

Modified-> El archivo esta modificado de forma local

Staged-> Preparada. Está lista para ser enviada

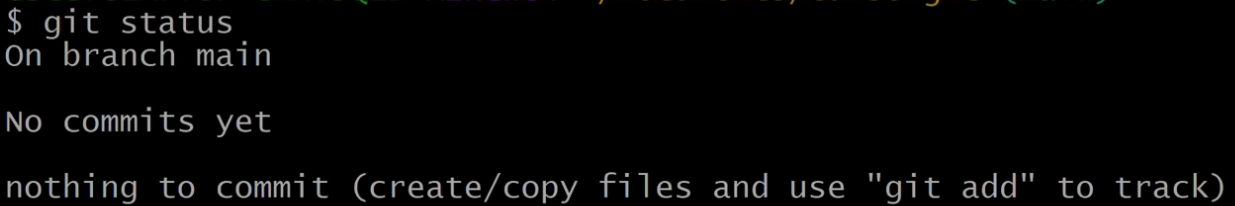
Committed-> La versión del archivo ya se encuentra en el directorio de git.

Como pasos previos, debe identificarse

Git config –global user.email “correo”

Git config –global user.name “usuario”

Para ver el estado puedo utilizar el comando “git status”

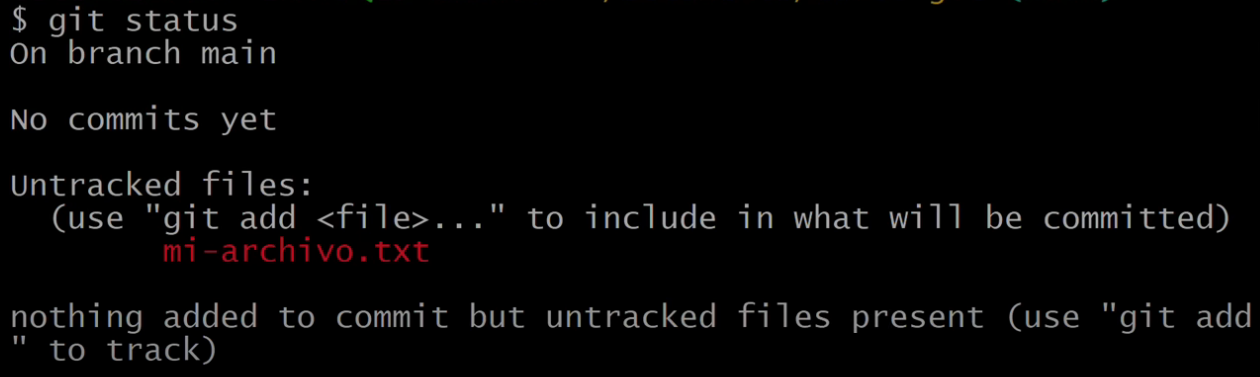


On branch main -> Me dice en que rama me encuentro. En este caso en la rama main

No commits yet -> Aún no se ha realizado ningún commit

Nothing to commit -> Me dice que no hay ningún archivo modificado, así es que no es necesario hacer un commit.

----------------------------------------------------------------------------------------------------------------------------------



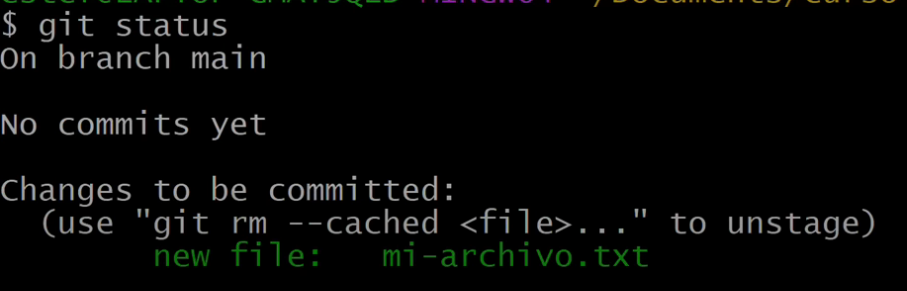
Si se modifica un archivo localmente y volvemos a realizar una consulta de estado vemos 2 diferencias:

Untracked files ->Archivos no rastreados, es decir que esta sin registros. Se muestra en rojo

La siguiente línea dice que no se ha realizado el commit, pero que hay archivos no rastreados.

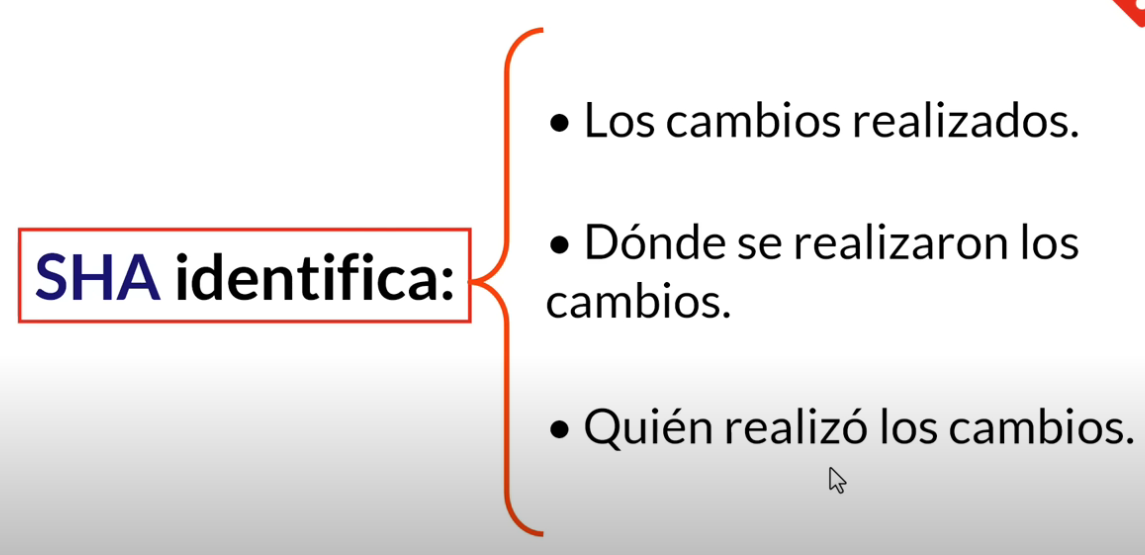
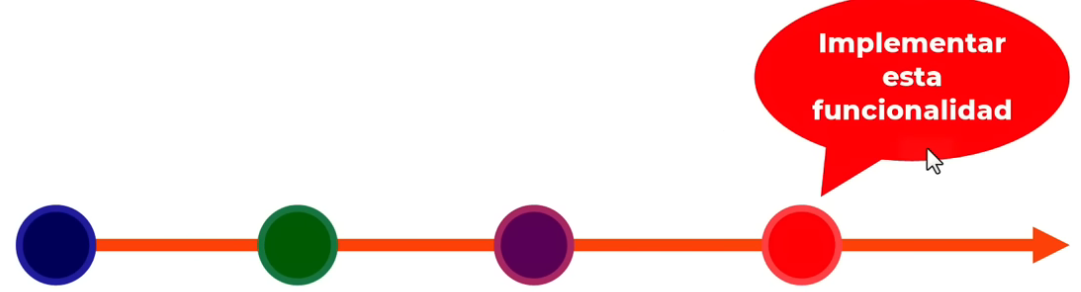
Git add “nombre del archivo” ->agrega el archivo

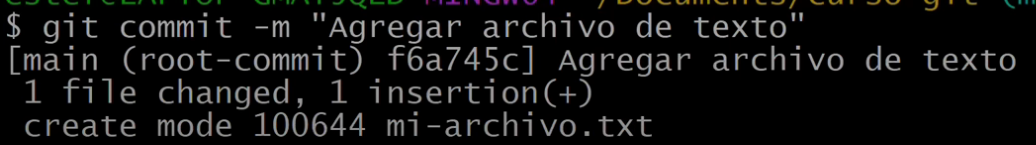
Git add . ->agrega todo lo que este en la carpeta



