TRABAJO E.D: CONTROL DE VERSIONES

GIT

REALIZADO POR:

ANTONIO MARTINEZ NAVARRO

CONTENIDO

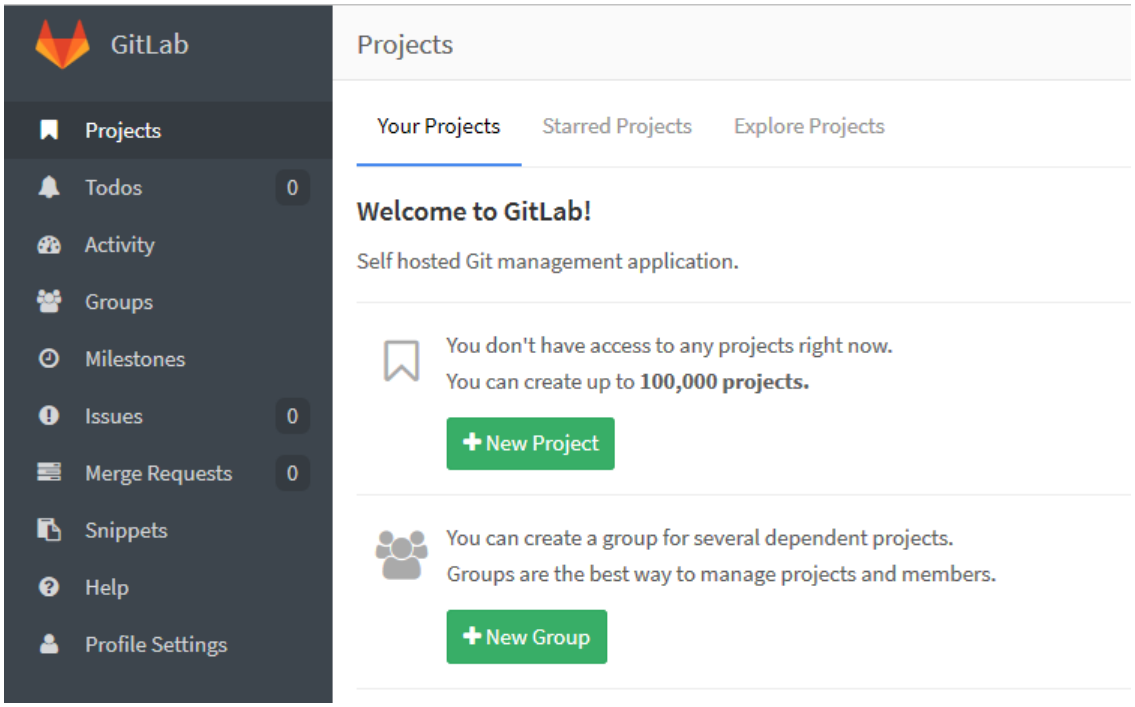
1. INTRODUCCIÓN

Esto es una demostración de cómo usar un software de Control de Versiones como GIT. Este software es muy útil a la hora de trabajar conjuntamente en un mismo proyecto.

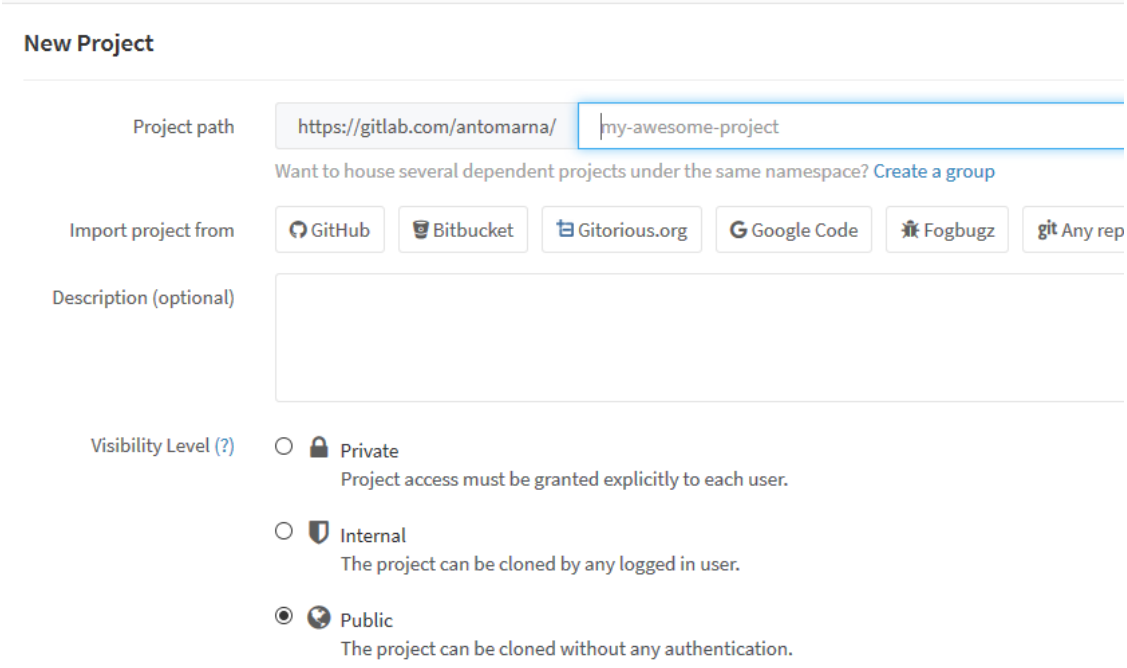
Vamos a ver unos pasos para entender mejor cómo funciona GIT.

