**ESCUELA COLOMBIANA DE INGENIERÍA**

**JULIO GARAVITO**

**Laboratorio N °1**

**CVDS, Grupo 1**

**Profesor:Oscar David Ospina**

**Juan José Mejía Celis**

**Laura Valentina Gutiérrez Rico**

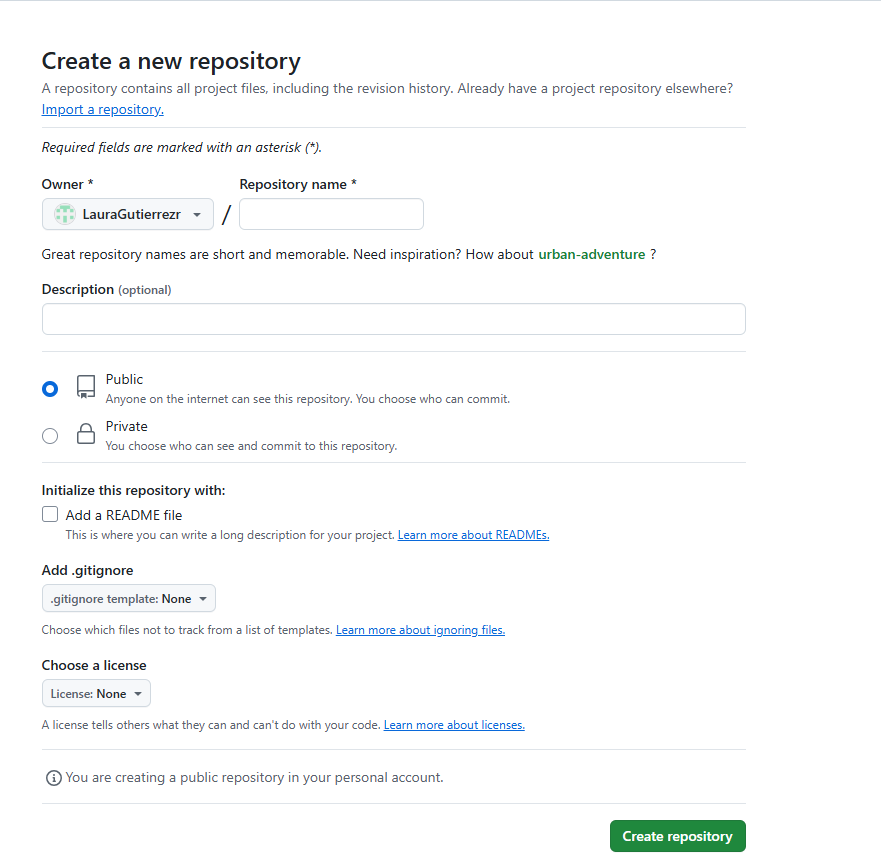
**Enero**

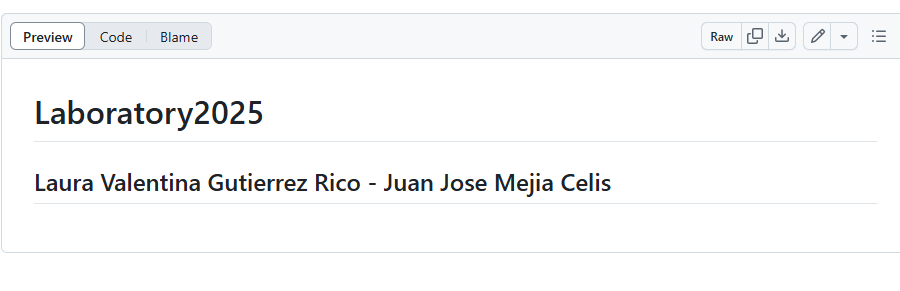
**2025-1**

## **PARTE I (Trabajo Individual)**

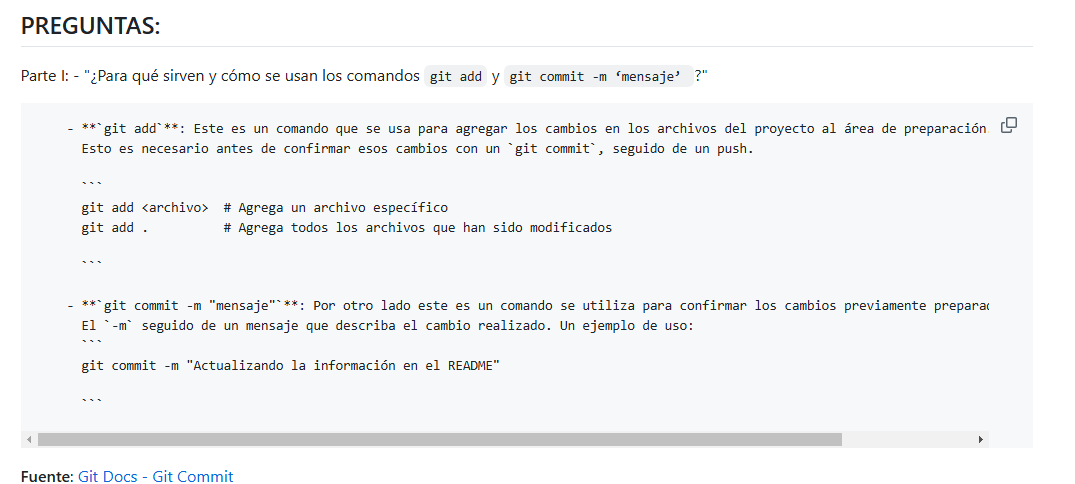
Creando el repositorio remoto:

1. **Como primer paso creamos un repositorio local en nuestro Github, tomamos como ejemplo el de LauraGutierrezr**



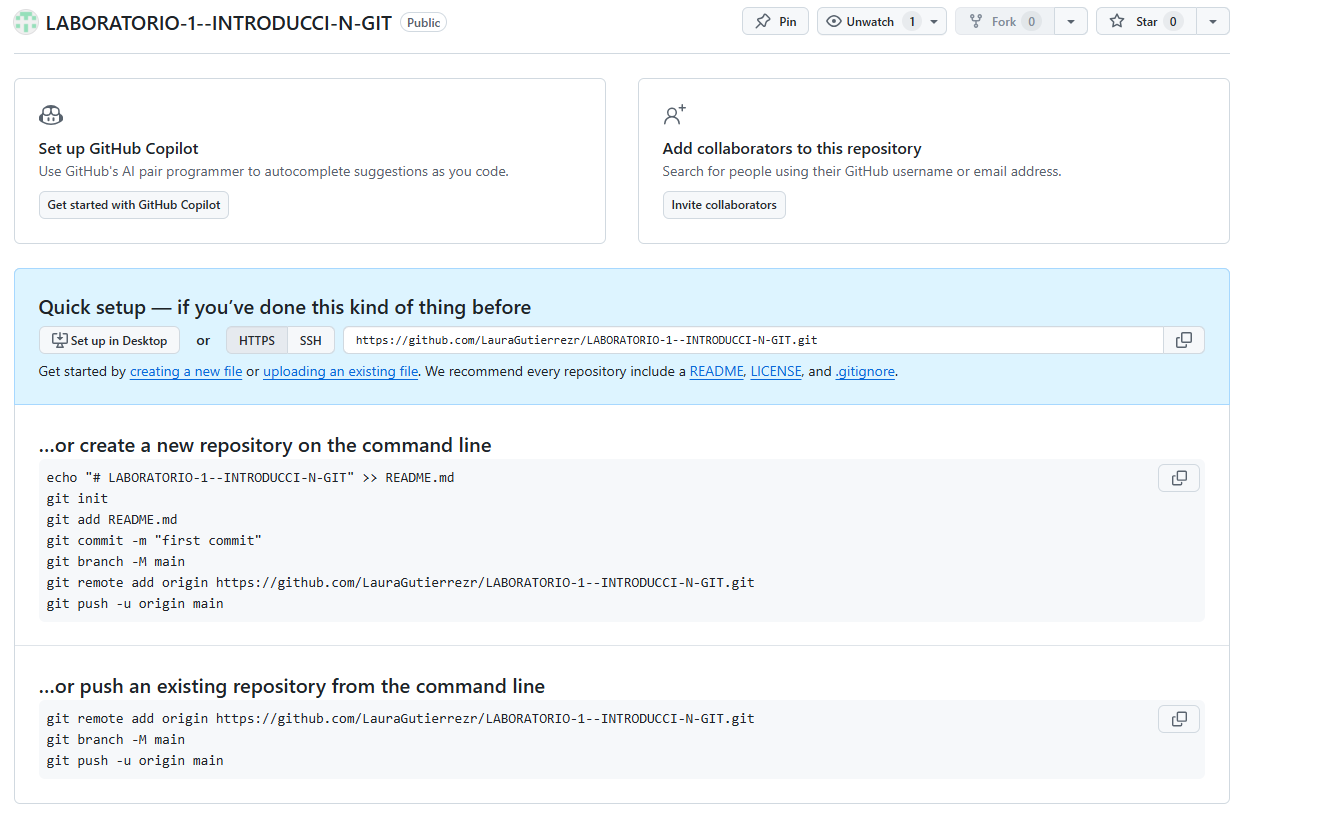
1. **Creamos en el repositorio un archivo README con nuestros nombres**

