# ¿Qué es Git y que nos aporta?

Git es una herramienta de control de versiones para trabajar en equipo siempre de manera distribuida, un trabajador puede desarrollar sus tareas dentro del proyecto y subirlas a un repositorio, otro trabajador podrá descargar dicho proyecto, realizar sus cambios y volverlo a subir. Cada trabajador puede también trabajar en su propio repositorio local y después sincronizarlo con un repositorio remoto. Se puede trabajar con Git de dos maneras:

* Instalando Git desde la web oficial y después utilizándolo desde el símbolo del sistema (cmd) o el powershell de Windows.
* Desde eclipse a través del plugin EGit.

**EGit** es un plugin que integra Git dentro de Eclipse y nos permite gestionar eficientemente el control de versionado de nuestros proyectos. Nosotros decidimos en que momento registramos una nueva versión del producto de modo que siempre tendremos un historial de versiones.

# ¿Qué es Github y que nos aporta a nosotros?

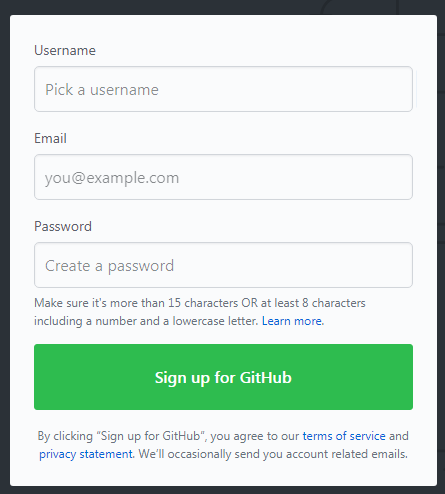
Es un gran repositorio remoto de uso público donde se pueden subir proyectos y compartirlos públicamente.

Cada equipo de trabajo podrá crear un repositorio remoto en Github para su proyecto y después sincronizarlo con los repositorios locales de cada uno de los componentes del grupo.

# Crear cuenta de Github

Para utilizar Github es imprescindible crearse una cuenta:

* Accedemos a la web <https://github.com/>
* Luego rellenamos el formulario de alta de la cuenta y pulsamos el botón “Sign up for GitHub”.

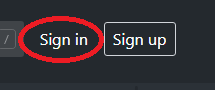


* Después tendrás que escoger el plan gratuito y rellenar alguna información adicional.
* Recibirás un correo a la dirección de email especificada para verificar la cuenta de correo. Verifícala.

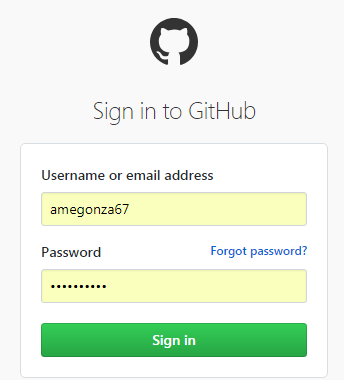
# Iniciar sesión en Github

Ahora que ya tenéis una cuenta iniciar sesión cuando sea necesario.

* Accedemos a la web <https://github.com/>



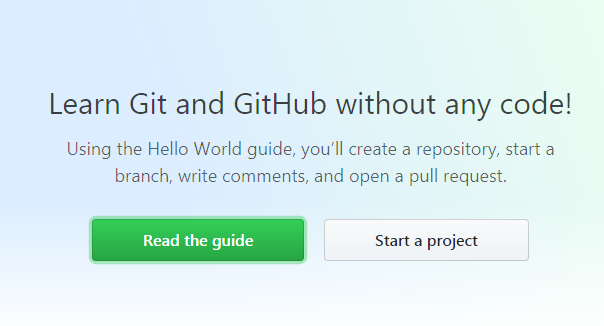
* Introducimos los datos de acceso y pulsamos *Sign in*:



