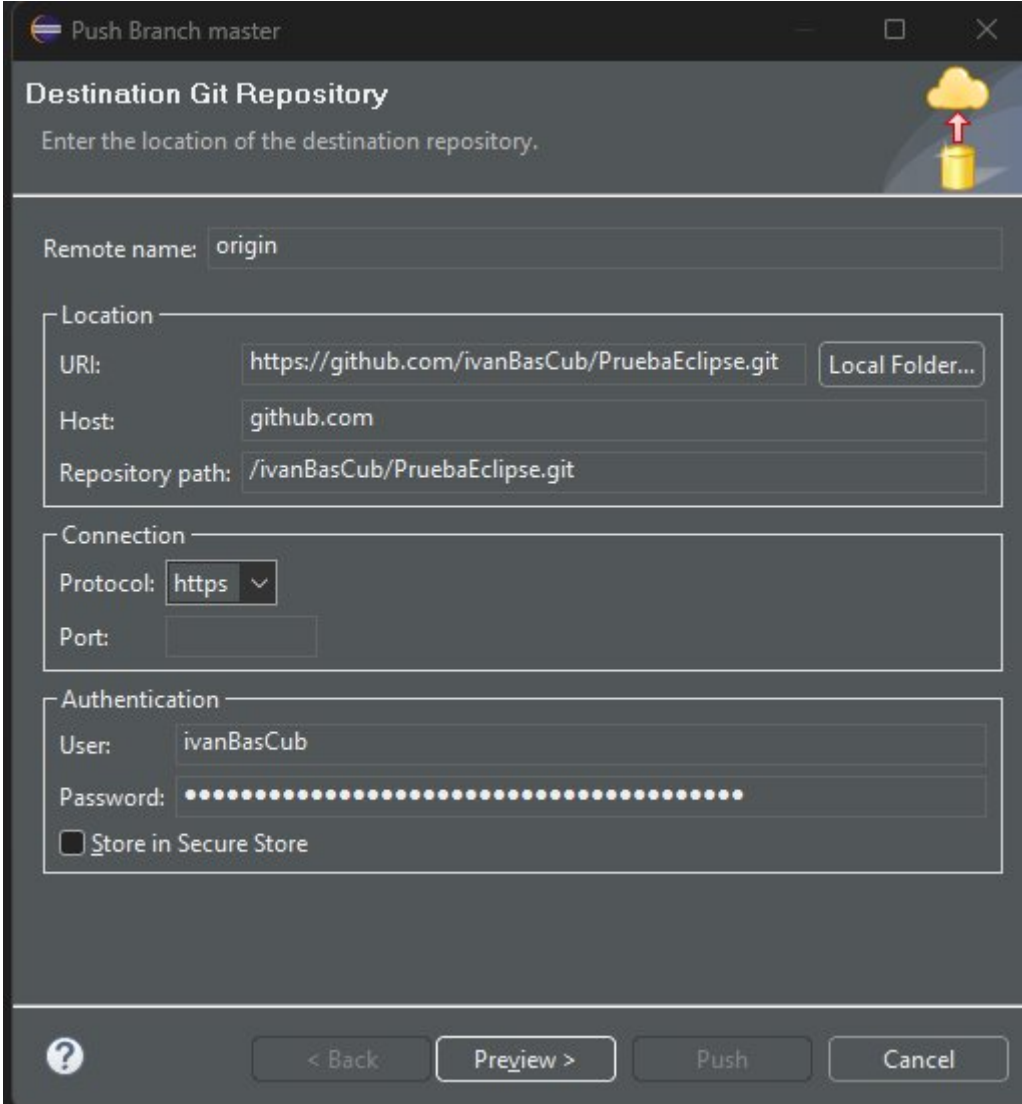
User:

Password:

☐ Store in Secure Store

? < Back Preview > Push Cancel

En esta pestaña pegamos la url que nos salia en el repositorio que hemos creado en github ademas de rellenar los datos de identificación. La contraseña es la id del token que hemos creado anteriormente



The screenshot shows a dialog box titled "Push Branch master" with a close button in the top right corner. The main heading is "Destination Git Repository" with a subtitle "Enter the location of the destination repository." and a cloud upload icon. The dialog is divided into four sections: "Remote name" with a text field containing "origin"; "Location" with fields for "URI" (https://github.com/ivanBasCub/PruebaEclipse.git), "Host" (github.com), and "Repository path" (/ivanBasCub/PruebaEclipse.git), plus a "Local Folder..." button; "Connection" with a "Protocol" dropdown set to "https" and an empty "Port" field; and "Authentication" with "User" (ivanBasCub), a masked "Password" field, and a checkbox for "Store in Secure Store". At the bottom are buttons for "?", "< Back", "Preview >", "Push", and "Cancel".

Push Branch master

Destination Git Repository

Enter the location of the destination repository.

