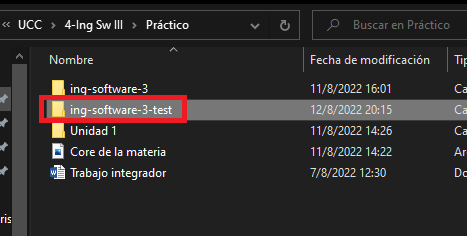
**1 – Instalar git**

Ya estaba instalado

**2 - Crear un repositorio local y agregar archivos**

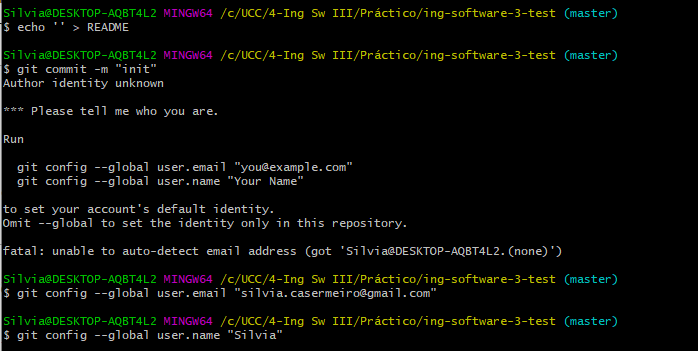
• Crear un repositorio local en un nuevo directorio.



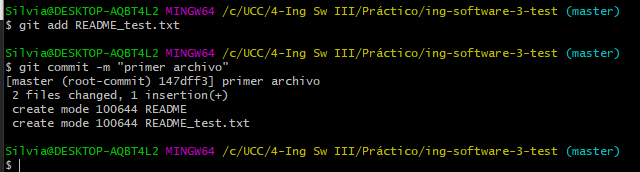
• Agregar un archivo Readme.md, agregar algunas líneas con texto a dicho archivo.

• Crear un commit y proveer un mensaje descriptivo.

Creado el archivo, trato de commitearlo. Me pide me autentique.



