**Actividad Evaluable**

|  |  |
| --- | --- |
| Módulo | Servicios de red |
| Nombre y Apellidos: | Diego Cota |
| Nombre y Apellidos: | Eva Pulla |
| Nombre y Apellidos: |  |
| Estas soluciones deben ser entregadas en PDF. | |

## FECHA DE ENTREGA: --/--/21

Objetivos

El objetivo de esta actividad individual es la de repasar, asentar y adquirir un mayor conocimiento de lo impartido en clase.

El archivo debe nombrarse como: **nombre\_apellido1\_apellido2.PDF**

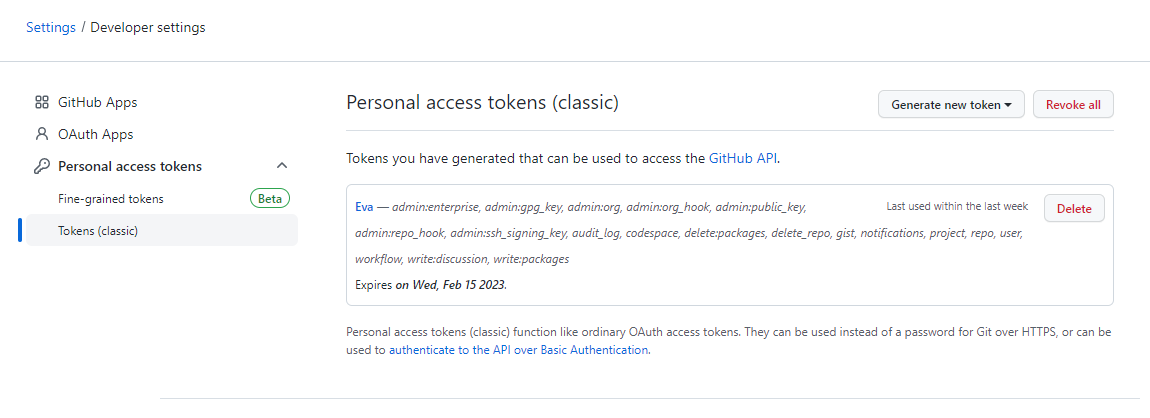
Ej: Belen\_Vargas\_Sabater.pdf

Cuestiones

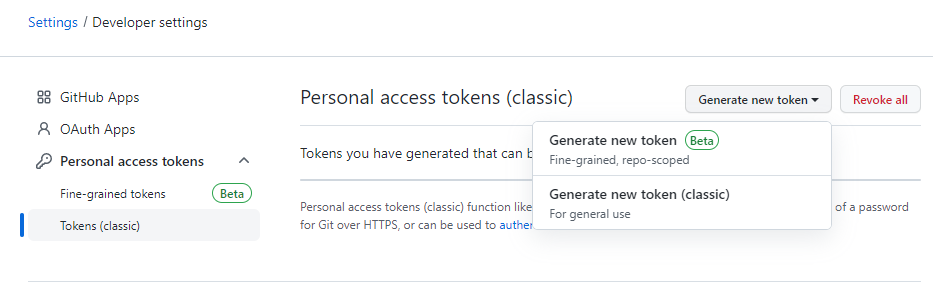
EJERCICIO 1: **Práctica**: Realizar una guía práctica a modo de tutorial, sobre el uso de GitHub, algunos de los elementos que debe cubrir son:

* Creación de repositorio y conexión con repositorio personal de GitHub mediante terminal con ejemplos
* Push Y pulls a los repositorios desde el terminal con ejemplos.
* Cada miembro del equipo debe realizar cambios en el repositorio propiedad de uno de los miembros, documenta el proceso.

