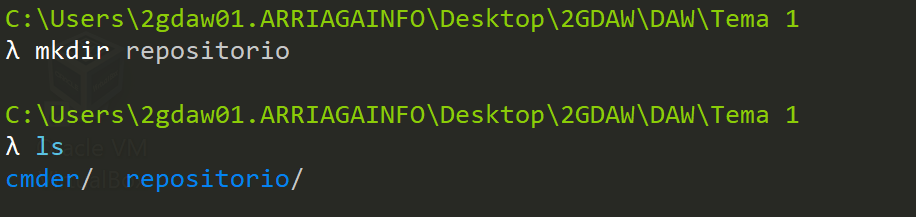
igTrabajo 1 - Introducción a Git

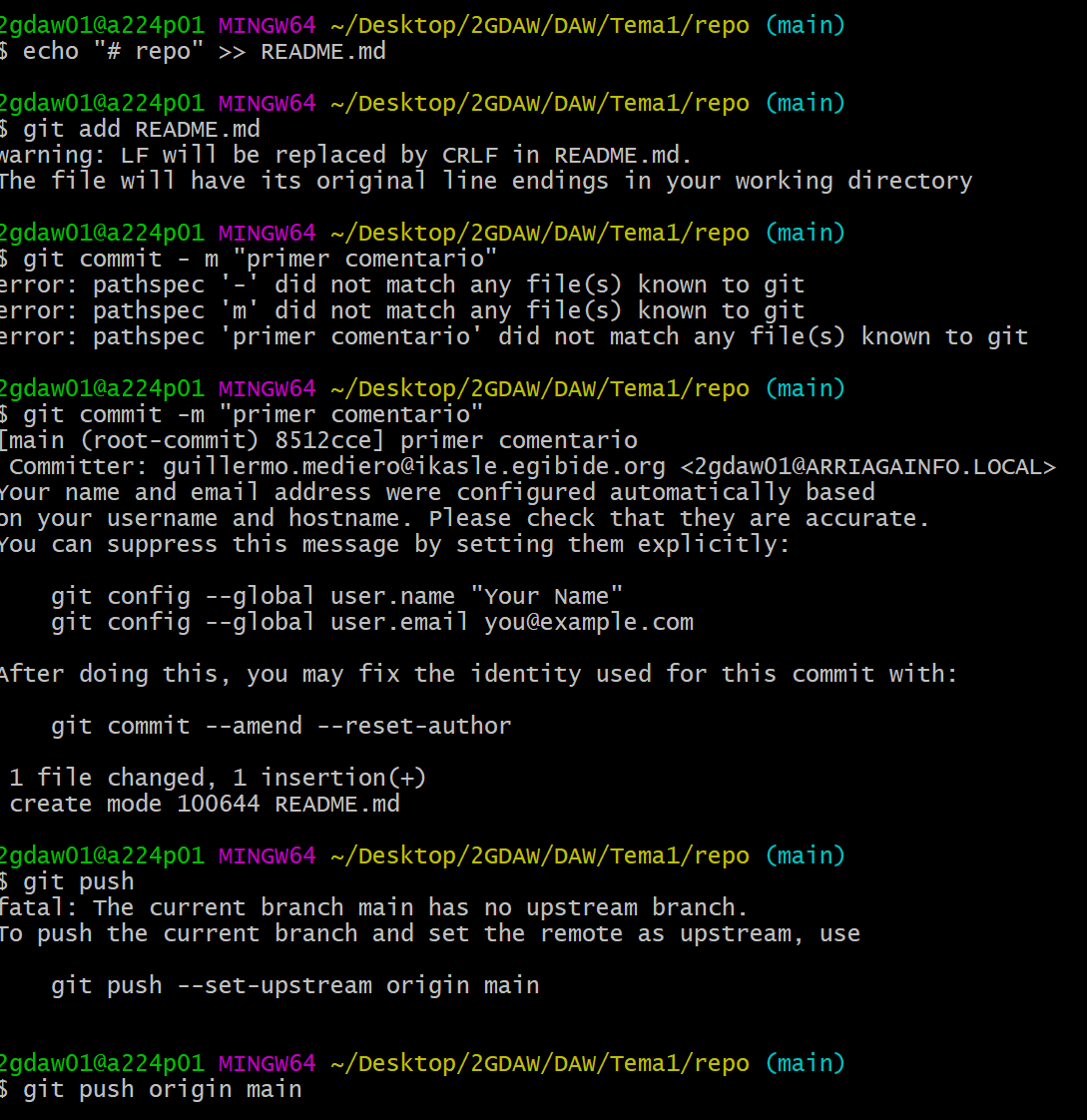
Tareas

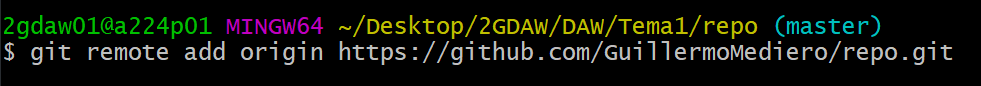
1. INICIAMOS NUESTRO REPOSITORIO REMOTO DESDE LOCAL

* Creamos una carpeta en donde nosotros no apetezca



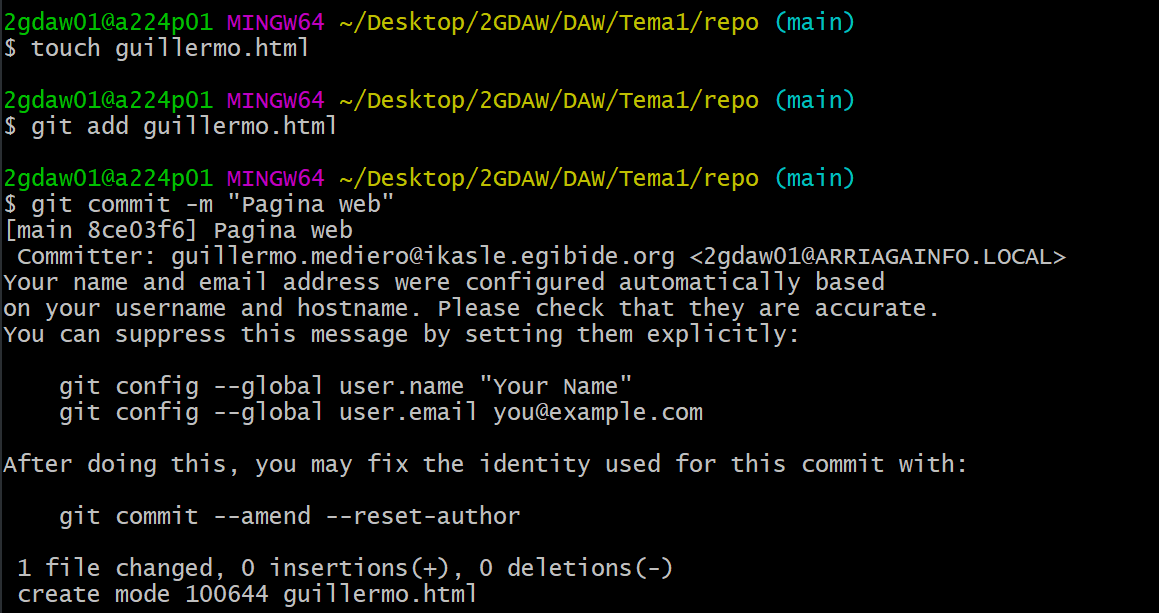