Agrego el archivo y commiteo

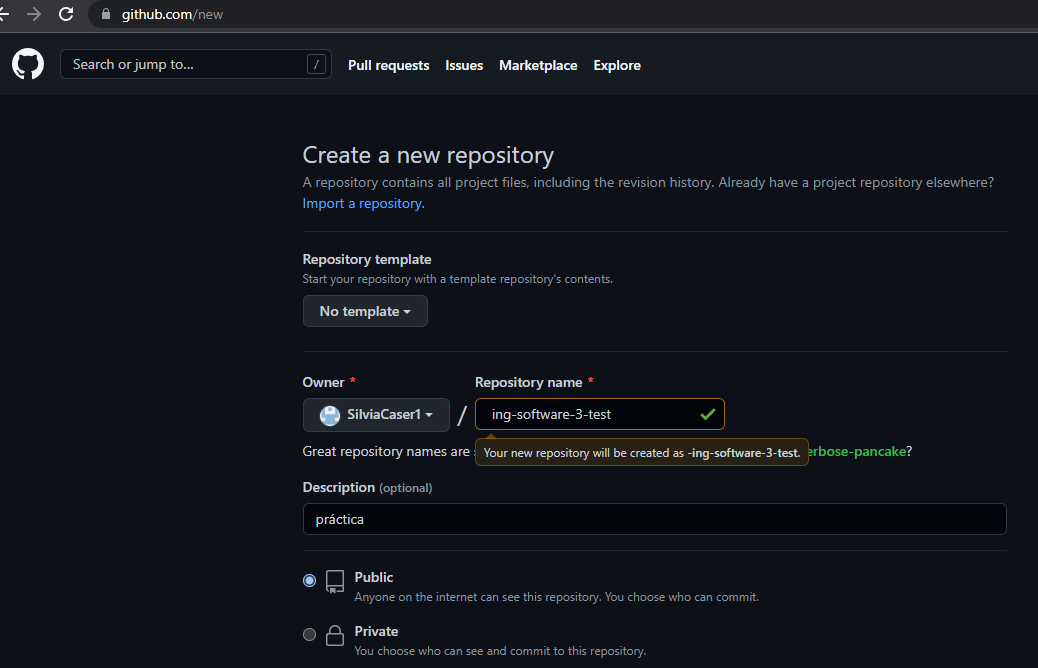


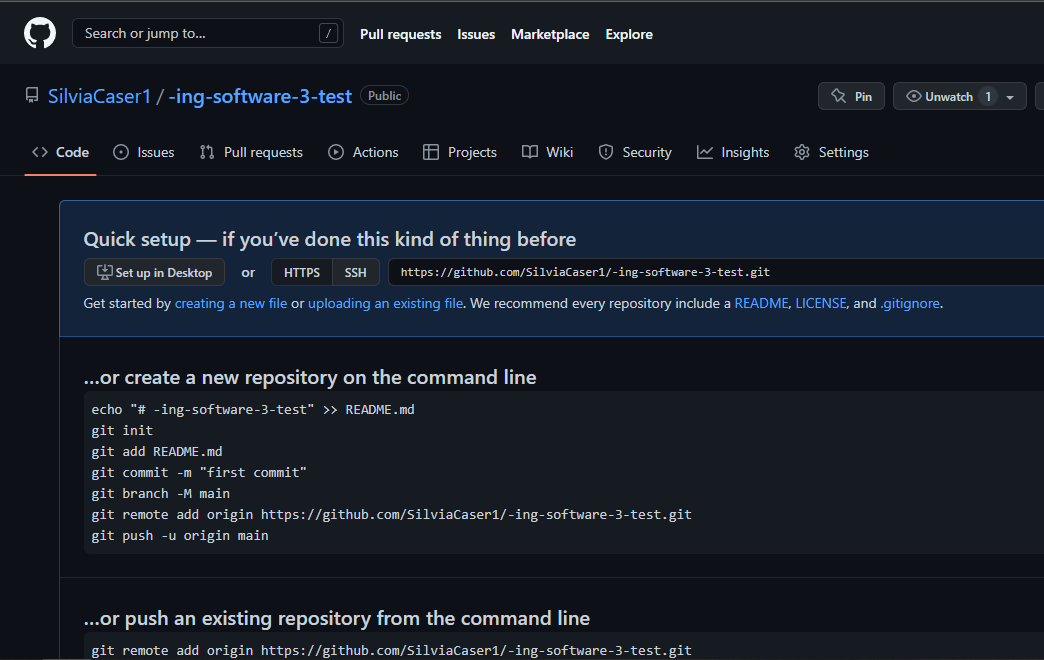
**3- Crear un repositorio remoto**

• Crear una cuenta en <https://github.com>

Ya la tenía creada

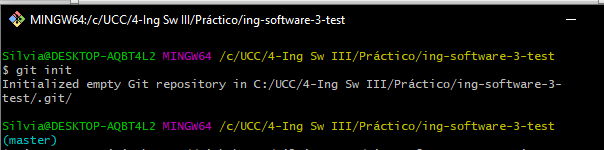
• Crear un nuevo repositorio en dicha página (vacío)

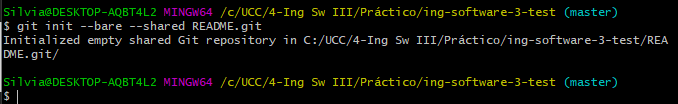




• Asociar el repositorio local creado en el punto 2 al creado en github.