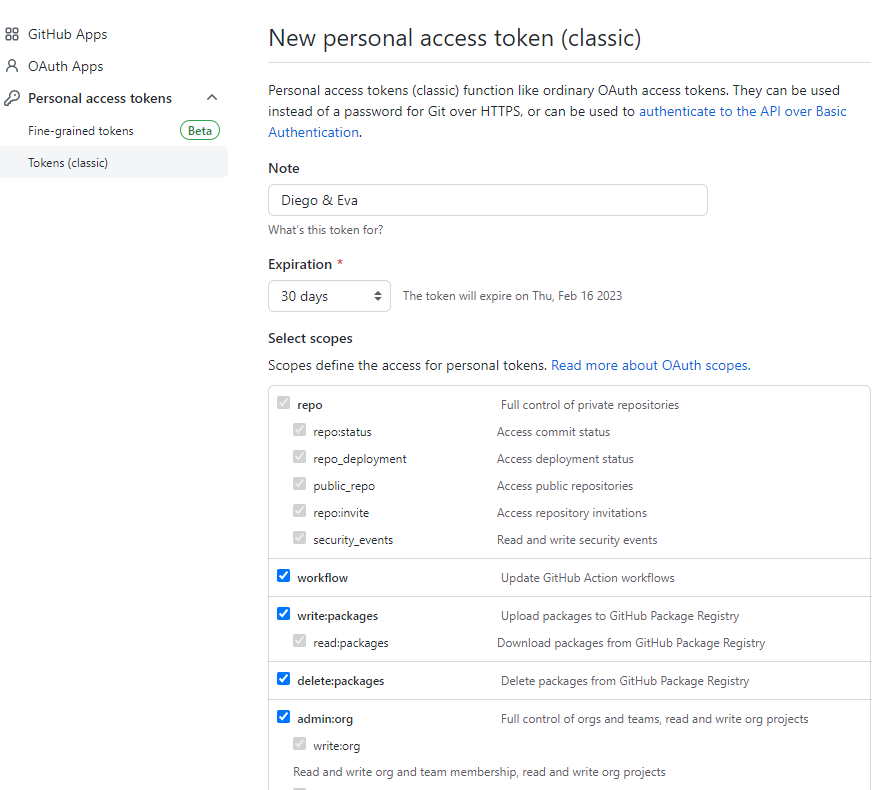
**1º** Entramos en GitHub con nuestra cuenta, nos vamos a settings y abajo del todo encontraremos la opción Developer settings y entramos en esa opción. Una vez dentro de Developer settings seleccionamos Tokens (classic). Lo que hacemos en tokens es eliminar los que haya generados dando en Delete que aparece en color rojo.



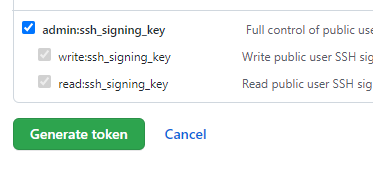
**2º** Ahora una vez eliminados los tokens, generamos un nuevo token pinchando sobre Generate new token y escogemos la opción Generate new token (classic) ya que es mas sencilla y requiere menos datos.



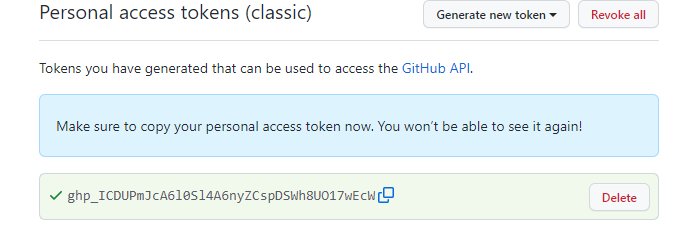
**3º** Añadimos una nota para identificar el token y marcamos todas las casillas para dar acceso a todo lo relacionado con el repositorio.



**4º** finalmente cuando ya hemos marcado todas las casillas damos a Generate token para que se nos genere.

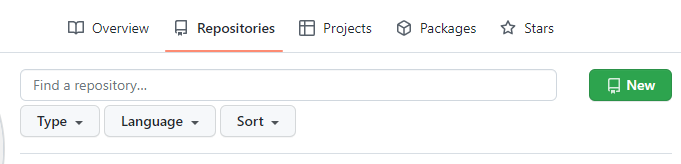


**5º** Se muestra que ya hemos generado el token y guardamos muy bien el token (se recomienda copiarlo y pegarlo en un archivo de texto) ya que no volverá a mostrarlo.