* Una vez creada utilizaremos el comando git init y añadimos el archivo README.md y le hacemos un comit



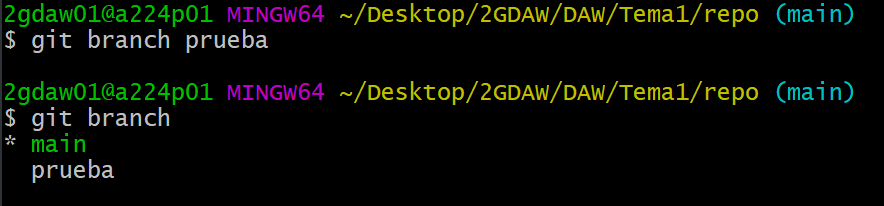


2- CREA UNA RAMA CON OTRA VERSIÓN DEL PROYECTO

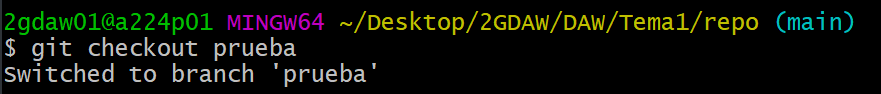
* Nos vamos a crear un archivo llamado Guillermo.html dentro del repositorio maestro con el comando git add, una vez creado le hacemos un commit



