



INSTALACIÓN Y CONFIGURACIÓN DE IDE'S

NETBEANS && ECLIPSE && VISUAL STUDIO CODE

ESTER HITOS GARCIA & JOSE ANTONIO RIOS
1º DAW I.E.S CAMPANILLAS

1. NETBEANS: INSTALACIÓN Y CONFIGURACIÓN DE PYTHON.....	2
01. PASO: Descarga la última versión Python Para Windows	2
02. PASO: Descarga la última versión Python Para Linux2	
2. ECLIPSE: INSTALACIÓN Y CONFIGURACIÓN DE PYTHON	4
01. Instalación del plugin Python	4
02. Instalación del Interprete Ejecutable Python.exe ..	7
03. Creación de Modulo “BasicoHolaMundo” en el proyecto Python.	11
04. Creación de Fichero “HolaMundo.py” en el proyecto Python.	12
3. ECLIPSE: UTILIZACIÓN DE GIT Y GITHUB.....	14
01. Proceso para activar git en un proyecto de java .	14
02. Inicializar el proyecto “git init” en eclipse ...	15
03. Crear fichero .class y guardar estado en git con commit	17
04. Comprobar el Staging Area del proyecto JavaGit en Eclipse	18
05. Realizar los commit en el repositorio git con Eclipse	21
06. Volver a los estados anteriores de los commits realizados.	25
07. Creación del Repositorio Remoto en Git Hub	27
08. Push Head en Eclipse al Repositorio Remoto	28
4. VISUAL STUDIO CODE: INSTALACIÓN Y CONFIGURACIÓN DE las extensiones XML Y XML TOOLS de Red Hat	32
01. VISUAL STUDIO CODE extensiones XML	32

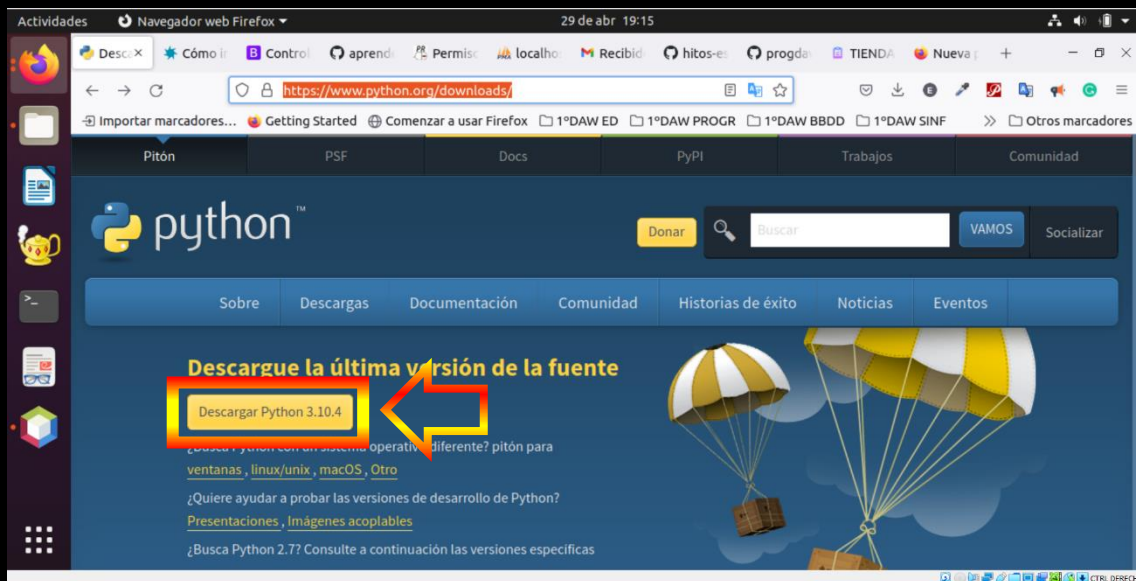
1.NETBEANS: INSTALACIÓN Y CONFIGURACIÓN DE PYTHON

01. PASO: Descarga la última versión Python Para Windows

- Si estás trabajando con Windows tienes que introducir en la barra del navegador esta dirección:

<https://www.python.org/downloads/>

- Y aparecerá esta pantalla que se ve en la imagen, tenemos que pulsar descargar Python 3.10.4



02. PASO: Descarga la última versión Python Para Linux

- En Linux, miramos primero la versión que tenemos instalada, porque en el S.O UBUNTU 2020 ya viene instalada por defecto y es el que utilizare para esta práctica.
- Abrimos la consola de "cmd" pulsando 'Crt+Alt+T' e introducimos estos comandos:
 - python --versión
 - python3 --versión

```
esther@esther-VirtualBox: ~  
esther@esther-VirtualBox:~$ python --version  
Orden «python» no encontrada. Quizá quiso decir:  
  la orden «python3» del paquete deb «python3»  
  la orden «python» del paquete deb «python-is-python3»  
esther@esther-VirtualBox:~$ python3 --version  
Python 3.8.10  
esther@esther-VirtualBox:~$
```

- 📁 Siguiente paso es actualizar la lista de repositorios, para poder descargar después la última versión de Python, con este comando
 - 🔑 Sudo apt update
- 📁 Ahora bajamos un soporte de software que nos ayudara a la configuración :
 - 🔑 sudo apt install software-properties-common
- 📁 E insertamos este comando para los repositorios
 - 🔑 sudo add-apt-repository ppa:deadsnakes/ppa
- 📁 Y como siempre en Linux después de toda instalación hay que actualizar los paquetes con:
 - 🔑 Sudo apt update
- 📁 Ahora que ya hemos actualizado todos los paquetes, ya podemos instalar la nueva versión de Python con este comando:
 - 🔑 Sudo apt install Python 3.9
- 📁 Pero cuando instalamos la nueva versión de Python, no la ejecuta ,a no ser que lo configuremos, para que no coja la versión 3.8 que tiene instalada por defecto, y nos coja la versión 3.9 que acabamos de instalar, con estos comandos:
 - 🔑 Sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.9 1

```
esther@esther-VirtualBox: ~  
esther@esther-VirtualBox:~$ python3 --version  
Python 3.8.10  
esther@esther-VirtualBox:~$ sudo update-alternatives --install /usr/bin/python3  
python3 /usr/bin/python3.9 1  
update-alternatives: utilizando /usr/bin/python3.9 para proveer /usr/bin/python3  
(python3) en modo automático  
esther@esther-VirtualBox:~$ python3 --version  
Python 3.9.12  
esther@esther-VirtualBox:~$
```

- 📁 Si este comando no te funciona, es porque cambiaste muchas veces de versión, este comando te ayudara a ejecutar la versión que te interese
- 📁 Sudo update-alternatives --set python3.9 /usr/bin/python3

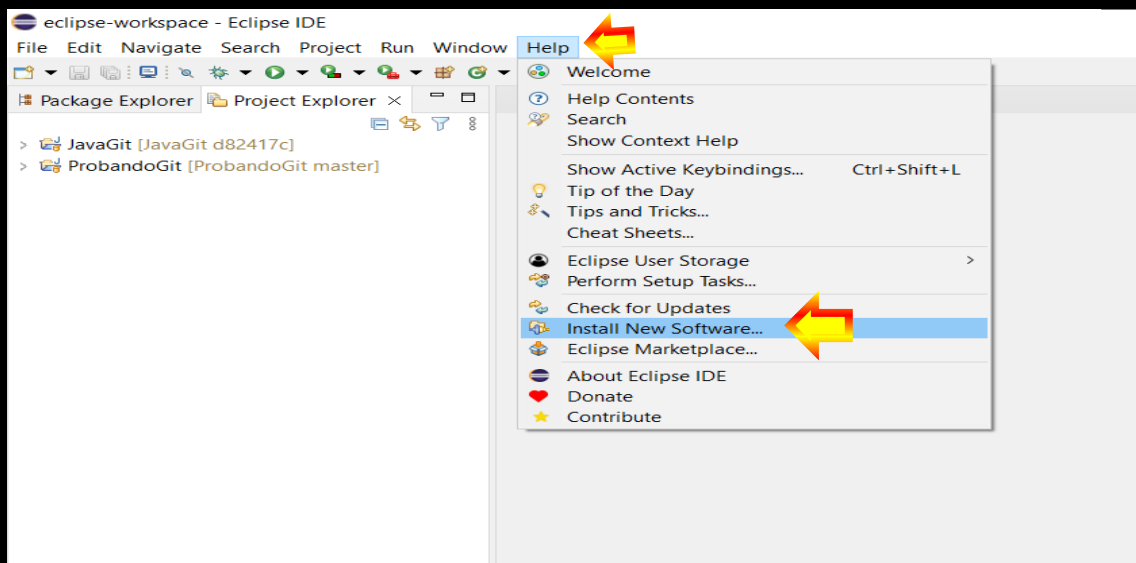
AVISO IMPORTANTE NO EXISTEN PLUGINNS DE PYTHON PARA NetBeans 12.0 SOLO PARA LA VERSION NetBeans 8.1 y 8.2 pirata

2. ECLIPSE: INSTALACIÓN Y CONFIGURACIÓN DE PYTHON

01. Instalación del plugin Python

- 📁 La versión en la que estoy trabajando de Eclipse es la 2022-03, lo especifico porque cada dos por tres actualizan versiones y creo que es importante que se sepa con que versión estoy trabajando.

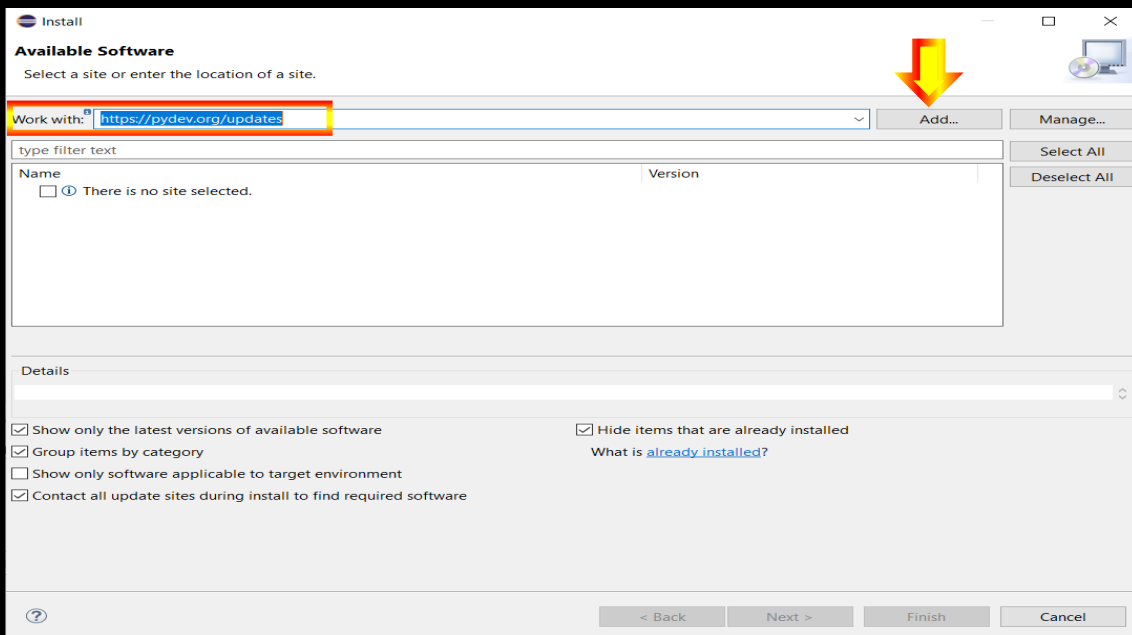
- 📁 Para instalar el plugin de Python tenemos que ir a la ventana de "Help" y después "Install New Software".



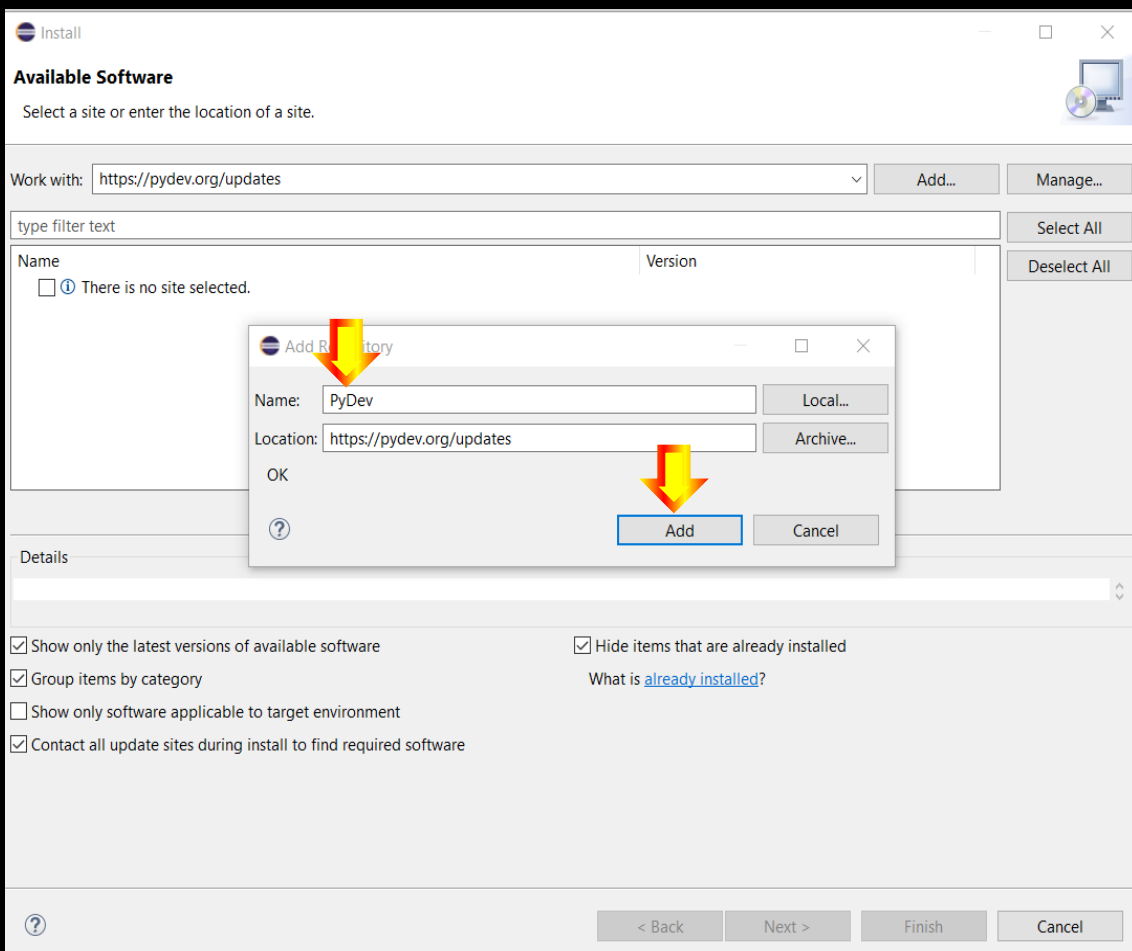
- 📁 En el siguiente paso le daremos la dirección web de donde poder descargar el plugin de Python en Eclipse: <http://www.pydev.org/updates>.

Es importante que sepas que "PyDev" es un IDE de Python para Eclipse, que se puede usar en el desarrollo de Python, Jython y IronPython.

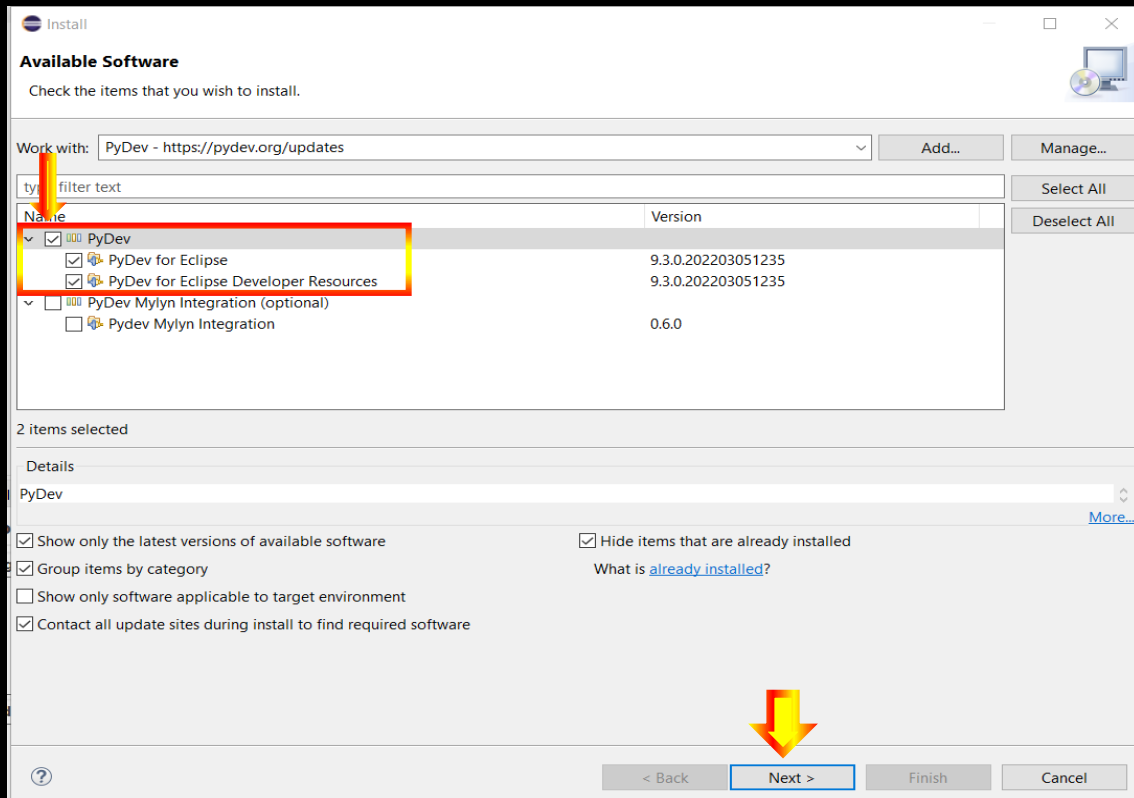
Insertamos la dirección en la barra "work with"
y pinchamos en "Add"



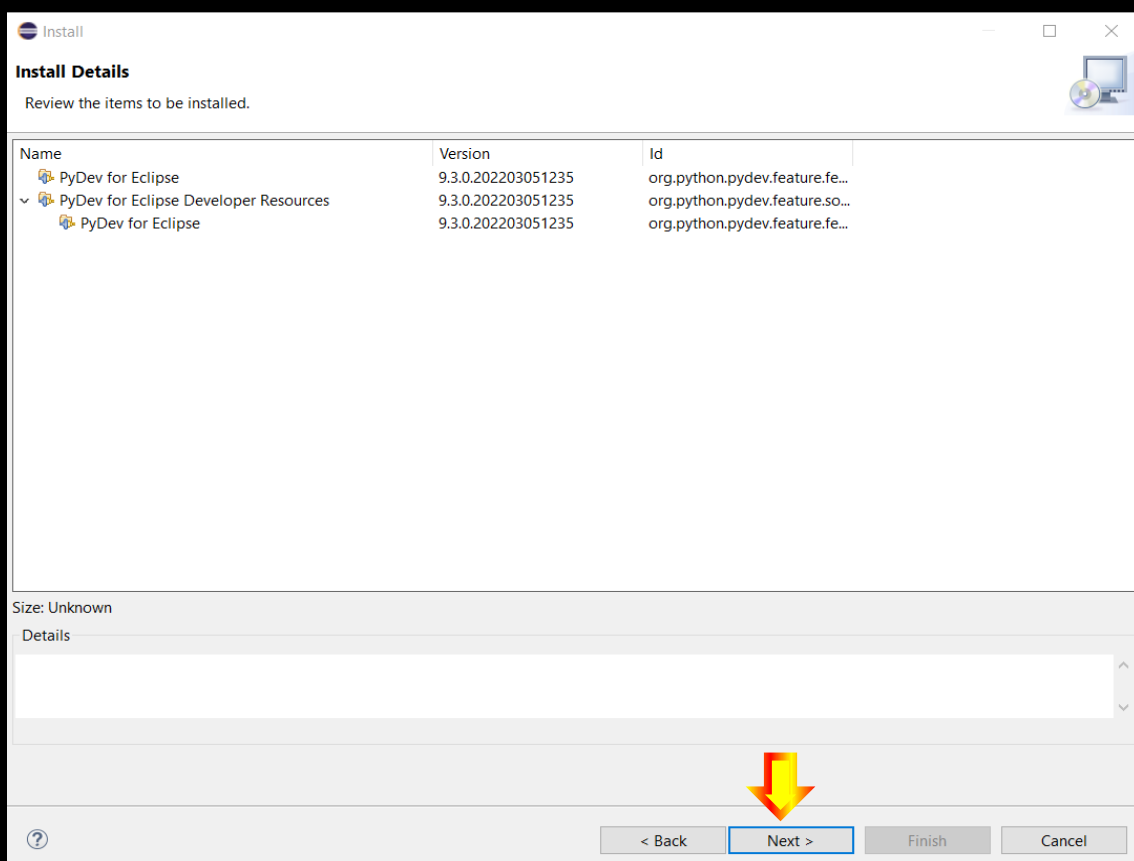
Nos saldrá una ventana para darle nombre al
repositorio de Python, Le he dado el nombre de
"PyDev" y pinchamos en "Add" y nos buscará los
paquetes correspondientes.