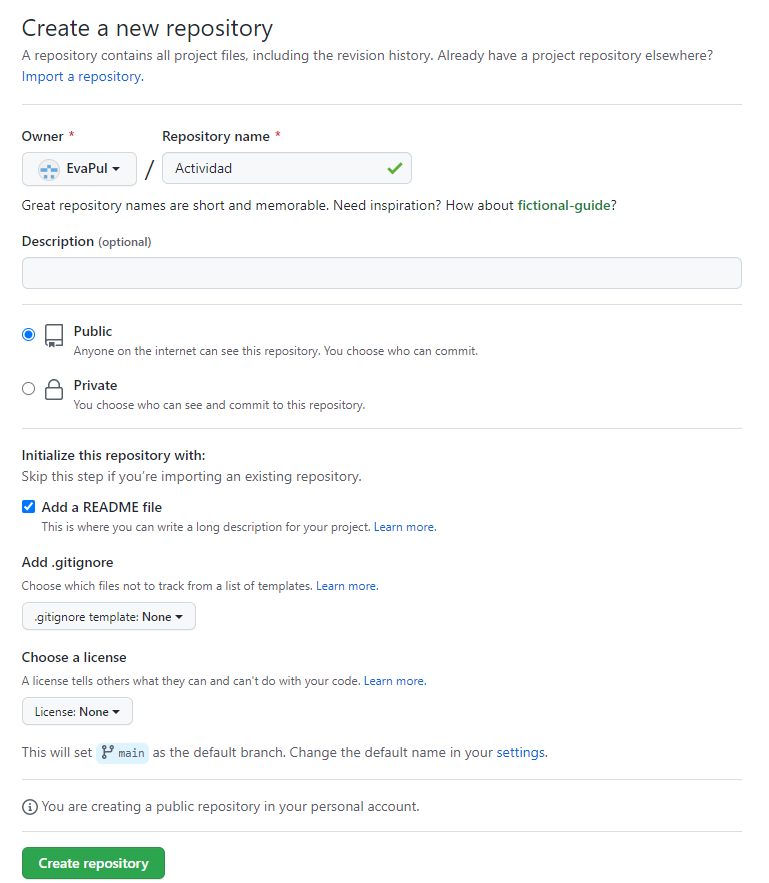


ghp\_ICDUPmJcA6l0Sl4A6nyZCspDSWh8UO17wEcW

**6º** Una vez que tenemos el token vamos con la creación de repositorio. Para crear el repositorio nos vamos a repositories y damos a New.



**7º** en el próximo paso vamos a darle nombre, visibilidad y otros detalles al repositorio. Cuando ya tenemos todo damos a créate repository.



**8º** Abrimos el terminal con Ctrl+Alt+T y empezamos con los comandos. Primero entramos en la carpeta anteriormente creada llamada prueba donde hemos ido trabajando.

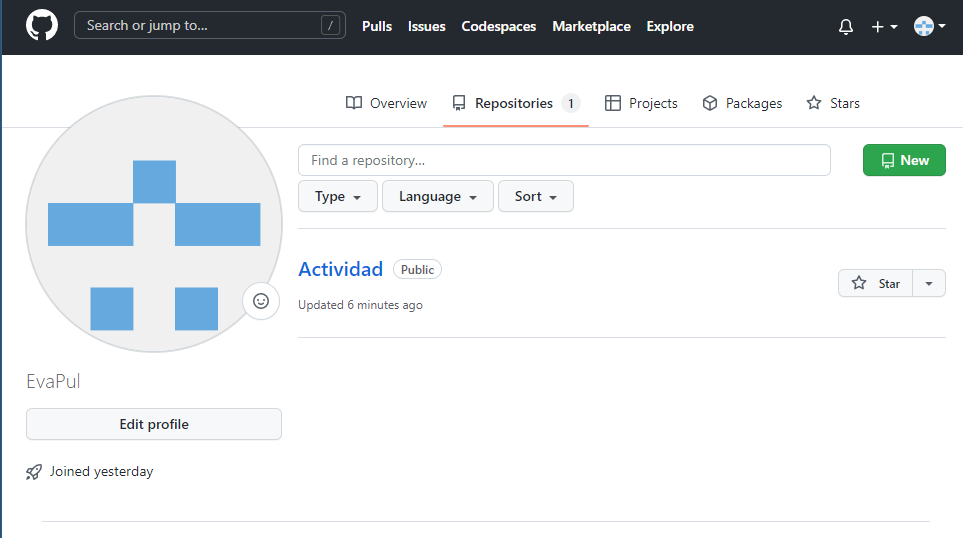
Comenzamos con el comando git remote add prueba <Url del repositorio> para poder subir los archivos al repositorio.