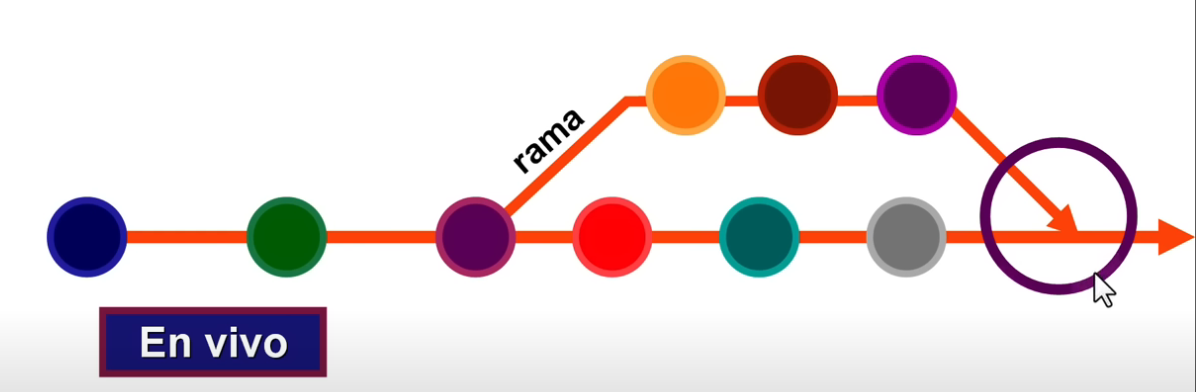
Ahora luego de realizar el “git add” y hacer una consulta de estado (git status) vemos que ya estamos en el área de preparación listos para enviar a la nube, pero también podemos retractarnos con el comando mostrado en la imagen (solo se quita del área de preparación y no borra el archivo).

El commit es un registro de lo que se ha realizado para comparar con otras versiones.





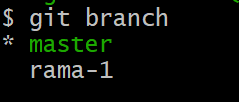
Las ramas (branch) son las versiones independientes del proyecto



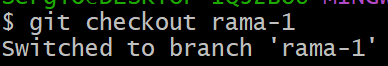
Para crear una nueva rama -> **git branch nombre-de-la-rama** (con minúsculas y sin espacios)

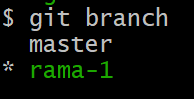
Para ver las ramas existentes -> **git branch**





Para cambiar de rama -> **git checkout nombre-de-la-rama**





Según donde yo esté ubicado es la rama que crearé, es decir que sí creo una rama estando dentro de una rama, se creará una rama más pequeña.

Para eliminar una rama -> git branch –d nombre-de-la-rama (hay que hacerlo desde otra rama)

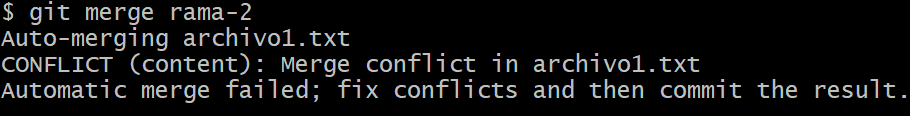
Para fusionar ramas (merge) debemos pararnos en la rama que recibirá los cambios y realizar el comando

Git merge nombre-de-la-rama-que-quiero-fusionar

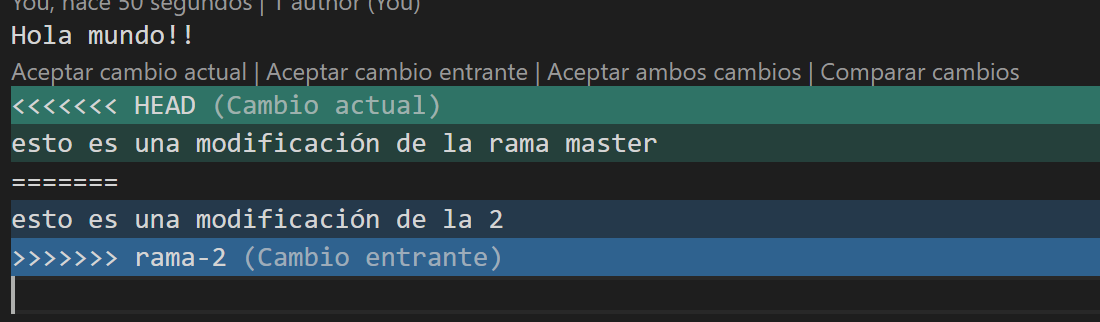
Con esto se agregarán todos los cambios de una rama a otra.

Los conflictos ocurren cuando 2 o más personas editan la misma línea.

En la siguiente imagen, podemos ver en las imágenes que se produjo un conflicto al tratar de juntar la rama-2 con la rama master y git nos pide solucionar el conflicto para luego hacer el commit correspondiente.



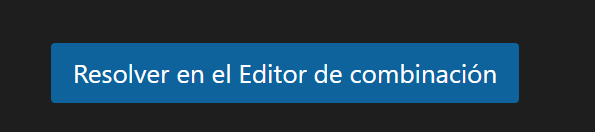
En la segunda imagen, estando en visual studio, podemos ver claramente el error, puesto que se editó la misma línea.



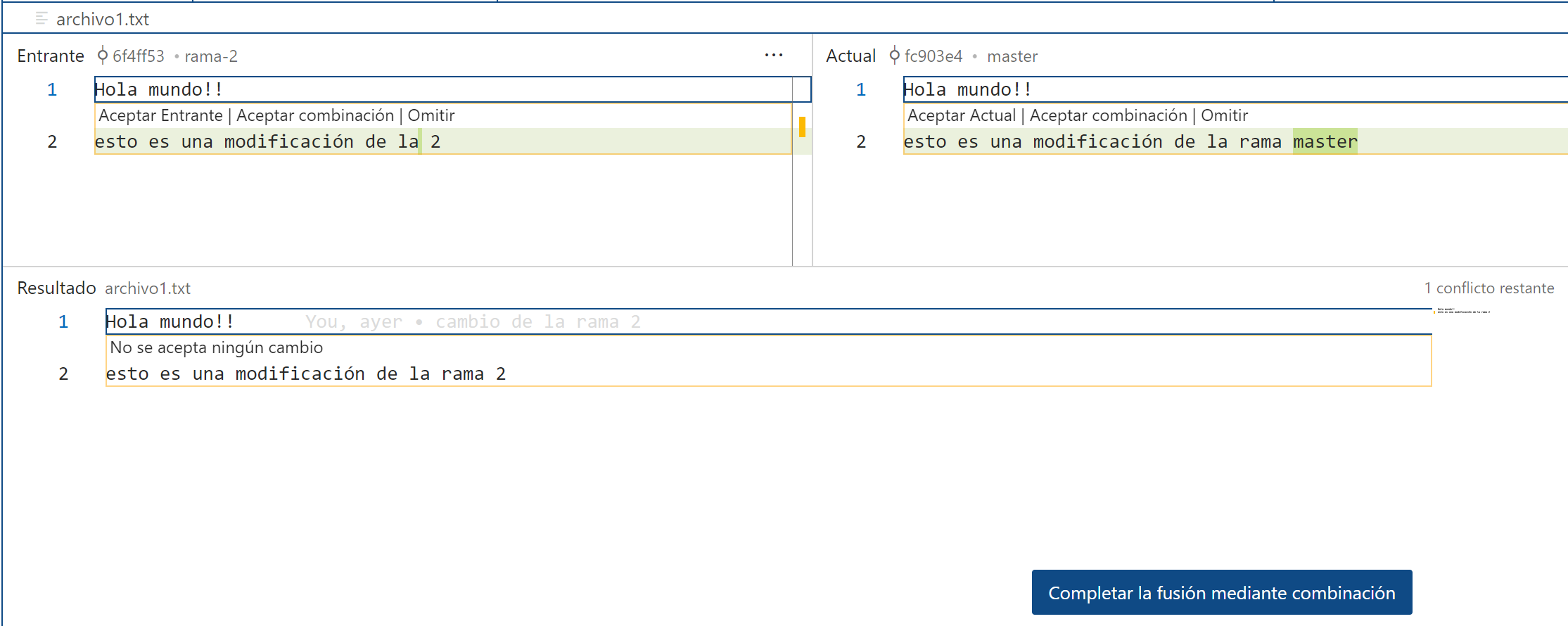
Aquí también de forma directa en el editor de texto se puede dar click en la opción que más me acomode.