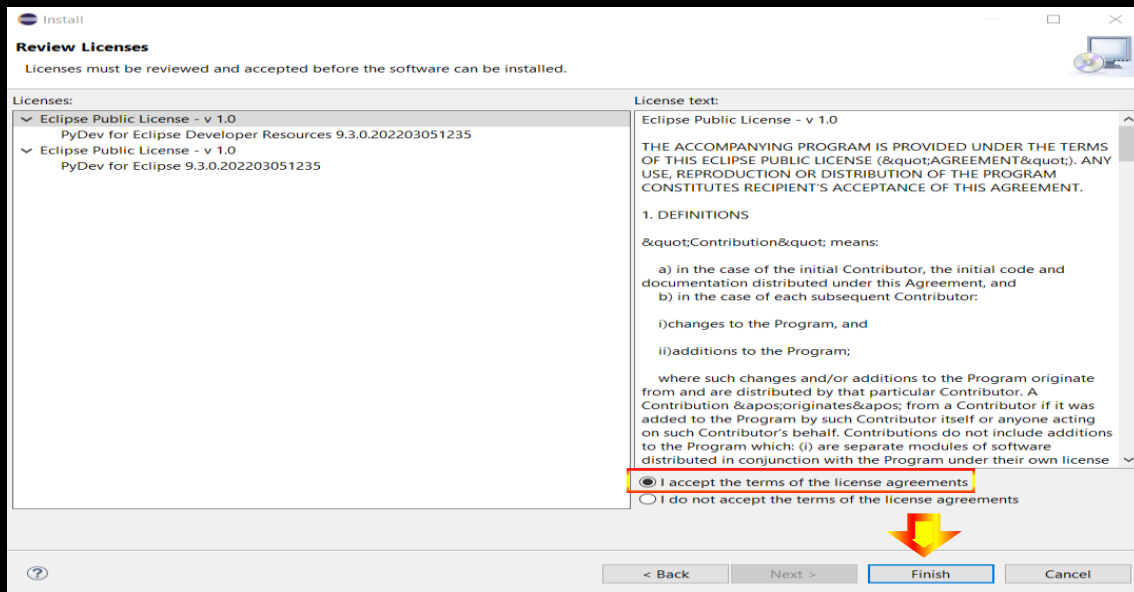
Una vez encuentra los paquetes de “PyDev”, seleccionamos el primero, y se autoselecciona los que dependen de él, y pinchamos en “Next”.



Te avisa de los paquetes que vas a instalar, le damos a “Next” de nuevo.



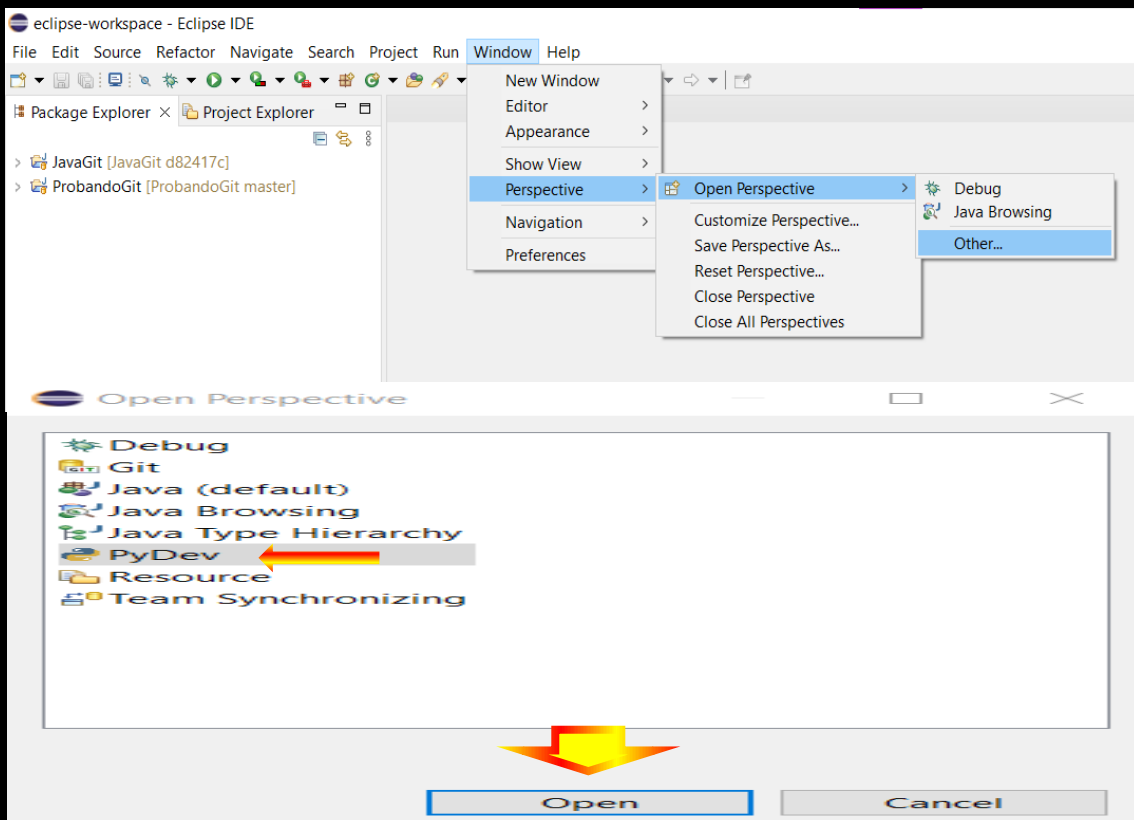
🖱️ En el siguiente paso, aceptamos la licencia del software y pinchamos en “Finish”



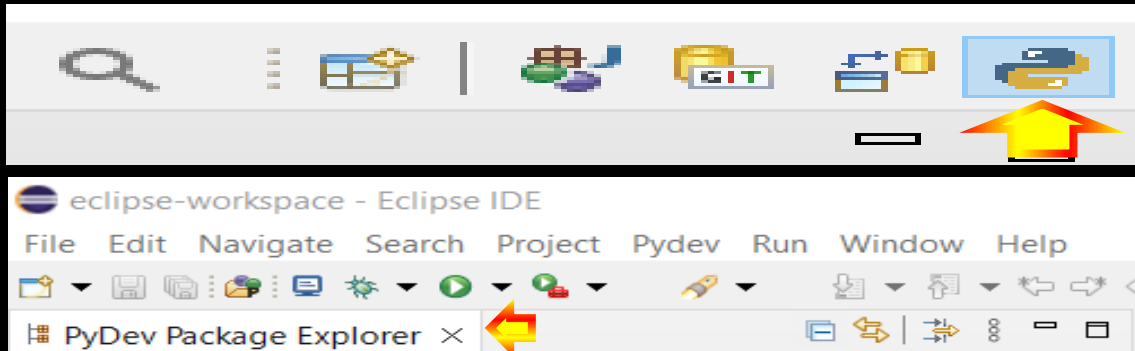
🖱️ ¡¡¡Ya tenemos la instalación del plugin!!!

02.Instalación del Interprete Ejecutable Python.exe

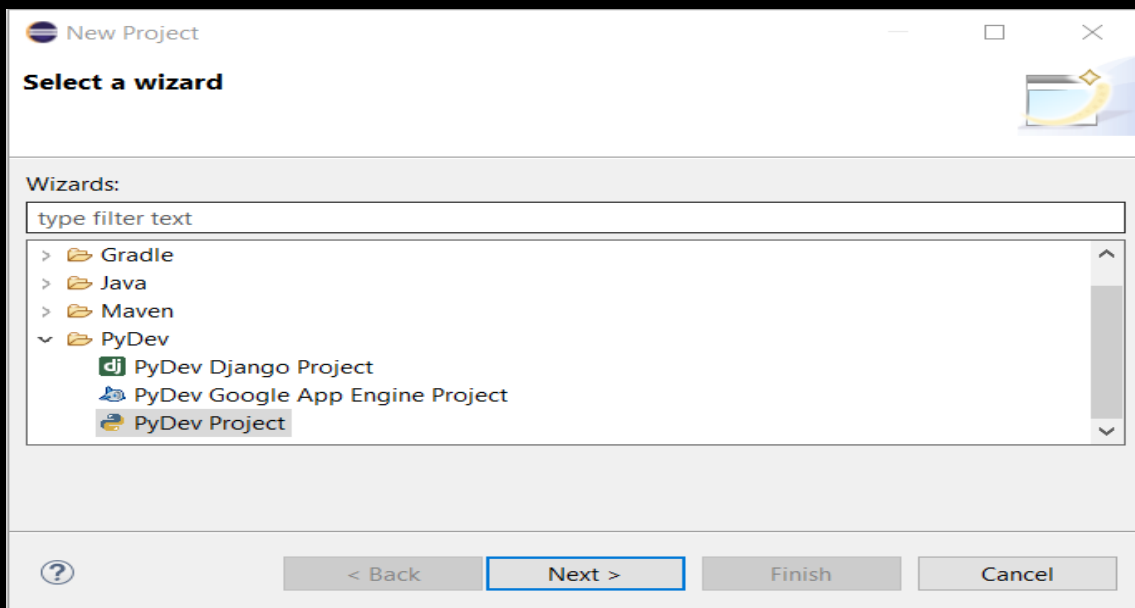
🖱️ Como ya tenemos Python instalado podemos crear un proyecto con él, lo primero que haremos es crear la vista en el IDE de “PyDev” siguiendo los pasos de las fotos que vienen a continuación.



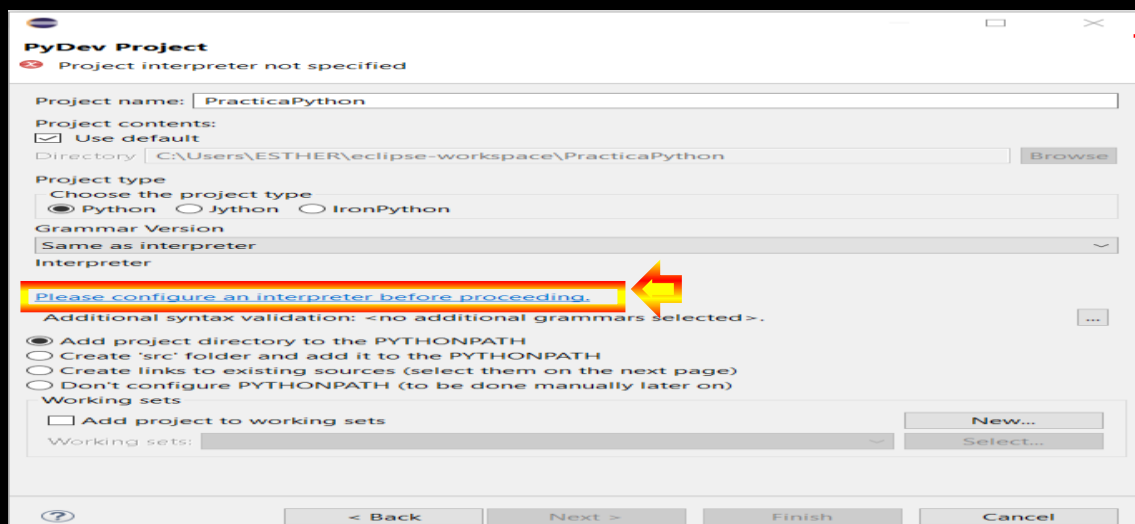
Como podemos observar en la parte superior derecha ya nos aparece el logotipo de Python y a la izquierda la perspectiva de PyDev.



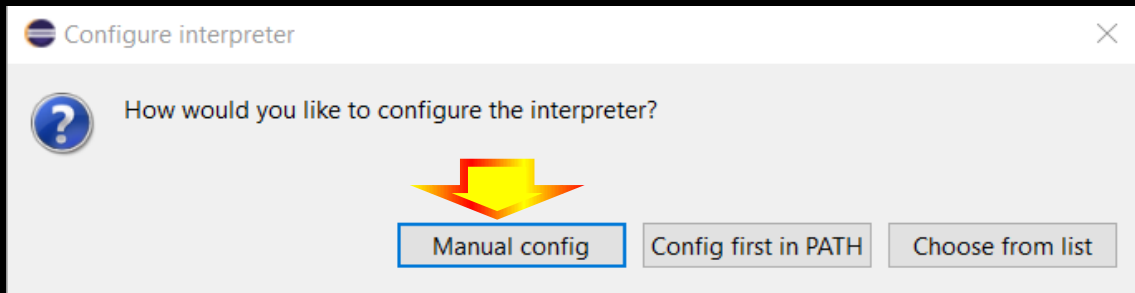
Ya podemos crear un nuevo proyecto con Python.



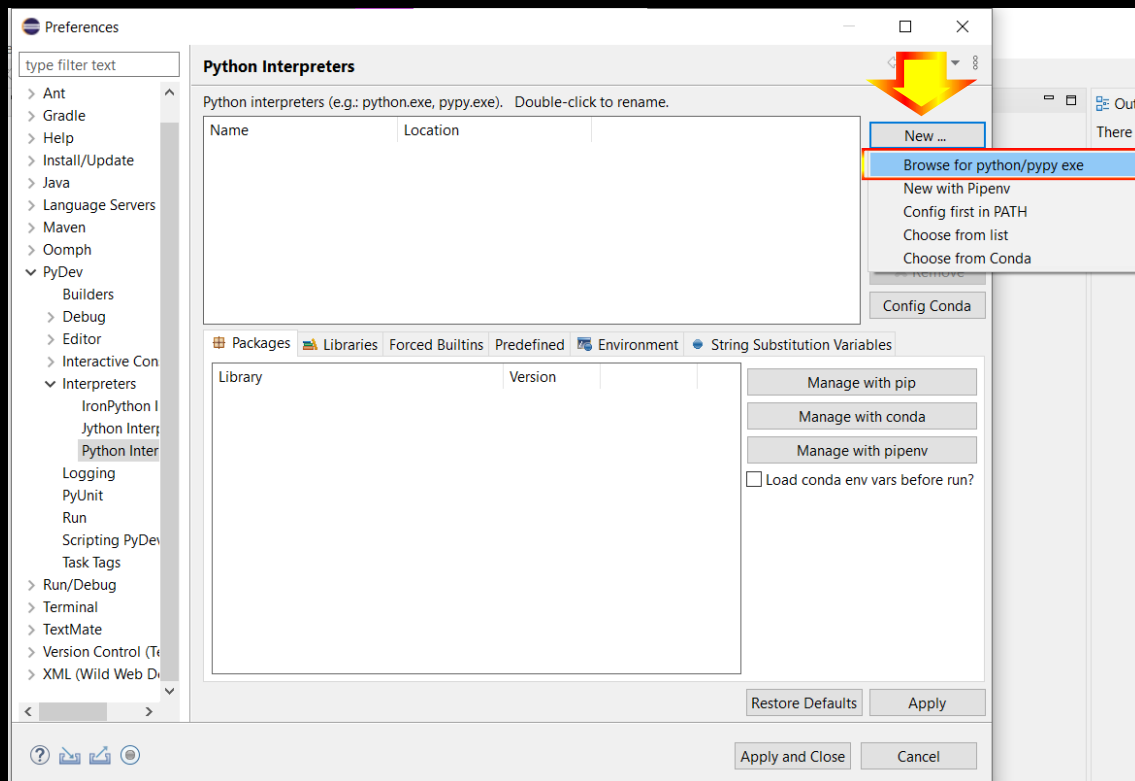
En el siguiente paso le tenemos que facilitar el ejecutable del Python que lo hicimos en el punto 1 que es: "Descarga la última versión Python Para windows". El ejecutable que nos descarguemos en su momento lo utilizaremos ahora es el compilador de Python.



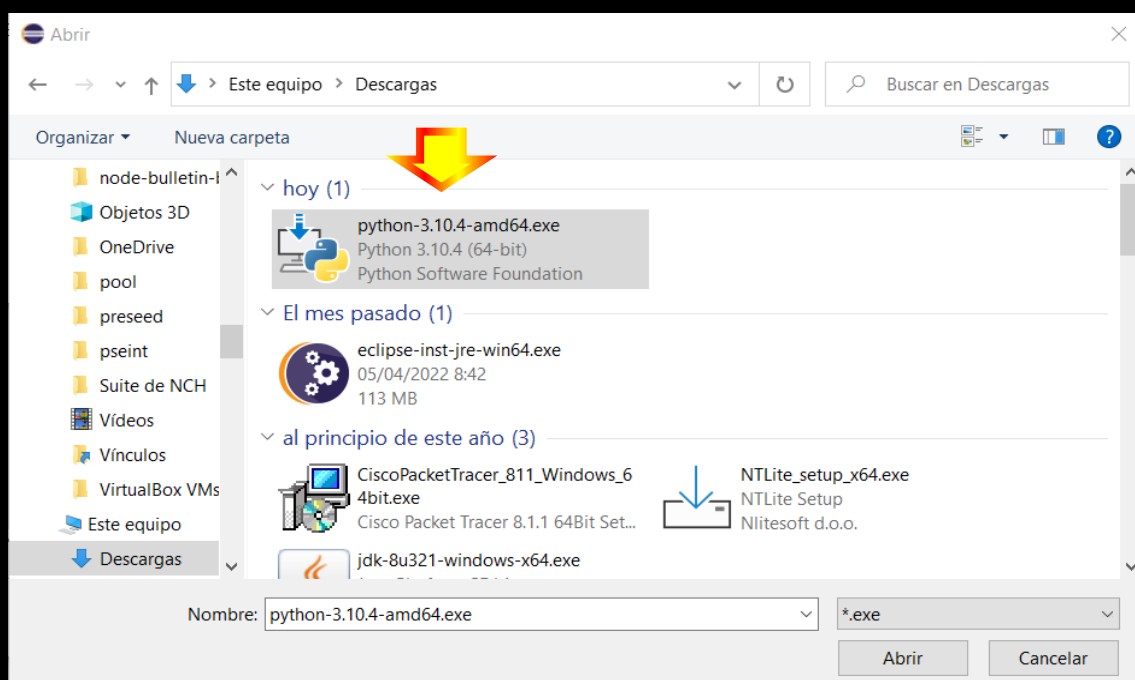
ESTER HITOS GARCIA & JOSE ANTONIO RIOS
INSTALACIÓN Y CONFIGURACIÓN DE IDE'S




Pinchamos en "New" y buscamos la descarga y la seleccionamos.