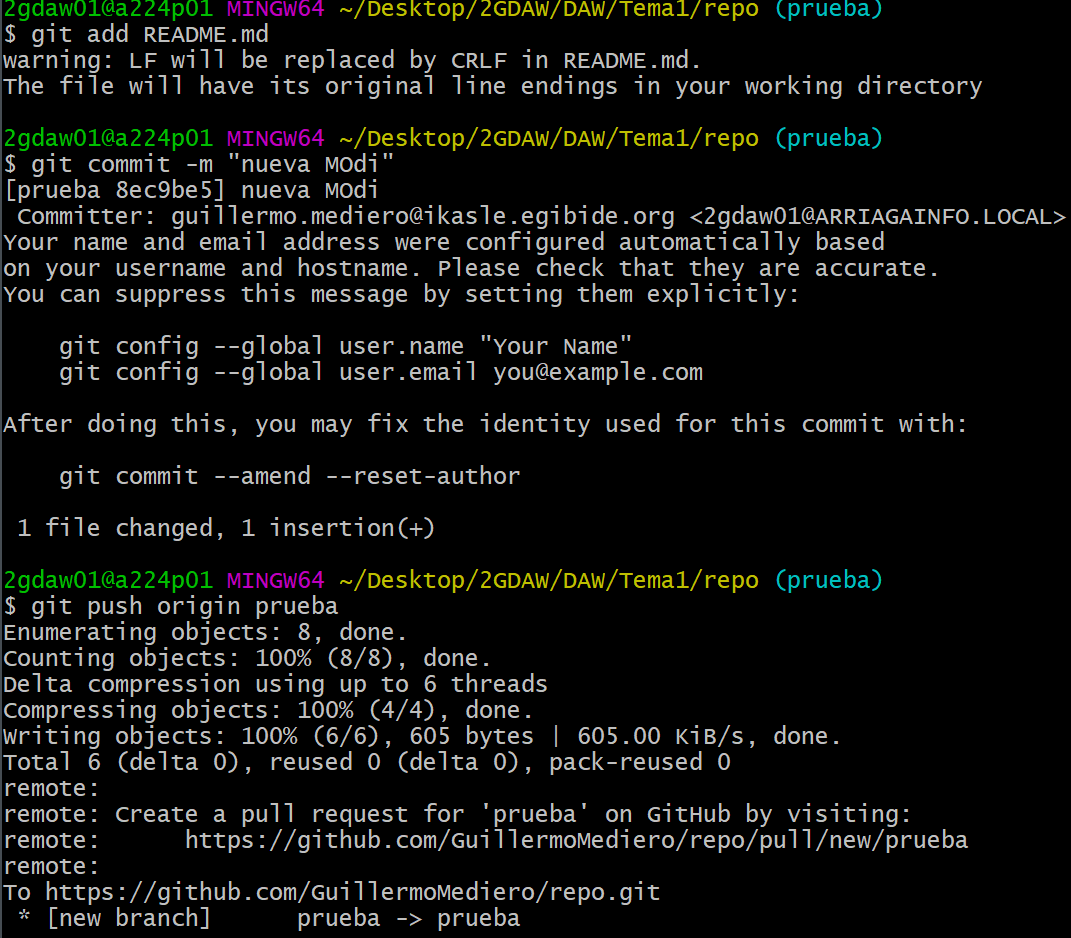
* Una vez creado vamos a generar una rama auxiliar con e comando git branch “nombre de la rama”.