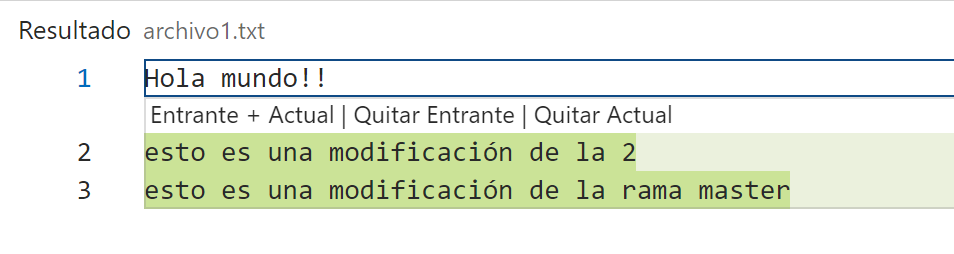
También podemos ir un poco más abajo y dar click en la siguiente opción



Donde nos aparecerá lo siguiente

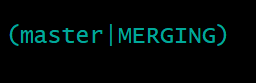


En esta pestaña puedo aceptar cualquiera de las 2 versiones y hasta incluso quedarme con ambas en paralelo



También podría hacer una combinación e ir mirando cómo va quedando todo. Al final se debe dar click en “Completar la fusión mediante combinación”

Luego de eso podemos ver un cambio en nuestro git

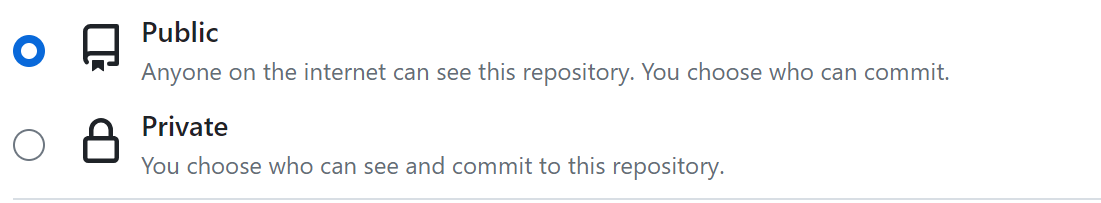


Para finalizar debemos ingresar el comando -> git merge –continue

**Git-Hub**

Es un hosting que nos permite poder almacenar diferentes proyectos en la nube para poder trabajar desde cualquier parte con uno o más colaboradores.

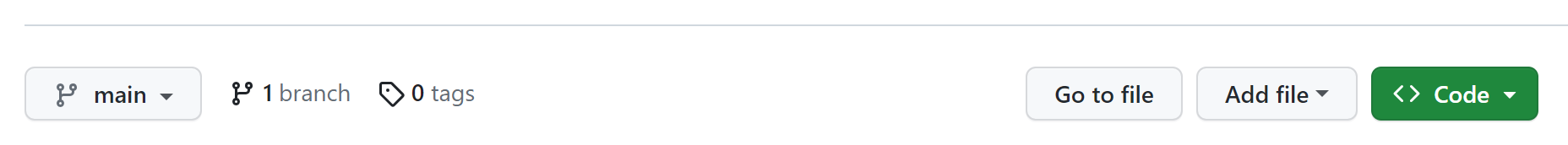
Al crear un repositorio en Git-Hub, lo primero es decidir si será público o privado



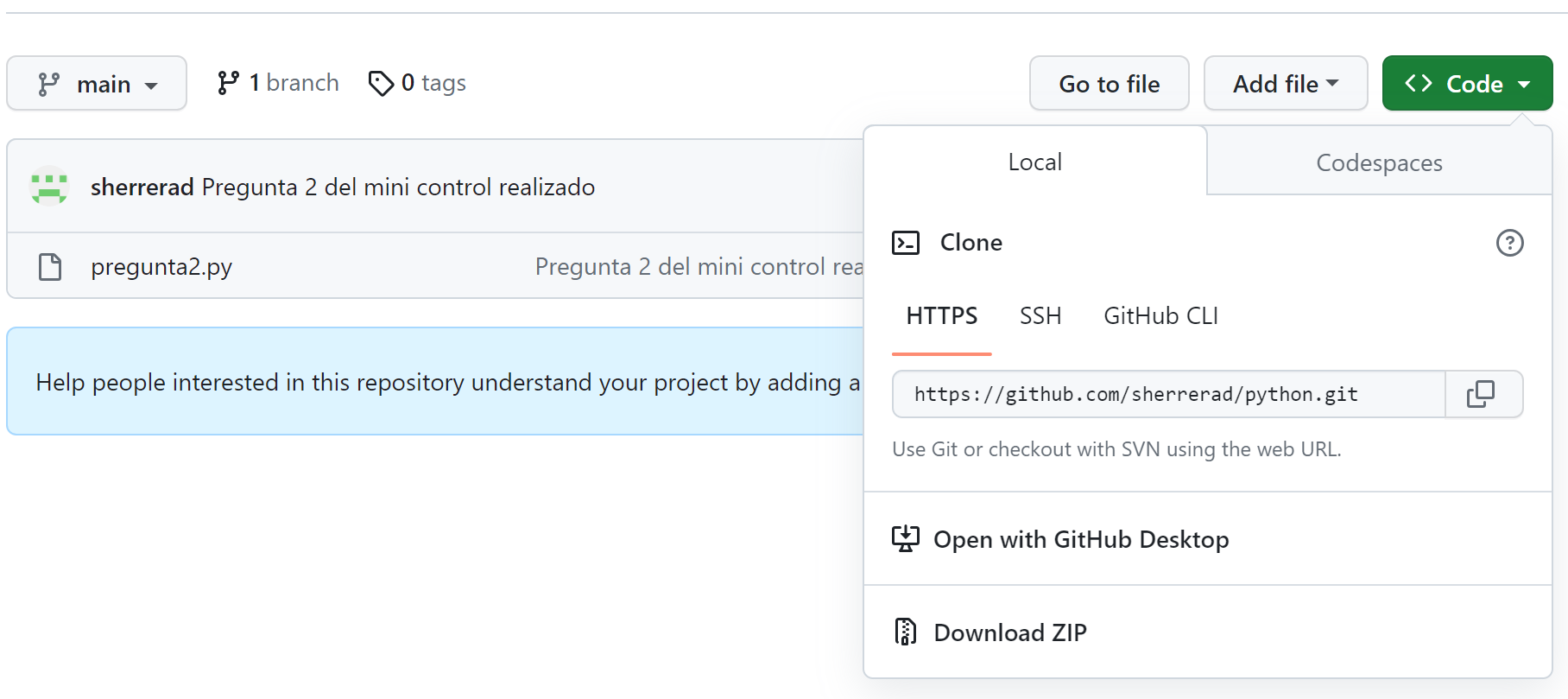
Si le damos a público, cualquiera podrá descargarlo, pero solo los usuarios seleccionados podrán hacer commit. De ser privado, solo quien creó el repositorio y quienes escojas, podrán verlo.

Por defecto git-hub trabaja con la rama principal como “main”, ya que es el nuevo nombre por defecto (antes era master).