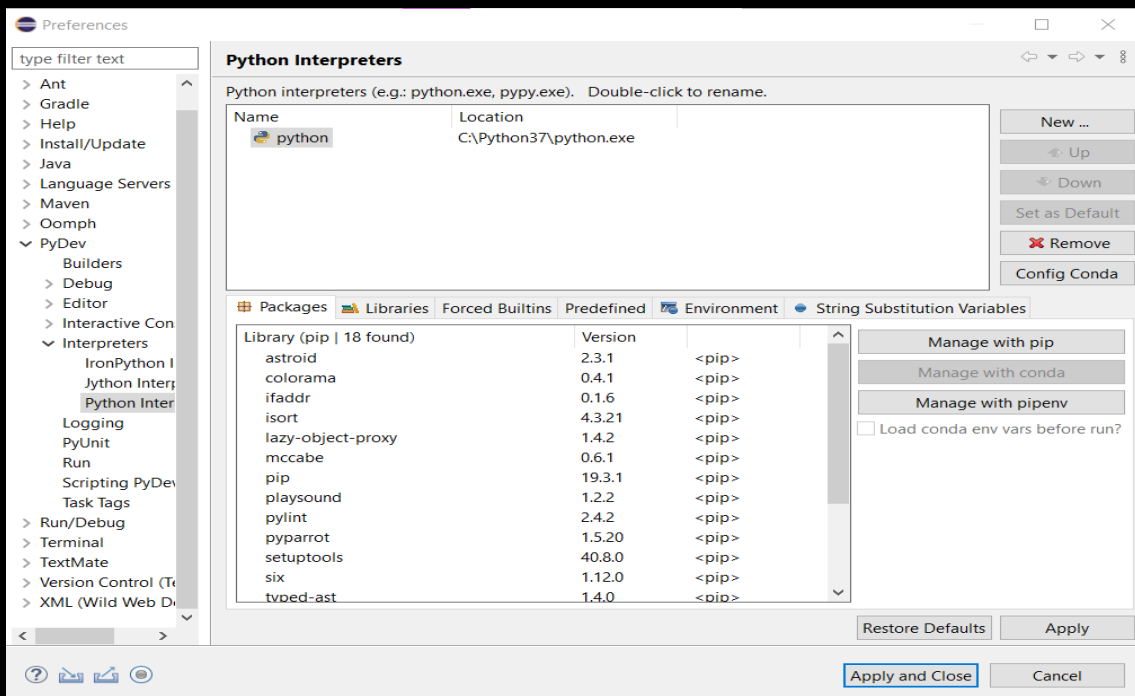



seleccionamos el ejecutable .exe

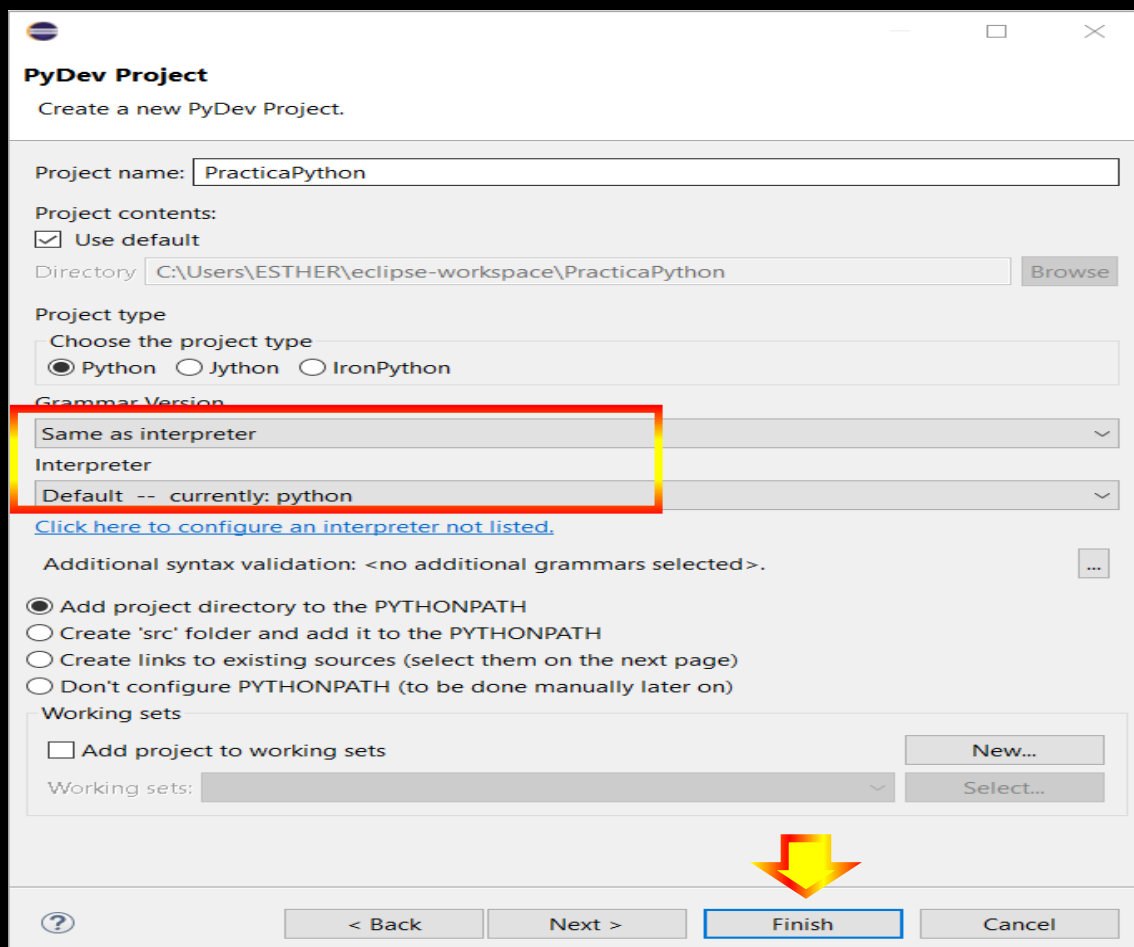


ESTER HITOS GARCIA & JOSE ANTONIO RIOS
INSTALACIÓN Y CONFIGURACIÓN DE IDE'S

 Y le pinchamos a "Apply and Close".

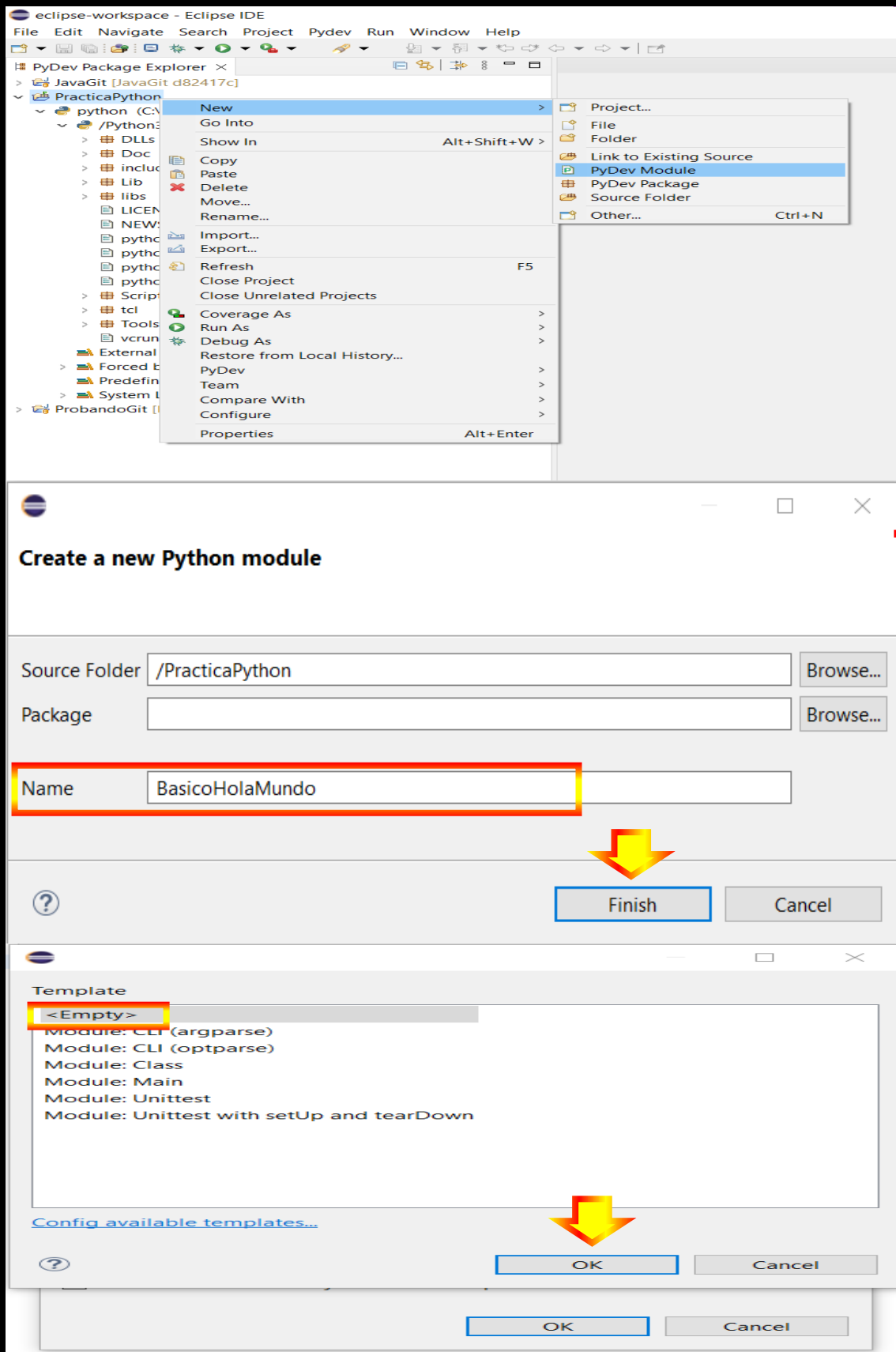


 El Firewall nos pedirá acceso para hacer los cambios en el disco duro y ya podemos dar a finalizar a la realización del proyecto en Python.

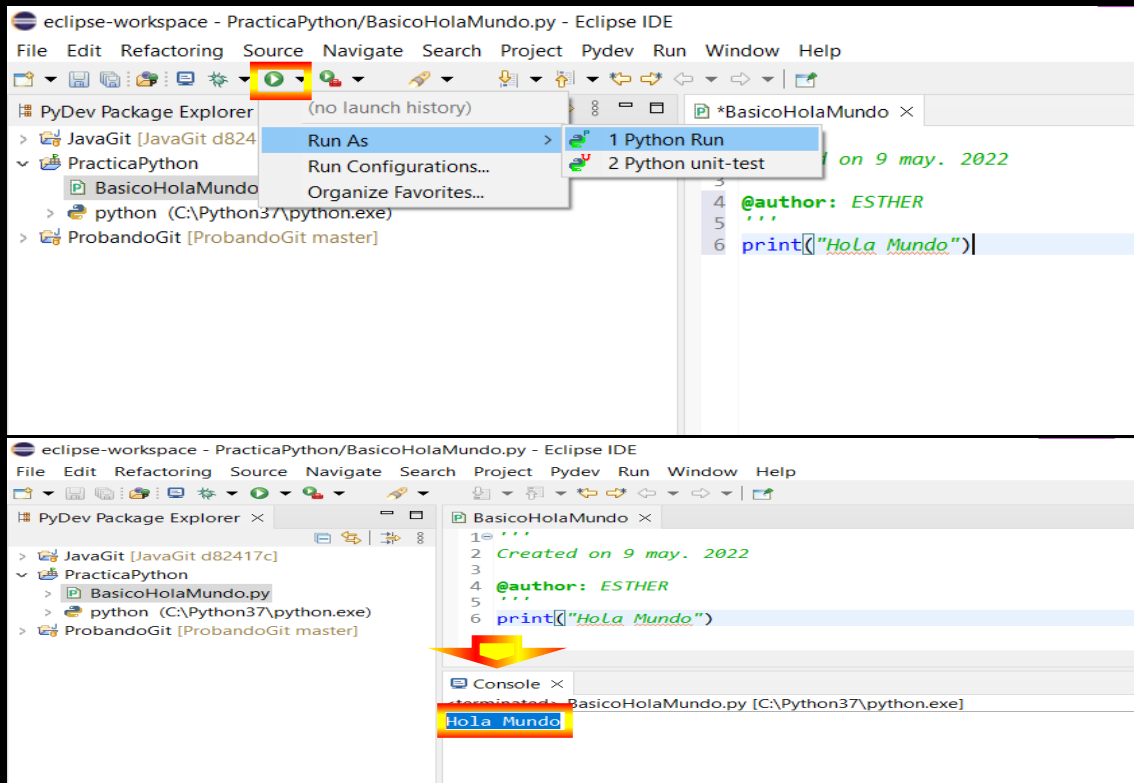


03. Creación de Módulo "BasicoHolaMundo" en el proyecto Python.

- 🖱️ Pincha con el botón derecho del ratón en "PracticaPython", escoge "New" y "PyDev Module"

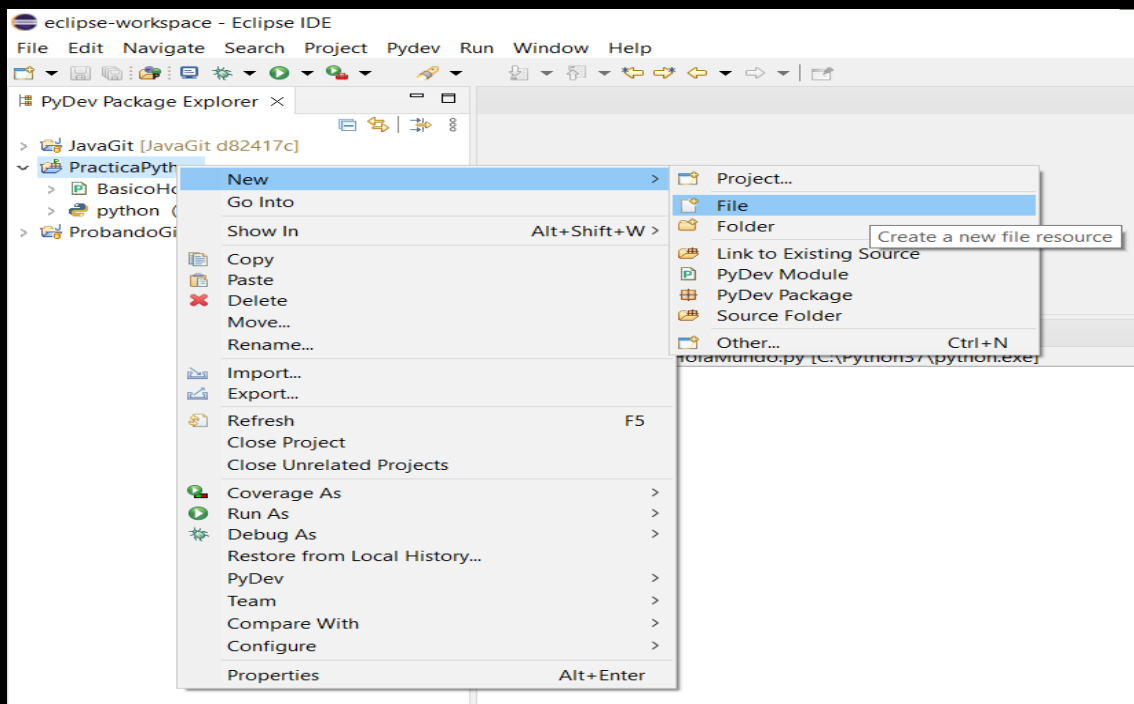


📁 Nos aparece un fichero llamado “BasicoHolaMundo.py”, con este archivo ejecutaremos un “Hola mundo” por consola, escribiendo “Print(“Hola Mundo”)” y ejecutándolo.

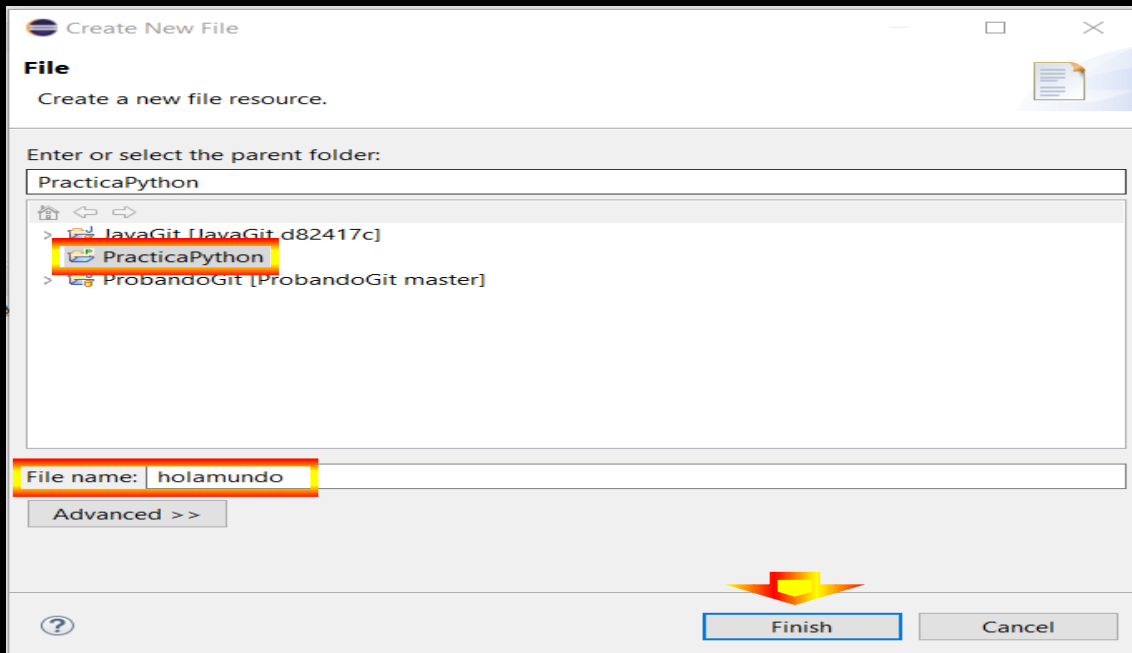


04. Creación de Fichero “HoLaMundo.py” en el proyecto Python.

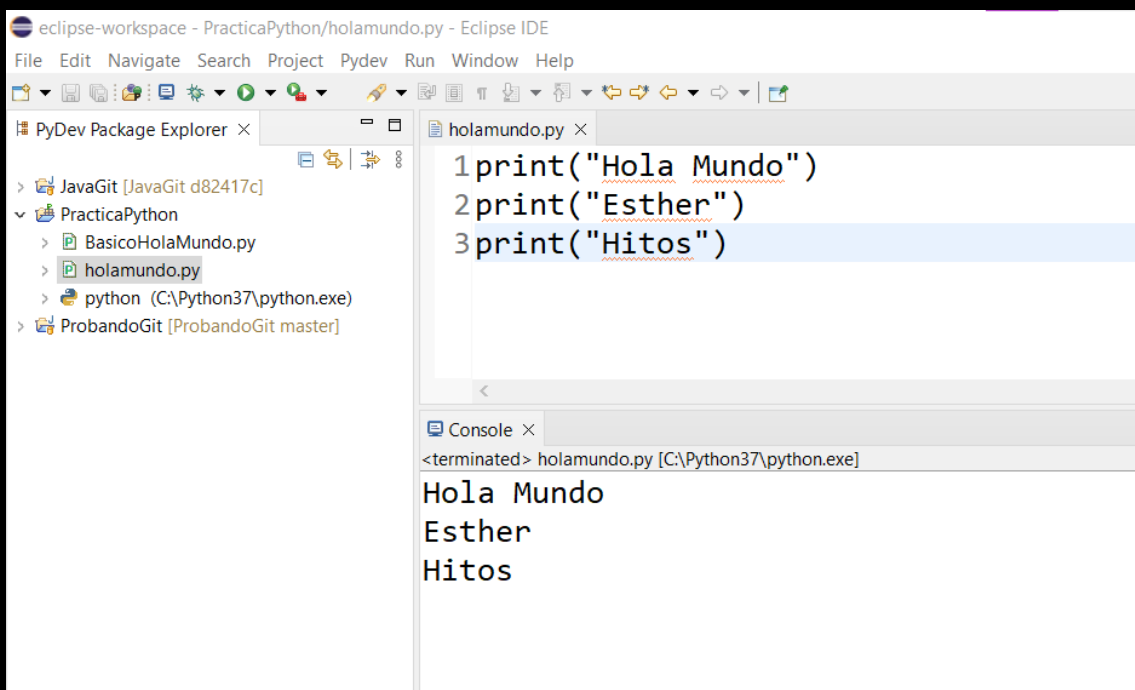
📁 En esta ocasión abriremos un fichero de una manera más simple.



ESTER HITOS GARCIA & JOSE ANTONIO RIOS
INSTALACIÓN Y CONFIGURACIÓN DE IDE'S



En este caso solo Escribimos la salida en consola y pondré dos salidas mas con mi nombre y apellido.

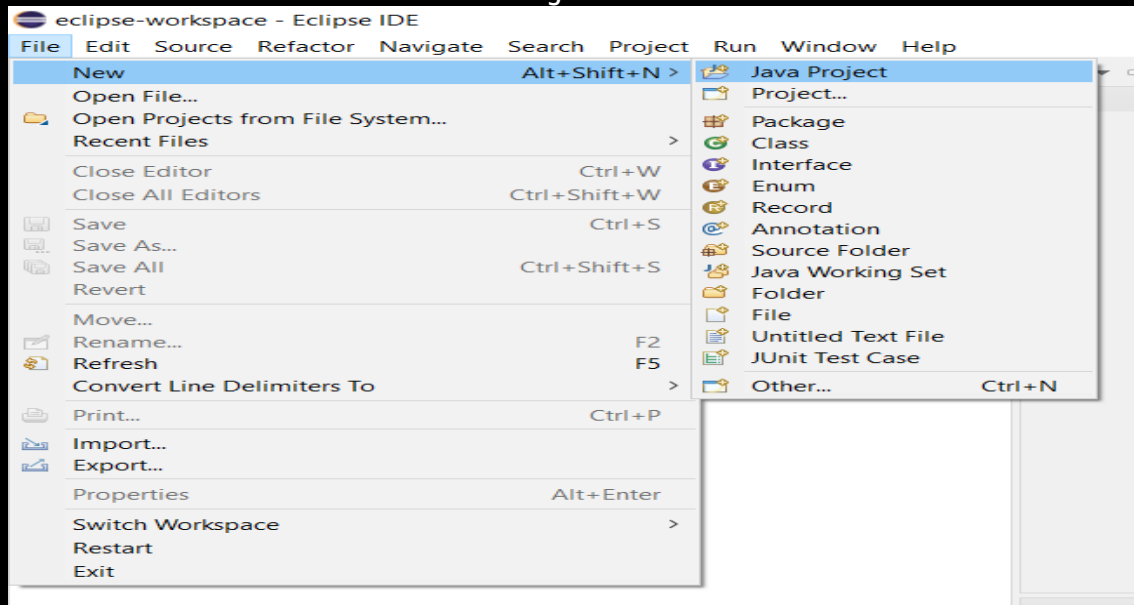


¡¡¡Ya sabes utilizar Python en Eclipse!!!

