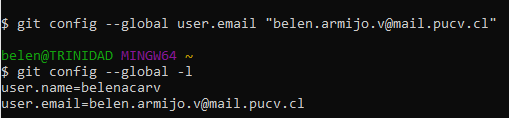
Tarea 4 FIS1220

La siguiente tarea muestra el trabajo que se realiza a través de comandos git, que es una herramienta reconocida por su preservación de versiones de código. Dentro de la tarea se crearán distintos archivos y se mostrara como se va actualizando su versión.

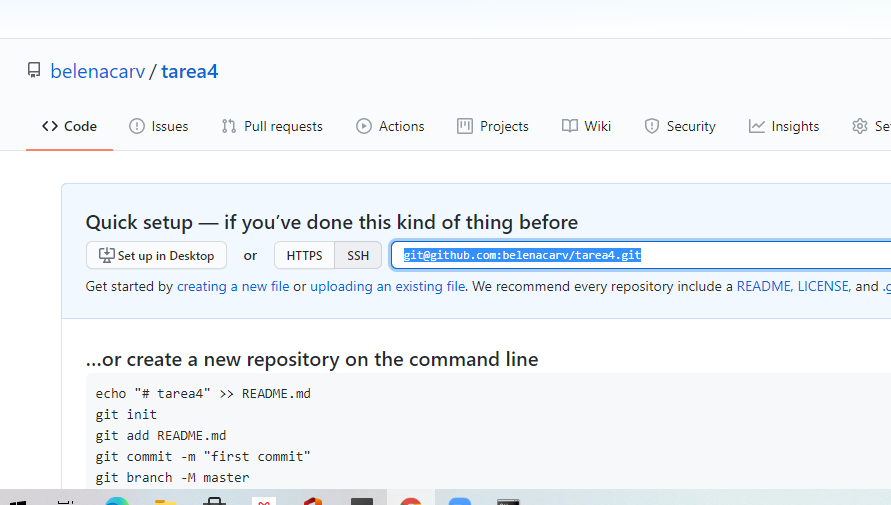
Todo cambio esta guardado en el repositorio :

Iniciando y creando las credenciales gitlab establecidas en el equipo de trabajo:

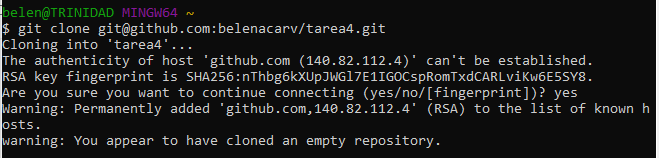


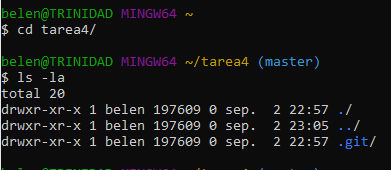


Proyecto creado en GitHub:

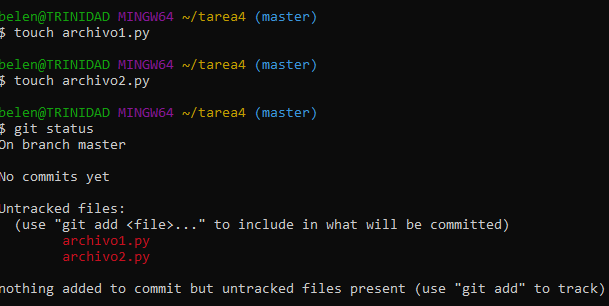


Se procede a realizar un clone del proyecto desde github para trabajar desde la estación de trabajo:





Creando los archivos que luego se subirán a GitHub, y se hace un status, vemos que aún los archivos no están subidos a GitHub:



Los comandos que se usarán para nuestro trabajo con git serán los siguientes:

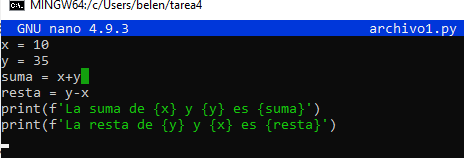
git status: iremos viendo el estado de rastreo de nuestros archivos con respecto al repositorio.

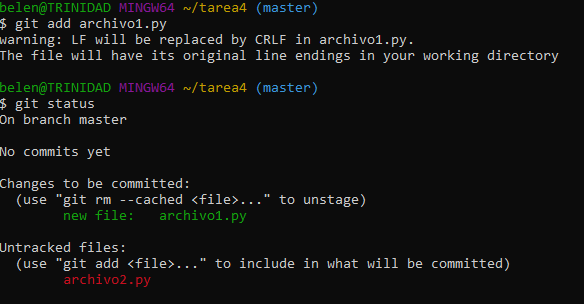
git add <archivo>: añadiremos a nuestra conveniencia el archivo a la cola que será actualizada posteriormente en github.

git commit -m «»: daremos el comentario de las modificaciones que se han realizado para mantener un seguimiento y trabajo transparente cuando se trata de compartir el repositorio con otras personas.