**3. Investigando definiciones**

Agregamos las definiciones de git add y git commit en el README actualizado

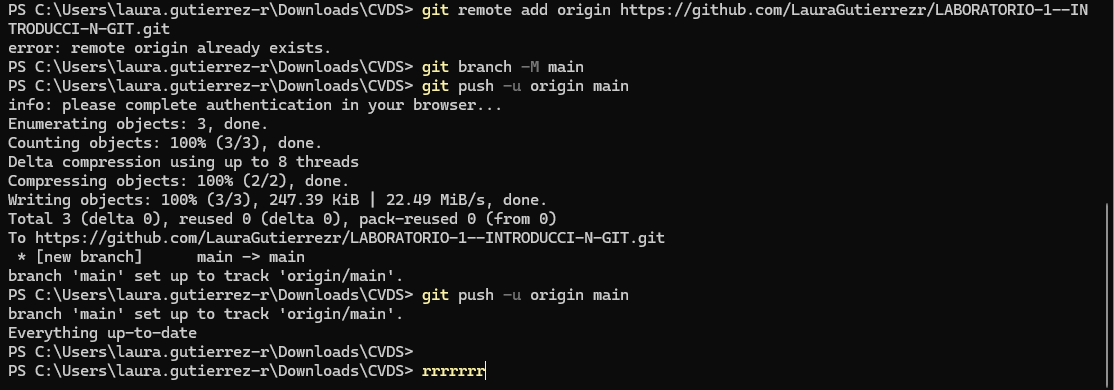
**4. Creamos nuestras cuentas enlazadas al correo institucional previamente**

**5. Creando un repositorio en blanco (vacío) e GitHub con los comandos específicos desde la terminal del computador**

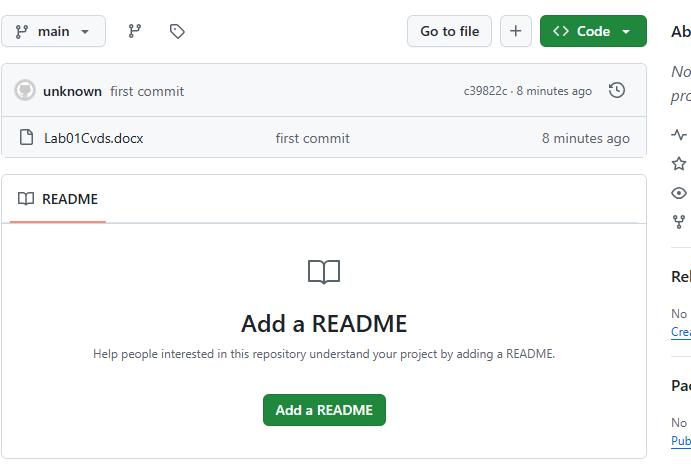


6. Configuramos el repositorio local con el repositorio remoto

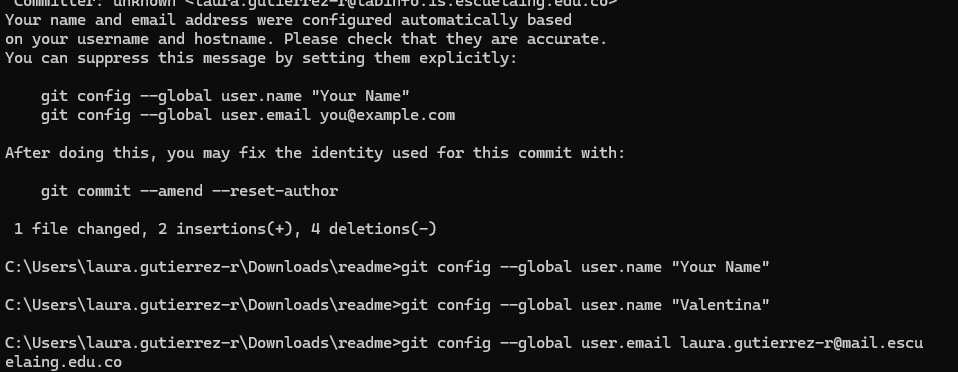
**6. Configuramos el repositorio local con el remoto**

En este caso tiene el nombre anterior del repositorio de la versión final.