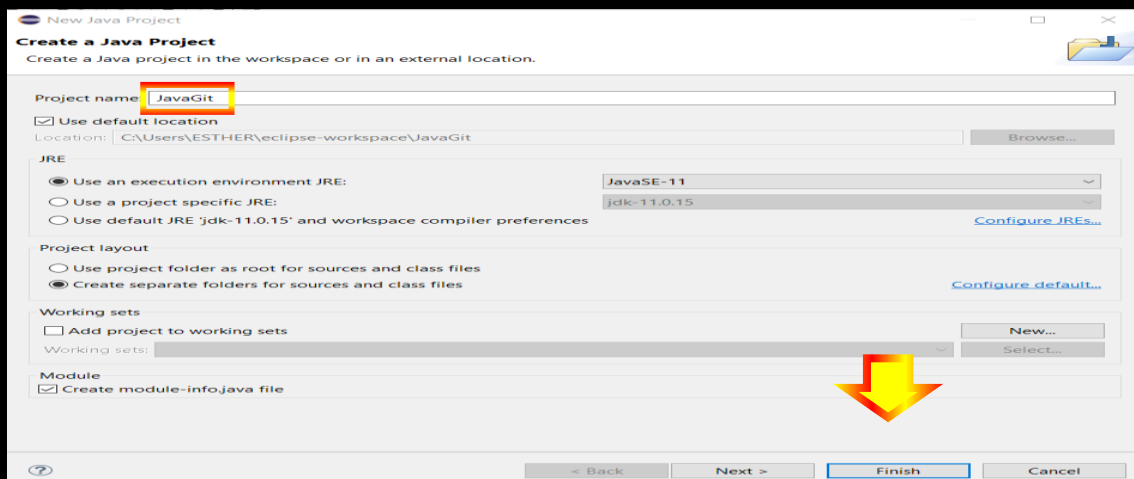
3.ECLIPSE: UTILIZACIÓN DE GIT Y GITHUB

01. Proceso para activar git en un proyecto de java

- ◆ Creamos un proyecto java, pulsando "File" → "New" → "Java Project"

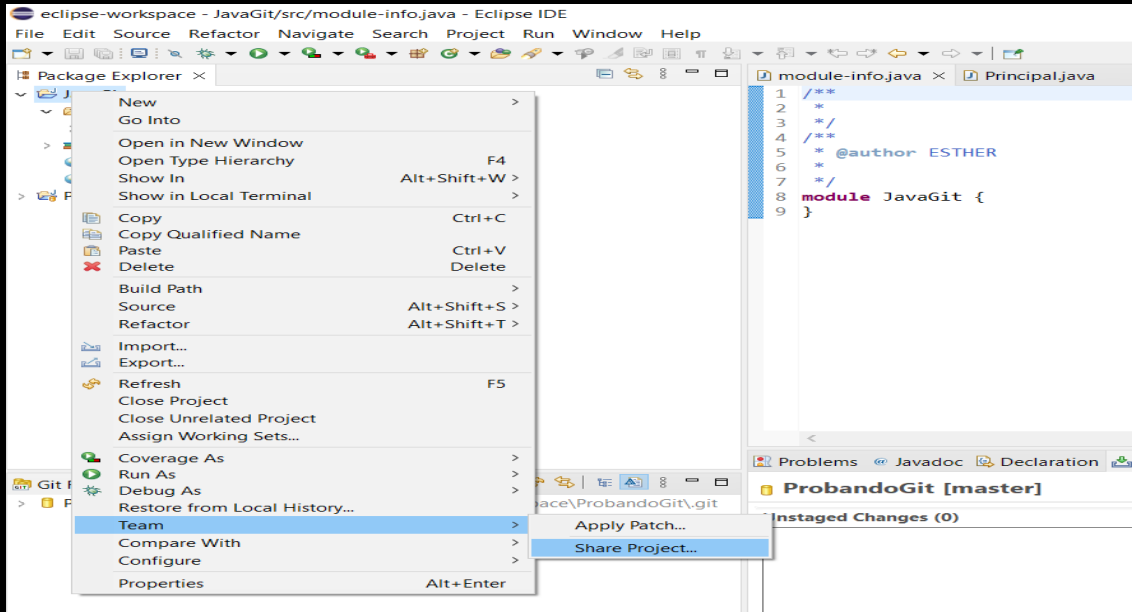


- ◆ Y le ponemos el nombre al proyecto "JavaGit"

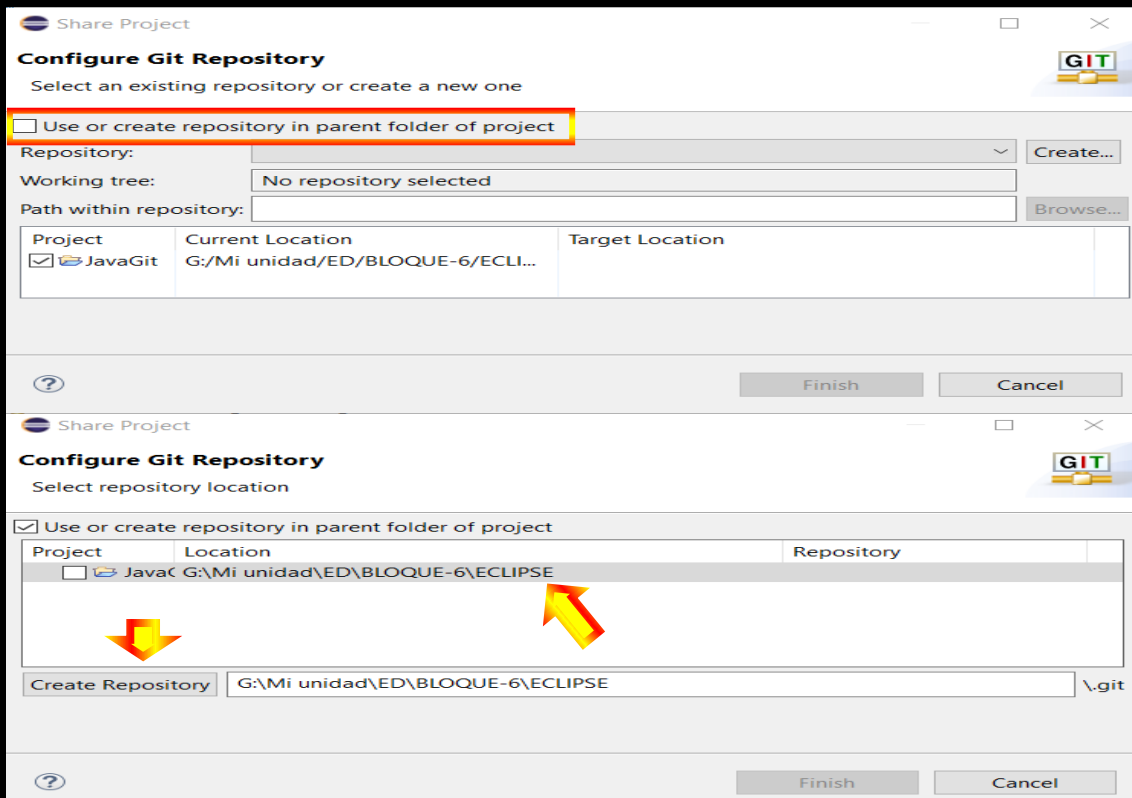


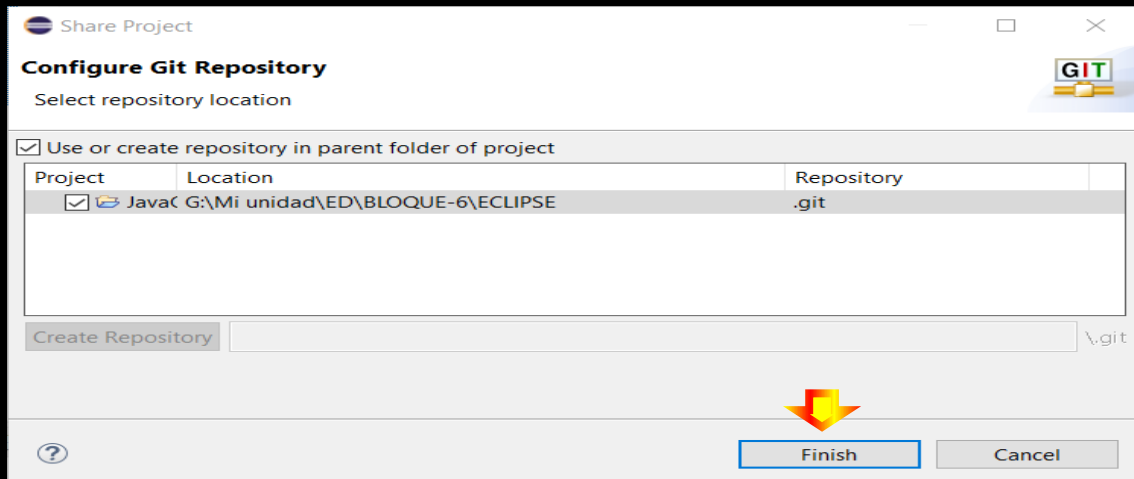
02. Inicializar el proyecto "git init" en eclipse

- Pinchamos en el proyecto y en el botón derecho del ratón y nos sale un menú desplegable, pinchamos la opción "Team" y a continuación "Share Project". De esta manera se hace un "git init" en eclipse.

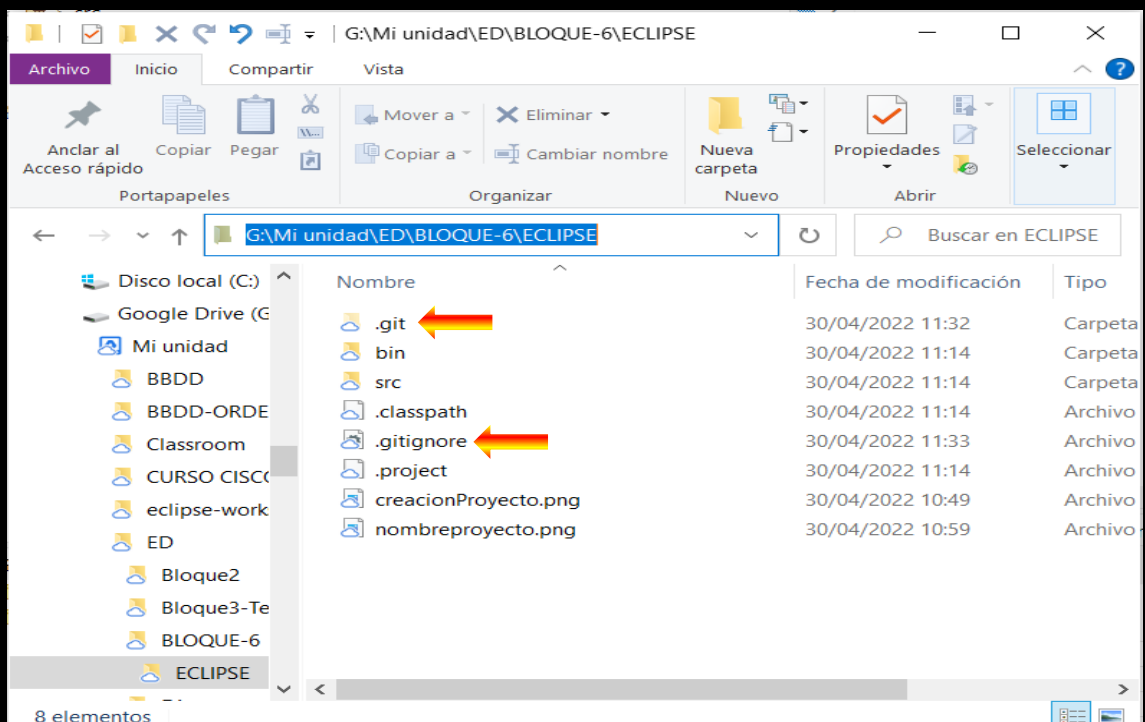


- Una vez que hemos hecho el "share Project" se nos abre una ventana de configuración del repositorio de git. Debemos pinchar en usar o crear el repositorio en la carpeta padre o proyecto.



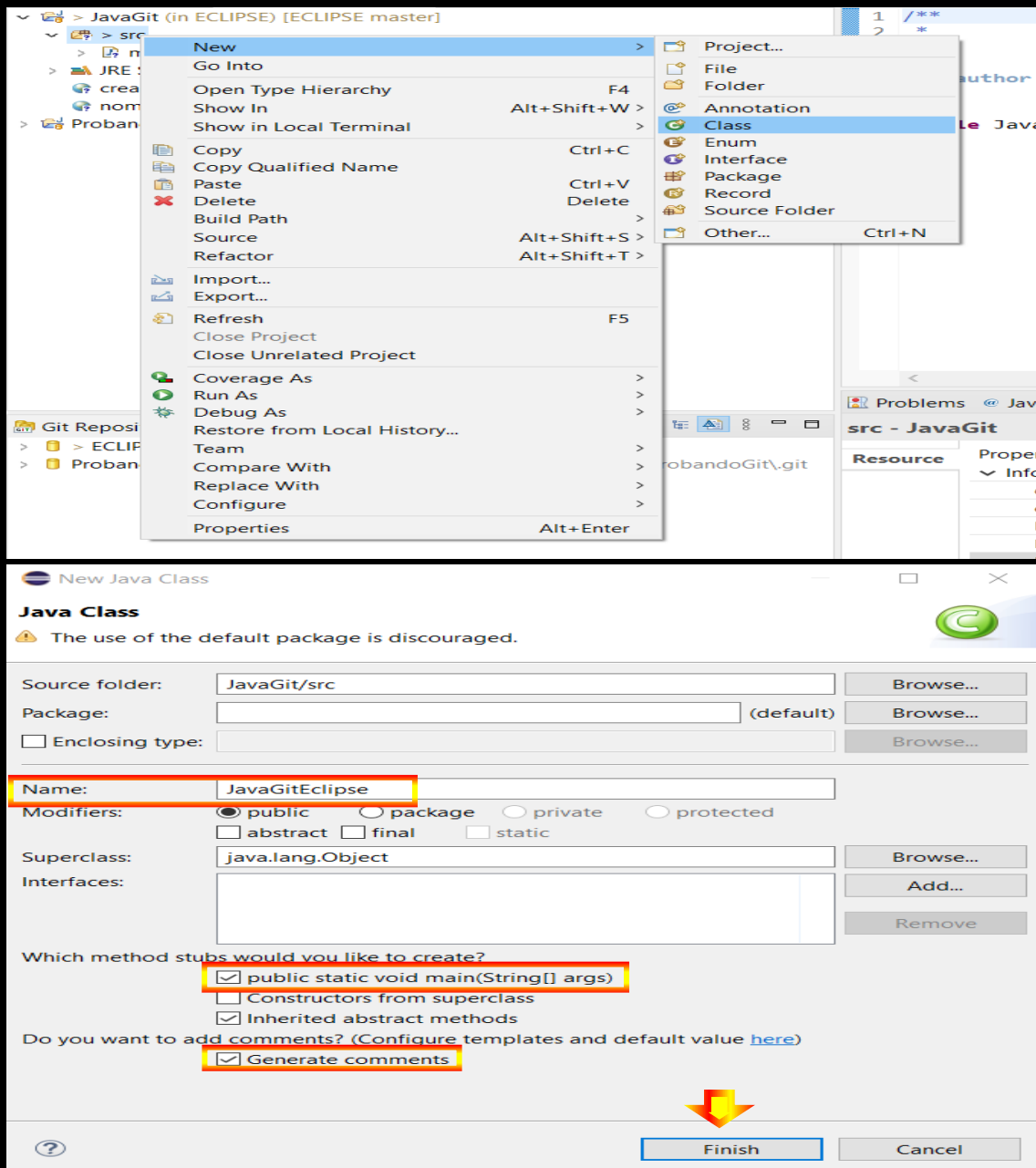


Comprobamos que en la carpeta del proyecto esté la carpeta oculta de git del proyecto ".git" y el fichero ".gitignore", si están es que ya se a creado el repositorio de git para el proyecto.

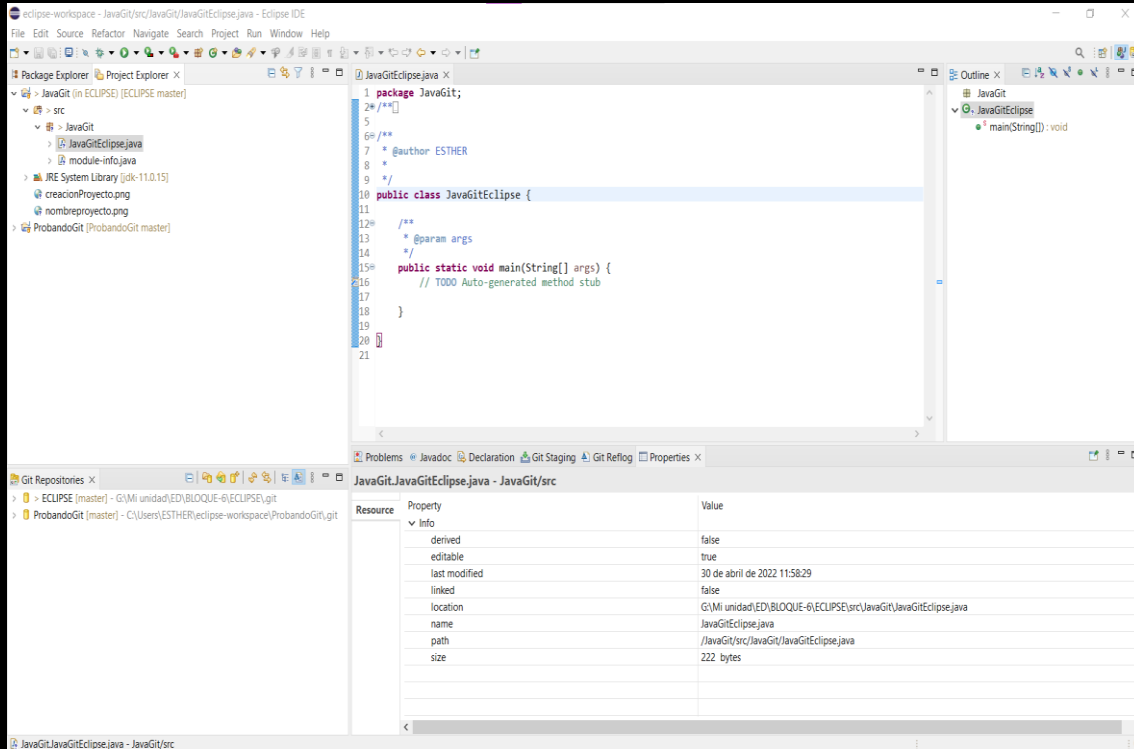


03. Crear fichero .class y guardar estado en git con commit

- ❖ Pulsamos en el proyecto "JavaGit" y en "src", creamos un nuevo fichero "JavaGitEclipse.class"

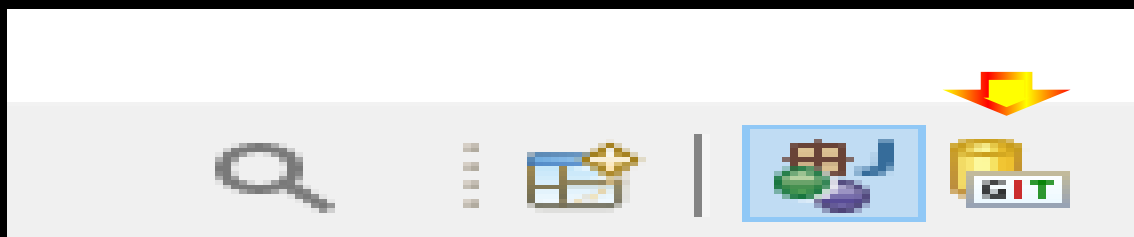


- Ya tenemos creado el fichero para empezar a trabajar con java y con el primer cambio ir guardando en el área de trabajo “staging área” de git y luego poder hacer los “commits”