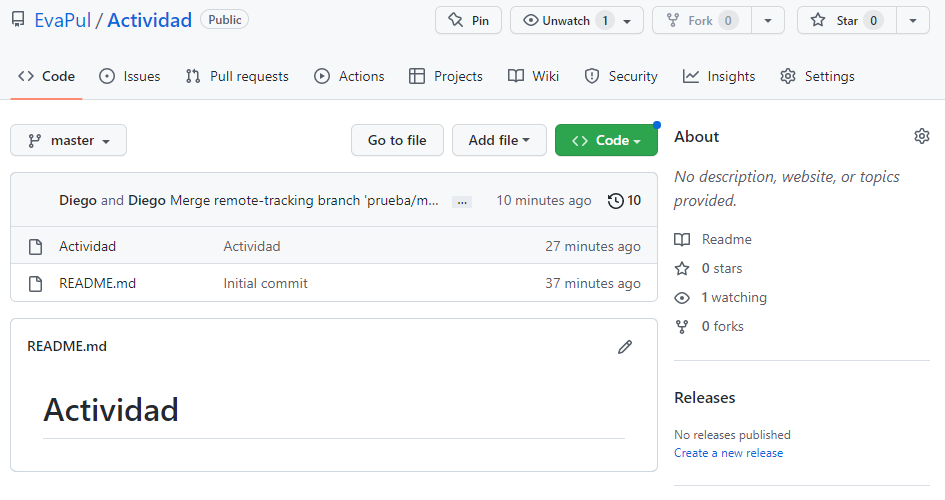
**9º** Utilizamos el comando git push -u prueba master para subir los archivos al repositorio indicado con el comando anterior.



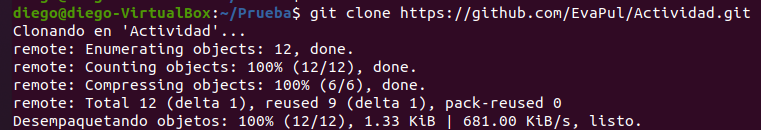
**10º** Si vamos a GitHub a repositorios podemos ver que ha tenido una actualización hace tan solo 6 minutos. Si pinchamos en Actividad podemos ver el repositorio.



**11º** Dentro podemos ver las actualizaciones realizadas en el repositorio.



**12º** Vamos a la terminal del compañero y ejecutamos el comando git clone <Url del repositorio> para descargar el repositorio en la máquina del compañero y poder trabajar en él.

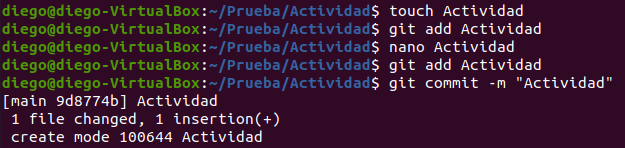


**13º** Una vez que hemos descargado el repositorio ejecutamos el comando touch <nombre fichero> para crear el fichero, en este caso Actividad.

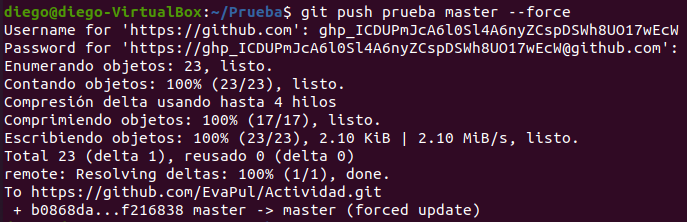
Después ejecutamos en comando git add <nombre fichero> para añadir stage. Una vez que lo tenemos en el stage ejecutamos el comando nano <nombre fichero> para editar el archivo.