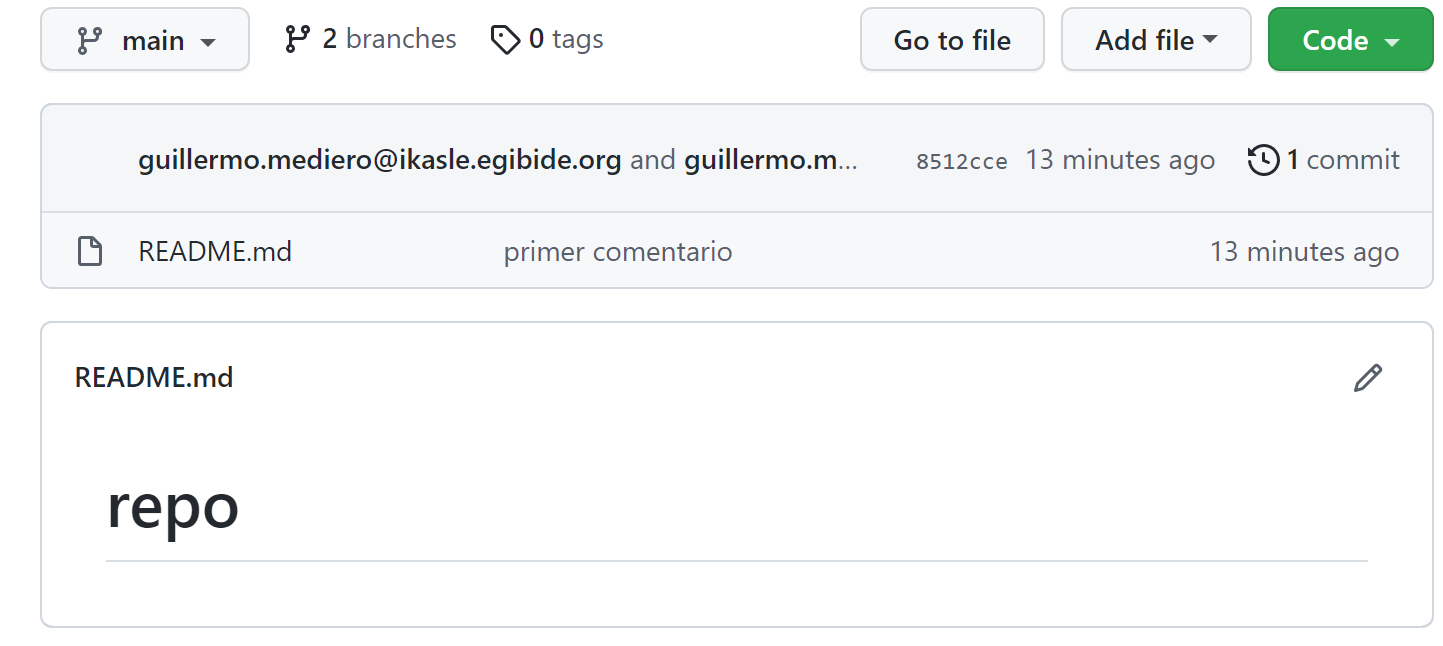
* Una vez creado la rama auxiliar nos movemos a ella con git checkout “nombre de la rama”