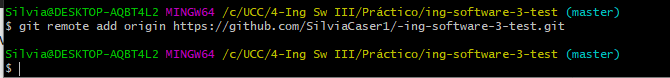
Se inicializa el repo local

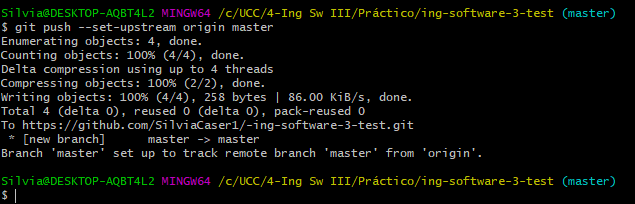


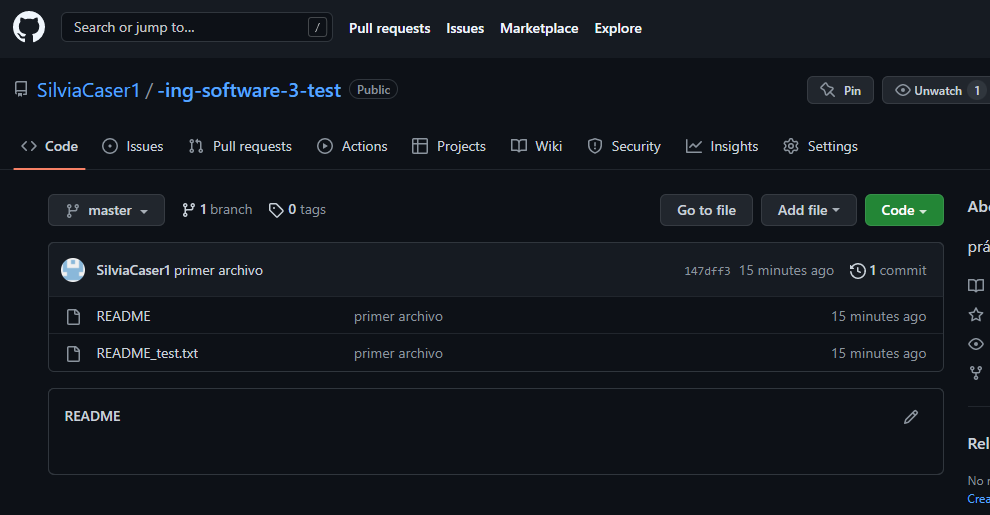


• Subir los cambios locales a github

Para poder subir los cambios, primero hay que agregar el repo como origen







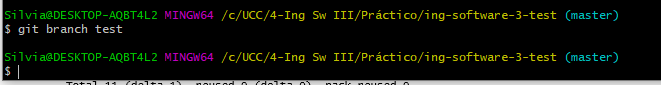
**4- Familiarizarse con el concepto de Pull Request**

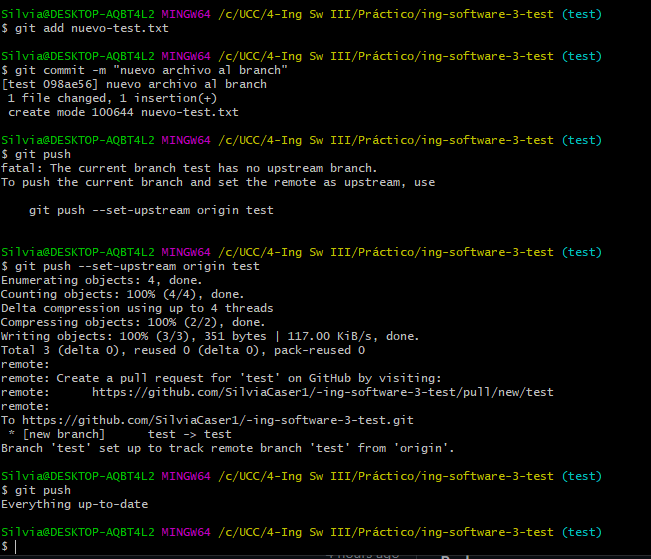
Para algunos de los puntos proveer imágenes.