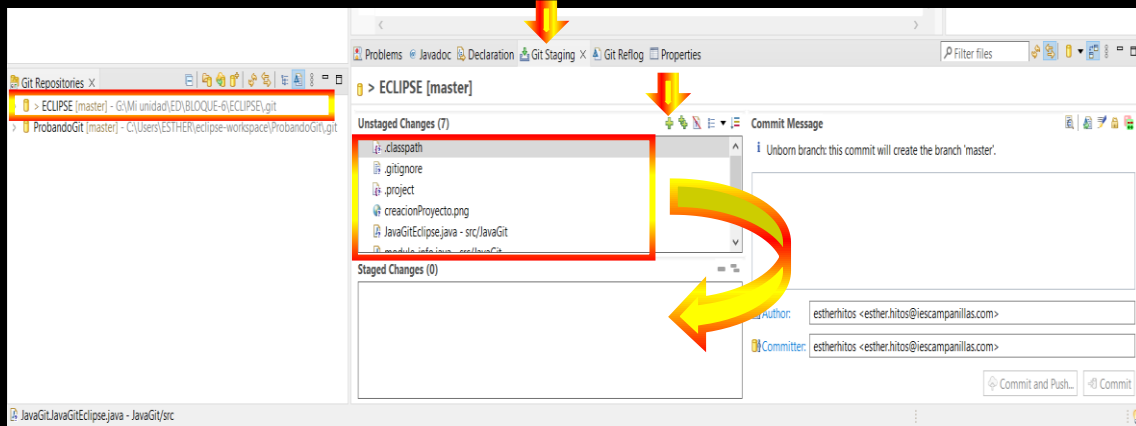


04. Comprobar el Staging Area del proyecto JavaGit en Eclipse

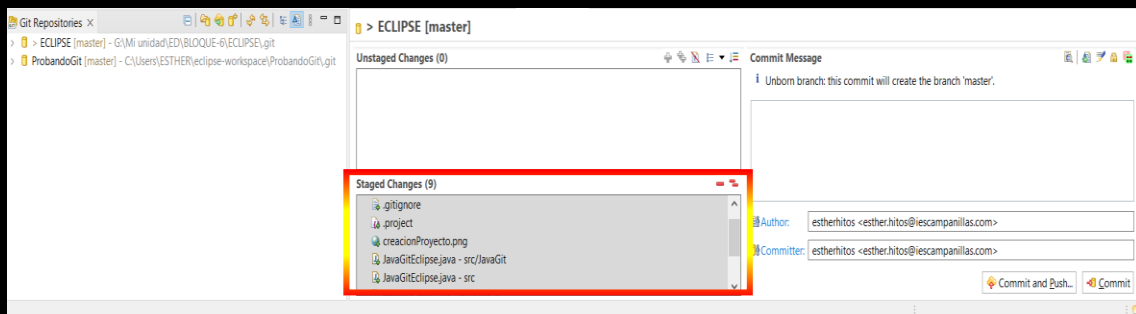
- Para poder visualizar el Area de trabajo de git, tenemos una pestaña en la parte superior derecha del IDE de Eclipse para visualizar el proyecto con utilidades para el repositorio de git.



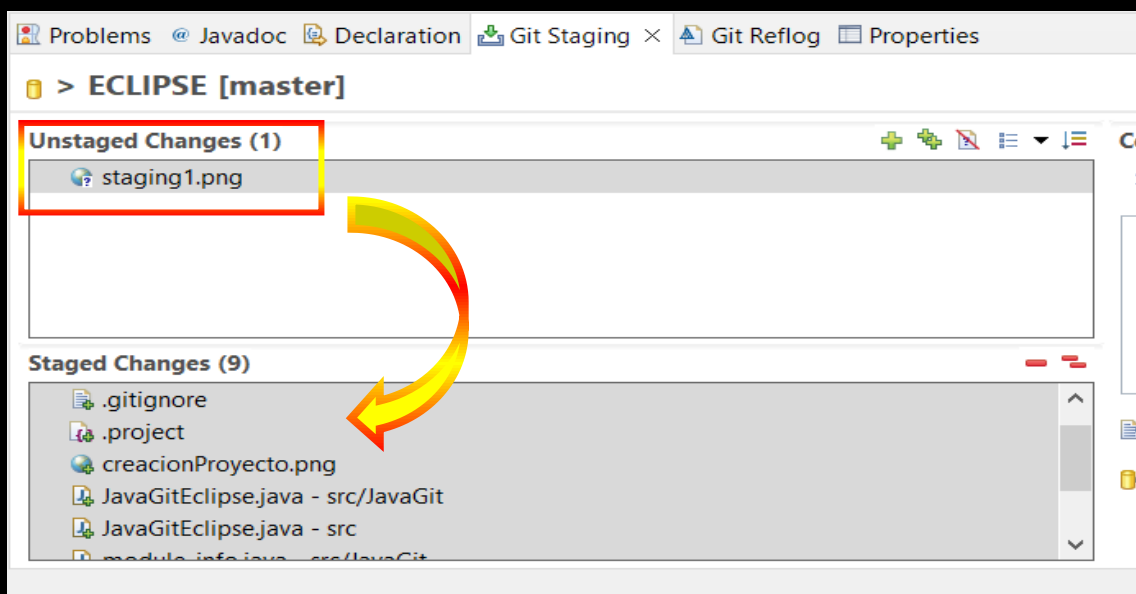
- Se nos abrirá una visualización diferente y en la parte de abajo del fichero nos indica los ficheros que están esperando para pasar al staging área en la opción "Git Staging".

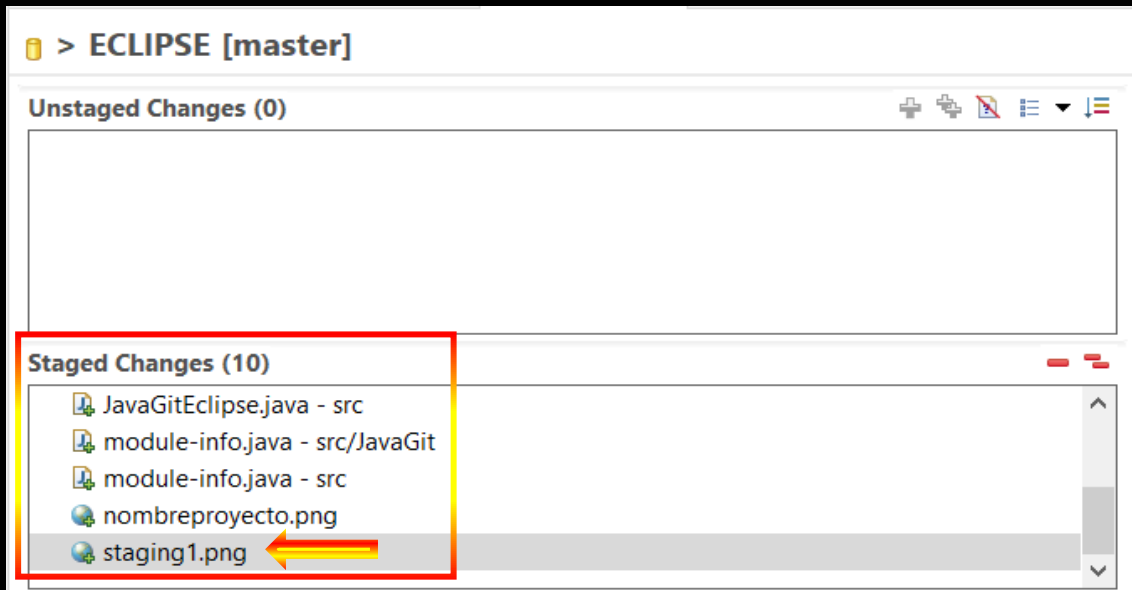


- Tenemos que seleccionar todo lo que queramos para dejarlo preparado para hacer los commit después para nuestro repositorio git, y pulsando al "+" de color verde o arrastrando para la casilla de "Staged Changes" justo abajo.

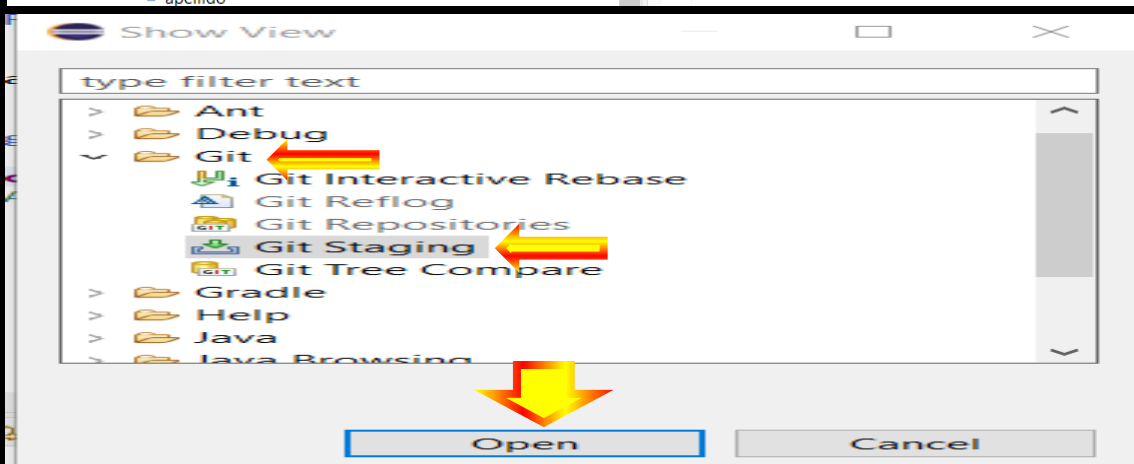
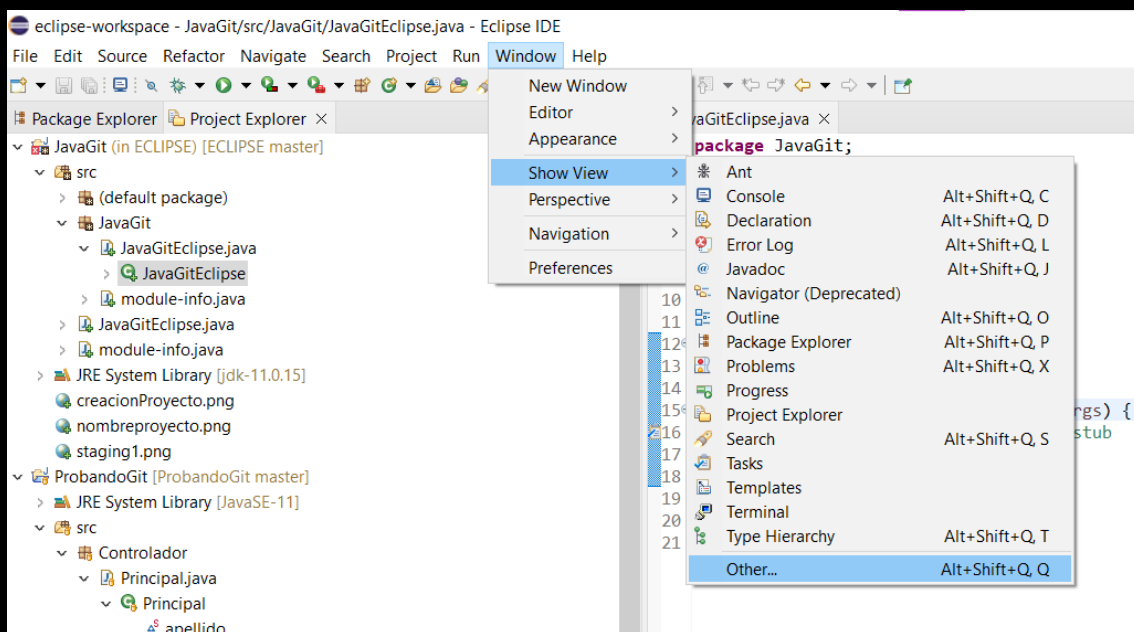


- Si seguimos insertando ficheros o paquetes al proyecto se quedarían en "Unstaged Changes" esperando que se pasaran al "Staged Changes" que es el sitio donde después hay que hacer un commit.



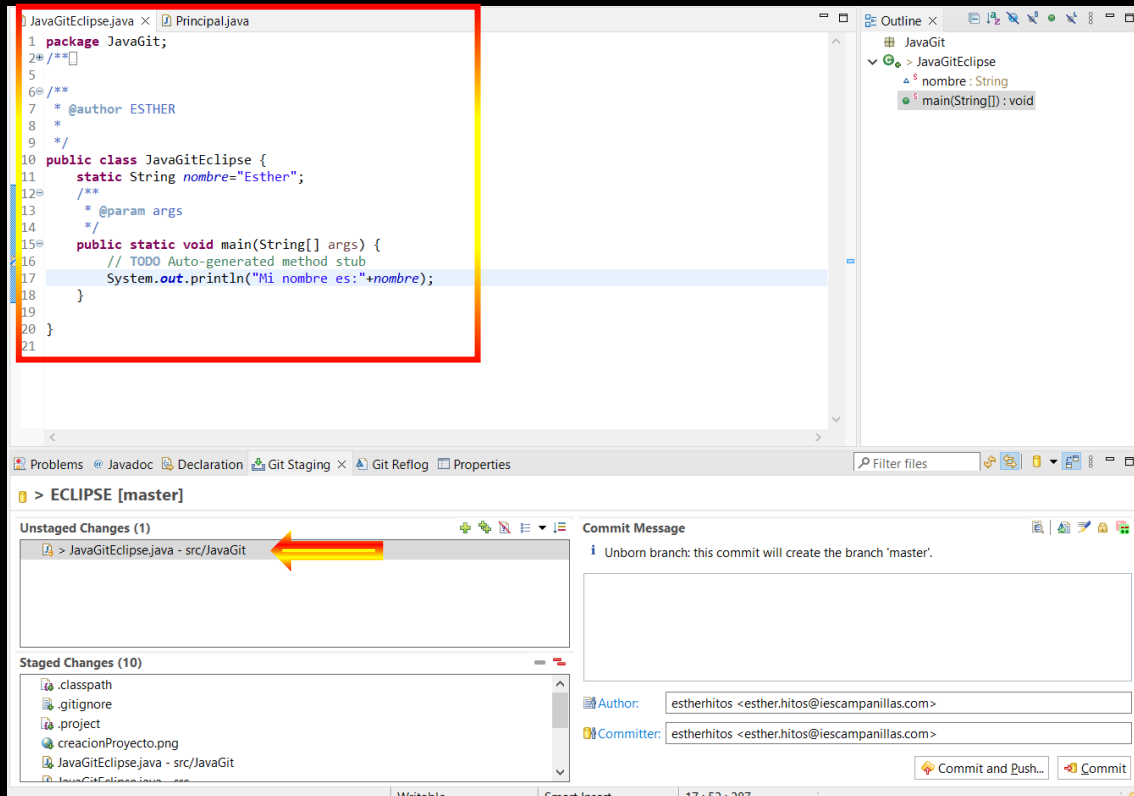


♦ También en vista de modo java, podemos abrir la ventana de “Git Staging” solamente para tenerlo a la vista mientras hacemos un pequeño programa java para probar los commit

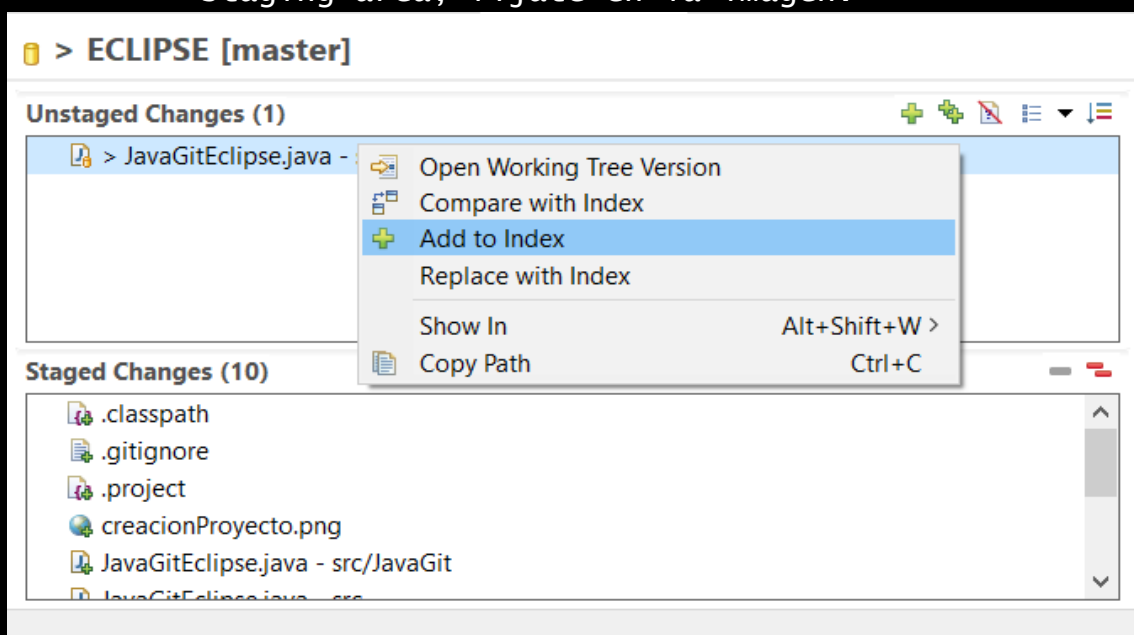


05. Realizar los commit en el repositorio git con Eclipse

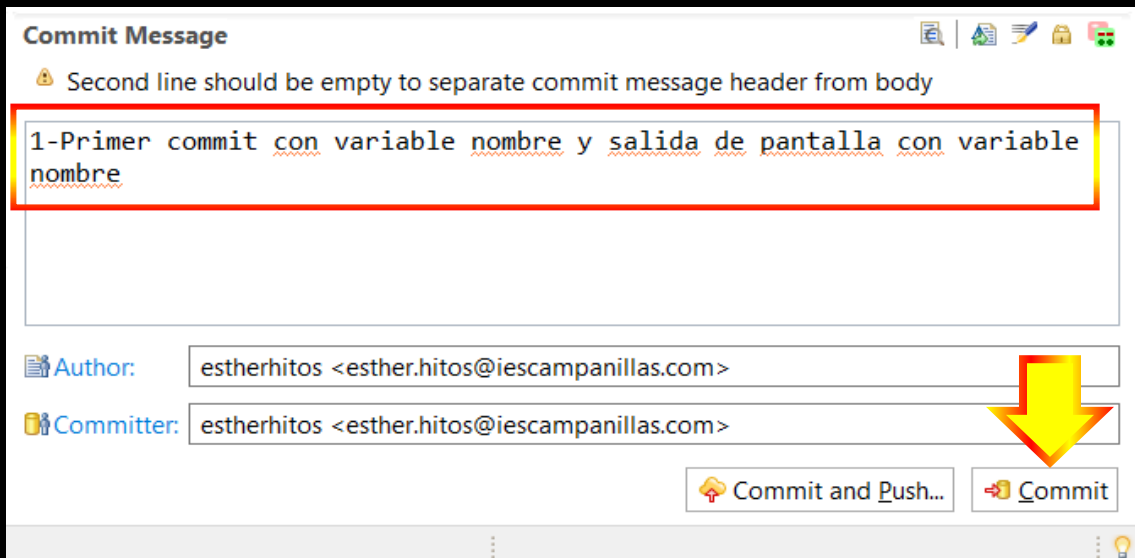
- He creado una variable y una salida de pantalla con esa variable en el .class y fíjate como el staging área esta a la espera de otro cambio!



- Antes del primer commit vamos a pasar de otra manera diferente la actualización del fichero al staging área, fíjate en la imagen.



- Una vez ya tenemos todos los cambios en el "Staging Changes" ya podemos hacer el primer "commit" copia de respaldo, en la parte inferior derecha, tenemos que realizar una descripción de nuestro commit, porque si queremos en algún momento regresar a nuestra copia de respaldo, a este punto tan inicial del proyecto, nos facilita mucho la vida, si le ponemos una descripción para saber que hicimos en ese commit y regresar a él para editarlo.