**7. Subimos los cambios realizados en el README y todo estaba correcto.**



**8. Configuramos los correos, enlazados a Git**

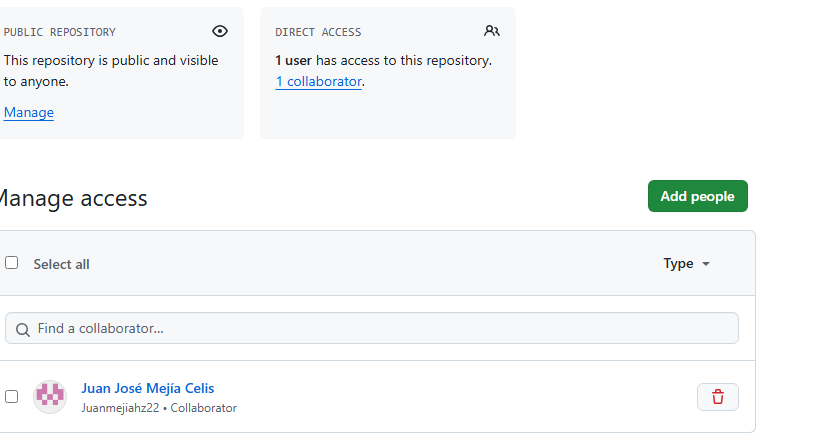
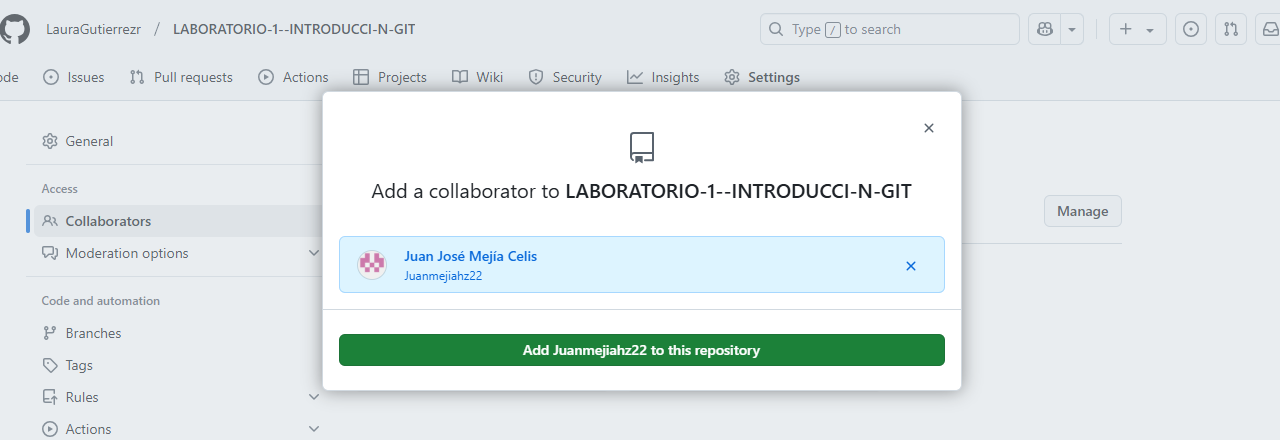


## **PARTE II (Trabajo en parejas)**

**Asignamos roles en nuestro equipo:**

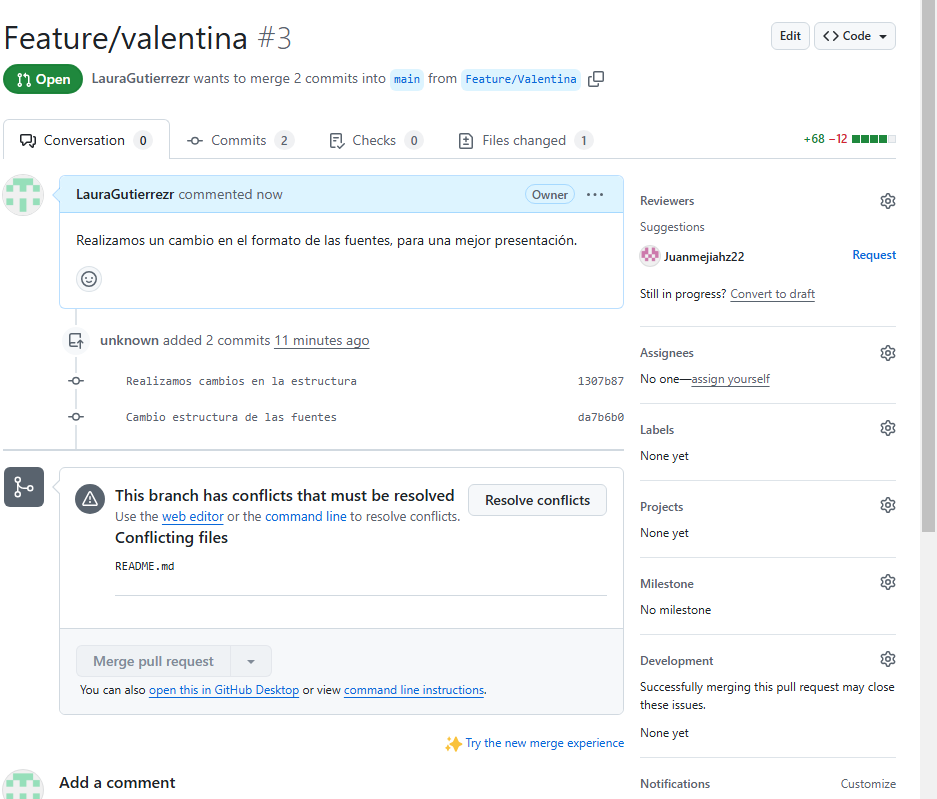
Owner es Laura y Colaborator es Juan. Usamos el repositorio creado en la cuenta de Laura.