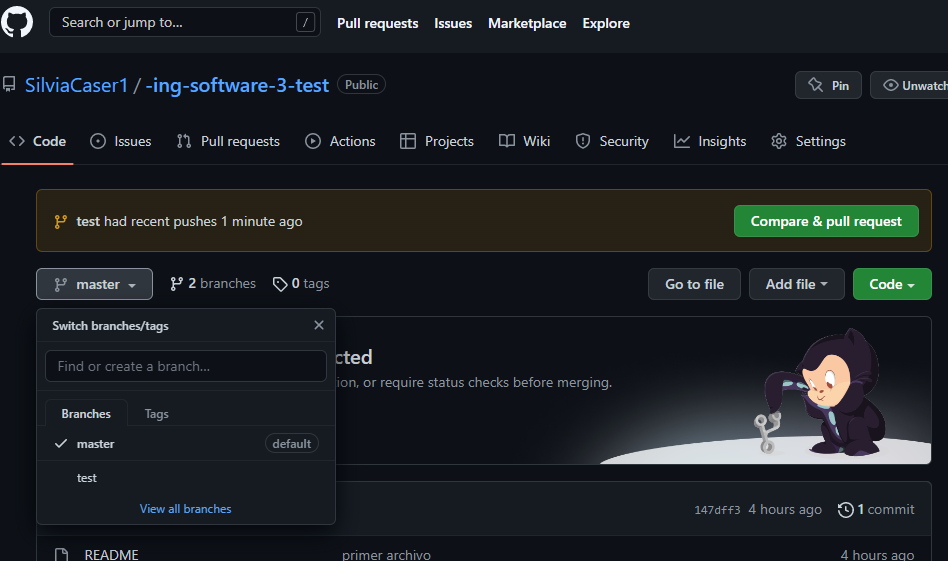
• Explicar que es un pull request:

Es un conjunto de archivos, relacionados a una feature, en la cual un desarrollador estuvo trabajando local y se quiere mergear con el resto del código en el repositorio.

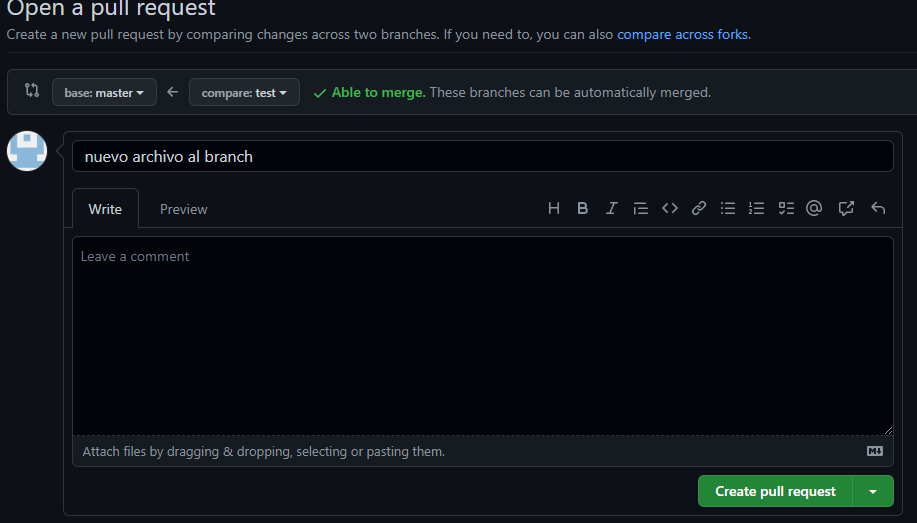
• Crear un branch local y agregar cambios a dicho branch.