Una vez editado el archivo volvemos a ejecutar el comando git add <nombre fichero> para actualizar el fichero que tenemos en stage.

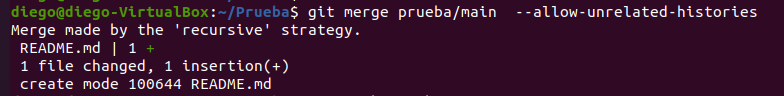
Finalmente ejecutamos el comando git commit -m “<nombre que se le quiera dar para identificarlo>” para generar un commit.



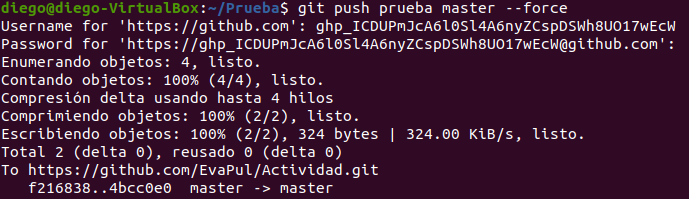
**14º** una vez creado el commit ejecutamos el comando git push prueba master --force para subir de forma forzada el commit al repositorio porque la página puede crear un main (como ha pasado en este caso).



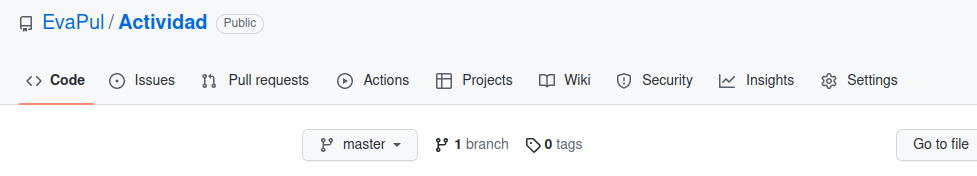
**15º** realizamos el comando git merge Prueba/main –allow-unrelated-histories para poder fusionar el branch main y master.



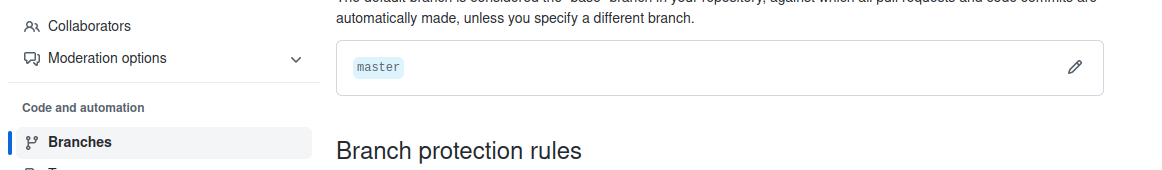
**16º** Una vez que hemos fusionado el main y master ejecutamos el comando git push prueba master --force para poder subir el repositorio.



**17º** Hemos tenido que cambiar la Branch principal y para ello hemos ido al repositorio y hemos pinchado sobre settings.



**18º** Una dentro de Settings podemos ver que nos aparece la opción Branches y pinchamos ahí. Automáticamente a la derecha nos aparece un recuadro donde podremos cambiar la Branch.

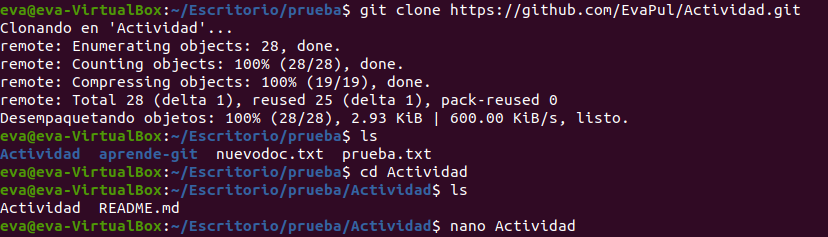


**19º** Una vez que hemos cambiado la Branch nos vamos al terminal (continuamos en la del compañero) y ejecutamos el comando git push prueba –-delete main para eliminarmain.