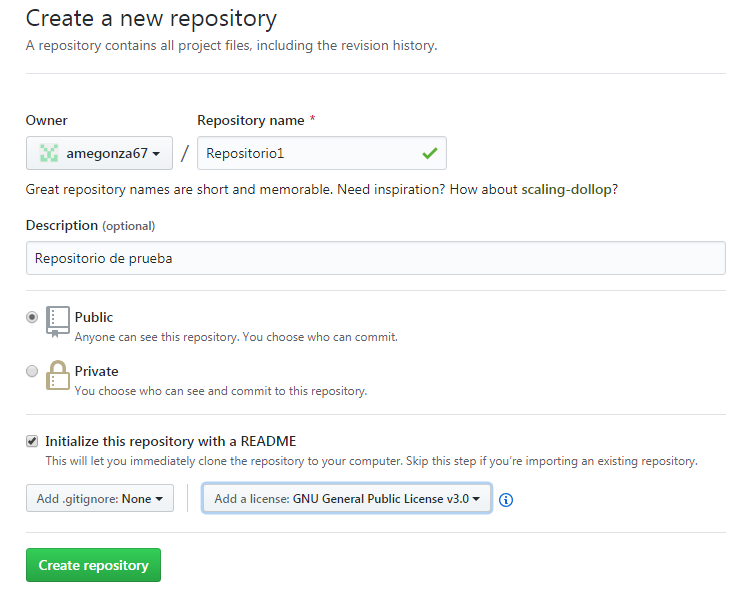
* Introducimos los datos de acceso y pulsamos *Sign in*:

# Crear el primer repositorio

Ahora tendrás la posibilidad de leer una guía (algo interesante si tienes un rato para hacerlo) o comenzar un proyecto.



* Haz clic en el botón “*Start a Project*”.
* Como todavía no tenéis ningún repositorio, os llevará directamente a crear el primero:

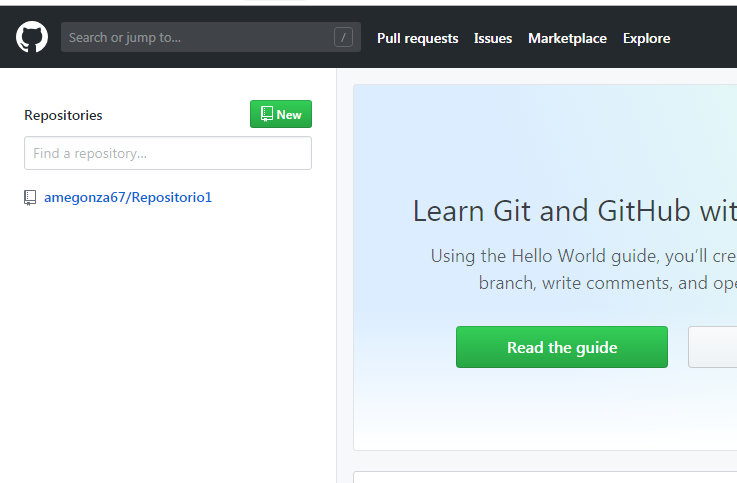


* Completa la información requerida y pulsa “*Create repository*”.

Ya tienes tu repositorio.

# Explorando el repositorio

Las próximas veces que inicies sesión en *Github*, verás a la izquierda los repositorios creados y podrás crear otro nuevo cuando quieras pulsando el botón *New*.

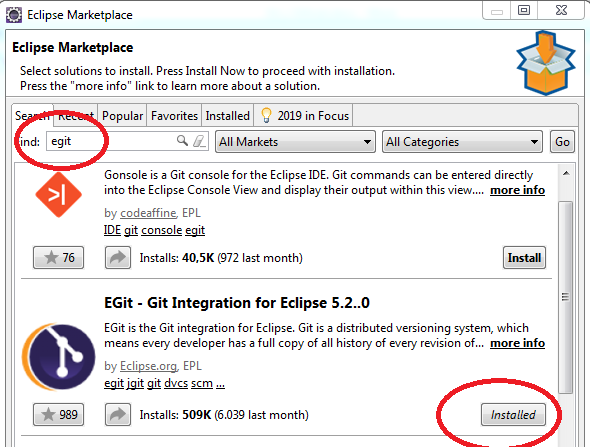


* Haz clic sobre el nuevo repositorio.
* La URL que está mostrando el navegador es importante para después poder configurar desde Eclipse nuestros proyectos para que puedan sincronizarse con el repositorio remoto que acabamos de crear. En mi caso es:

https://github.com/amegonza67/Repositorio1

# Instalar el plugin EGit en Eclipse

Las últimas versiones de Eclipse ya tienen el plugin de EGit instalado. Para comprobarlo podemos seleccionar en el menú “Help / Eclipse markitplace…” y buscar *EGit*.