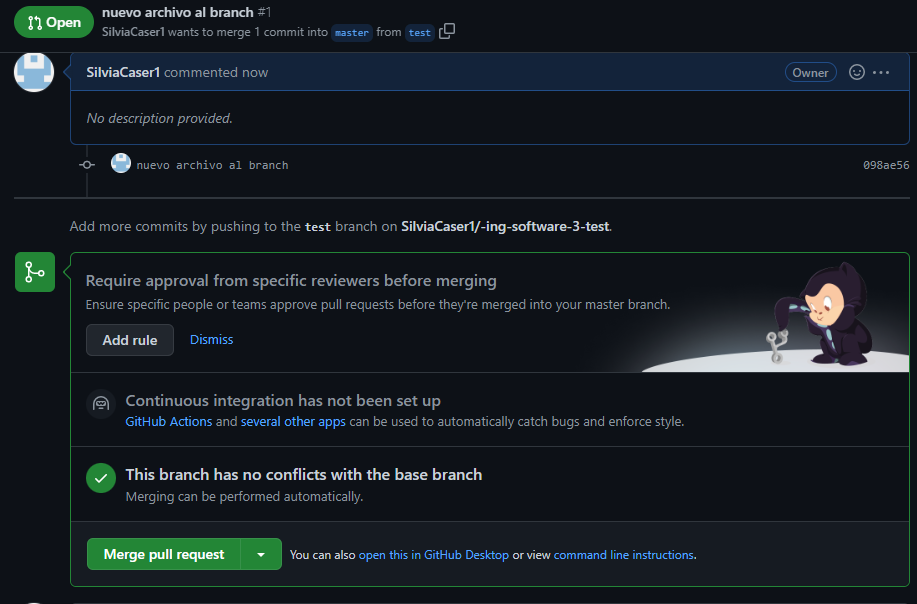




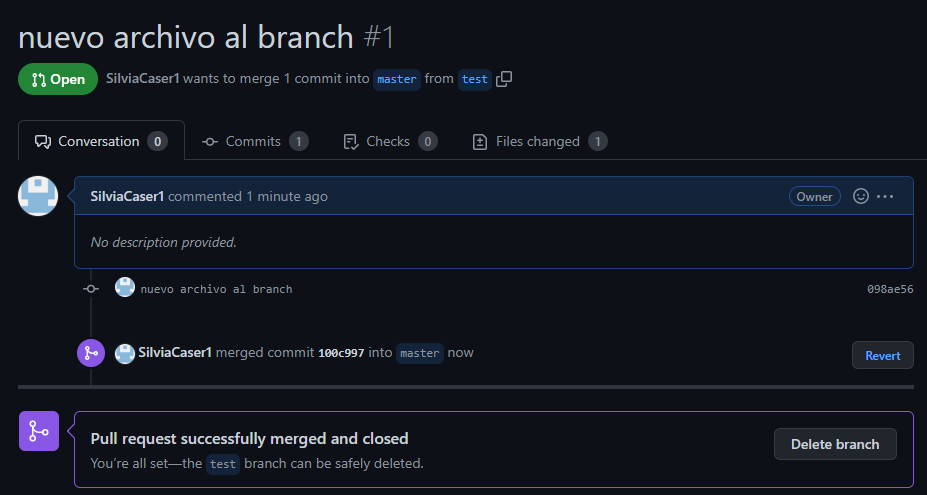


• Subir el cambio a dicho branch y crear un pull request.





• Completar el proceso de revisión en github y mergear el PR al branch master



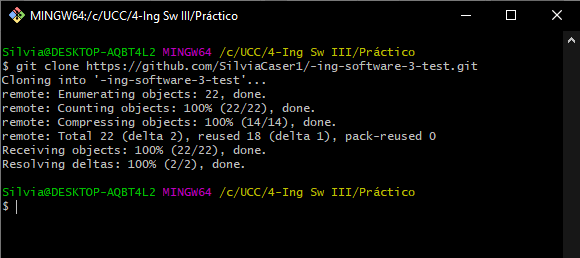
**5- Mergear código con conflictos**

* Instalar alguna herramienta de comparación. Idealmente, una 3-Way:

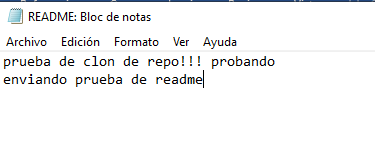
o P4Merge

o ByondCompare trial

* Clonar en un segundo directorio el repositorio creado en github

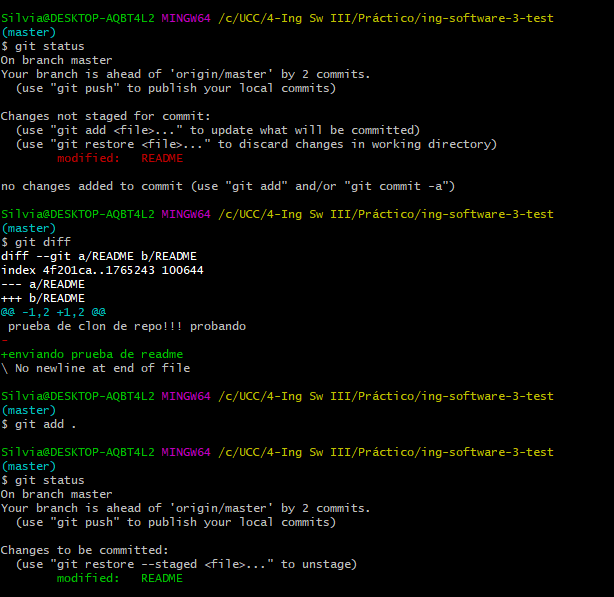


* En el clon inicial, modificar el Readme.md agregando más texto (en ing-sw-3-test)