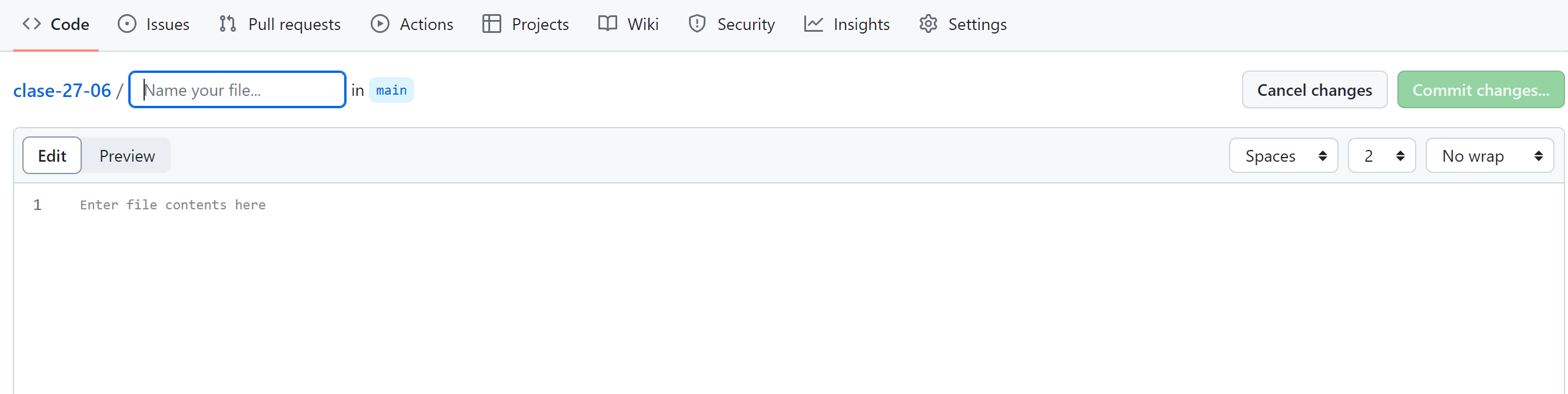
En Add file puedo agregar un archivo directamente, sin necesidad de git



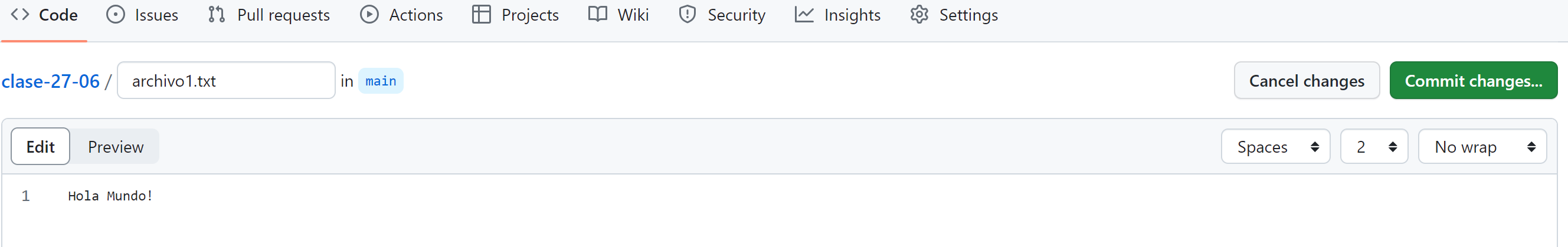
Y en Code puedo ver los datos para clonar o descargar como un .zip

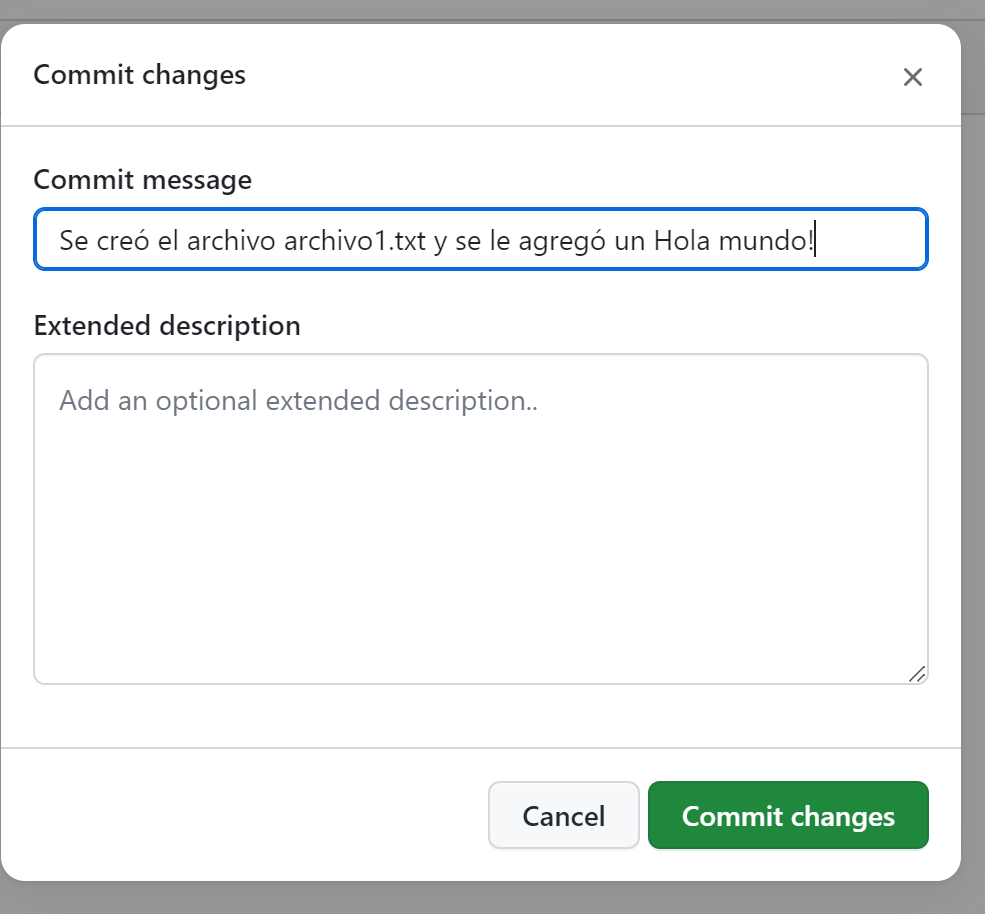


En git-hub puedo crear directamente un archivo de forma mucho más grafica

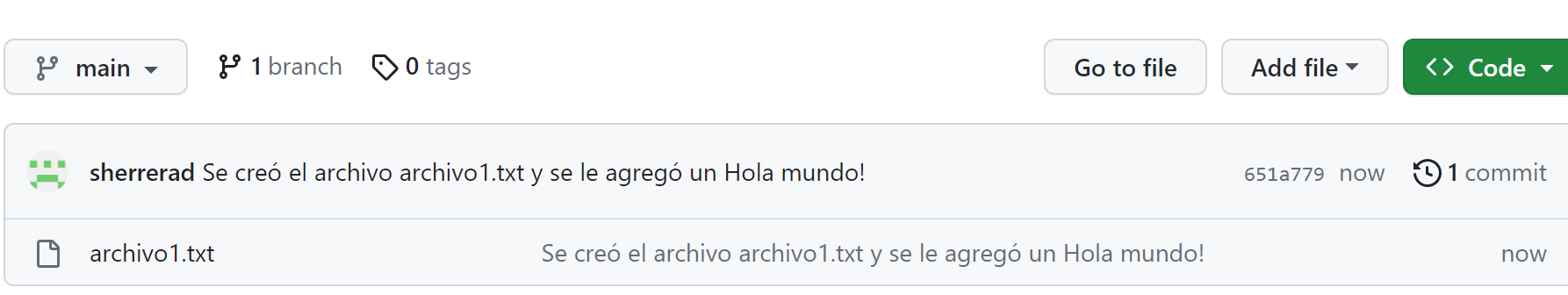


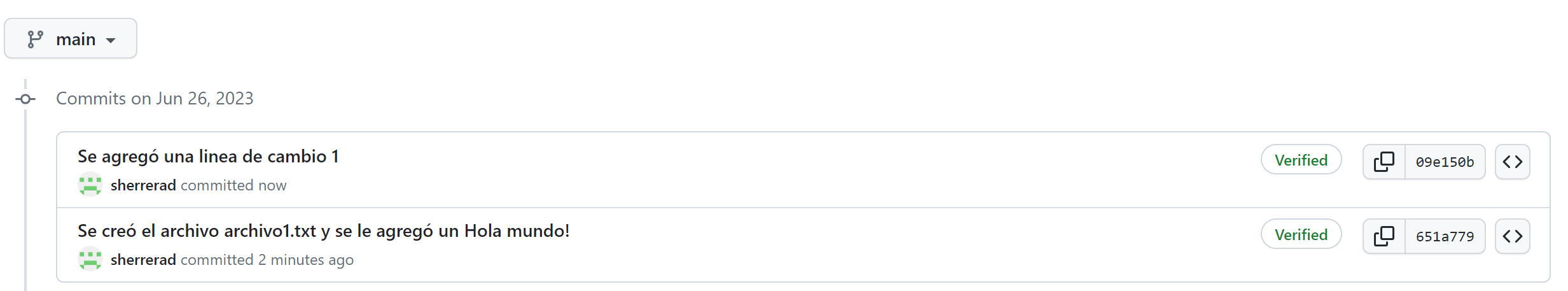
Se debe dar un nombre y editar. Podemos ver como por defecto estamos en la rama main. Luego debemos dar click en el recuadro verde