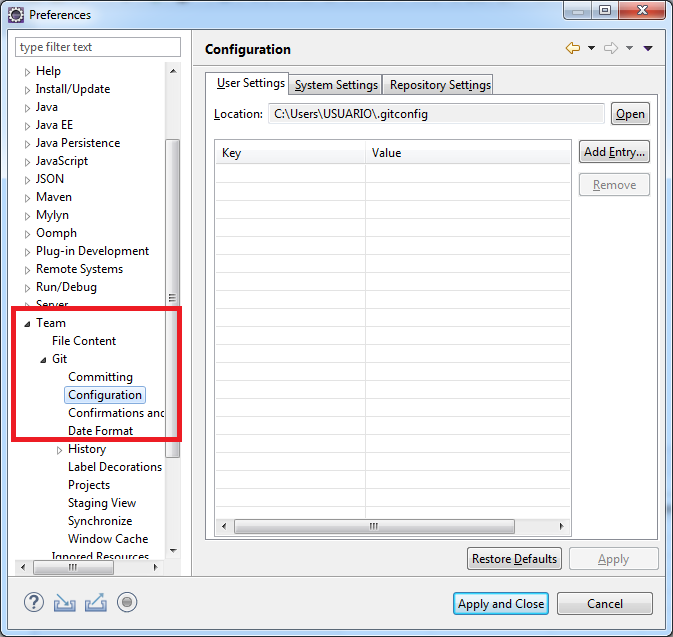


Si nos aparece el botón *Installed* en lugar de *Install* es que ya está instalado.

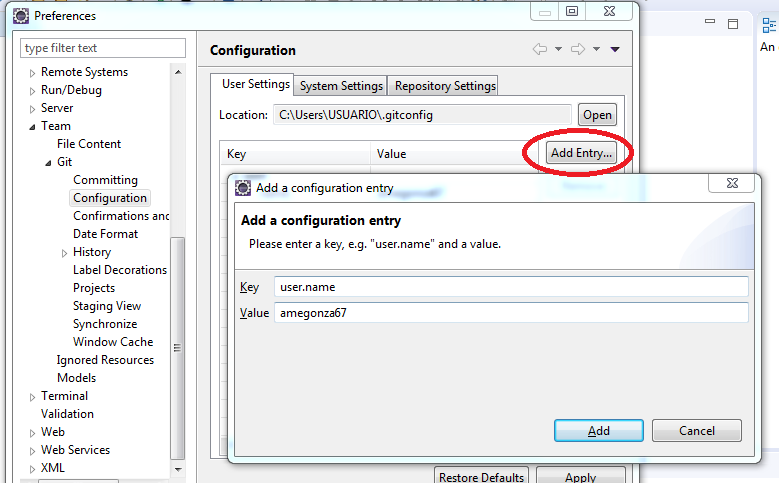
# Añadir datos identificativos para Git

Lo primero que debemos hacer en nuestro Eclipse es configurar el entorno de *Git* para indicar unos datos de identificación.

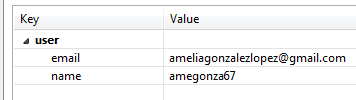
* Accede en los menús a *Windows / Precerences* y en las distintas opciones de la izquierda ve desplegando *Team / Git / Configuration*.