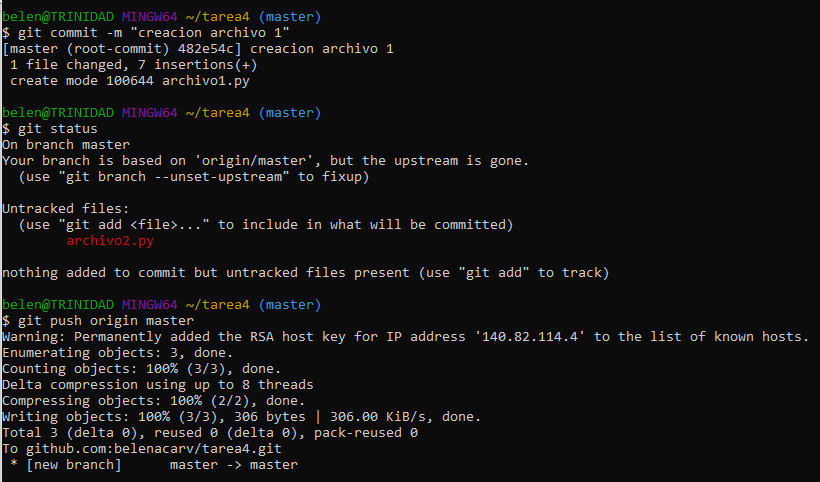
git push origin master: confirmar todo lo realizado anteriormente y subir los nuevos cambios a nuestra rama master.

Primero se trabajará en el archivo1 de acuerdo con lo solicitado en la tarea, en el procedimiento se irán subiendo los cambios solo de este archivo al repositorio, por lo que el archivo2 se verá constantemente como «sin seguimiento» hasta que se termine con este primer punto.

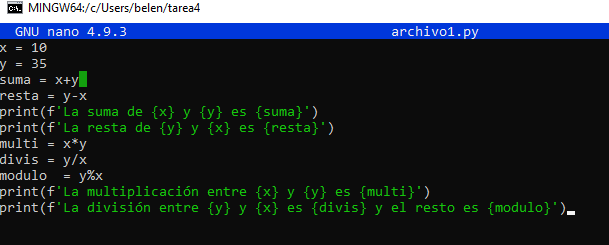


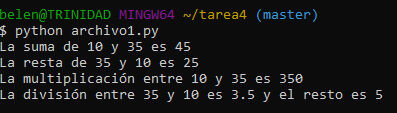


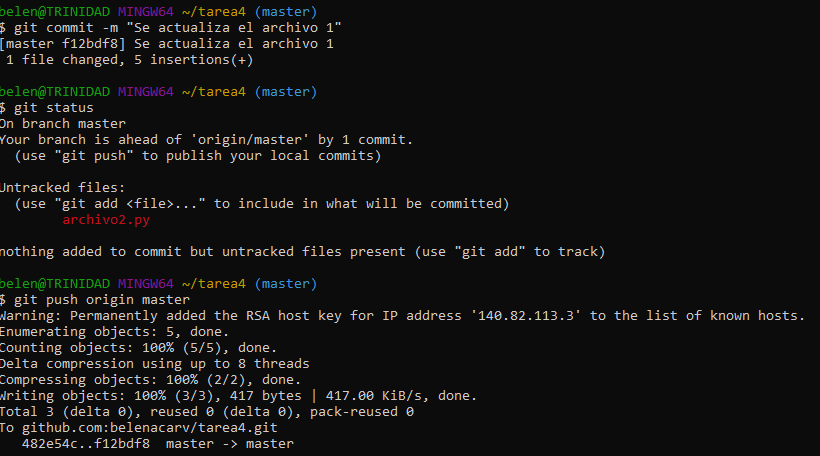
Luego al archivo se le añade un comentario con el comando **git commit -m “comentario**”:



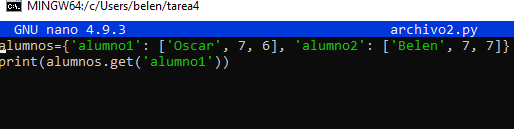
Se actualiza el archivo 1:



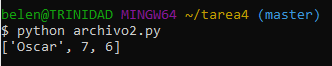


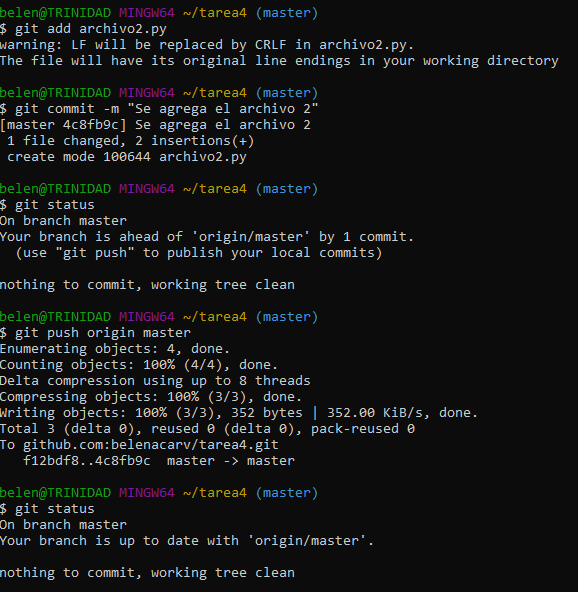
Se le agrega el comentario con **git** **commit,** para luego subirlo a GitHub actualizado con la línea de comando ***git push origin master***:

Ya terminado con el archivo1, se procede con el archivo2, de acuerdo a lo solicitado en la tarea, se hace el código en nano:

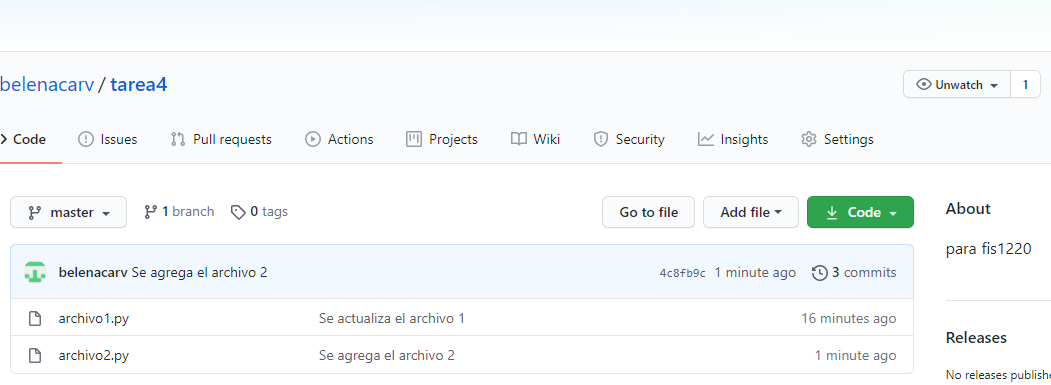


Se prueba su ejecución en la terminal:

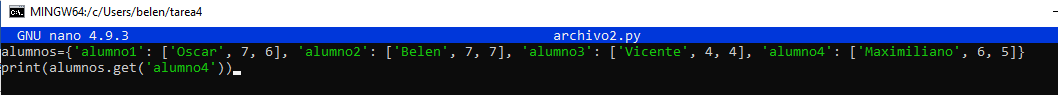


Luego se agrega el archivo para luego subirlo a GitHub:

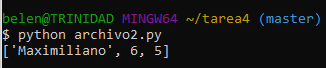
Se verifica que se cargó en la plataforma:

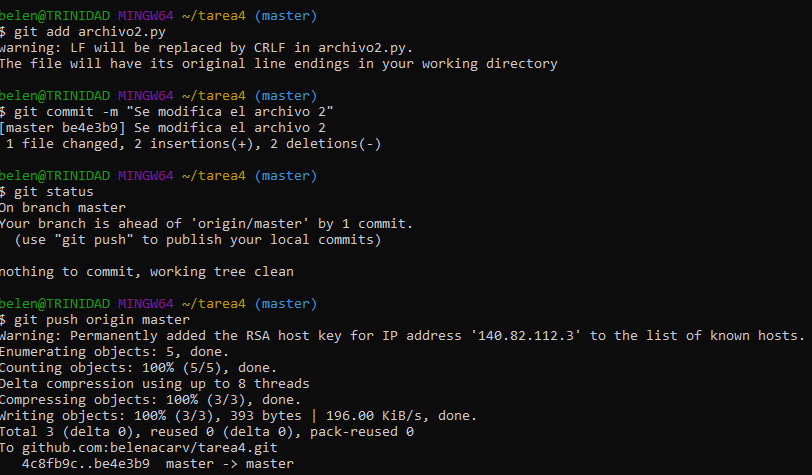


Ahora se hace su actualización, para utilizar el segundo ‘commit’:



Ahora operamos con él para ejecutarlo y hacer la revisión:





Los archivos terminan en su versión final de acuerdo con la tarea propuesta como se encuentra actualmente el repositorio.

También se muestran los commits hechos en el historial del repositorio:  
  