Commit Message

⚠ Second line should be empty to separate commit message header from body

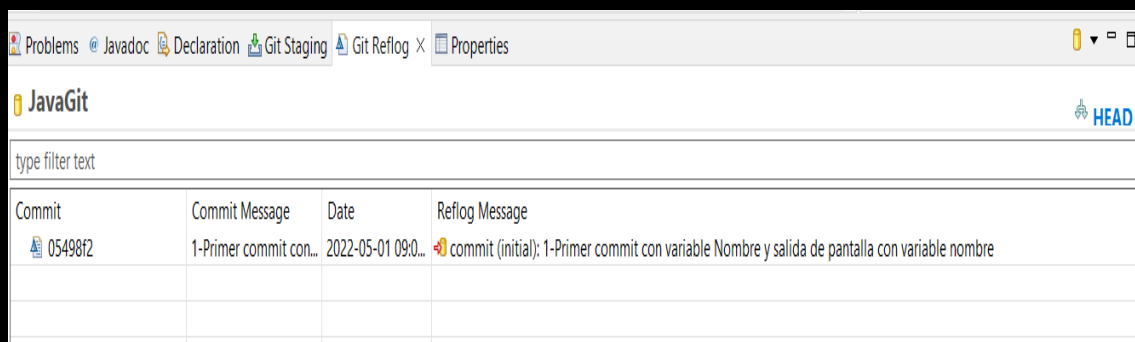
1-Primer commit con variable nombre y salida de pantalla con variable nombre

Author: estherhitos <esther.hitos@iescampanillas.com>

Committer: estherhitos <esther.hitos@iescampanillas.com>

Commit and Push... Commit

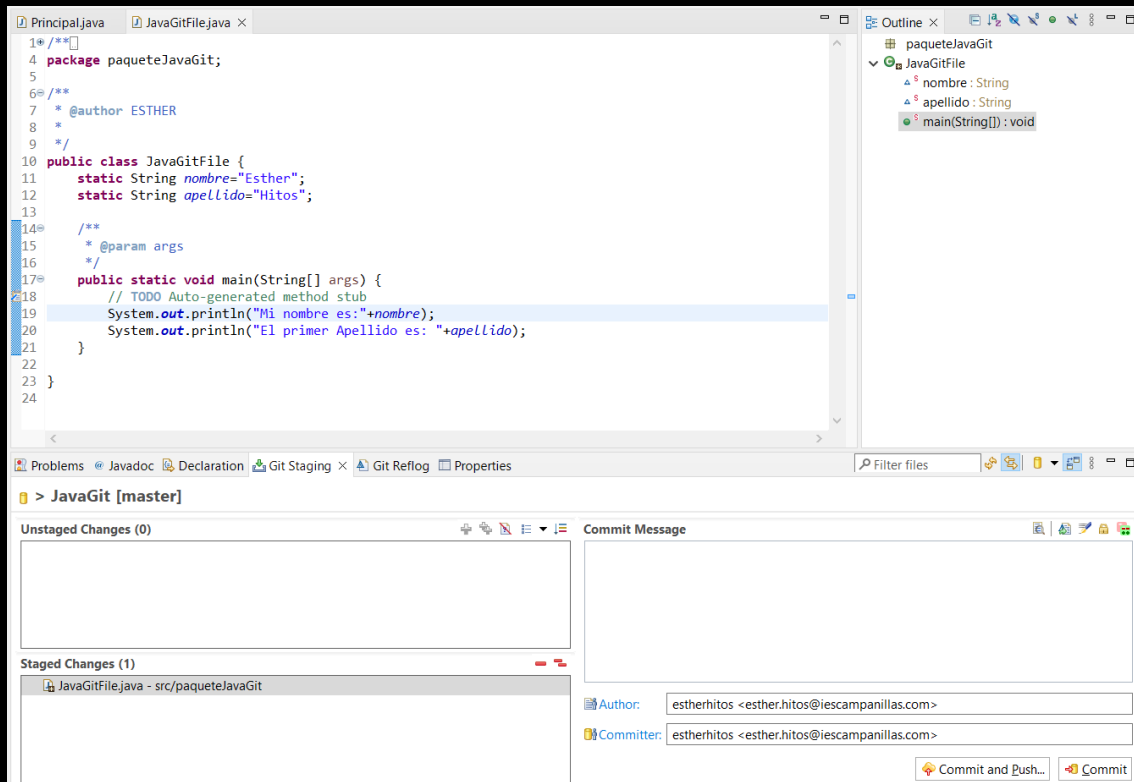
- Como podemos observar en la pestaña "Git Reflog" nos indica el id del primer commit y su comentario.



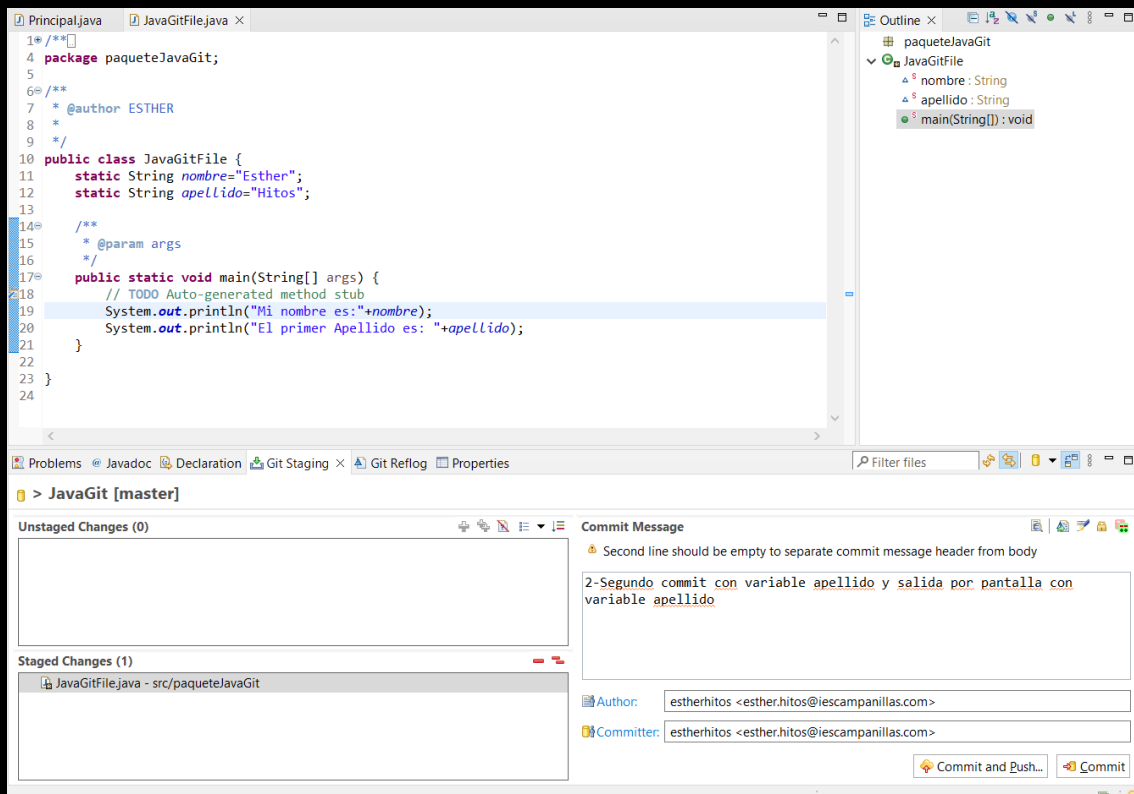
Commit	Commit Message	Date	Reflog Message
05498f2	1-Primer commit con...	2022-05-01 09:0...	commit (initial): 1-Primer commit con variable Nombre y salida de pantalla con variable nombre

ESTER HITOS GARCIA & JOSE ANTONIO RIOS INSTALACIÓN Y CONFIGURACIÓN DE IDE'S

➡ Seguimos trabajando en el proyecto, y añado la variable "Apellido" con su salida de pantalla, automáticamente repito la operación que hemos realizado con el primer commit.

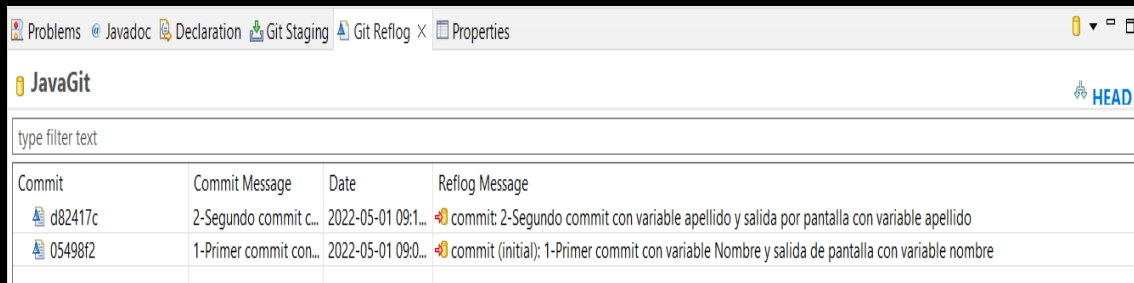


➡ Como se puede observar en la imagen hemos tenido que repetir el movimiento del staging Area y ahora realizaremos el 2º commit.



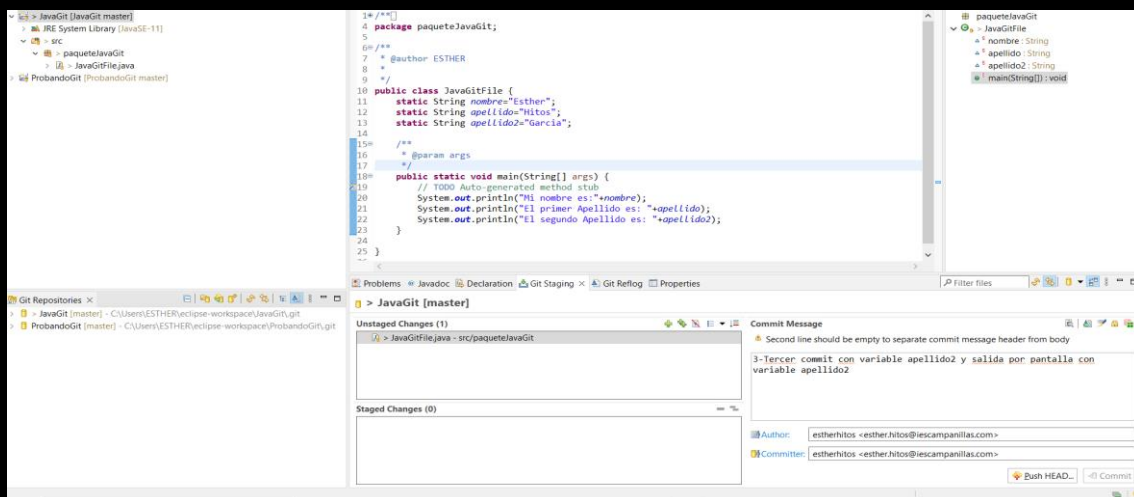
ESTER HITOS GARCIA & JOSE ANTONIO RIOS INSTALACIÓN Y CONFIGURACIÓN DE IDE'S

- Como podemos observar en la pestaña “Git Reflog” nos indica el id del primer commit y segundo commit con sus comentarios.

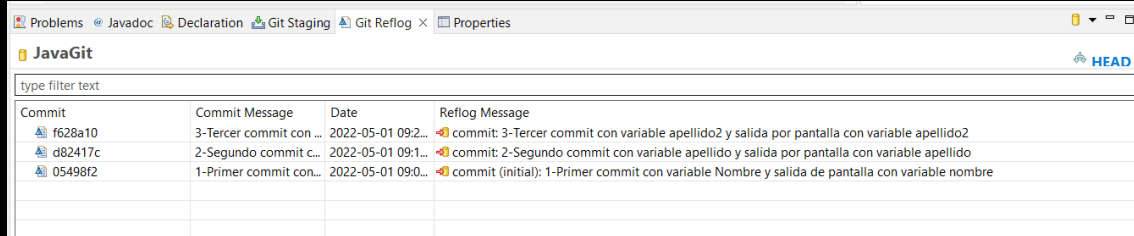


Commit	Commit Message	Date	Reflog Message
d82417c	2-Segundo commit con...	2022-05-01 09:1...	commit: 2-Segundo commit con variable apellido y salida por pantalla con variable apellido
05498f2	1-Primer commit con...	2022-05-01 09:0...	commit (initial): 1-Primer commit con variable Nombre y salida de pantalla con variable nombre

- Tercer commit y último, repetimos “Staging Area”, comentario en “commit” y le damos al botón de commit.



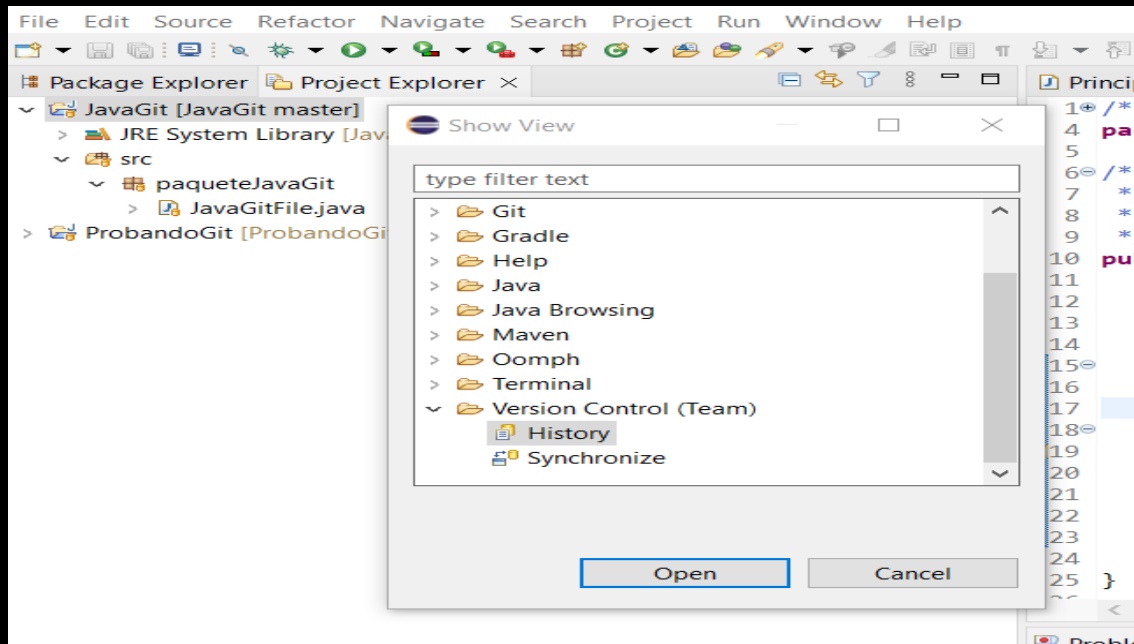
The screenshot shows the Eclipse IDE with the 'Commit' dialog box open. The 'Commit Message' field contains the text: '3-Tercer commit con variable apellido2 y salida por pantalla con variable apellido2'. The 'Author' and 'Committer' fields are both filled with 'estherhitos <esther.hitos@iescampanillas.com>'. The 'Push HEAD' button is visible at the bottom right of the dialog box.



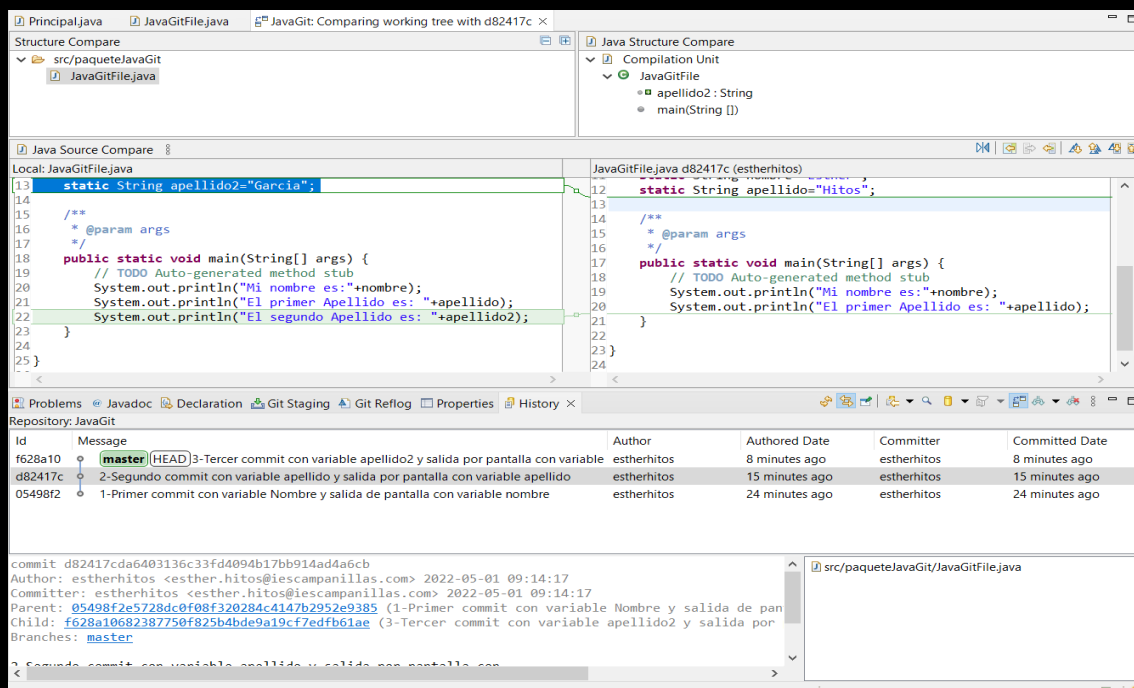
Commit	Commit Message	Date	Reflog Message
f628a10	3-Tercer commit con ...	2022-05-01 09:2...	commit: 3-Tercer commit con variable apellido2 y salida por pantalla con variable apellido2
d82417c	2-Segundo commit con...	2022-05-01 09:1...	commit: 2-Segundo commit con variable apellido y salida por pantalla con variable apellido
05498f2	1-Primer commit con...	2022-05-01 09:0...	commit (initial): 1-Primer commit con variable Nombre y salida de pantalla con variable nombre

06. Volver a los estados anteriores de los commits realizados.

- Si en algún momento quisieras ir para atrás en tu proyecto y quisieras ver los commit que has realizado para empezar a trabajar con el y modificarlo desde ese punto del commit en la pestaña "Windows", "Sow View", puedes mirar en la carpeta "Versión Control (Team)" el Historial de los commit.

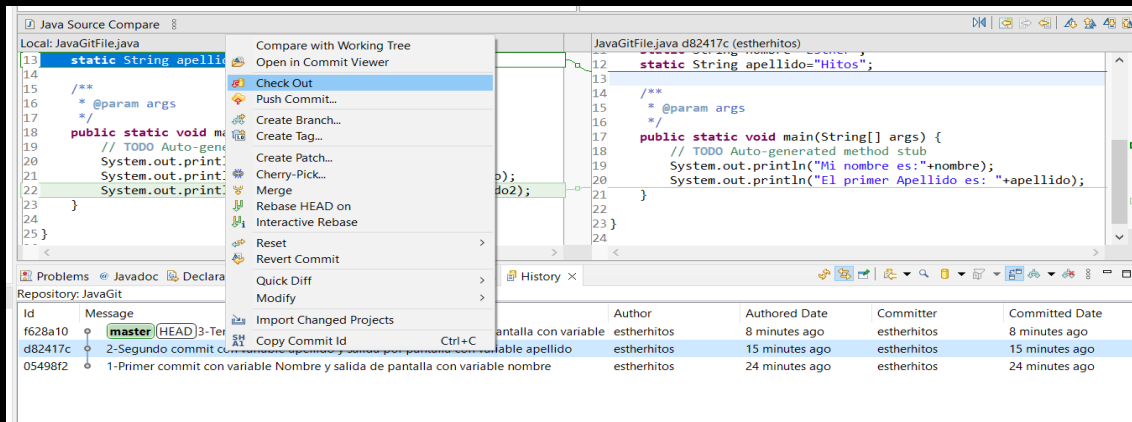


- Como se puede observar en la primera pantalla tenemos el estado actual de nuestro proyecto y en el lado derecho tenemos el estado de como quedo el proyecto en el 2º commit.