2. CREAR UN REPOSITORIO REMOTO EN GITLAB.

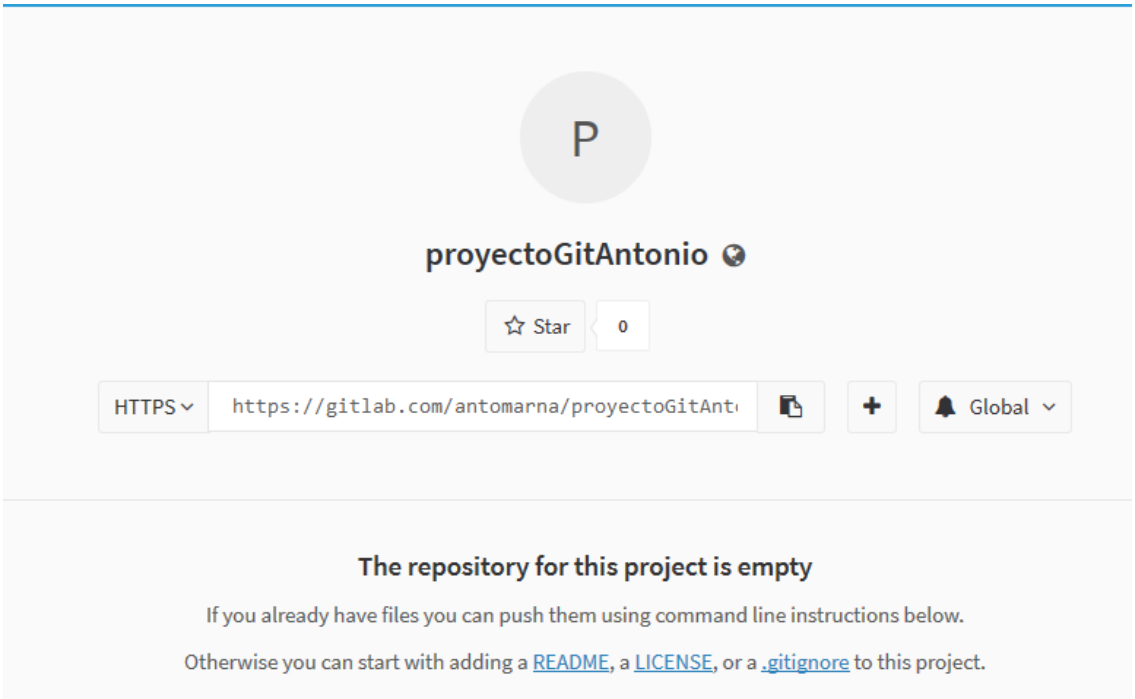
Lo primero que tenemos que hacer es entrar en la página de GITLAB y darse de alta para después crear un repositorio en el cual vamos a trabajar.



Pinchamos en New Proyect.



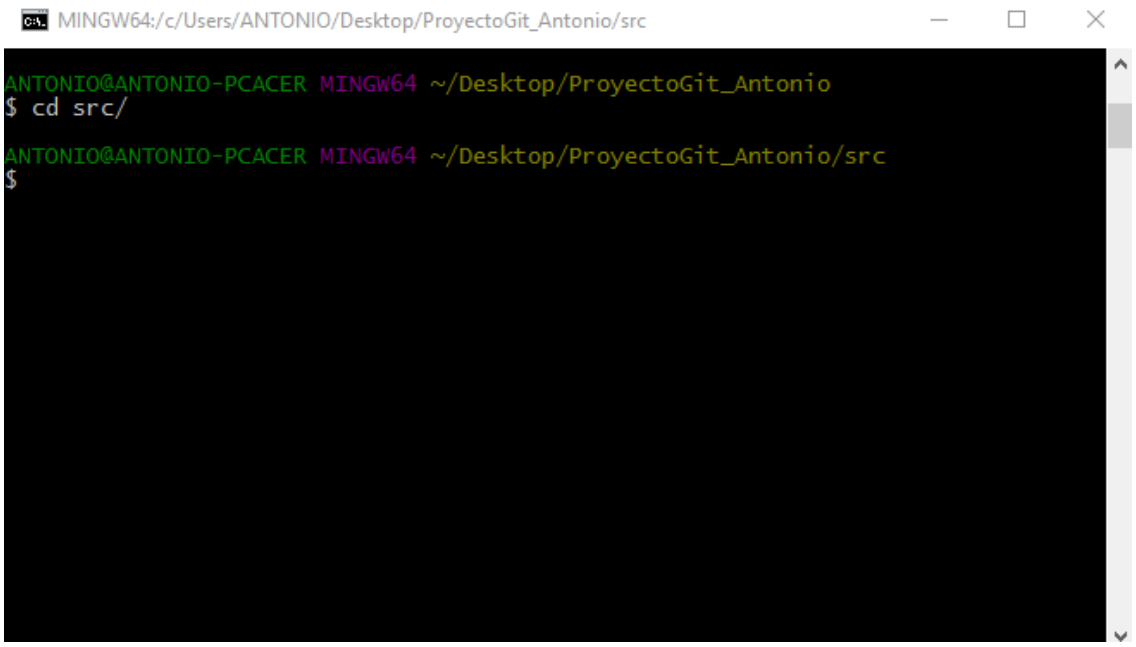
En el cuadro de texto ponemos el nombre a nuestro repositorio. Pulsamos en **Create Proyect** y nos aparece esta pantalla:



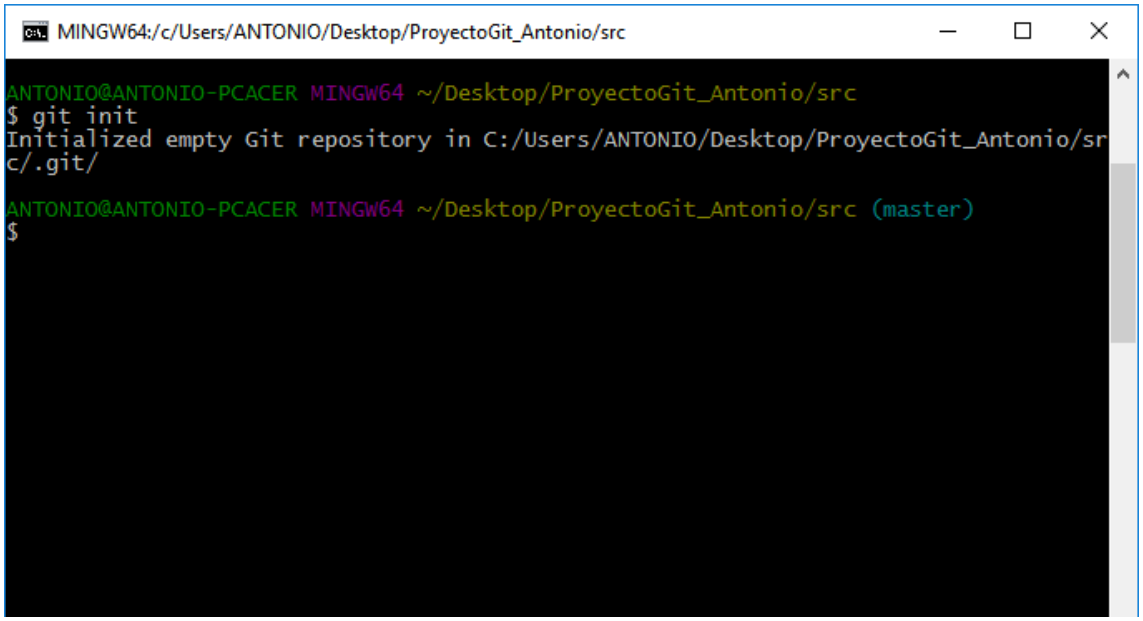
Ya tenemos el repositorio de trabajo creado y la URL que utilizaremos desde GIT Bash para subir cambios.

3. EMPEZAR A USAR GIT BASH DENTRO DE NUESTRO PROYECTO.

Abrimos GIT Bash y nos vamos dentro del directorio **src** del proyecto.