**20º** Ahora nos volvemos a la terminal del compañero inicial y ejecutamos el comando git clone <Url repositorio> para descargar en la maquina el repositorio actualizado por el otro compañero.

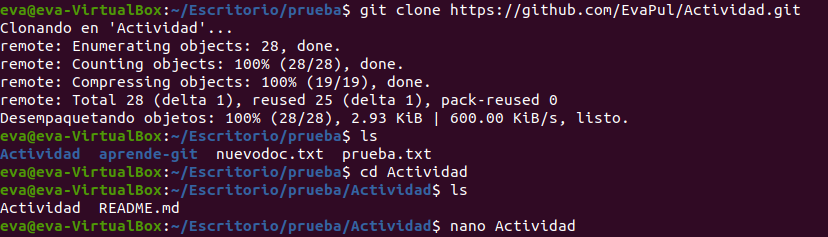
En este caso se realizó el comando git clone en vez de git pull porque estaba dando una serie de problemas de sincronización de archivos.



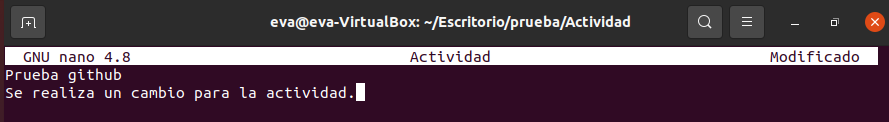
**22º** Ahora que ya tenemos el repositorio actualizado descargado en la maquina vamos a realizar una nueva modificación.

Para realizar la modificación nos situamos en prueba y ejecutamos el comando ls para listar y así poder ver los ficheros que tenemos en prueba. Podemos ver que tenemos el fichero Actividad y ejecutamos el comando cd Actividad para entrar en Actividad. Volvemos a realizar el comando ls para poder ver los ficheros que se encuentran dentro de Actividad y como vemos que tenemos el fichero Actividad que ha creado el compañero y ha modificado vamos a realizarle una nueva modificación.

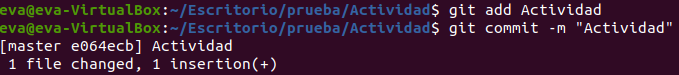
Ejecutamos el comando nano <nombre del fichero> para abrirlo y modificarlo.



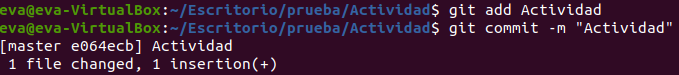
**23º** Una vez abierto el fichero podemos ver la frase escrita por el compañero “Prueba github” y nosotros vamos a añadir una segunda frase “Se realiza un cambio para la actividad”. Una vez realizado el cambio pulsamos Ctrl+O para guardar el cambio y cerrar el archivo.



**24º** Una vez editado el archivo volvemos ejecutamos el comando git add <nombre fichero> para añadirlo al stage.



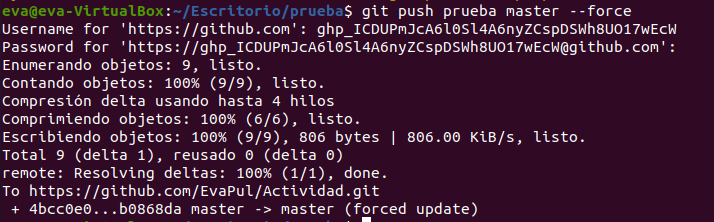
**25º** Ahora ya ejecutamos el comando git commit -m “<nombre que se le quiera dar para identificarlo>” para generar un commit.