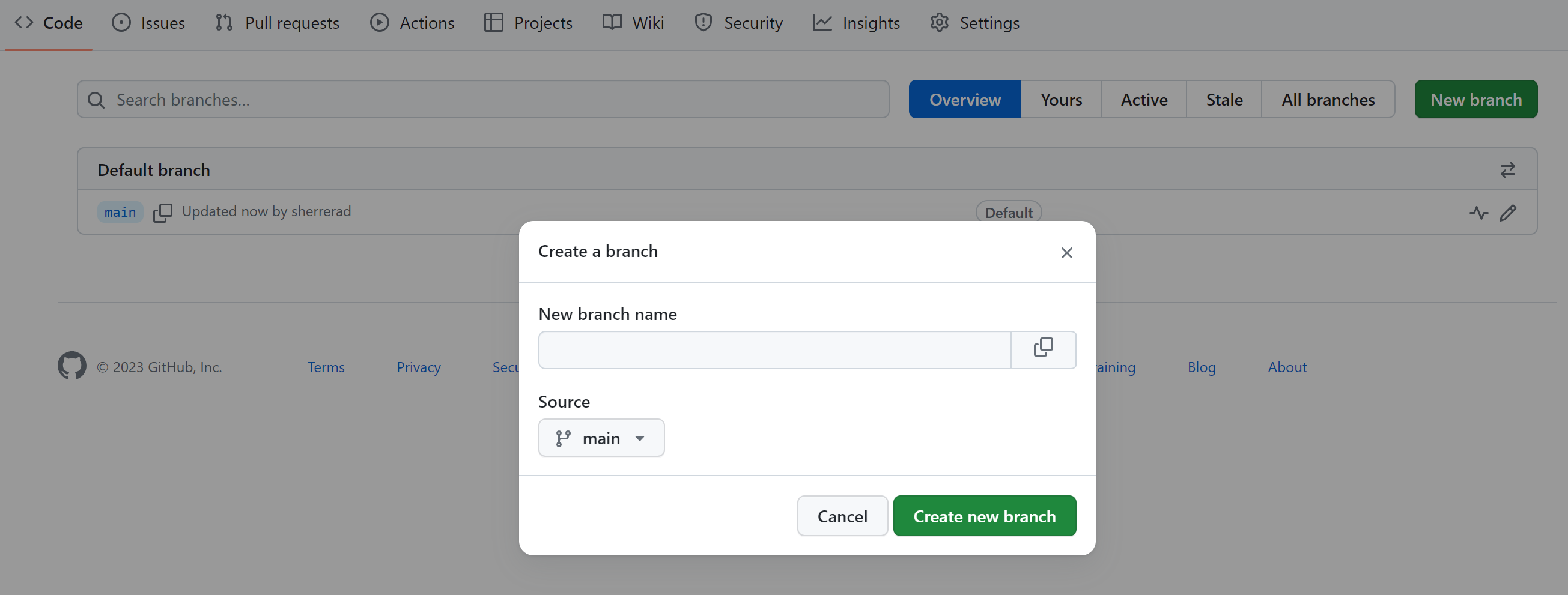




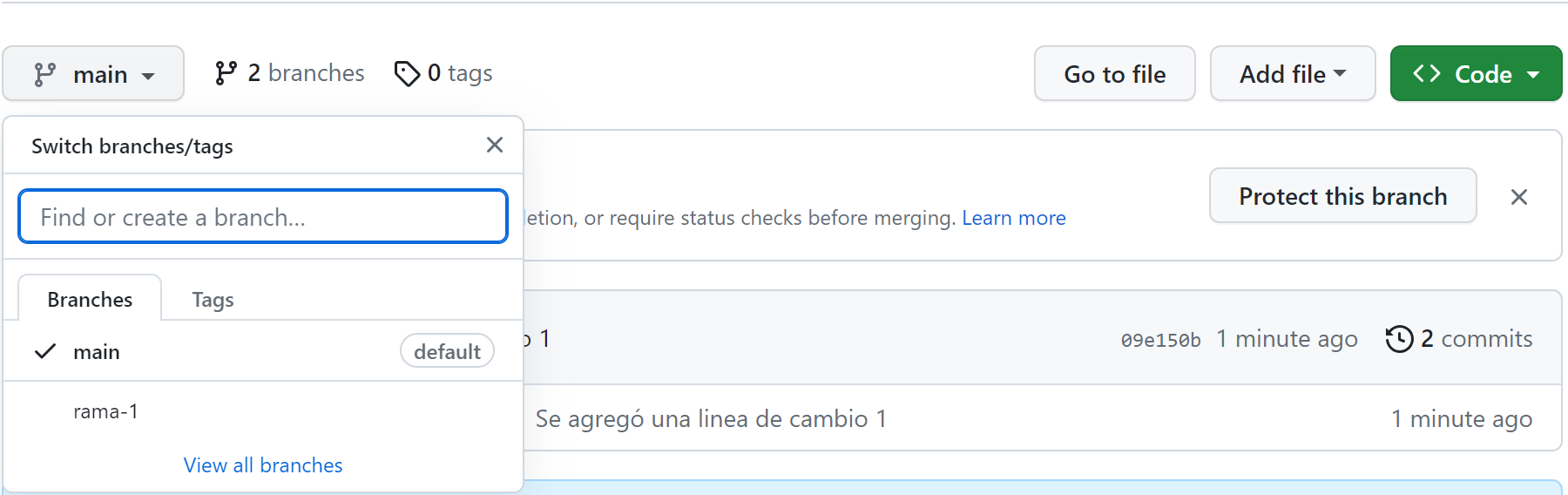
Una vez creado, podríamos editarlo, agregar ramas, entre otras funcionalidades de git



