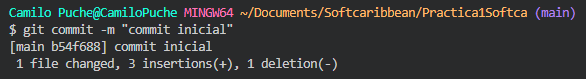
Luego de la creación del repositorio en GitHub se clona en la carpeta deseada con el comando

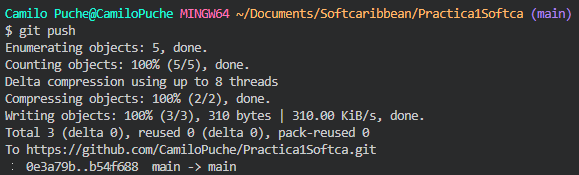
Git clone, seguido del URL donde este alojado el repositorio



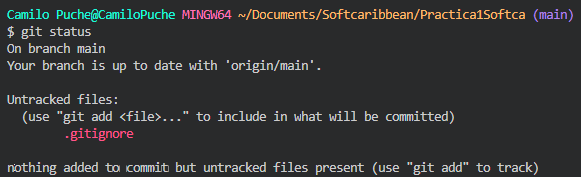
Se ejecuta el comando git add . para agregar los cambios realizados al archivo en el staging área

seguido del comando git commit -m “mensaje” para confirmar los cambios realizados

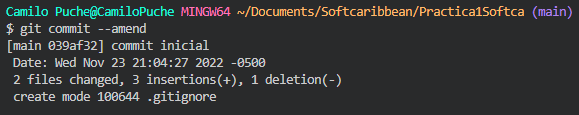
 y por ultimo el comando git push para subir los cambios al repositorio remoto alojado en GitHub



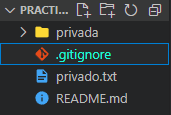
Para ignorar archivos y carpetas en nuestro repositorio debemos tener el archivo .gitignore en nuestro repositorio donde se escribirán los nombres de los archivos o carpetas que van a ser ignorados por git



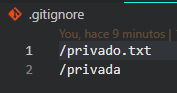
Luego de crear el archivo se agrega al staging y para no crear commit innecesarios se ejecuta el comando git commit –amend para que el cambio se fusione con el commit anterior



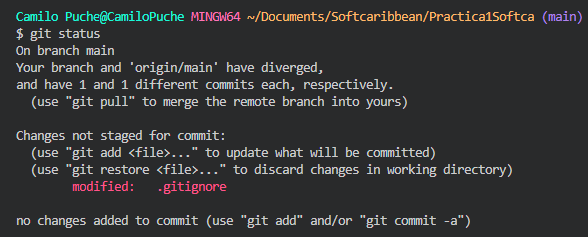
Ya creados el archivo y la carpeta a ignorar



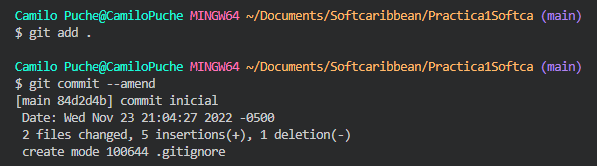
Se pasa a escribirlos en el archivo .gitignore para que git no les haga seguimiento



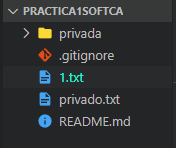
Al hacer un git status el archivo y la carpeta escritos en el archivo .gitignore no aparecen trackeados, solo que el archivo .gitignore ha sido modificado



Se realiza el paso anterior para no crear commit innecesarios y quedan los archivos ignorados en el repositorio local



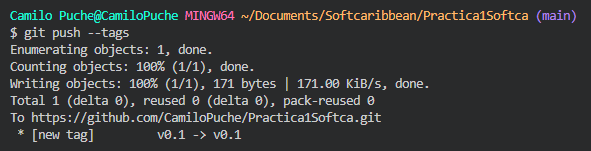
Añadimos el archivo 1.txt al repositorio local