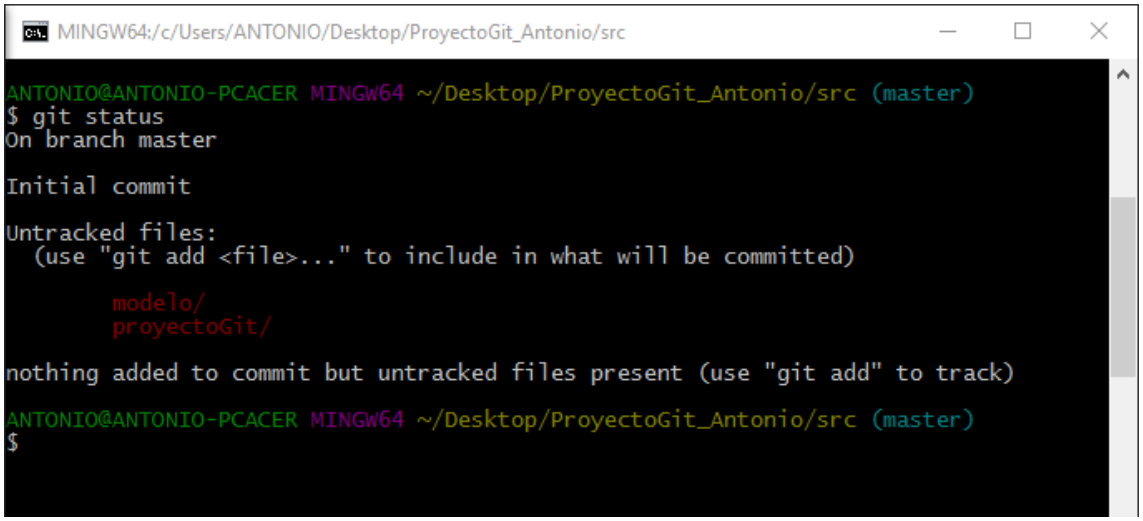


Ahora ejecutamos el comando **git init.**



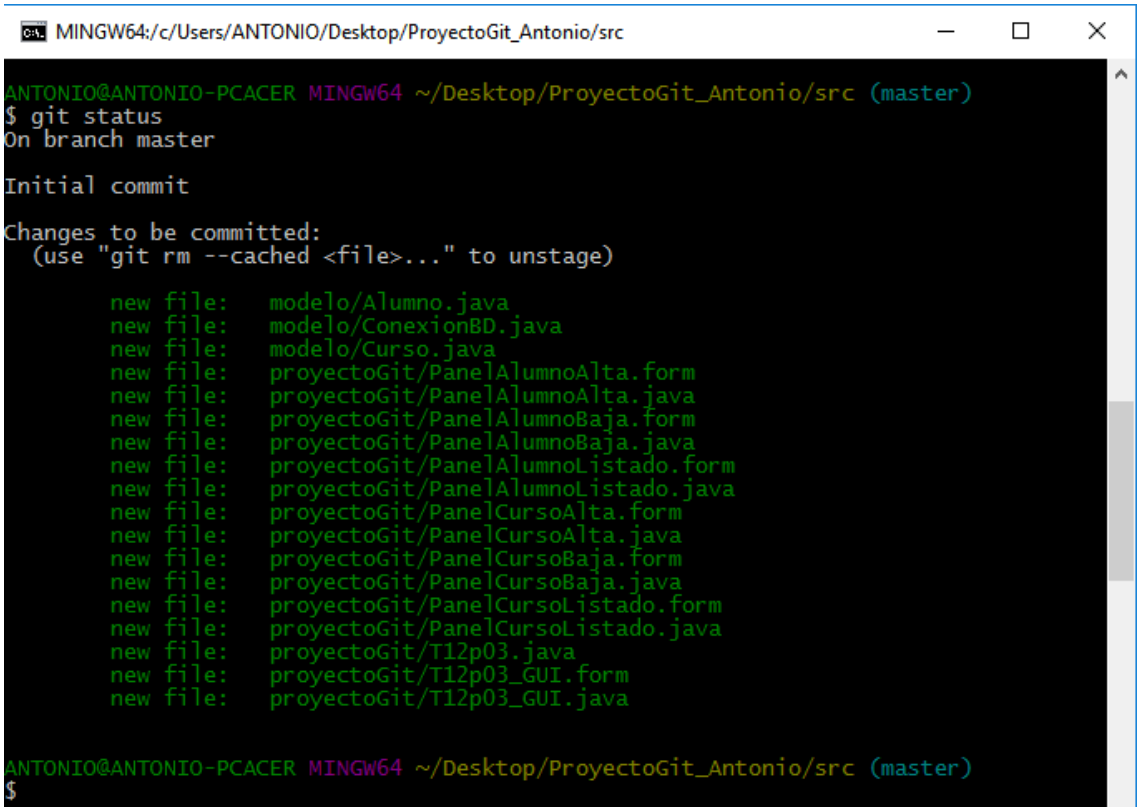
Automáticamente se crea la rama **master.**

Ahora podemos ejecutar **git status** para ver el estado de nuestro proyecto.