Remote name:

Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

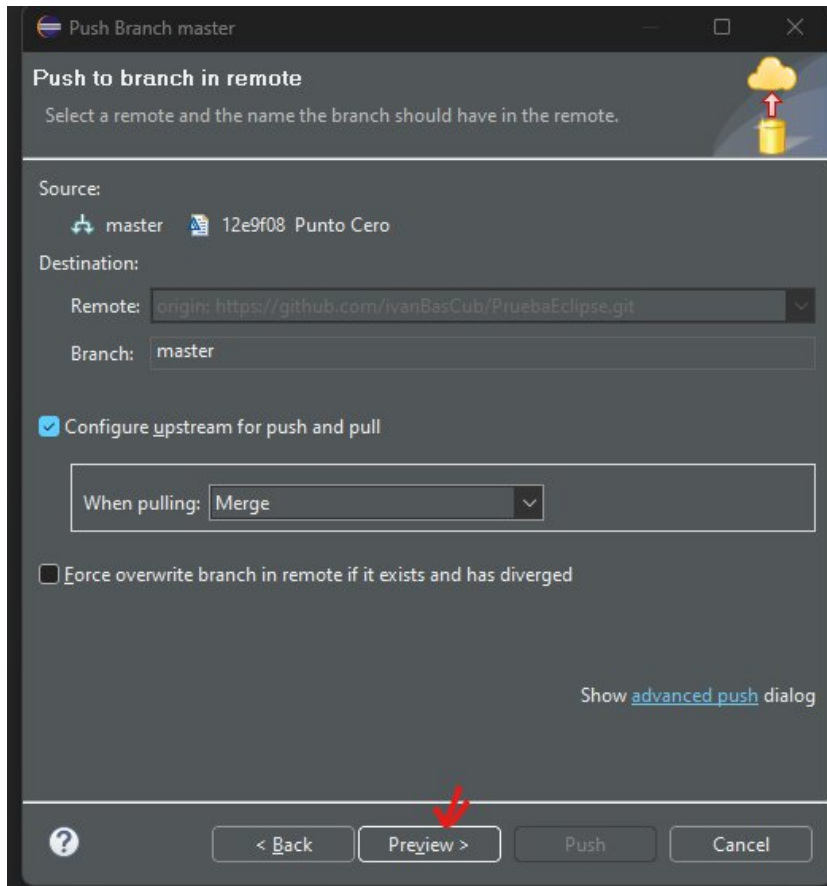
User:

Password:

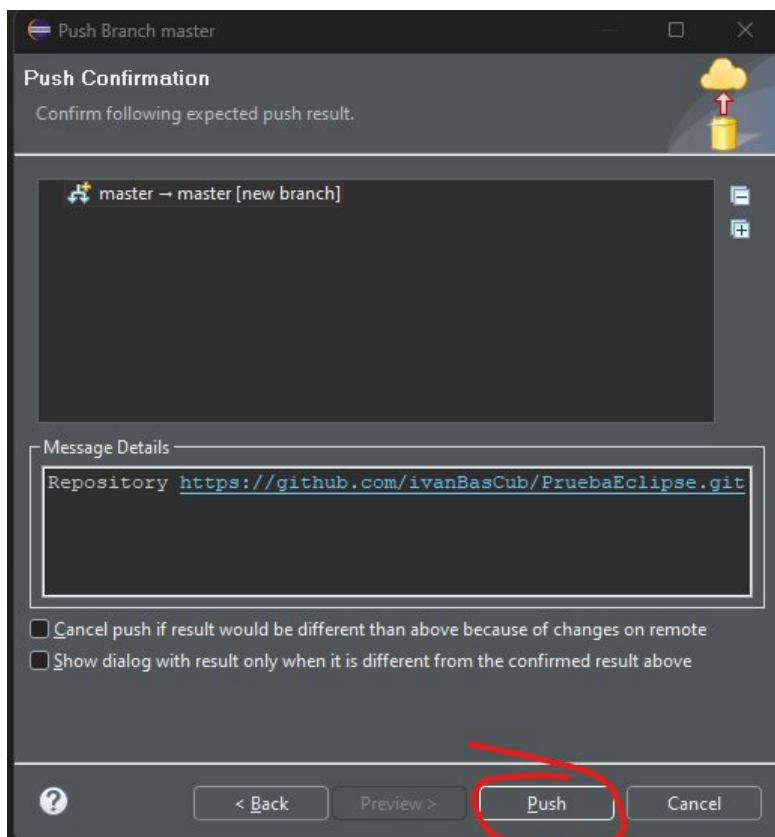
☐ Store in Secure Store

? < Back Preview > Push Cancel

Le damos a siguiente



Y le damos a Push



Nos dirigimos a GitHub y comprobamos para ver si se ha subido