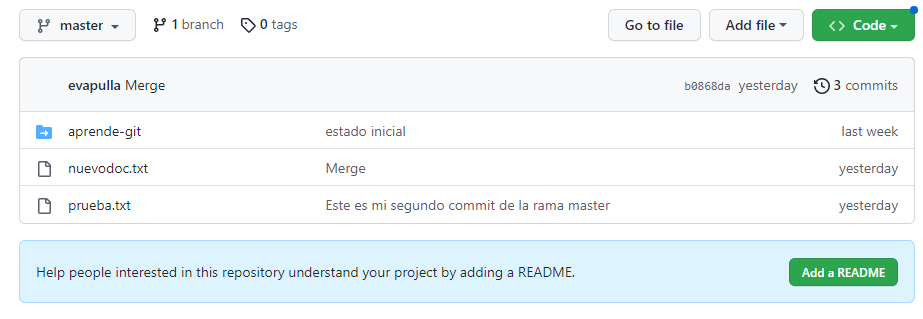
**26º** Realizamos el comando cd . . para salir de Actividad e ir a prueba.



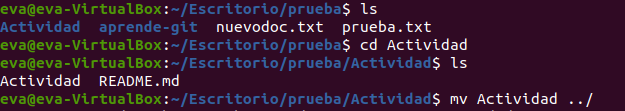
**27º** Ahora ya desde prueba realizamos el comandogit push prueba master –-force para poder subir el repositorio ya editado.



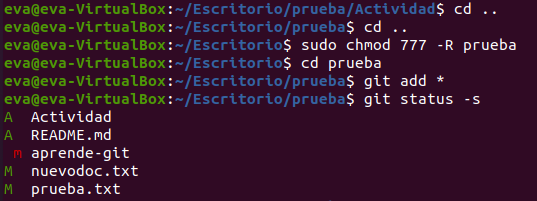
**28º** Si vamos a GitHub podemos ver las actualizaciones en el repositorio.



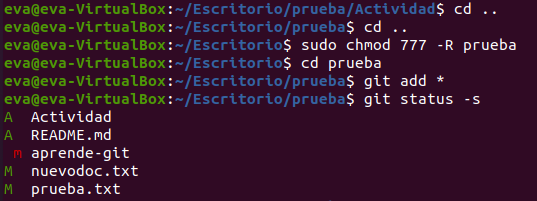
**28º** En el siguiente paso lo que vamos a hacer es mover los ficheros que tenemos en Actividad a prueba y para ello realizamos el comando mv Actividad . . /



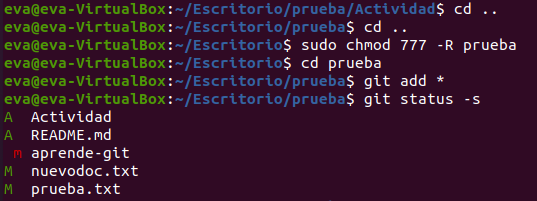
**29º** En nuestro caso tuvimos problemas de permisos en la carpeta y lo solucionamos con el comando sudo chmod 777 -R para poder dar permisos de todo tanto al directorio como a los ficheros.



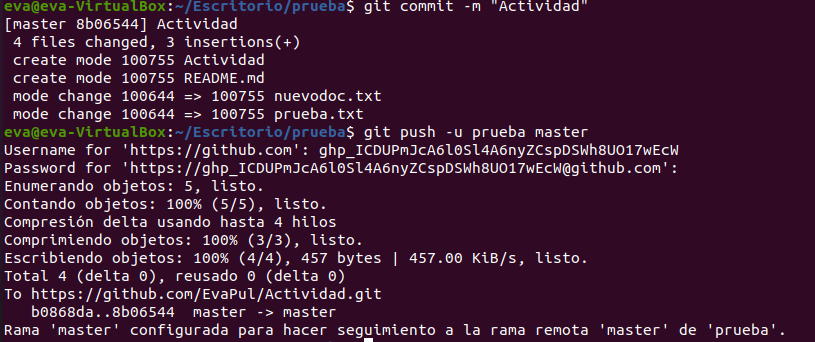
**30º** Luego ejecutamos el comando git add \* para agregar todos los archivos al stage.