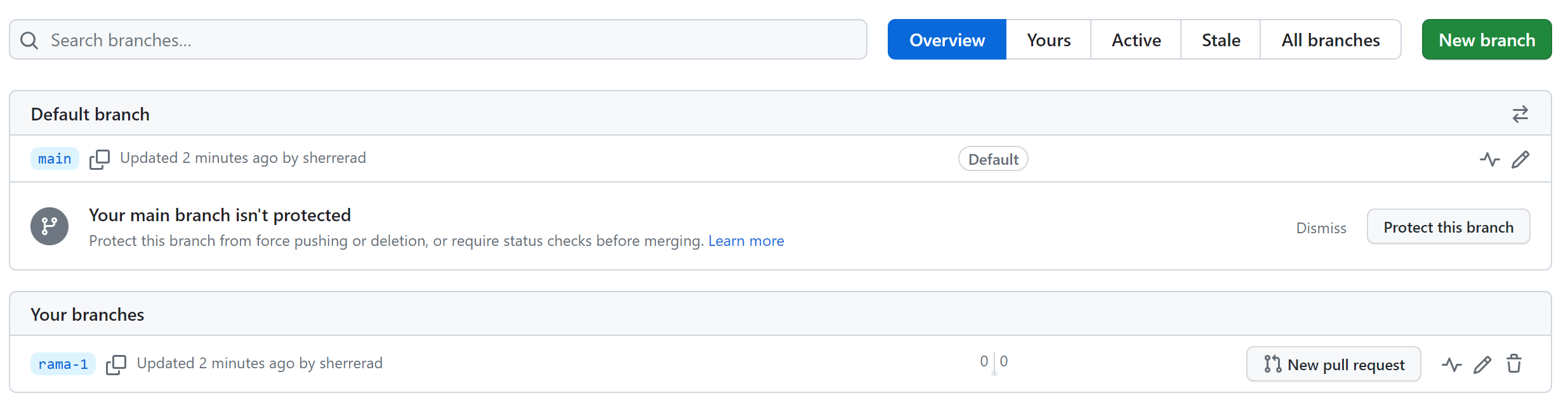




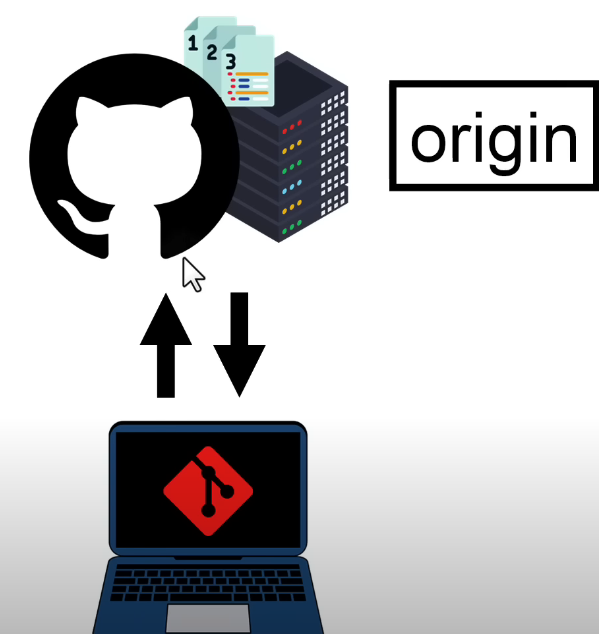
Al crear una nueva rama, ocurré lo mismo que se vio en git, es decir que la nueva rama será una copia exacta de la versión actual que se tenga.



Y asi puedo crear más ramas, editarlas, cambiarme o eliminar

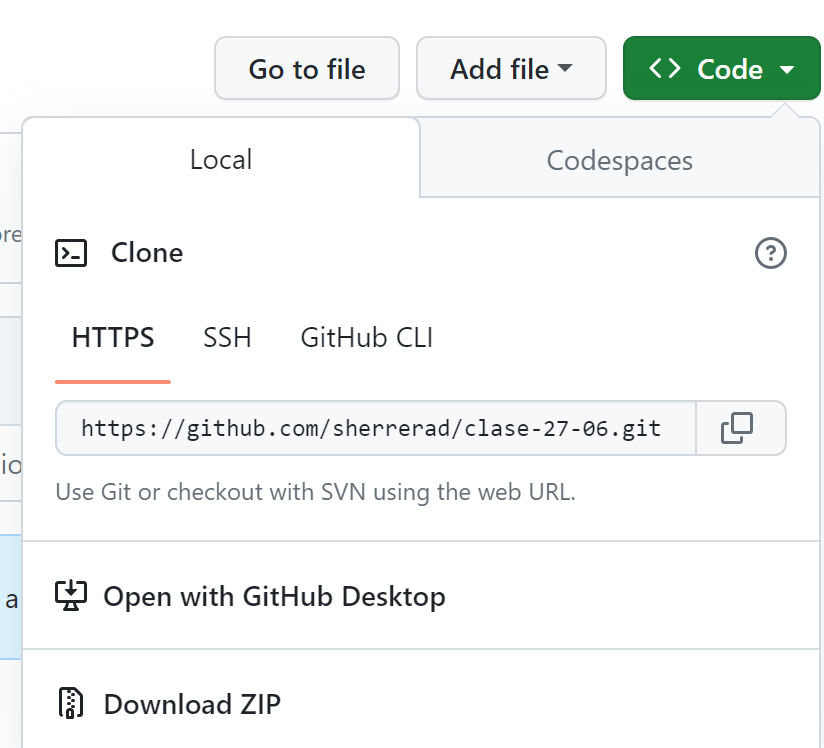


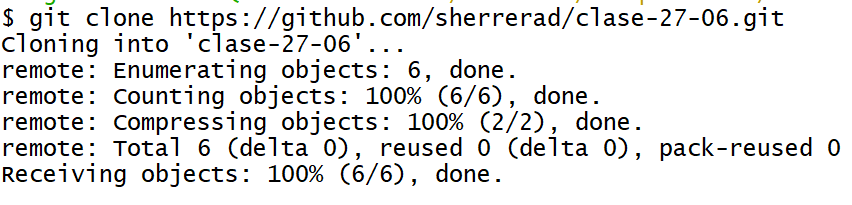
Al clonar un repositorio, realizaremos una copia en nuestro pc (copia local). Al repositorio remoto le llamaremos origin, pero solo es un nombre que le daremos de forma local y es solo un estándar, es decir que no necesariamente debe ser así. Solo servirá para referirse al repositorio remoto.