Como primer paso, agregamos a Juan a ser colaborador en el repositorio de Owner



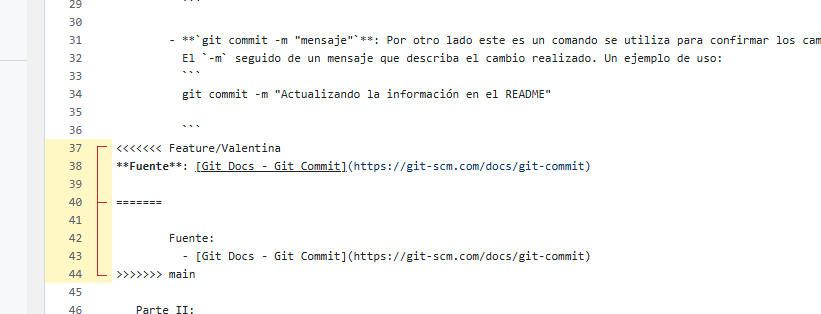
**2. Owner y Colaborador editan el archivo README.md al mismo tiempo e intentan subir los cambios al mismo tiempo. ¿Que sucedió?**

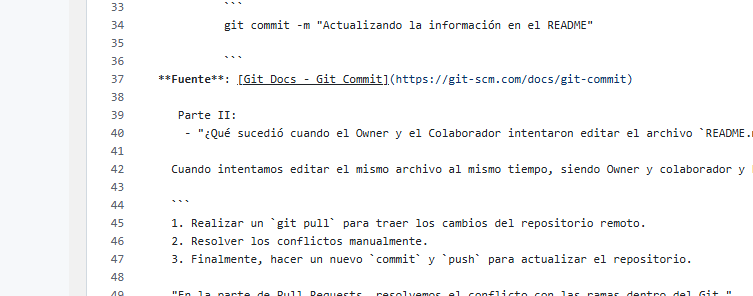
Al realizar el push en el repositorio, se presentó un conflicto de fusión (merge conflict) Cuando se producen estos conflictos, Git no sabe automáticamente qué cambios aceptar, por lo que nos muestra las secciones conflictivas de esta manera

Nos aparece un aviso en la rama

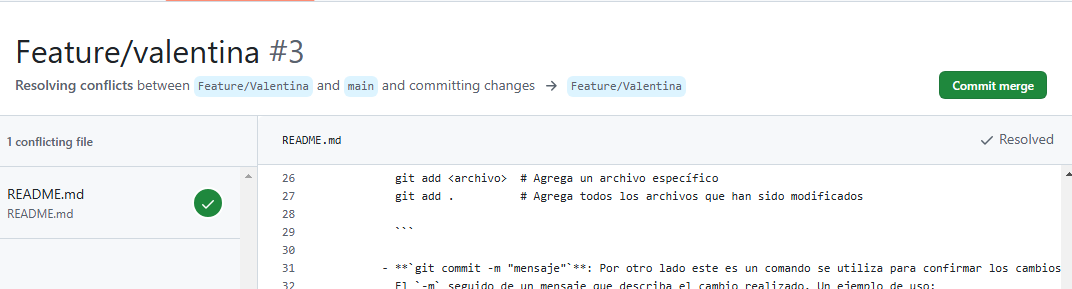
La versión de la rama Feature/Valentina (antes de =======)

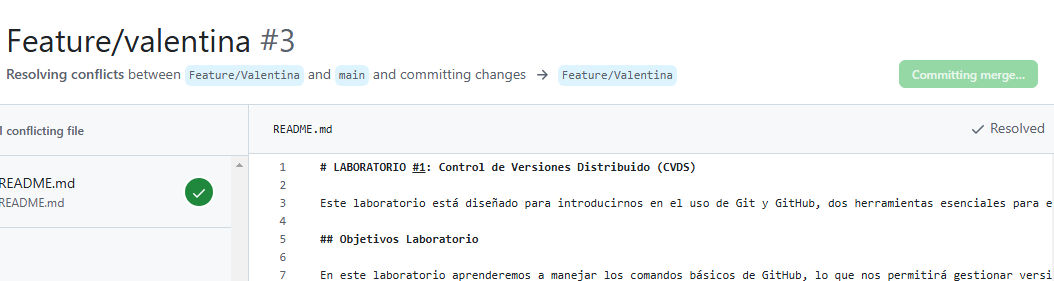
La versión de la rama main (después de =======)



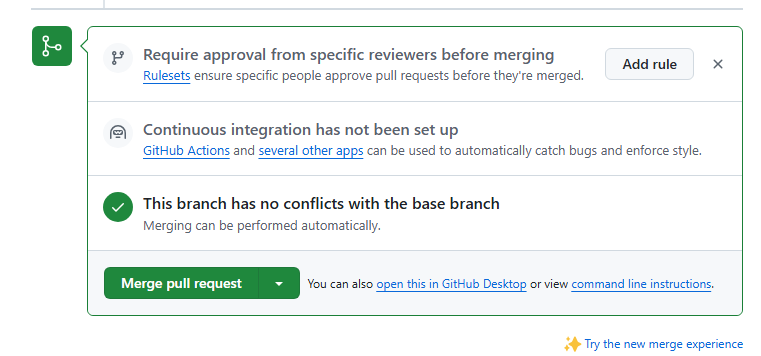
Dejamos la versión correcta en la rama, solucionando el error proporcionado por Git

Una vez realizado el cambio en el archivo, Git verifica que todo está correcto y verifica el commit del merge