* Accede en los menús a *Windows / Precerences* y en las distintas opciones de la izquierda ve desplegando *Team / Git / Configuration*.
* Ahora debemos crear una serie de variables:

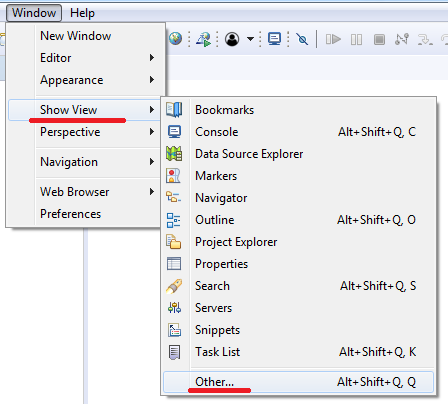


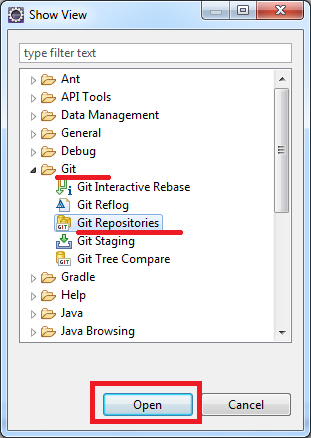
Finalmente dejaremos creados dos datos identificativos:



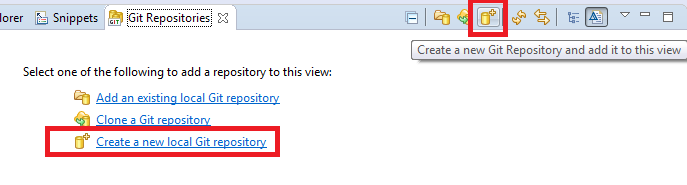
# Crear repositorio local

El repositorio local será el lugar donde guardemos los proyectos en nuestro equipo.

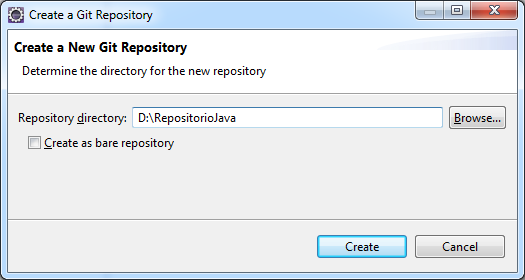




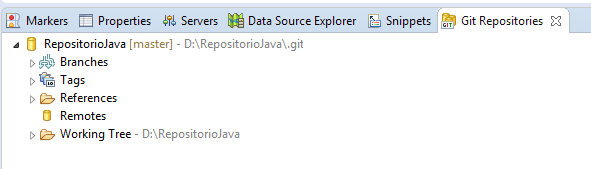
Ahora se ha añadido la pestaña “*Git Repositories*”, debemos crear uno nuevo y podemos hacerlo desde dos sitios distintos:



Ahora debemos escoger la ubicación deseada para el repositorio y pulsar el botón “*Create*”.



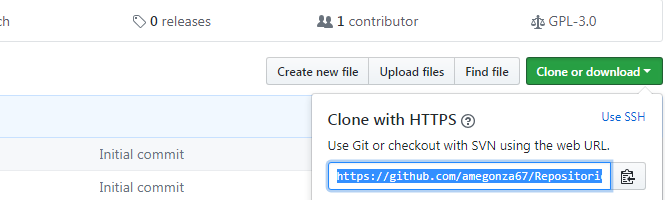
Ahora la pestaña “*Git Repositories*” nos está mostrando el nuevo repositorio.

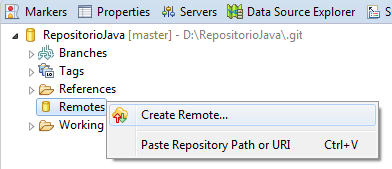


# Vincular nuestro repositorio local con el repositorio remoto

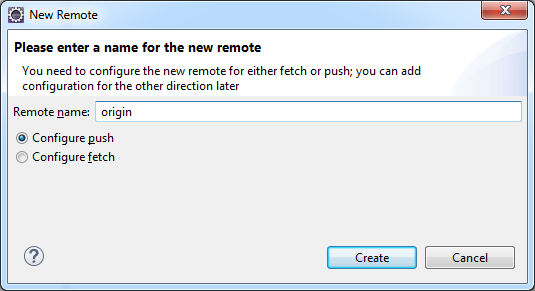
De nuevo nos ayudaremos de la solapa “*Git repositories*” para indicar la información de acceso al repositorio remoto de Github.