Para clonar debemos indicar el código -> **git clone “repositorio”**

El repositorio lo obtenemos directamente desde git-hub

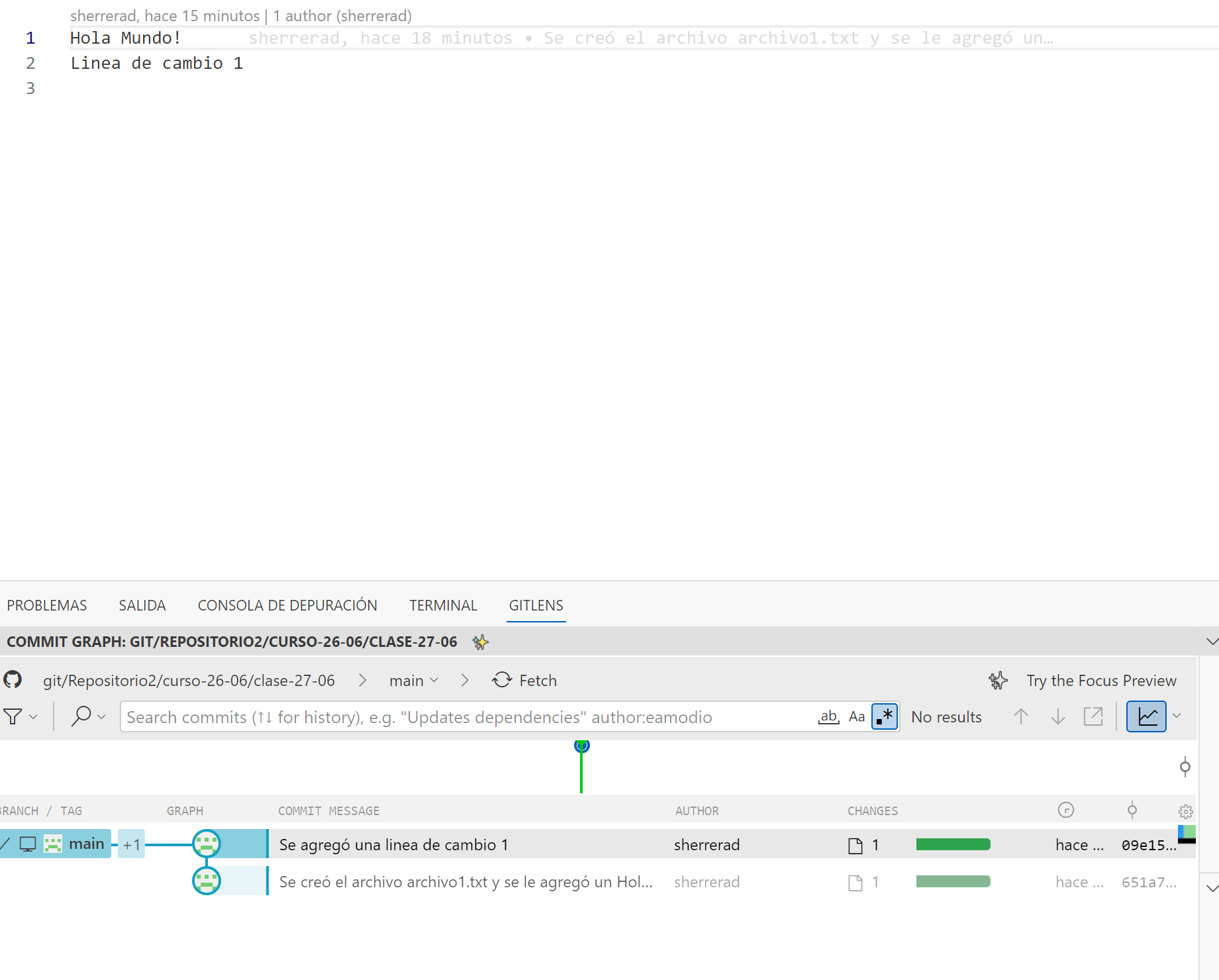
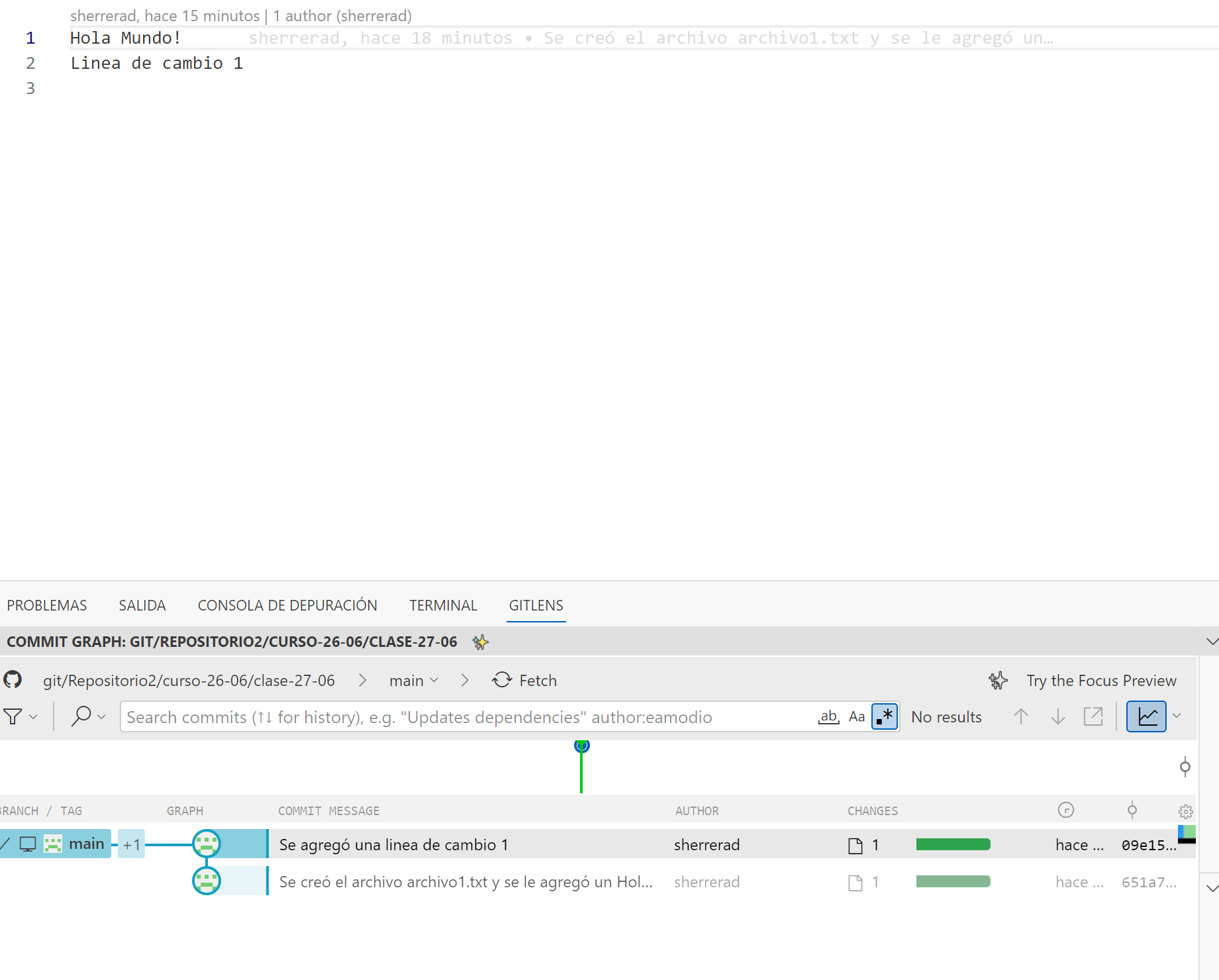




Se creará la carpeta, por lo que debemos entrar

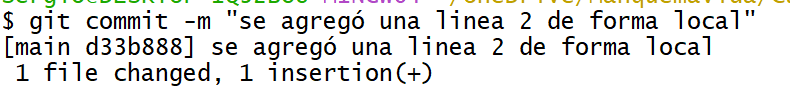


Si vamos a visual studio y abrimos la carpeta, podemos ver que se copió todo lo que teníamos en git-hub

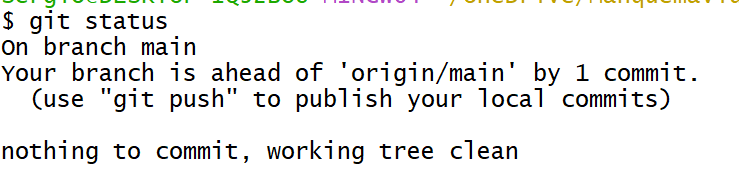


De esta manera ya podemos trabajar de forma local y cuando tengamos todos los cambios realizados y seguros, podemos enviar todo al repositorio remoto

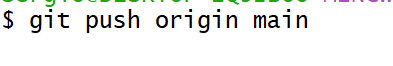
Realizamos un cambio, luego git add y luego un commit, pero todo de forma local



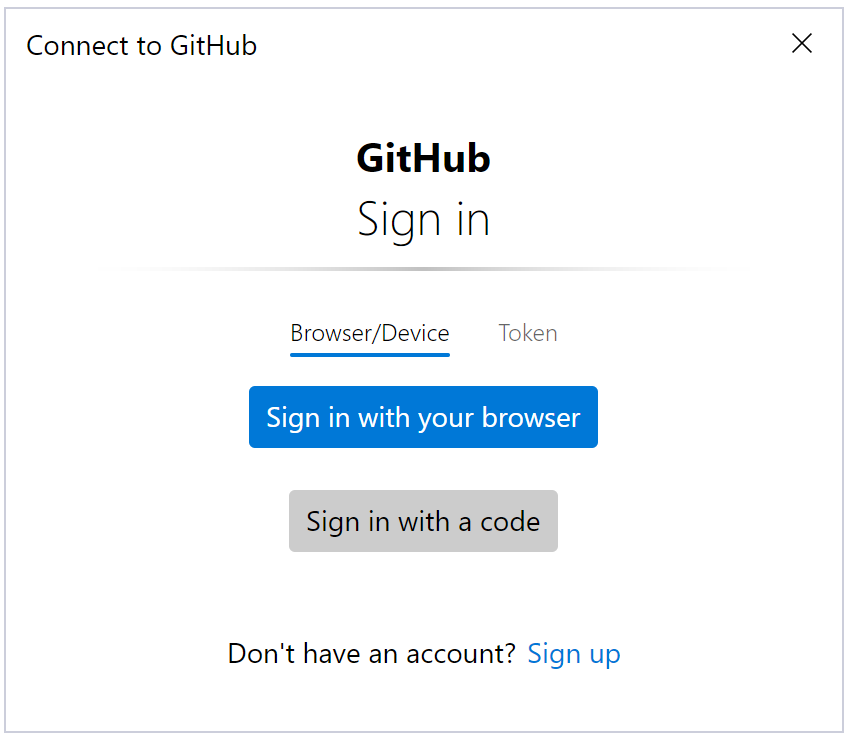
Al realizar una revisión de estado, veremos que se nos informa que realizamos cambios respecto de la versión de origen y que se recomienda subir los cambios con un git push



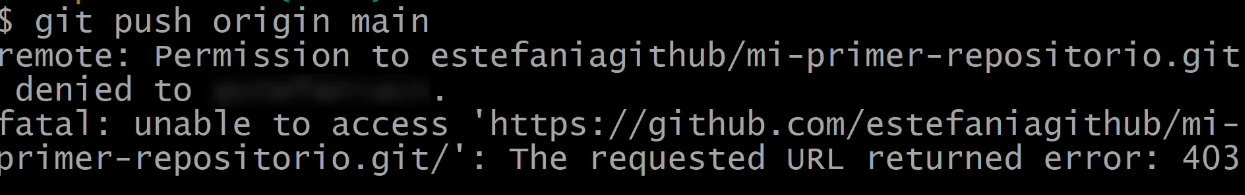
Para enviar los cambios lo hacemos con git push