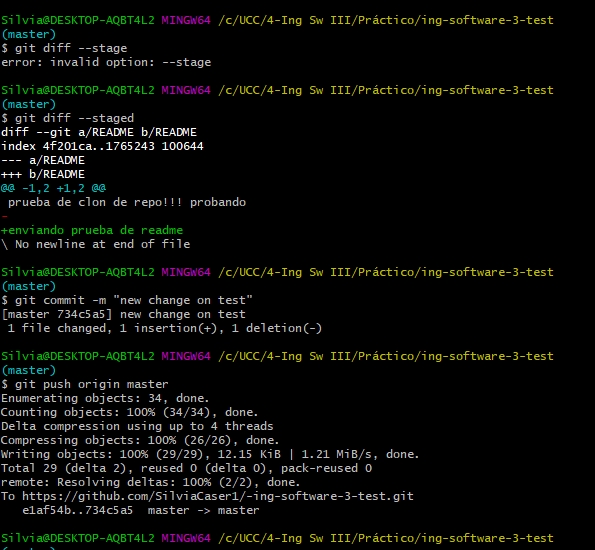


* Hacer commit y subir el cambio a master a github

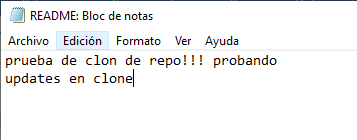
1. Con git status, obtengo que un archivo fue modificado en master
2. Con git diff, obtengo las diferencias entre el actual y el nuevo archivo
3. Con git add ., agrego el nuevo readme con la modificación última
4. Git status de nuevo veo que está en master el archivo con el cambio



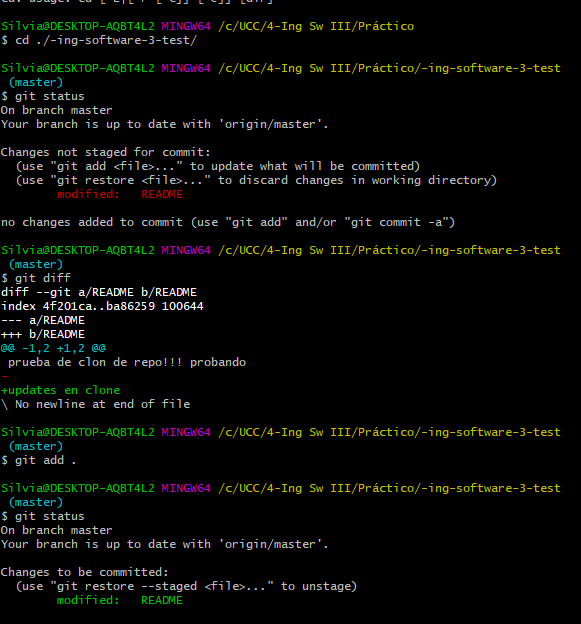
1. Se vuelve a hacer git diff para verificar la diferencia entre el anterior y el actual archivo
2. Con git commit –m “comments” confirmo mis cambios
3. Con git push origin master, subo los cambios a master



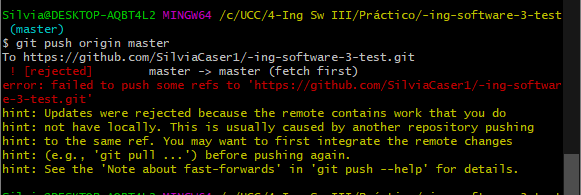
* En el segundo clon también agregar texto, en las mismas líneas que se modificaron el punto anterior.