Pero antes necesitamos la URL de acceso al repositorio de Github, para obtenerla, de nuevo en Github, accedemos al repositorio, hacemos clic en el botón “*Clone or download*” y copiamos la URL que aparece.

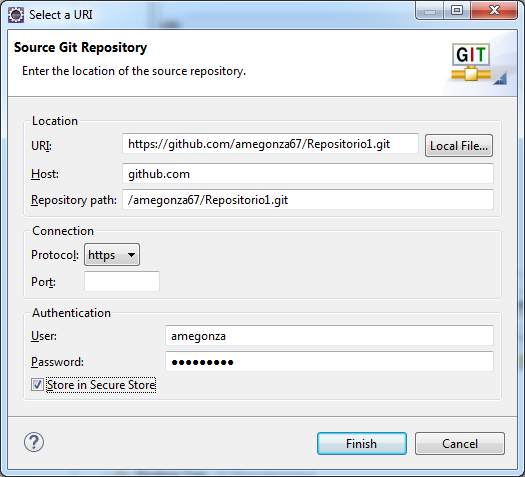
  
Ahora desplegamos la carpeta del repositorio en Eclipse y hacemos clic derecho en la carpeta *Remotes* para seleccionar la opción “*Create Remote ….* “.



Dejamos el nombre que da por defecto al origen remoto del repositorio y hacemos clic en el botón *Create*:



Rellenamos todos los datos de acceso al repositorio:



Y terminamos pulsando el botón ***Finish***. En la última pantalla, pulsamos el botón ***Save***.

# Gestionar nuestros proyectos con Git y Github

Hemos configurado el entorno de Eclipse para trabajar con Git y Github, ahora toca aprender cómo debemos trabajar a partir de ahora para que nuestros proyectos estén correctamente gestionados por Git. Sigue estos pasos:

* Crea un proyecto Java estándar llamado *PruebaGit*.
* Crea una clase sencilla como por ejemplo esta:

**public** **class** Principal {

**public** **static** **void** main(String[] args) {

**int** numero = 5;

**for** (**int** i=1; i<=10; i++) {

System.***out***.println

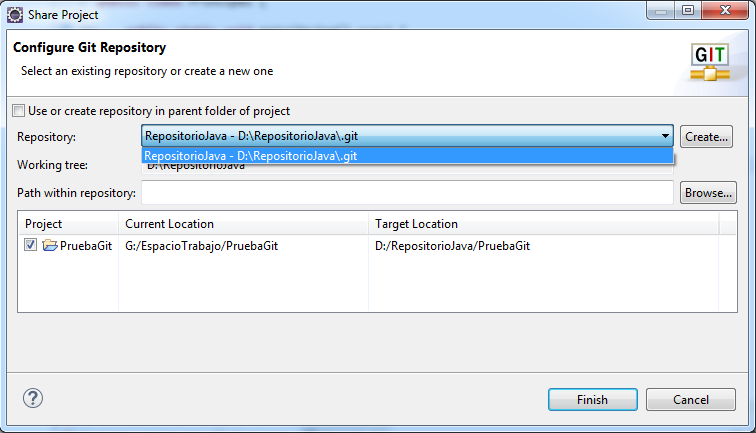
(numero + " x " + i + " = " + (numero\*i));