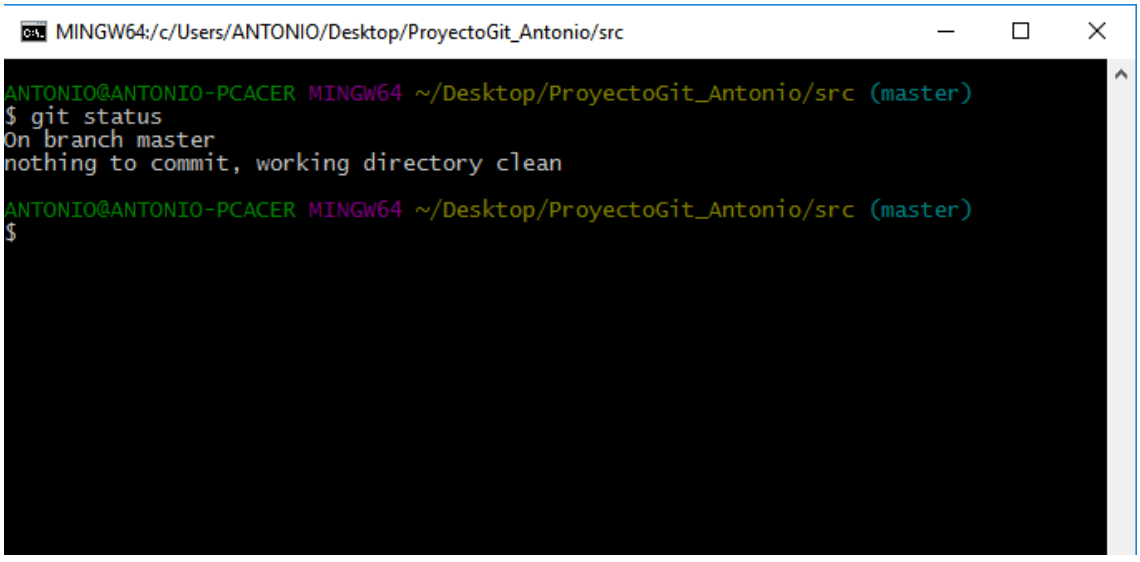
Como podemos ver hay dos directorios en rojo, eso significa que no se han añadido a la rama en la que estamos actualmente (master). Para añadirlos ejecutamos **git add .**

Podemos comprobar el estado actual con **git status.**



Ahora se han añadido todos esos ficheros. Lo que debemos hacer a continuación es confirmar los cambios. Para ello ejecutamos **git commit -m "*mensaje*".**

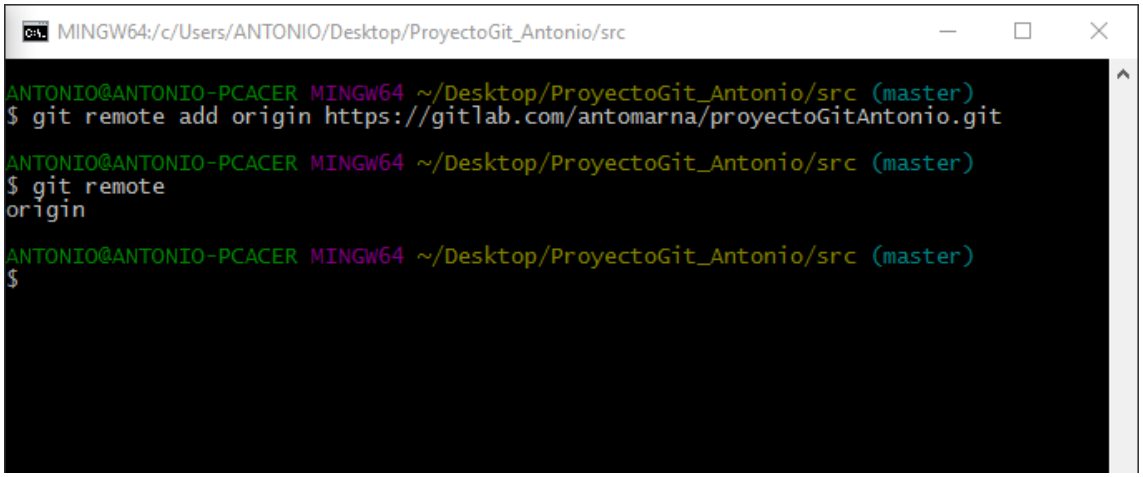
Comprobamos el estado y nos debe salir lo siguiente:



4. SUBIR NUESTRO PROYECTO ACTUAL AL REPOSITORIO REMOTO.

Lo primero que tenemos que hacer es decirle la ruta en la que se encuentra el repositorio remoto. Para esto ejecutamos:

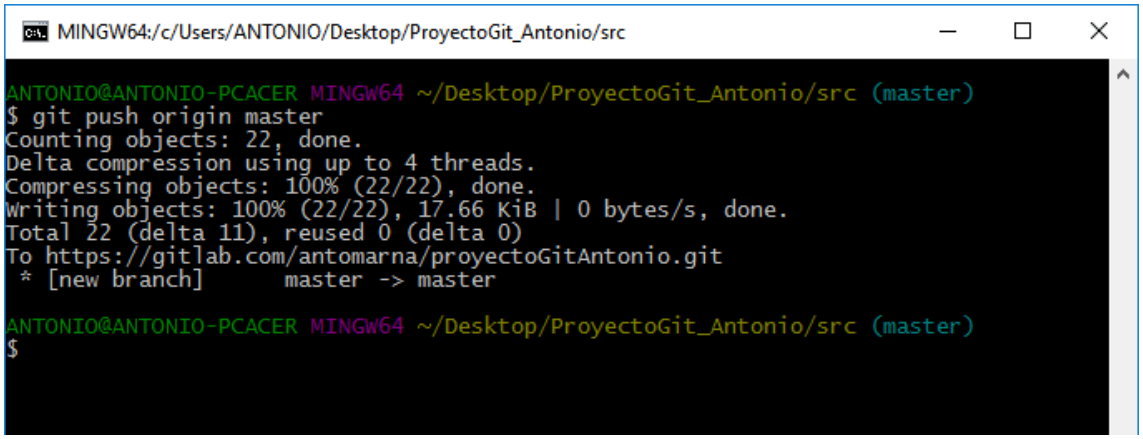
git remote add <nombre> <url>



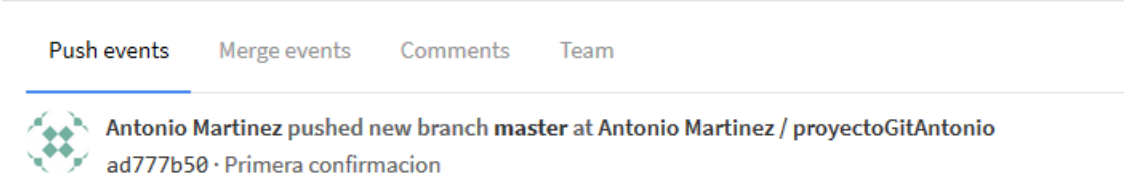
Yo he utilizado de nombre **origin.** A partir de ahora utilizaré este nombre en vez de la url.

Ahora ya podemos subir el proyecto a nuestro repositorio remoto ejecutando el siguiente comando:

git push <nombre> <nombre-rama>



Si entramos en la página de gitlab podemos ver lo que acabamos de subir.



5. CREAR UNA RAMA NUEVA PARA HACER MODIFICACIONES.

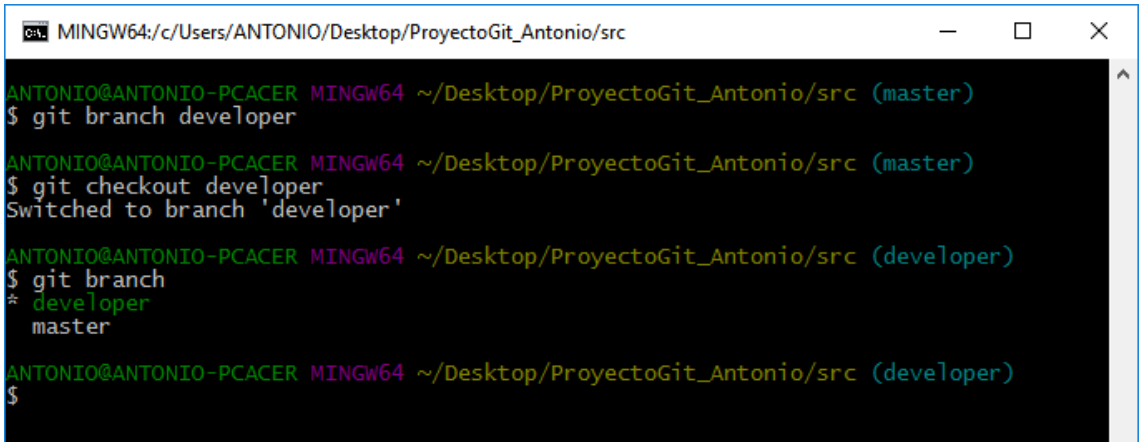
Si queremos hacer modificaciones a nuestro proyecto sin afectar al proyecto original podemos crear una nueva rama de trabajo, y al final, si queremos, fusionar el trabajo modificado al original (master).

Para crear una nueva rama tenemos que ejecutar el comando:

git branch <nombre-rama>

Para posicionarnos en esa rama ejecutamos el comando:

git checkout <nombre-rama>



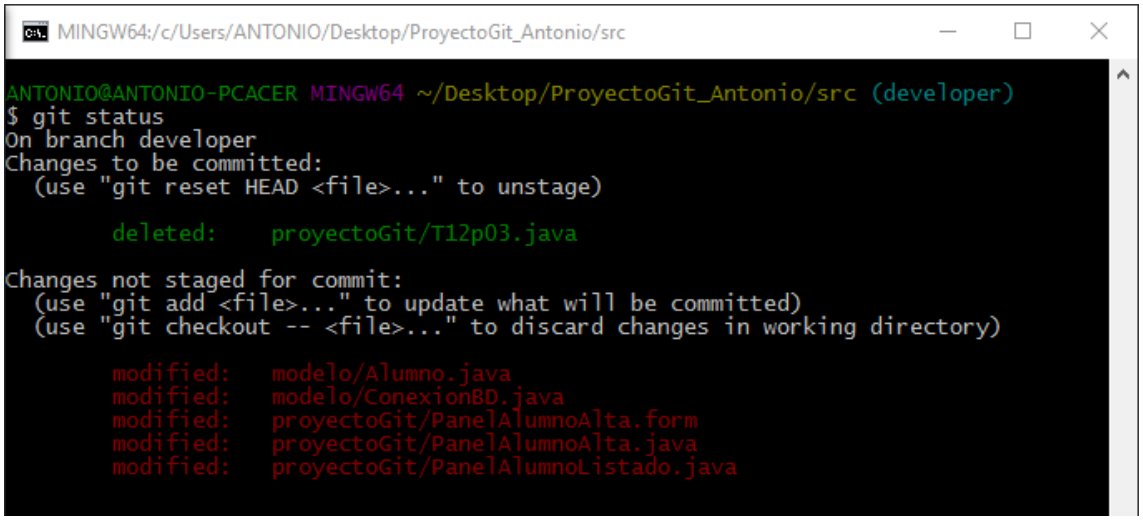
Para ver las todas las ramas ejecutamos **git branch.** El asterisco nos indica en que rama nos encontramos actualmente.

Ahora ya podemos empezar a hacer nuestras modificaciones en el proyecto desde nuestro entorno de desarrollo.

6. MODIFICACIÓN DEL PROYECTO ORIGINAL.

En el proyecto utilizado para esta práctica quiero añadir un nuevo atributo a los alumnos, que será si es repetidor o no. Para ello añado un check-box en el JPanel del alta de alumno y realizo las modificaciones necesarias en las clases involucradas.

Una vez realizados los cambios, volvemos al bash y ejecutamos **git status** para ver el estado del proyecto en la rama actual.



Nos muestra todos los ficheros que han sido modificados y están por confirmar.

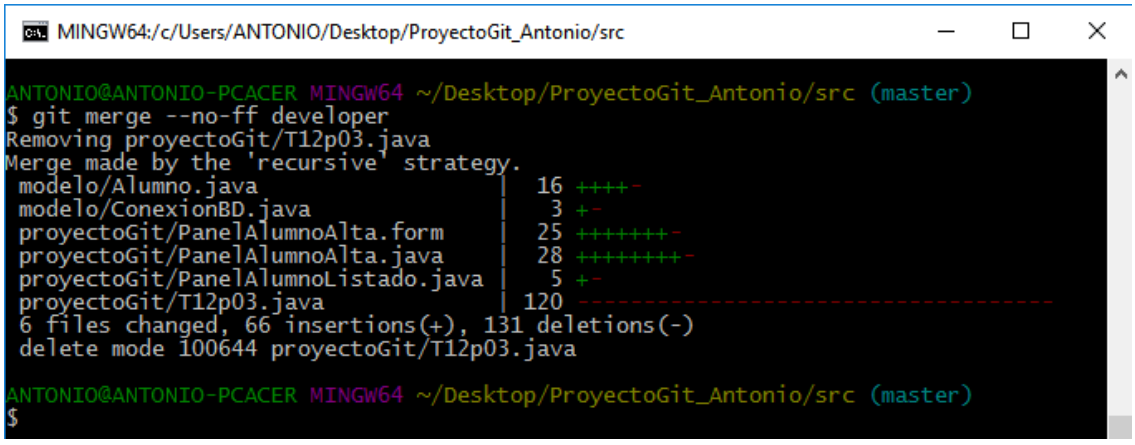
Lo que tenemos que hacer a continuación es añadir los cambios con el comando **git add .** y confirmar los cambios con **git commit -m "*mensaje*".**

Ahora lo que podemos hacer es, o bien subir el proyecto desde la rama actual (developer), o fusionar esta rama y subir el original (master).

Para fusionar la rama ***developer*** con la rama ***master,*** nos situamos en la rama master (**git checkout master**) y ejecutamos el siguiente comando:

git merge --no-ff <nombre de la rama a fusionar>

En mi caso el nombre de la rama a fusionar es **developer.**



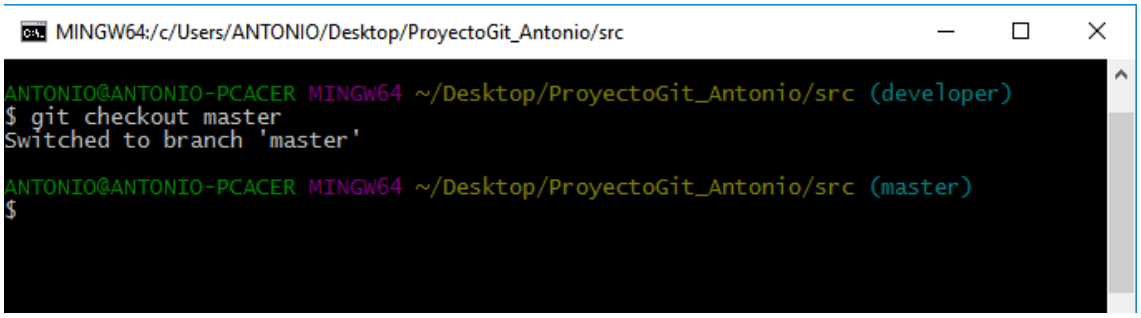
Para subir el proyecto actual ejecutamos **git push origin master**