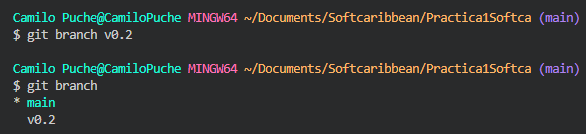
Se crea el tag v0.1



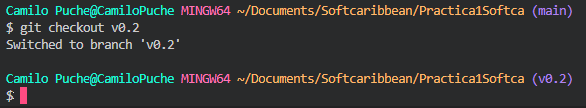
Se sube el tag al repositorio remoto



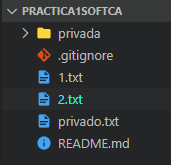
Se crea la nueva rama v0.2



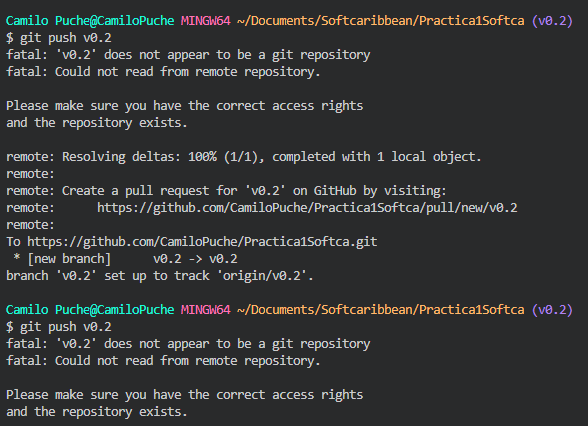
Pasamos el área de trabajo a la nueva rama v0.2



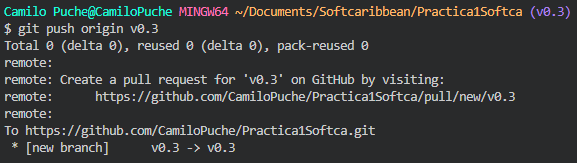
Creando el fichero 2.txt en la rama v0.2



Subimos la rama con sus cambios al repositorio remoto



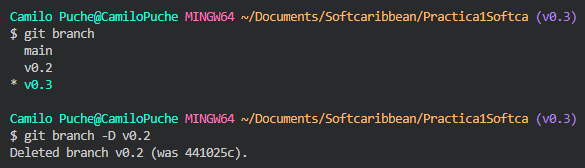
Arrojo un error que tuve que solucionar desde GitHub y me dejo hacer el push



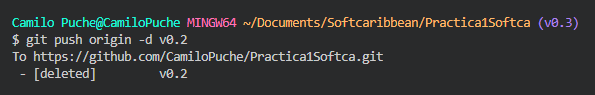
Pude solucionar el error, me equivoqué en el comando, para subir una rama a GitHub el comando push debe especificar el repositorio en este caso es origin seguido del nombre de la rama

Lo que hare será borrar la rama v0.2 y renombrar la rama v0.3 llamándola v0.2

Borrando la rama v0.2 localmente



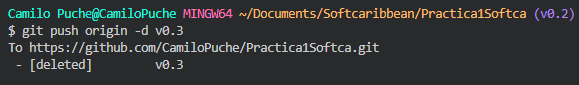
Borrando la rama v0.2 del repositorio remoto



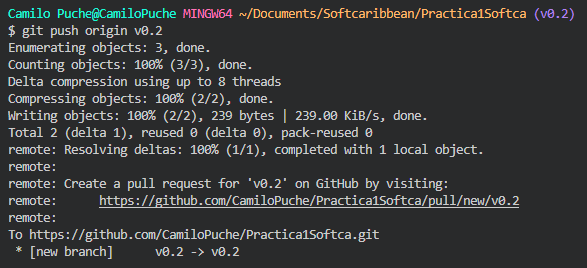
Renombrando la rama v0.3 con el nombre v0.2 localmente



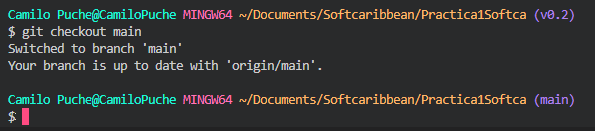
Borrando la rama v0.3 del repositorio remoto para luego subir la rama renombrada como v0.2



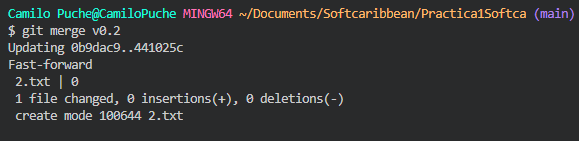
Ahora se sube la rama v0.2 al repositorio remoto