}

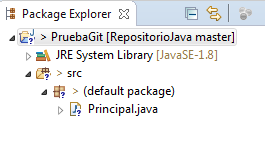
}

}

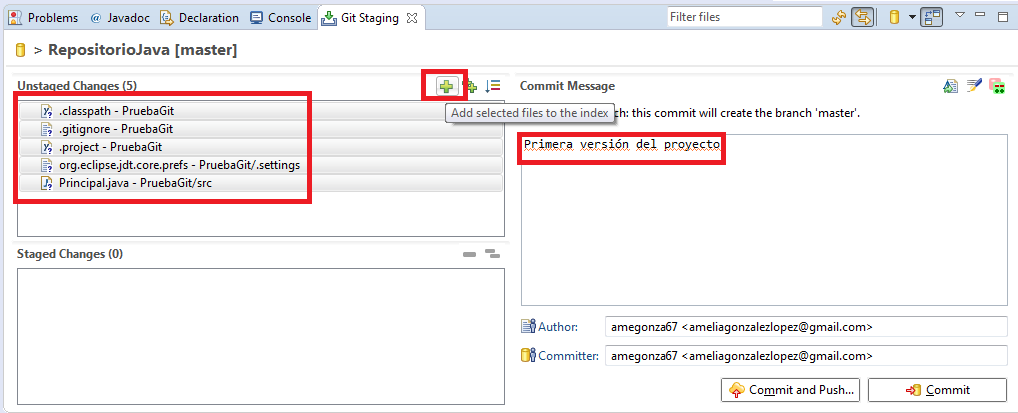
* Para poner el proyecto bajo el control de Git, haz derecho sobre su nombre y selecciona en el menú “*Team / Share Project*”.
* En el cuadro de diálogo “*Share Project*” seleccionamos el repositorio y pulsamos *Finish*.

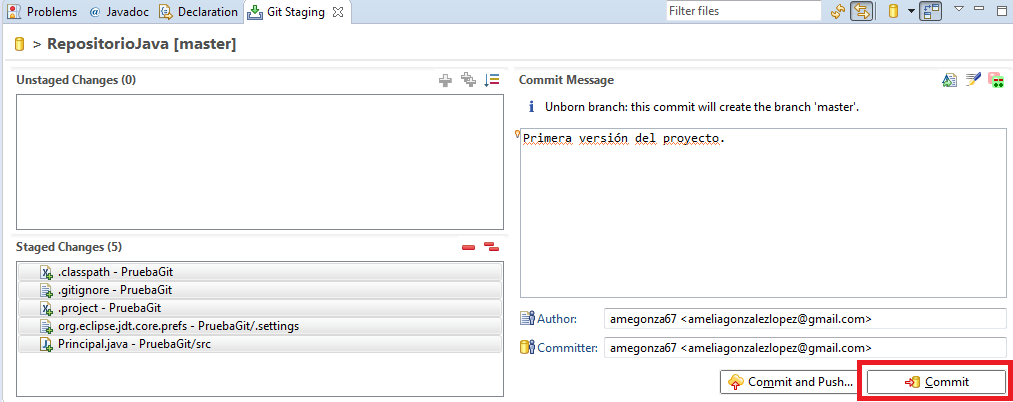


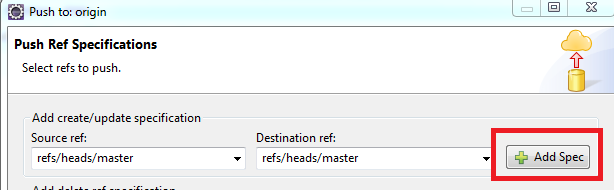
* Si ahora examinas la carpeta del espacio de trabajo de Eclipse, verás que el proyecto se ha movido de allí, ahora está en el repositorio local.
* La interrogación que ahora ves junto a todos los elementos de tu proyecto indica que todavía no se ha creado ninguna versión del proyecto.