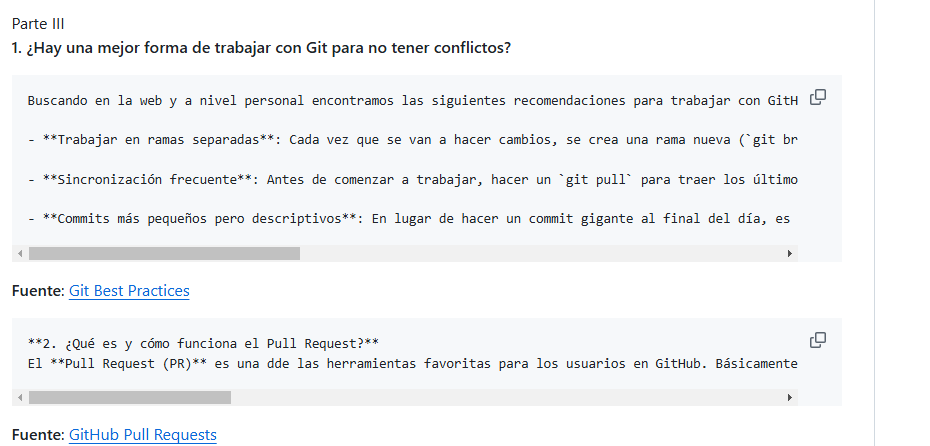


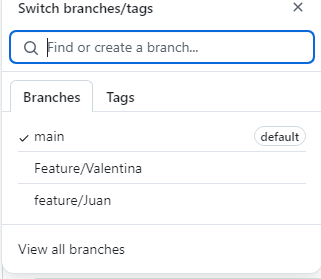
Luego de confirmar el commit debemos terminar el proceso del merge.

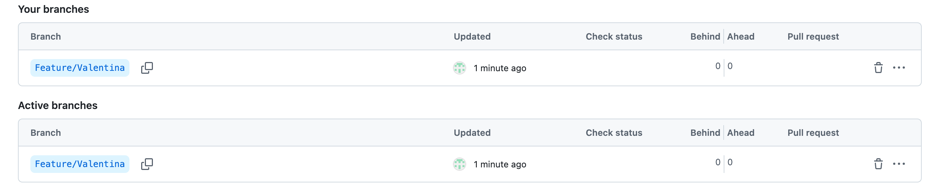


## **PARTE III (Trabajo de a parejas)**

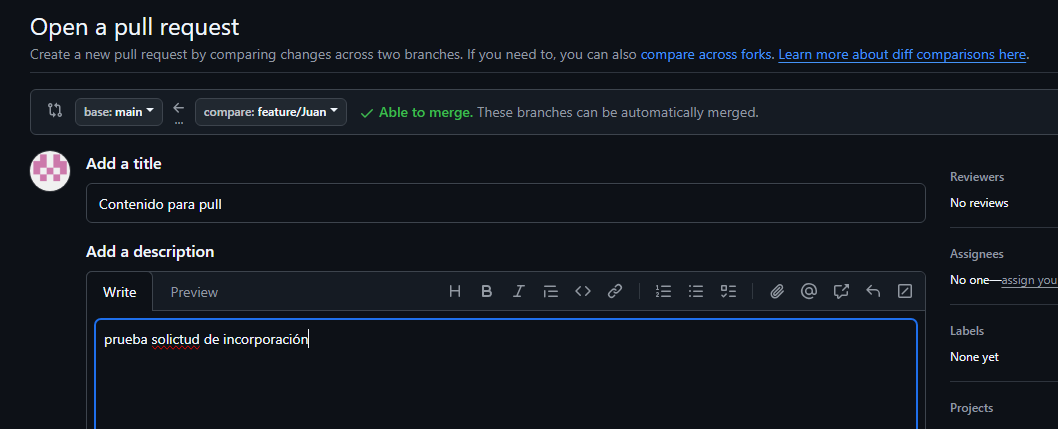
Preguntas:

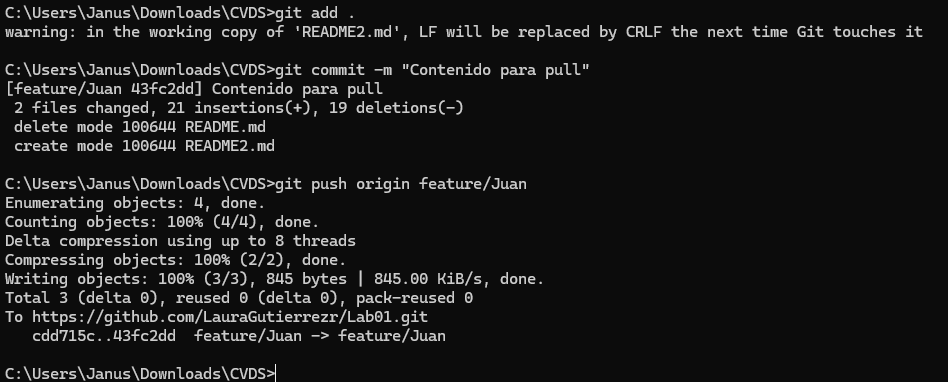
Las solucionamos en el doc README

**2. Creando una rama única para cada uno**

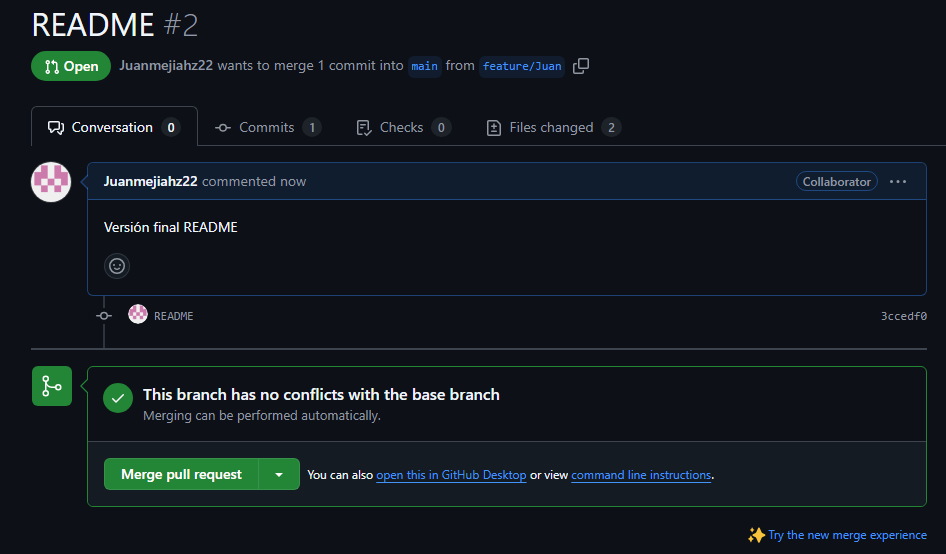


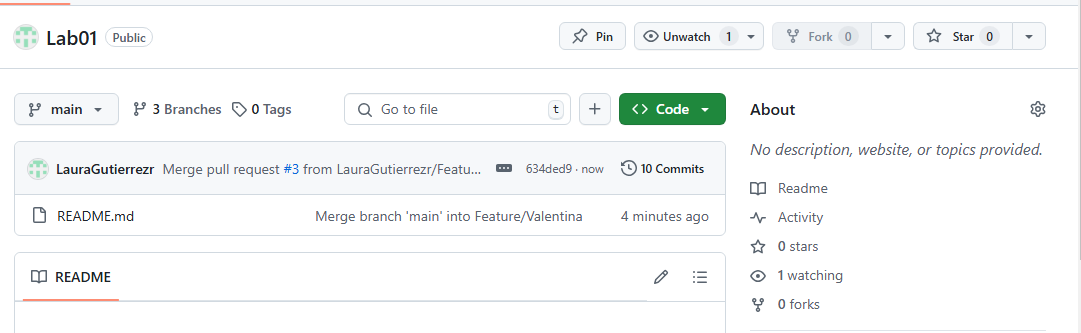
Realizamos el primer pull usando la rama feature/Juan, subimos una de las versiones finales del readme.





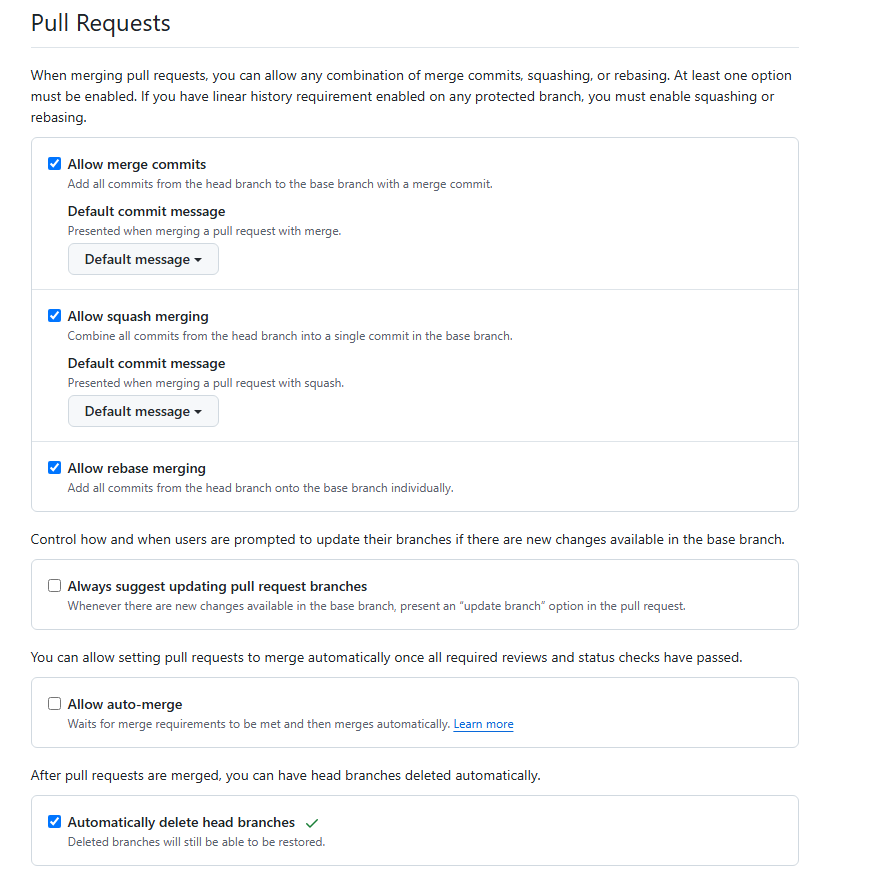
En este caso no hay problemas de versiones por lo que no se habían hecho cambios previamente, por lo que github nos verifica que todo está bien.