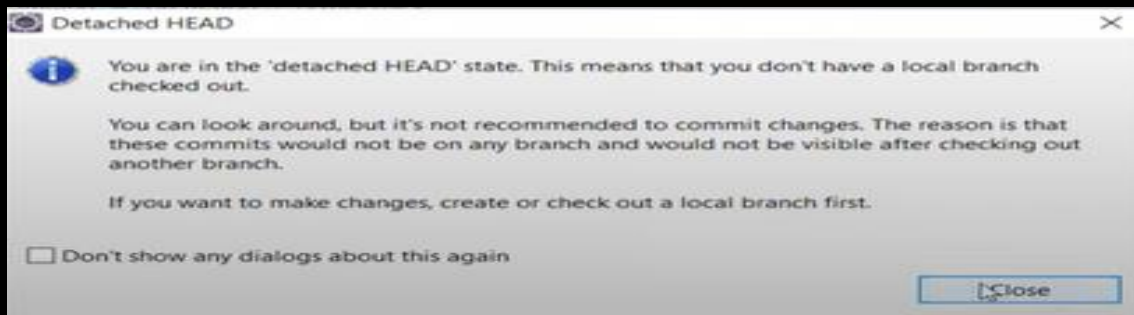


ESTER HITOS GARCIA & JOSE ANTONIO RIOS INSTALACIÓN Y CONFIGURACIÓN DE IDE'S

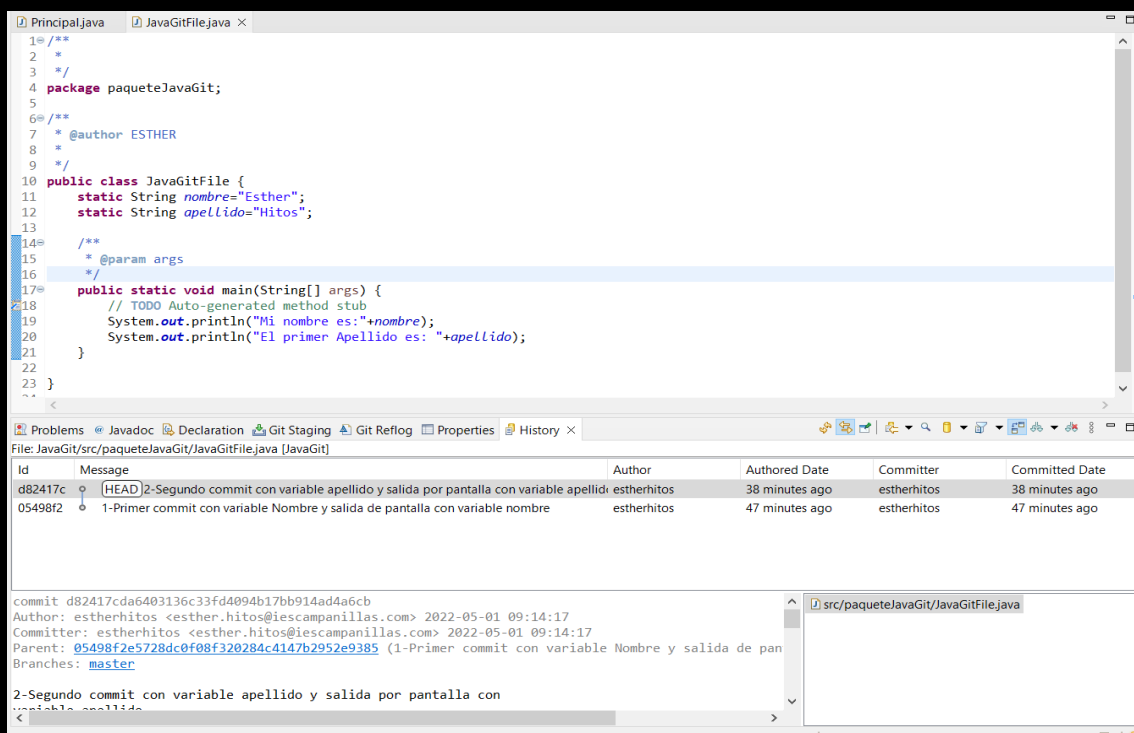
- Para poder modificar el estado de tu proyecto y regresar al commit anterior, tienes que darle al botón derecho del ratón en el commit al que quieres regresar y seleccionas un "check out"



- cuando lo pulsas sale un mensaje de aviso, para informarte de que si regresas a ese punto del proyecto y lo modificas ya no hay vuelta atrás.

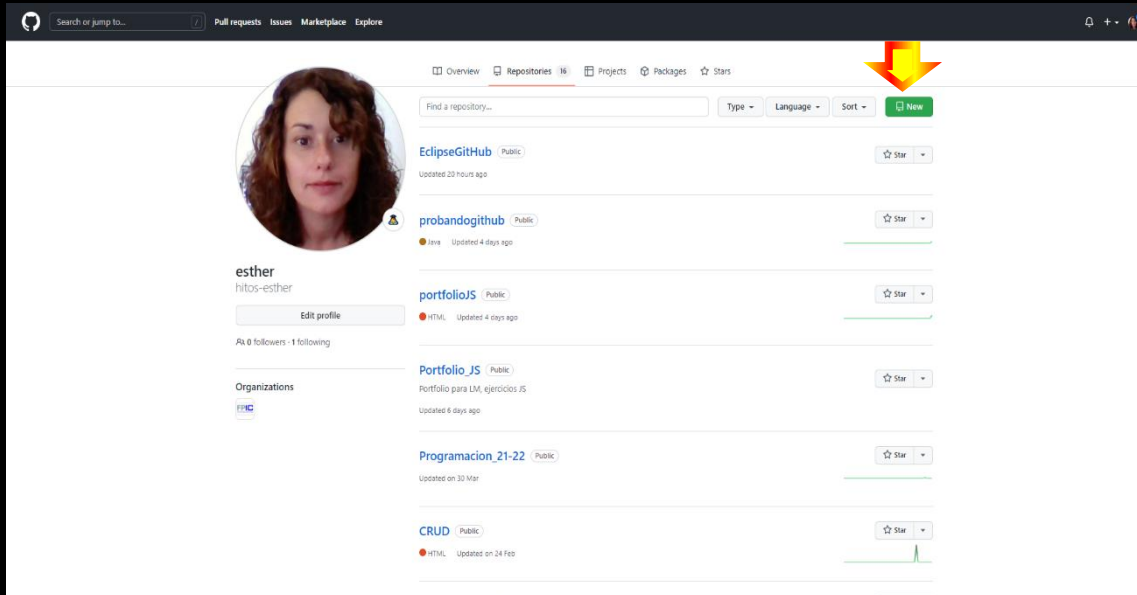


- Como se ve en la imagen de abajo ya no está el "Apellido2", hemos regresado al 2º commit que solo tenía el "Apellido".



07. Creación del Repositorio Remoto en Git Hub



- 👤 Ahora para poder subir nuestro proyecto al escritorio remoto primero tenemos que crear un repositorio remoto en git Hub.
- 👤 Nos vamos a git hub y en nuestro usuario creamos un nuevo repositorio.



Create a new repository


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


Owner * **Repository name ***

 hitos-esther / EclipseGit 

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

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:
Skip this step if you're importing an existing repository.


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


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

.gitignore template: None

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

License: None

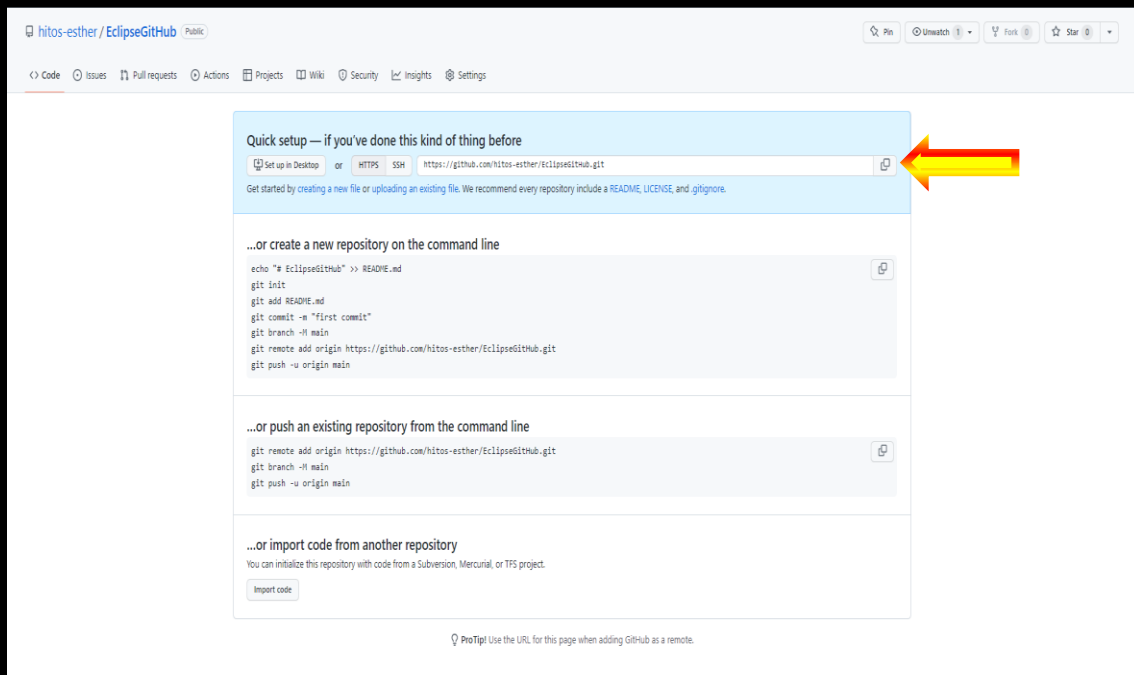
 You are creating a public repository in your personal account.



Create repository

ESTER HITOS GARCIA & JOSE ANTONIO RIOS INSTALACIÓN Y CONFIGURACIÓN DE IDE'S

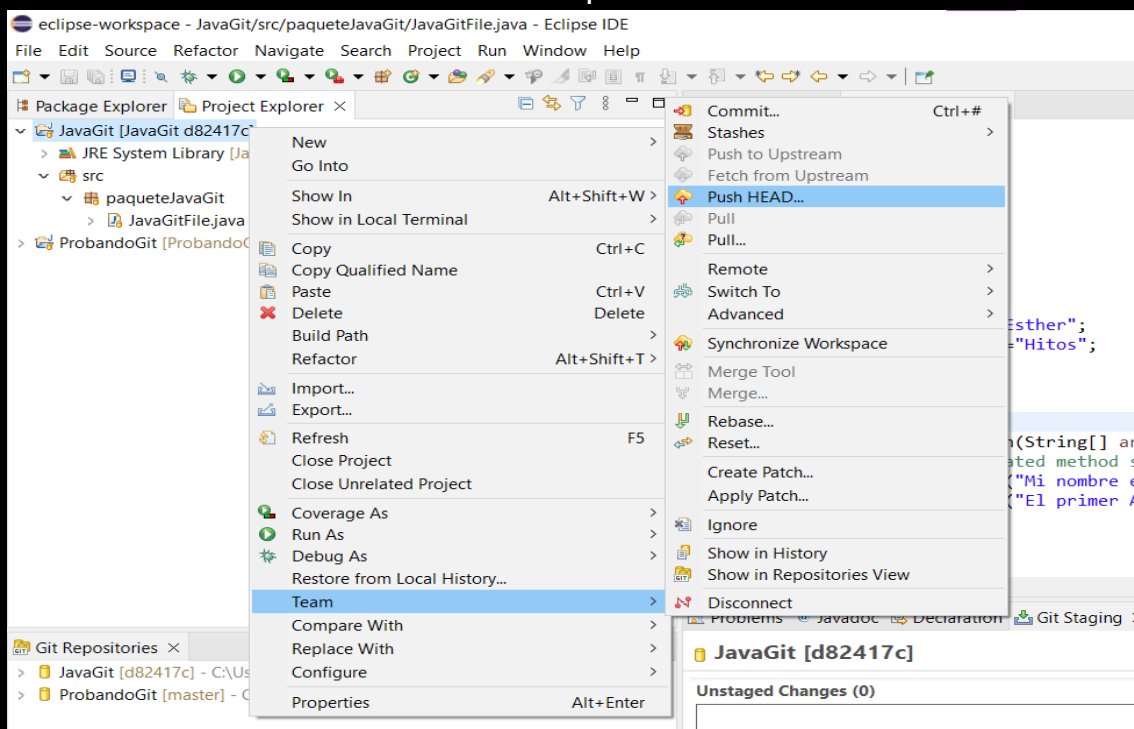
cuando se cree el nuevo repositorio te dará una dirección remota para poder direccionar luego nuestro proyecto de eclipse al repositorio remoto que tiene que subir el trabajo realizado.



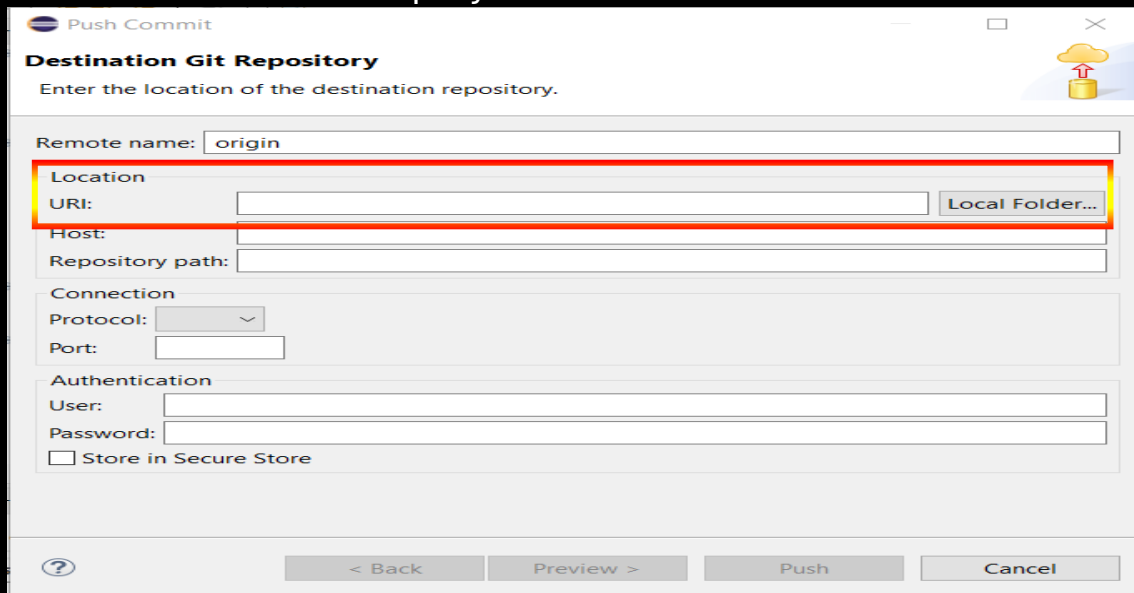
`https://github.com/hitos-esther/EclipseGitHub.git`

08. Push Head en Eclipse al Repositorio Remoto

En eclipse, en la vista de java, pinchamos en nuestro proyecto JavaGit, y seleccionamos "Team" y "Push Head", para indicarle la dirección remota de nuestro repositorio.



- ◆ Se nos abre una ventana donde pondremos la url de nuestro proyecto en "URI".



Push Commit

Destination Git Repository

Enter the location of the destination repository.

Remote name:

Location

URI:

Host:

Repository path:

Connection

Protocol:

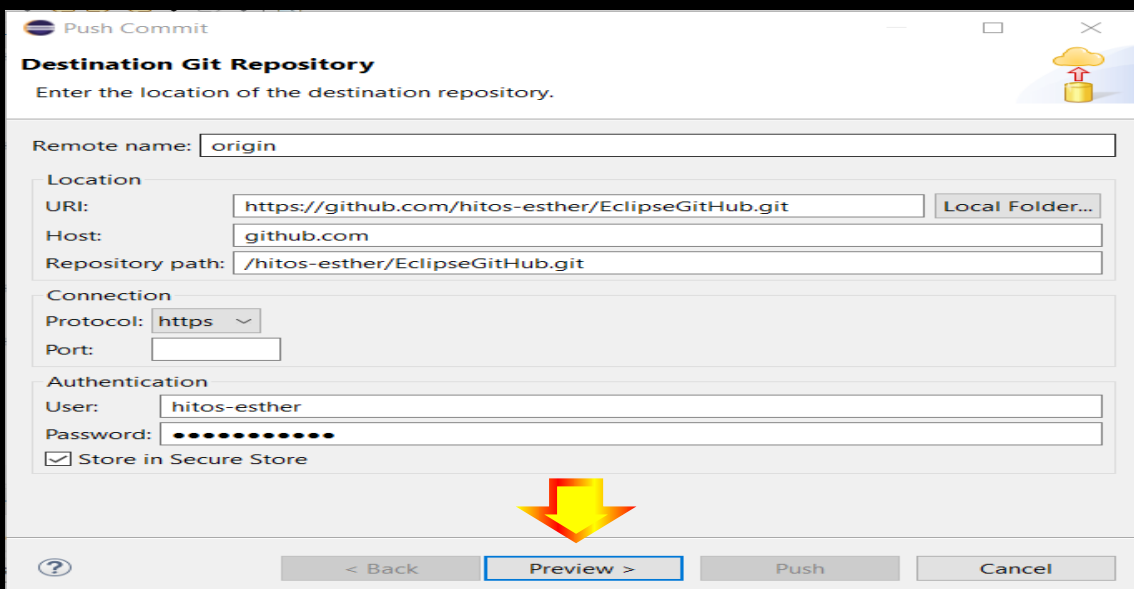
Port:

Authentication

User:

Password:

☐ Store in Secure Store



Push Commit

Destination Git Repository

Enter the location of the destination repository.

Remote name:

Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

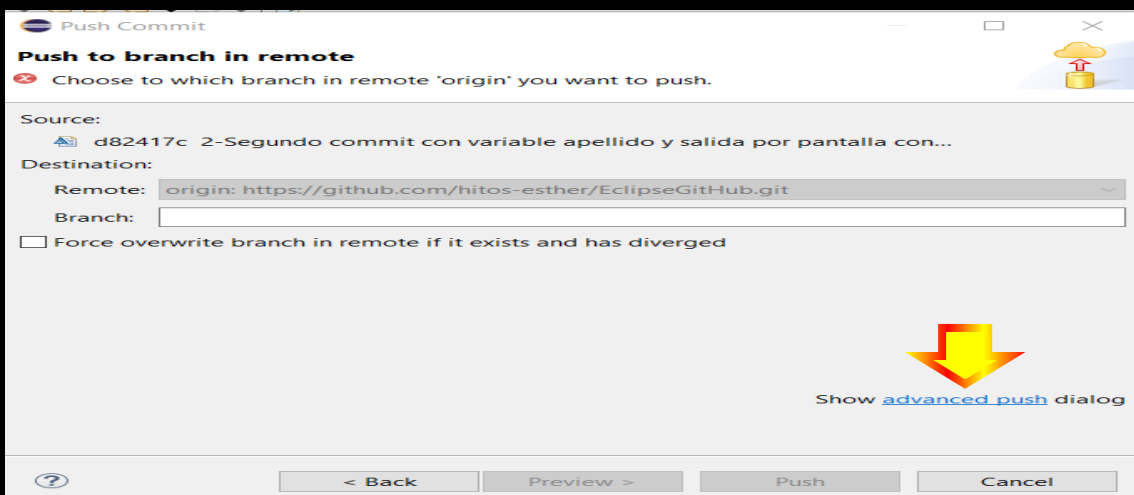
Authentication

User:

Password:

☒ Store in Secure Store

- ◆ En el siguiente paso agregaremos todas las ramas para que no haya conflicto en remoto a la hora de subirlo al repositorio.



Push Commit

Push to branch in remote

Choose to which branch in remote 'origin' you want to push.

Source:

Destination:

Remote:

Branch:

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

Push to Another Repository

Destination Git Repository

Enter the location of the destination repository.

Location

URI: Local Folder...

Host:

Repository path:

Connection

Protocol:


Port:

Authentication

User:

Password:

☒ Store in Secure Store



Push to: https://github.com/hitos-esther/EclipseGitHub.git

Push Ref Specifications


Select refs to push.

Specifications for push

Source ref: Destination ref:

Remote ref to delete:

Mode	Source Ref	Destination Ref	Force Update	Remove



Push to: https://github.com/hitos-esther/EclipseGitHub.git

Push Ref Specifications


Select refs to push.