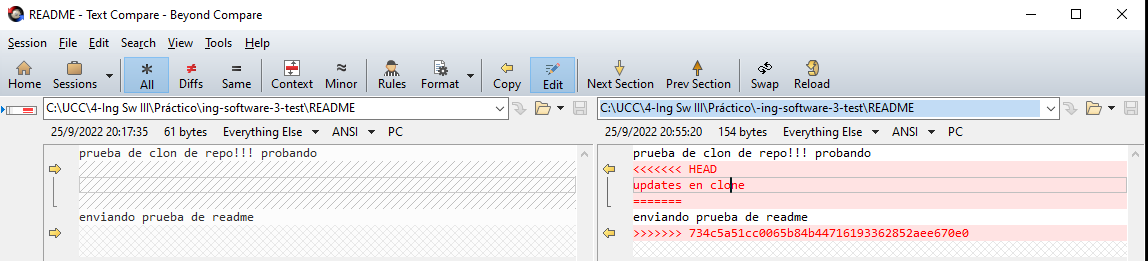
* Intentar subir el cambio, haciendo un commit y push. Mostrar el error que se obtiene.

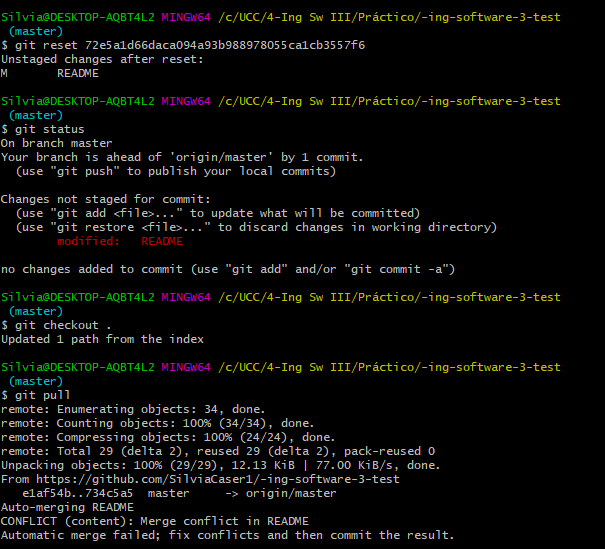


* Hacer pull y mergear el código (solo texto por ahora), mostrar la herramienta de mergeo como luce.

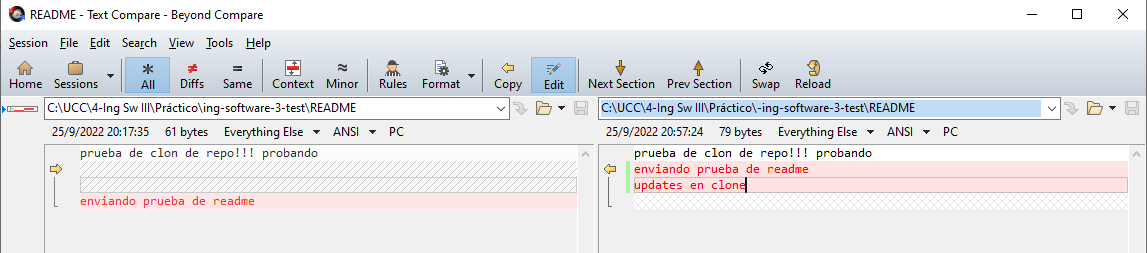


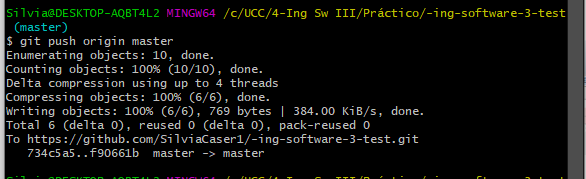
En la tool Beyond compare muestra





* Resolver los conflictos del código.





* Explicar las versiones LOCAL, BASE y REMOTE.
  + Remote: archivo remoto dentro del repo en línea (cambios realizados por otros usuarios). Estamos del lado de las personas externas que están haciendo los cambios del conflicto. En la rama o branch que se va a funcionar o mergear con el HEAD
  + Base: ancestro común, origen de los archivos local y remote (sin modificaciones)
  + Local: archivo local, incluidas las propias modificaciones. Versión nuestra del conflicto, nuestra rama o branch HEAD que va a contener los resultados del merge.
* Pushear el cambio mergeado