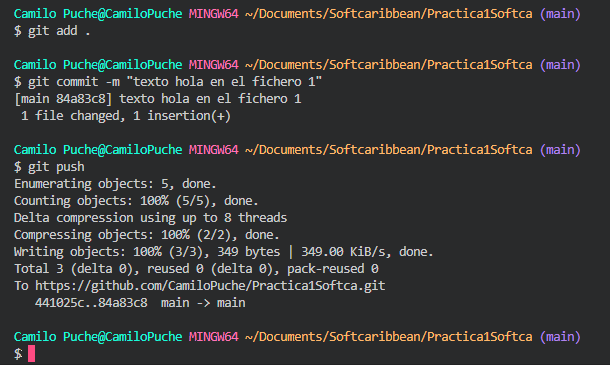
Me posiciono en la rama master (actualmente rama main)



Haciendo un merge directo de la rama v0.2 con la rama main

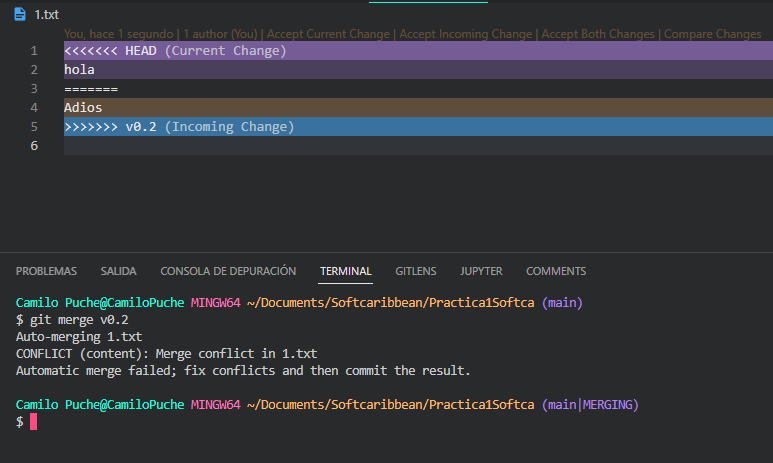


Hacer un merge que intencionalmente tenga conflicto, primero edito el fichero 1 y hago un commit en la rama main



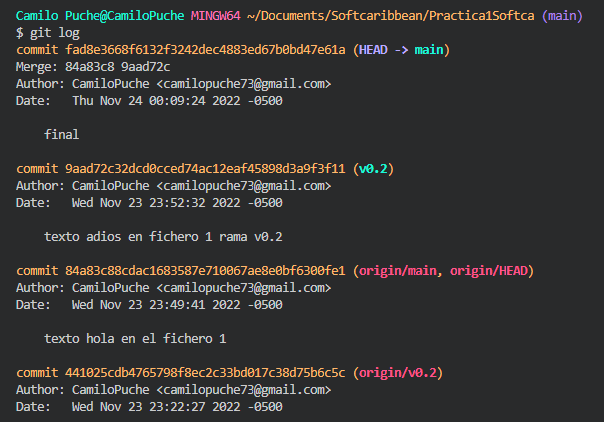
Me posiciono en la rama v0.2, modifico en la misma línea el fichero 1 y vuelvo a posicionarme en la rama main

Luego hago nuevamente un merge entre las dos ramas esperando el error



Antes de listar las ramas arregle el conflicto en el merge ya que investigue y no pude encontrar una solución mientras se esta en un conflicto como listar las ramas

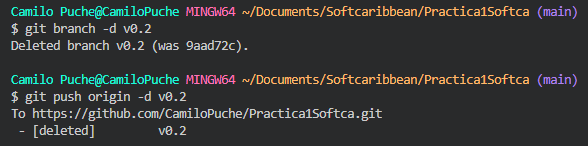
Ramas listadas (o eso creo)



Se crea el tag v0.2



Se borra la rama v0.2 en local y remoto



Listado de todos los commit son ramas y tags, de forma un poco grafica para ver el movimiento de las ramas