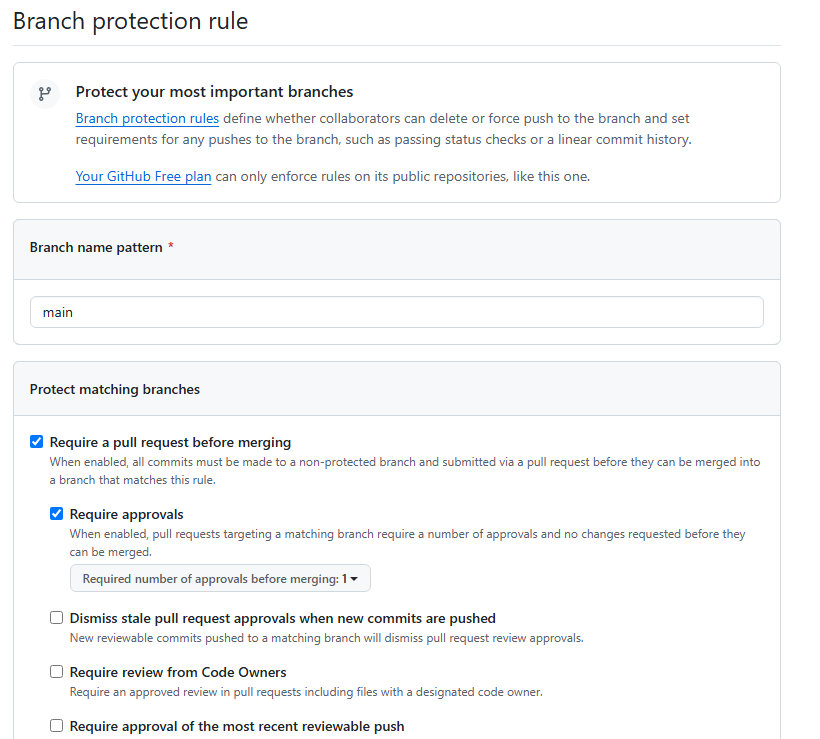
Usando la rama de Feature/Laura subimos la versión final del README**.**

**ÚLTIMO PASO**

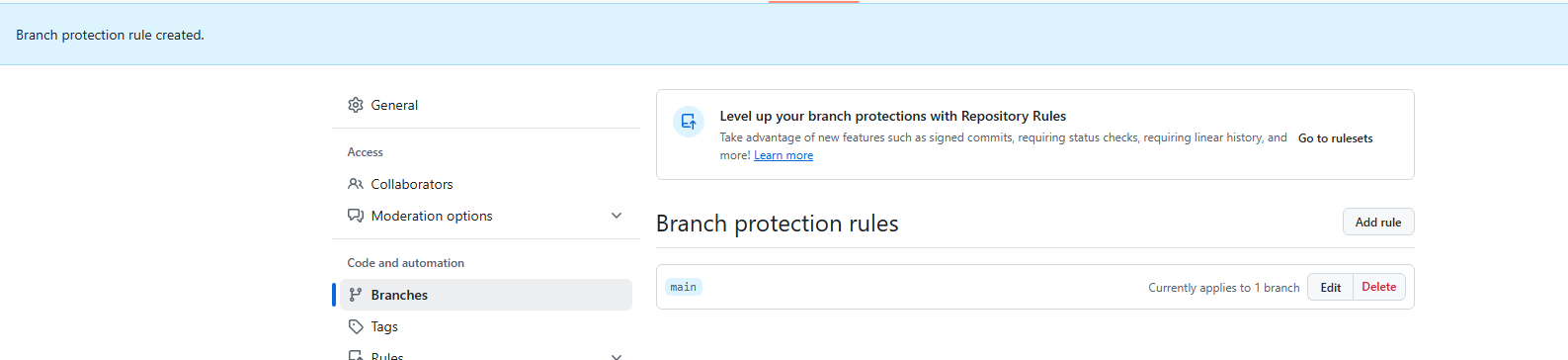
Realizamos el paso a paso para borrar las ramas al hacer los pull Request en configuración.



Especificamos la rama y indicamos las opciones necesarias



Creamos y nos damos cuenta de que todo haya salido bien.



**BIBLIOGRAFÍA**

* <https://aulab.es/articulos-guias-avanzadas/62/commit-en-git#:~:text=Un%20commit%20in%20Git%20representa,repository%20en%20un%20momento%20espec%C3%ADfico>.
* <https://git-scm.com/docs/git-add>