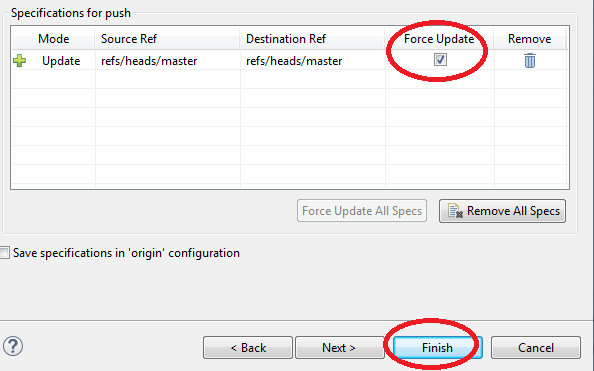
* Para generar la primera versión del proyecto, vamos a ejecutar el primer comando de Git, se trata del comando **COMMIT**. Ejecutar COMMIT nos crea una versión del proyecto. Haz derecho sobre su nombre y selecciona en el menú “*Team / Commit*”.
* Ahora la pestaña “*Git Sharing*” nos permitirá completar la operación. En primer lugar es muy conveniente escribir un comentario que quedará asociado a la versión. Después selecciona todos los archivos y llévalos al área de preparación (Staged Changes) pulsando el botón “*Add selected files …* “.



* Una vez colocados los archivos en el área de preparación, podemos hacer clic en el botón ***Commit*** para generar la primera versión en el repositorio.
* Ahora también queremos subir esta primera versión al servidor remoto de Githab. Para ello pulsamos clic derecho sobre el nombre del proyecto y seleccionamos *Team / Remote / Push*. En la primera pantalla habrá que seleccionar el repositorio remoto, en nuestro caso sólo hay uno y pulsar *Next*.
* En la siguiente pantalla habrá que seleccionar las ramas de origen y destino. Después hay que pulsar el botón “*Add Spec*”.



Después de esa operación, el origen y destino quedarán reflejados en la lista inferior donde habrá que marcar la pestaña “*Force Update*” y terminar pulsando el botón *Finish*.



* Comprueba en tu cuenta de *Github* como el proyecto se ha subido correctamente.

# Generar la segunda versión en el repositorio local

* Modifica la clase principal para poder mostrar la tabla de multiplicar de cualquier número introducido por teclado. Quedaría así:

**import** java.util.Scanner;

**public** **class** Principal {

**public** **static** **void** main(String[] args) {

Scanner lector = **new** Scanner(System.***in***);

System.***out***.println("¿La tabla de que número deseas? ");

**int** numero = lector.nextInt();

lector.nextLine();

lector.close();

**for** (**int** i=1; i<=10; i++) {

System.***out***.println

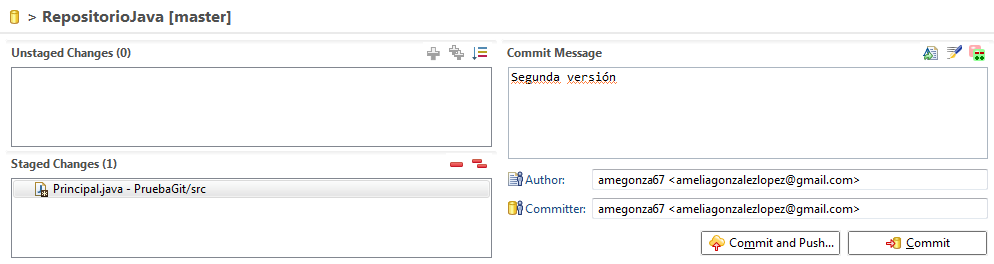
(numero + " x " + i + " = " + (numero\*i));

}

}

}

* Vuelve a realizar un COMMIT para crear una segunda versión. Igual que antes, lleva el único archivo modificado al área de preparación, escribe el comentario y pulsa el botón *Commit*.



# Mostrar historial

Clic derecho en el nombre del proyecto / Team / Show in history.

# Recuperar versiones antiguas de un archivo

Clic derecho sobre el archivo / Compare to / Commit.

# Importar proyecto Github

File / Import / Clone

# Actualizar proyecto a la última versión del repositorio remoto

Clic derecho / Team / Pull

# Videos sobre Git y Github (para trabajar desde Eclipse):

<https://www.youtube.com/watch?v=XMq6L_OIjzg>

<https://www.youtube.com/watch?v=VYHX5twLExs&t=617s>

<https://www.youtube.com/watch?v=xnkC3KqYdV8&t=31s>