Si queremos ver el registro de nuestra actividad ejecutamos el comando **git log.**

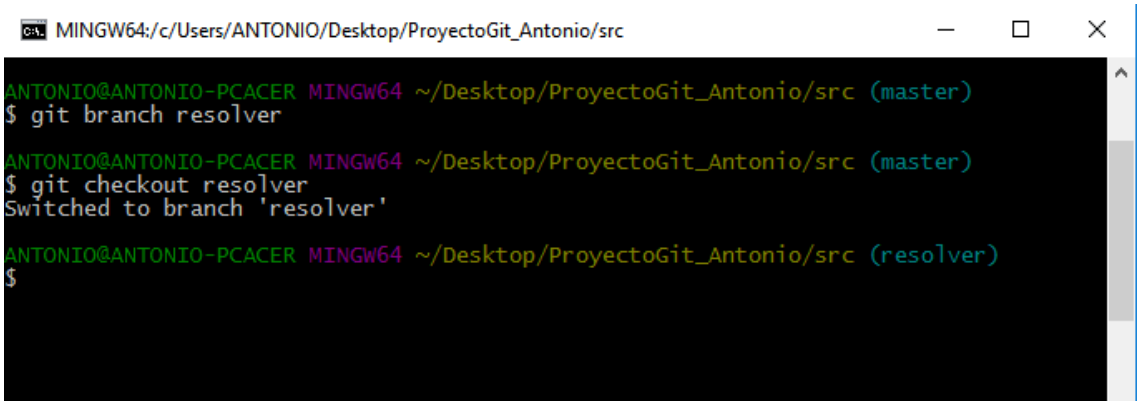
7. POSIBLE CASO EN LA VIDA REAL

Supongamos que estamos trabajando en una rama para hacer algún cambio en nuestro proyecto original y nos llaman para decirnos que tenemos que arreglar un problema en el original.

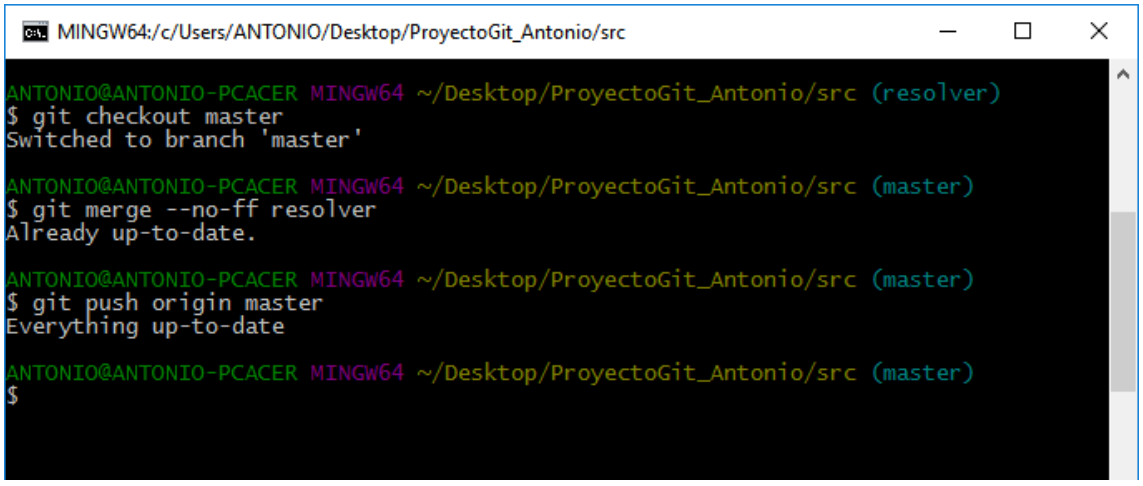
Lo primero que debemos hacer el volver a la rama de producción original (master).



Lo siguiente es crear una nueva rama para solucionar el problema.



Hacer todos los cambios para resolver el problema y tras esto fusionar esta rama con el comando **git merge** y subirlo a la rama de producción.



Una vez hecho esto podemos volver a nuestra rama de desarrollo y continuar con nuestro trabajo.

8. BIBLIOGRAFIA.

- Apuntes de Cubichilazaro:
  http://www.piradoiv.com/blog/aprende-usar-git-con-este-ejemplo-real
  http://www.piradoiv.com/blog/usando-git-con-tus-companeros-de-proyecto

- Documentación de la página oficial de GIT
  https://git-scm.com/book/es/v1/Fundamentos-de-Git

9. ENLACE AL PROYECTO

- https://gitlab.com/antomarna/proyectoGitAntonio.git