Specifications for push

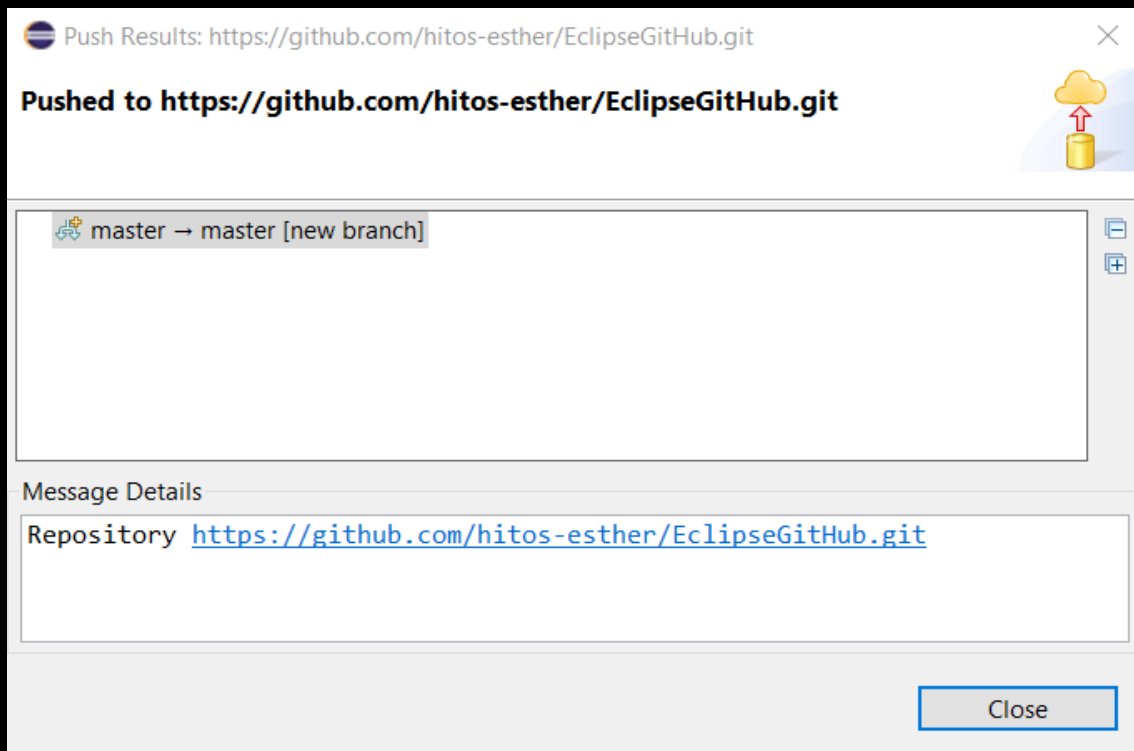
Source ref: Destination ref:

Remote ref to delete:

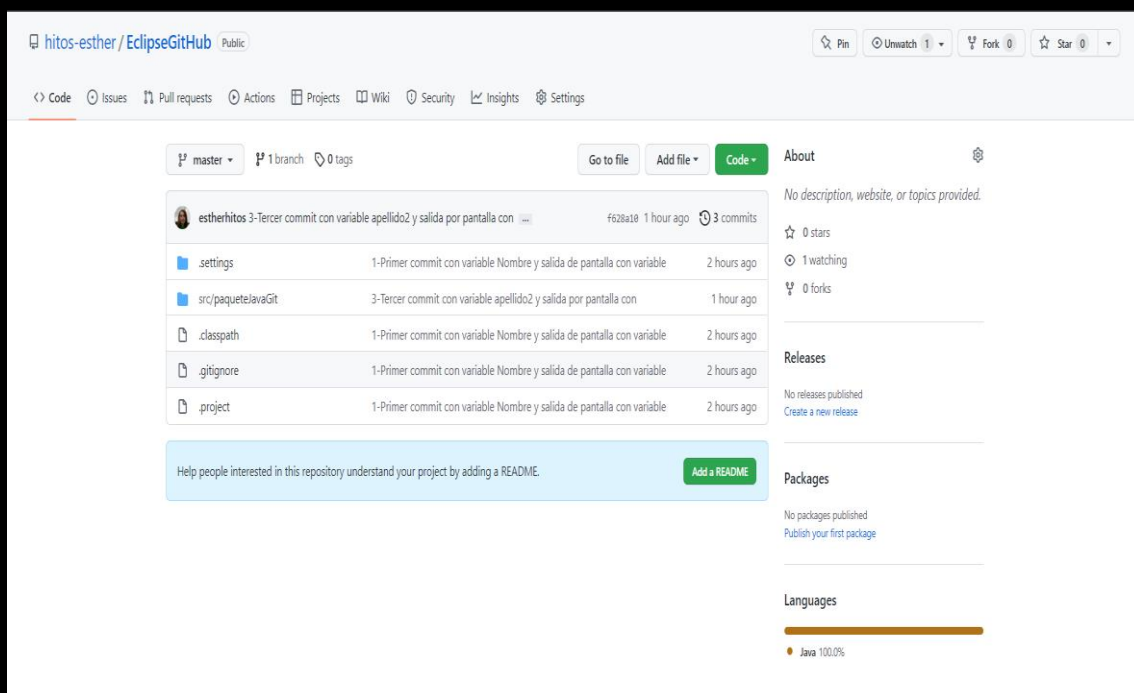
Mode	Source Ref	Destination Ref	Force Update	Remove
Update	refs/heads/*	refs/heads/*	<input type="checkbox"/>	



- ❖ Cuando acabamos todos los pasos te información sobre tu rama master



- ❖ Si ahora vamos a git Hub y vemos nuestro repositorio, comprobamos que esta todo el proyecto subido en nuestro repositorio.

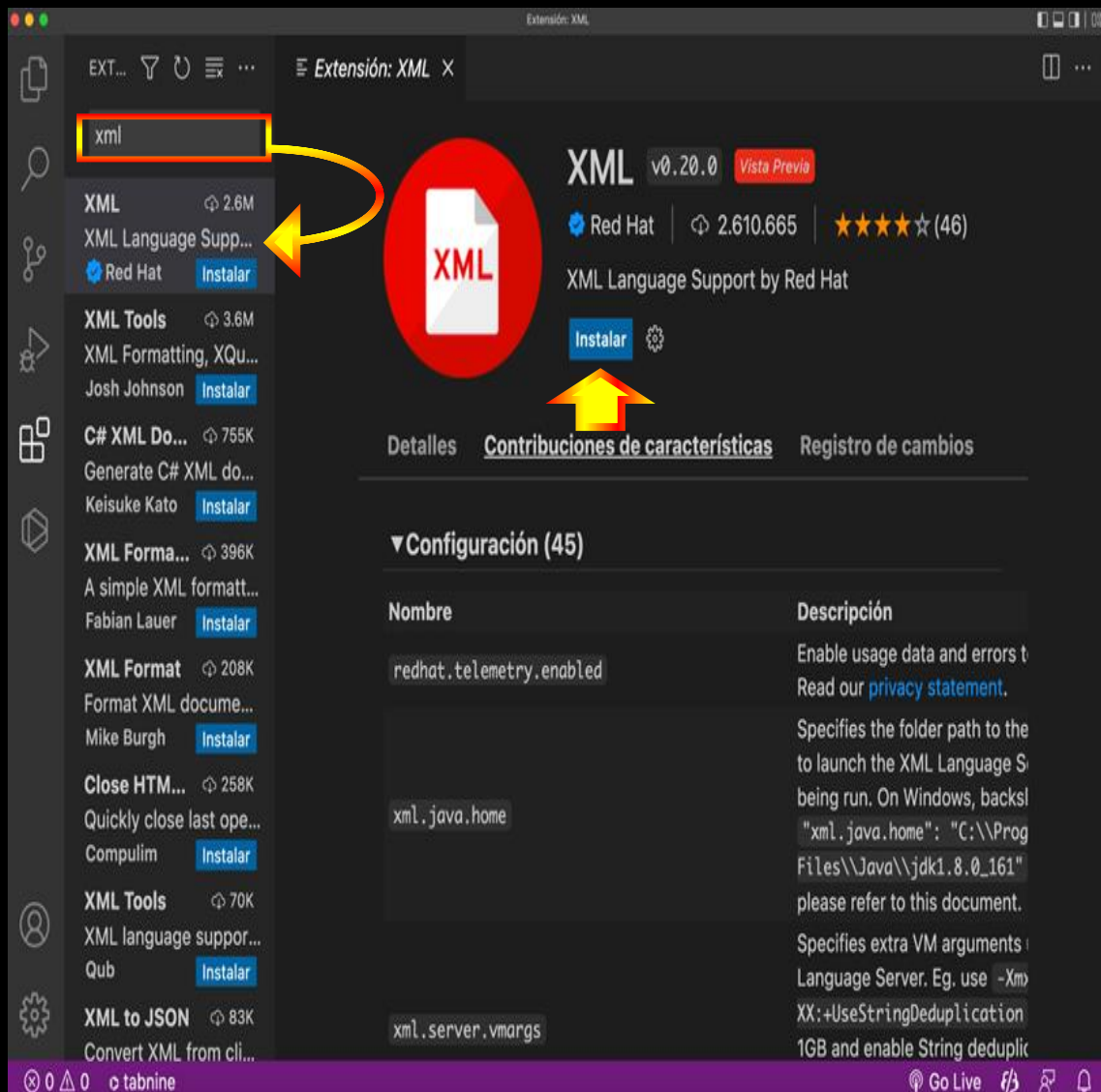


- ❖ Ya hemos terminado y sabemos todos los pasos para trabajar con Eclipse y .Git y en remoto con GitHub.

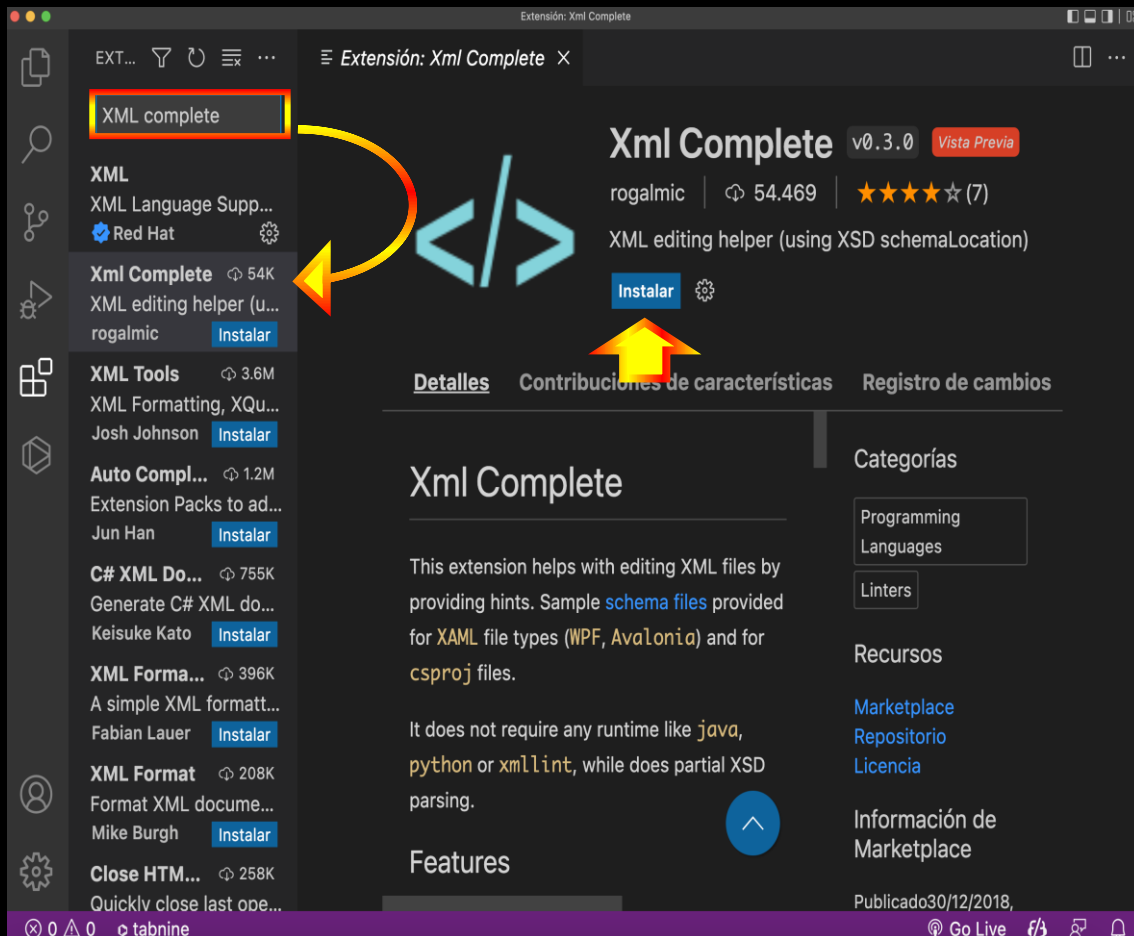
4. VISUAL STUDIO CODE: INSTALACIÓN Y CONFIGURACIÓN DE las extensiones XML Y XML TOOLS de Red Hat

01. VISUAL STUDIO CODE extensiones XML

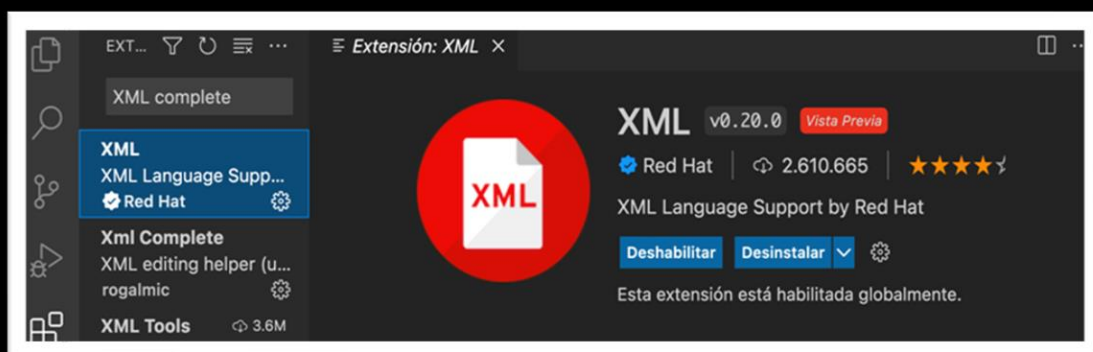
- ✧ A continuación, vamos a ocuparnos del editor de código fuente desarrollado por Microsoft Visual Studio Code.
- ✧ En primer lugar, vamos a proceder a la búsqueda, instalación y configuración de varios Plugins usa la herramienta de búsqueda que trae VSC. Estos Plugins en cuestión, denominados “extensiones” por VSC, son XML y XML Complete.
- ✧ Para ello, primeramente, y dentro de la barra lateral izquierda de VSC, pulsamos sobre el botón Extensiones. A continuación, sobre la barra de búsqueda tecleamos *XML*. En la primera posición nos aparece la extensión XML Language Support by Red Hat. Esta opción es la que procederemos a instalar.



- Una vez instalada, procedemos a hacer lo mismo con la extensión XML Complete: En la barra lateral izquierda, tecleamos XML Complete. Seleccionamos la extensión y la instalamos.

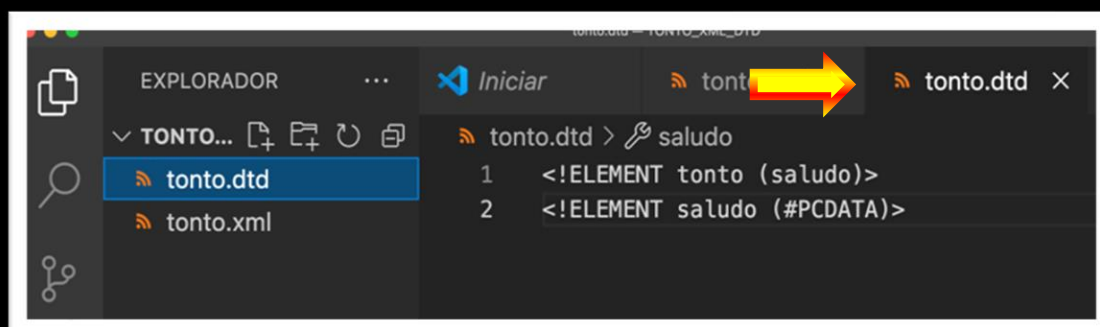
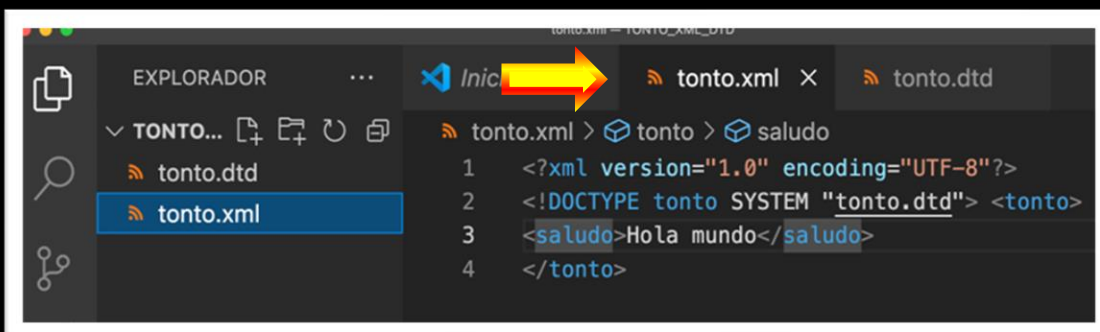


- Comprobamos que ambos plugins han sido instalados correctamente observando tanto debajo de la barra de búsqueda, donde ya no debería aparecer la opción Instalar, como en el menú descriptivo de la extensión, donde debería aparecernos en ambos plugins las opciones Deshabilitar y Desinstalar (indicando ello que ambas extensiones han sido instaladas correctamente).

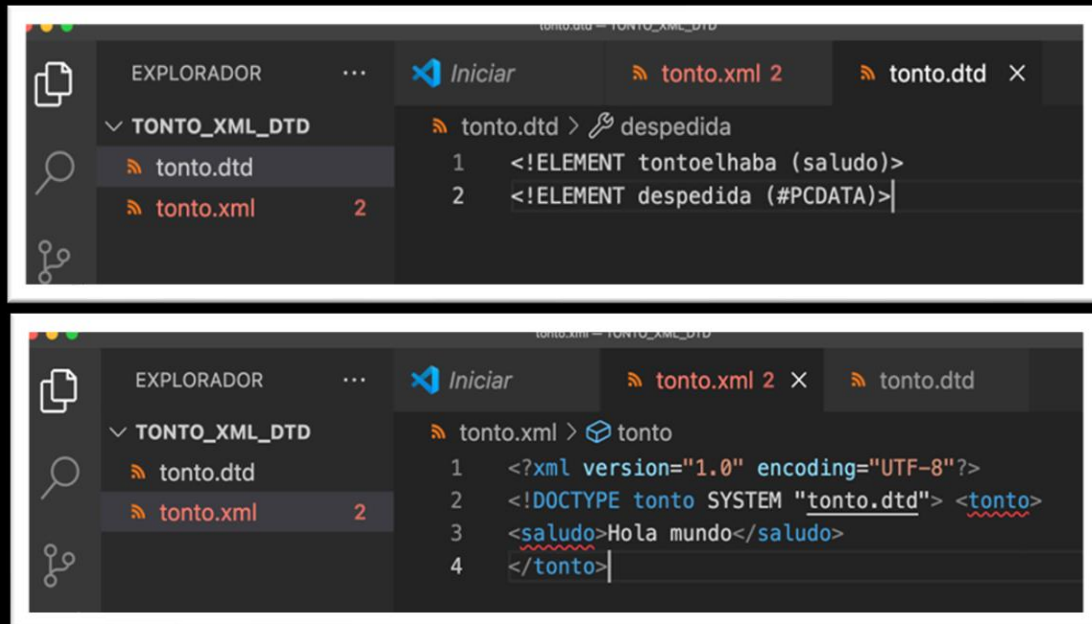




- ✦ Una vez realizados estos pasos previos, nos dirigiremos a probar ambas extensiones usando Visual Studio Code, donde podremos constatar con la conjunción de este Entorno de Desarrollo Integrado más los Plugins que acabamos de instalar podremos trabajar con apoyo y ayuda usando el lenguaje de marcado XML.
- ✦ Para ello, primeramente, creamos una carpeta en nuestro escritorio llamada "TONGO_XML_DTD". Dentro de la misma, valiéndonos del VSC, crearemos los archivos "tonto.xml" y "tonto.dtd".
- ✦ Dentro de cada archivo, deberá encontrarse el siguiente código que usaremos como ejemplo para constatar que la validación es correcta:



- ✎ Por último, y para cerciorarnos de que las extensiones que acabamos de descargar e instalar están realizando la validación de manera correcta, realizaremos algunos cambios en el archivo con extensión .dtd (tonto.dtd).
- ✎ En los cambios que hemos realizado nosotros, cabe señalar que hemos cambiado ,dentro del archivo .dtd, tonto por tontoelhaba (en la primera línea de código) y saludo por despedida (en la segunda línea).



- ✎ Tal y como podemos observar, VSC nos señala que se han detectado dos errores en el archivo tonto.xml tras realizar los dos cambios arriba señalados, marcándonos con subrayado de color rojo exactamente donde tendríamos el fallo. Ello nos indica, consecuentemente, que las extensiones están realizando la validación de manera correcta.