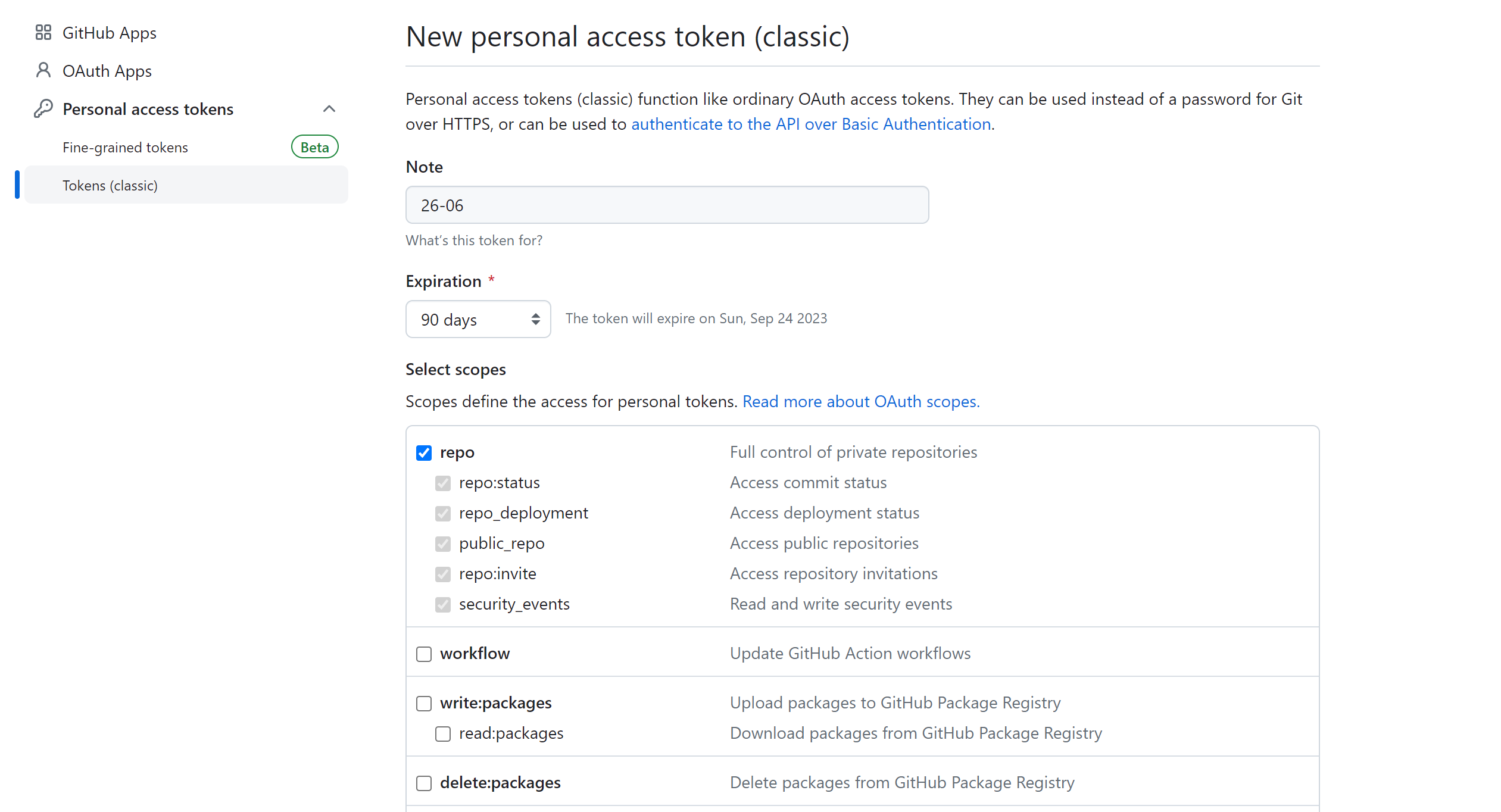
Se desplegará un cuadro para autenticarse

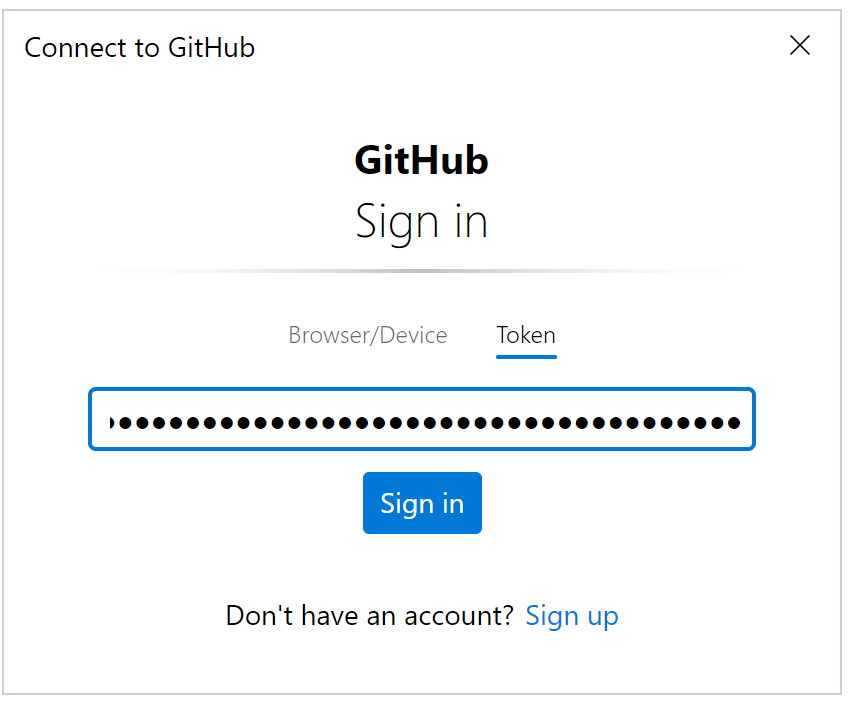


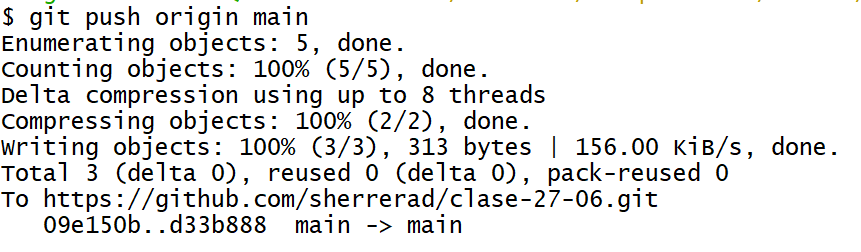
Si es que estas accediendo a un repositorio que no te ha asignado los permisos necesarios, te aparecerá el siguiente mensaje:



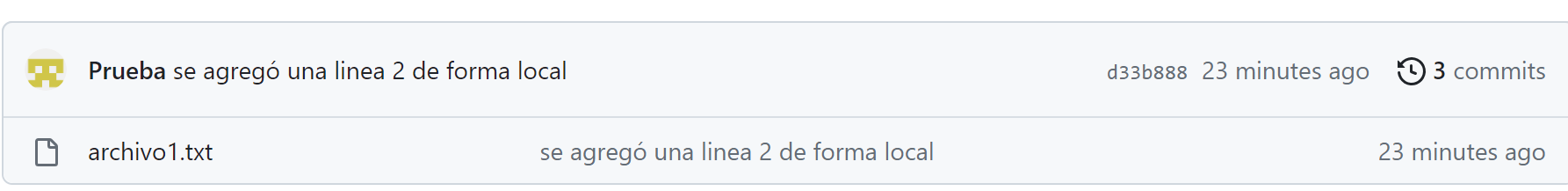
Se puede ingresar usuarios al repositorio o asignar un token, donde daremos permisos por cierta cantidad de tiempo







Si revisamos nuestro git-hub, veremos que los cambios se han subido



Ahora si queremos agregar cambios que se hayan realizado de forma remota a nuestra versión local, lo hacemos con **git pull origin main (**donde main es el nombre de la rama**)**