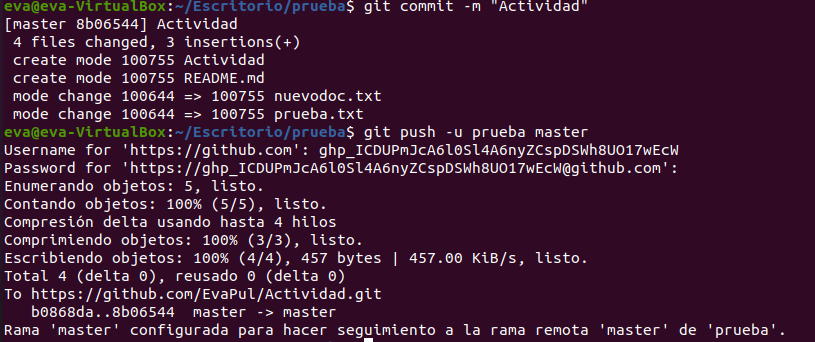
**31º** una vez que ya está subido al stage realizamos el comando git status -s para comprobar que se ha subido correctamente.



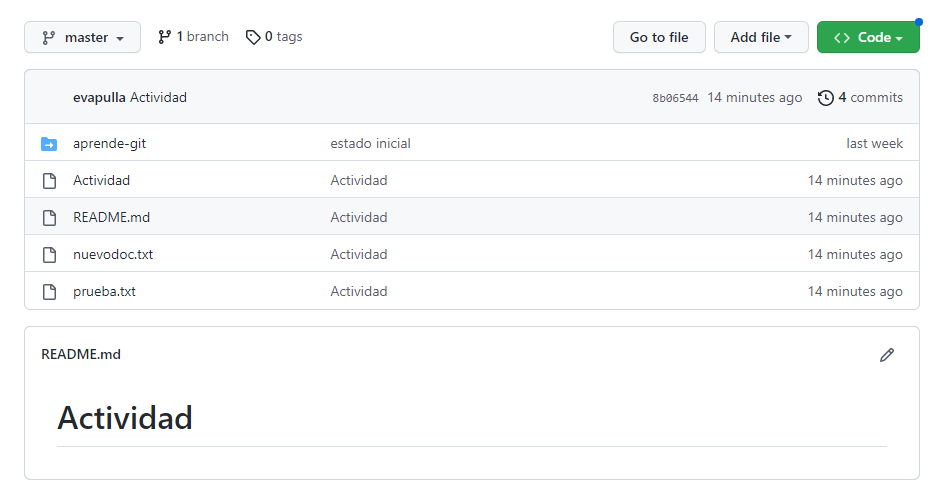
**32º** Ejecutamos el comando git commit -m “<nombre que se le quiera dar para identificarlo>” para generar un commit.



**33º** Finalmente ejecutamos el comandogit push prueba master –-force para poder subir el repositorio.



**34º** Si vamos a nuestro repositorio en GitHub podemos visualizar las actualizaciones del repositorio.



**35º** Para el siguiente paso nos vamos al terminal del otro compañero, en este caso Diego, para realizar unos nuevos cambios (meter una línea más de texto).

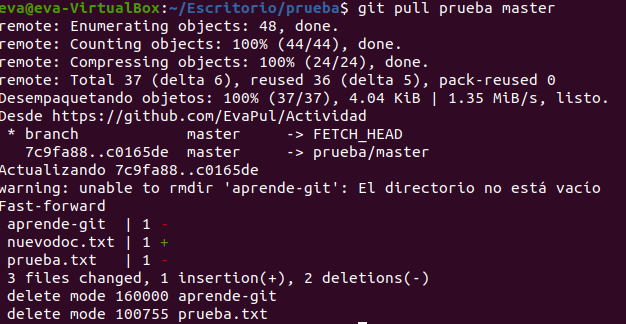
Una vez que hemos realizado los cambios ejecutamos el comando git add <nombre archivo> para subirlo al stage y finalmente realizamos el comando git commit -m “<nombre que se le quiera dar para identificarlo>”.



**36º** Una vez realizado el commit ejecutamos el comando git push prueba master para volver a subirlo al repositorio.



**37º** Vamos al terminal del otro compañero, en este caso Eva y realizamos el comando git pull para bajarnos las modificaciones del repositorio.



**38º** En GitHub vemos finalmente como quedaría nuestro repositorio.