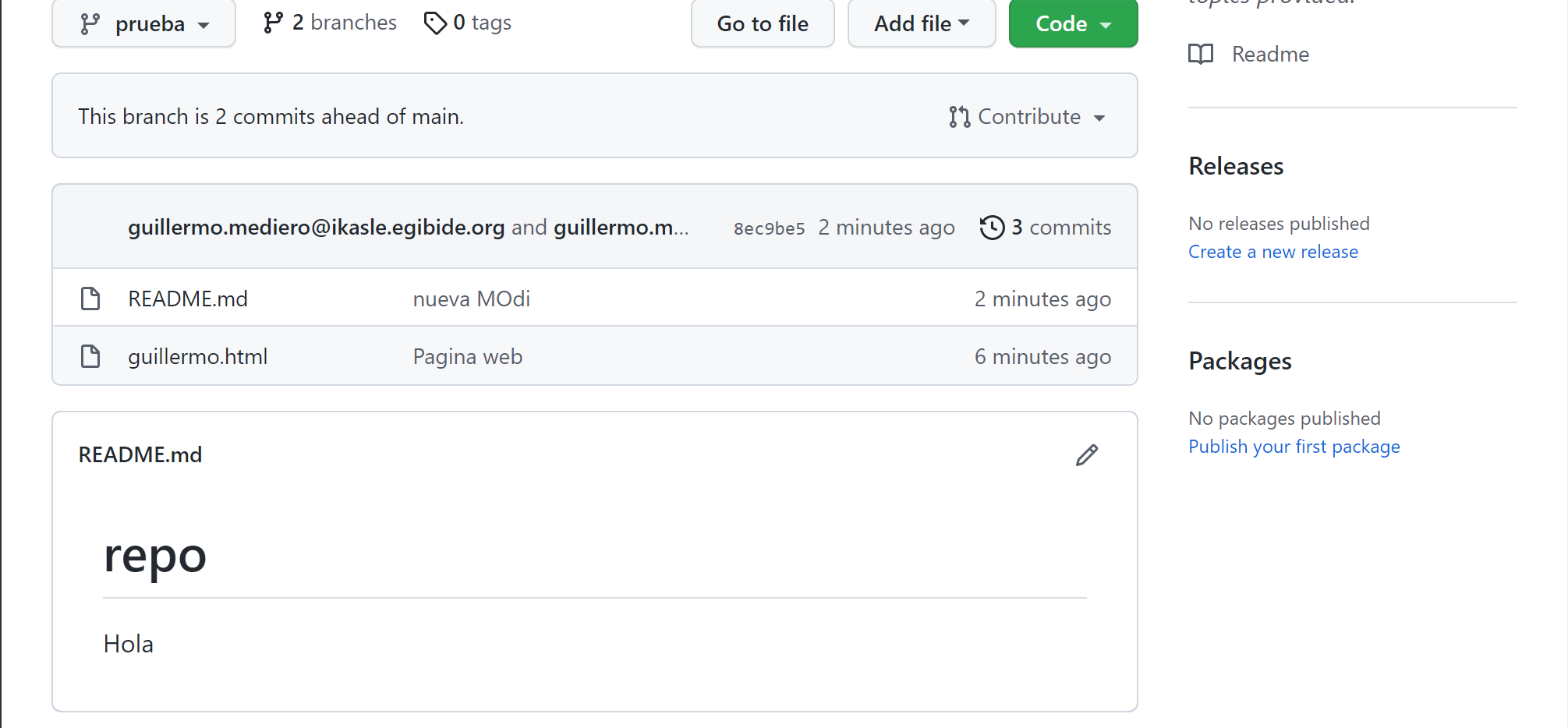


* Modificamos el archivo generado anteriormente y lo añadimos a la rama



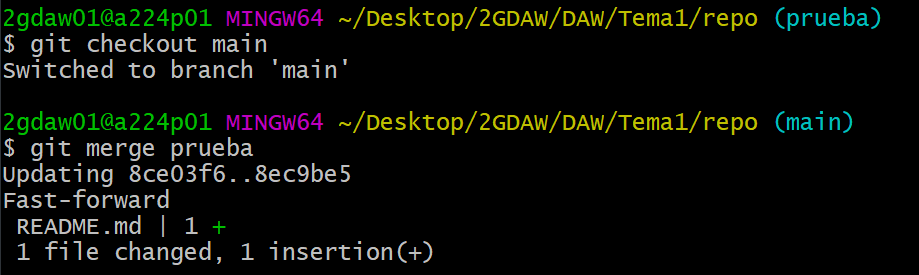
* Ahora lo comprobamos en GitHub





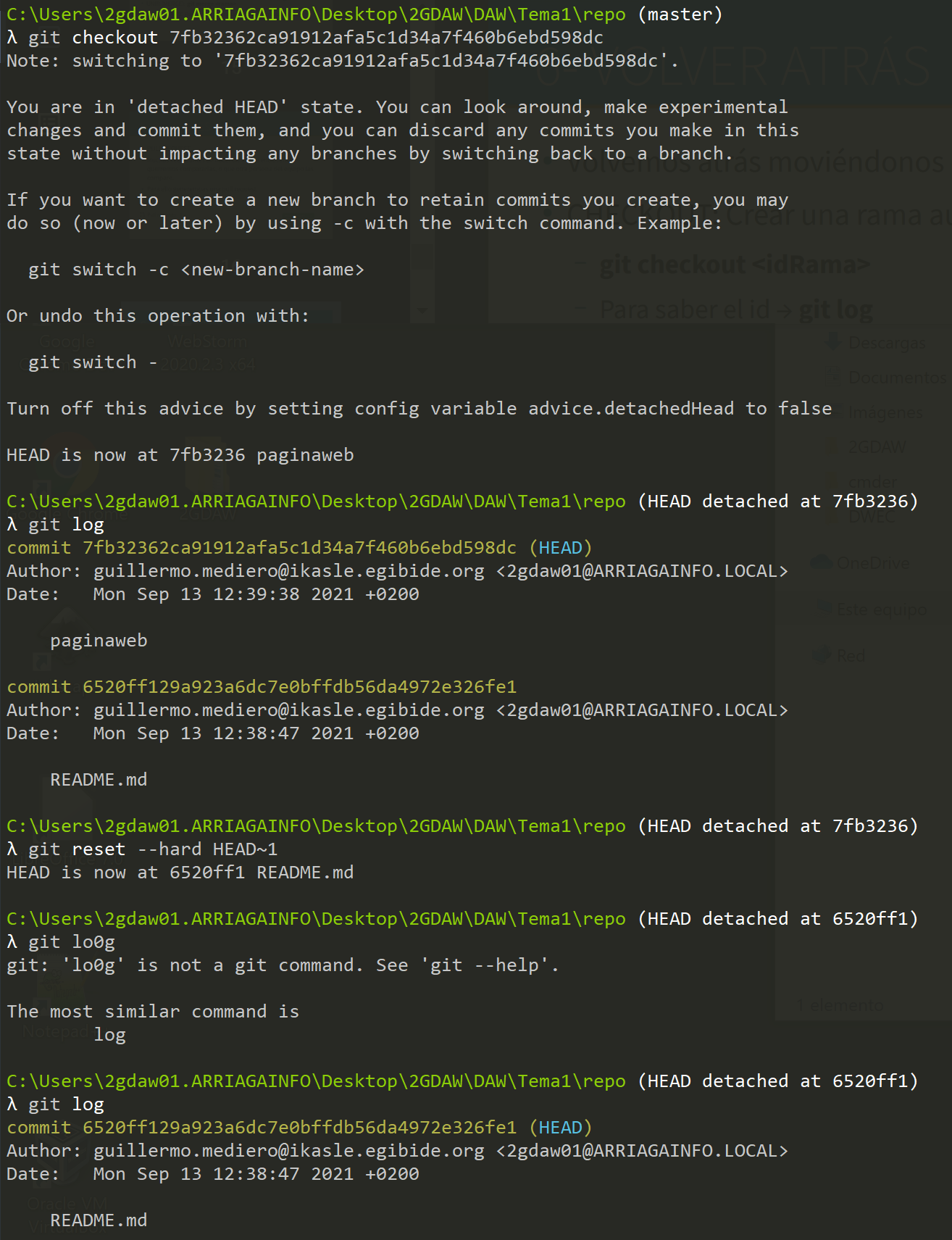
¿Cómo veo qué ramas existen? Usando el comando git branch ¿Cómo se cuál es la rama activa? La rama la cual ponga Head

3- UNE LA RAMA DEL EJERCICIO 2 A LA RAMA MASTER

-Vamos a unir la rama auxiliar con la master con el comando git merge “rama la cual queremos unir”

4- ¡LA HEMOS LIADO! VUELVE HACIA ATRÁS EN LOS COMMITS

* Si por cosas de la vida hemos hecho commit y parte de nuestro trabajo se nos ha perdido no tenemos que preocuparnos, siempre podemos volver atras en algun punto de commit, para ello podemos usar tanto el git checkout “id de la Rama”(para saber la id de un commit con git log) o git reset --hard HEAD~1



* ¿Qué diferencia hay entre una forma y la otra?

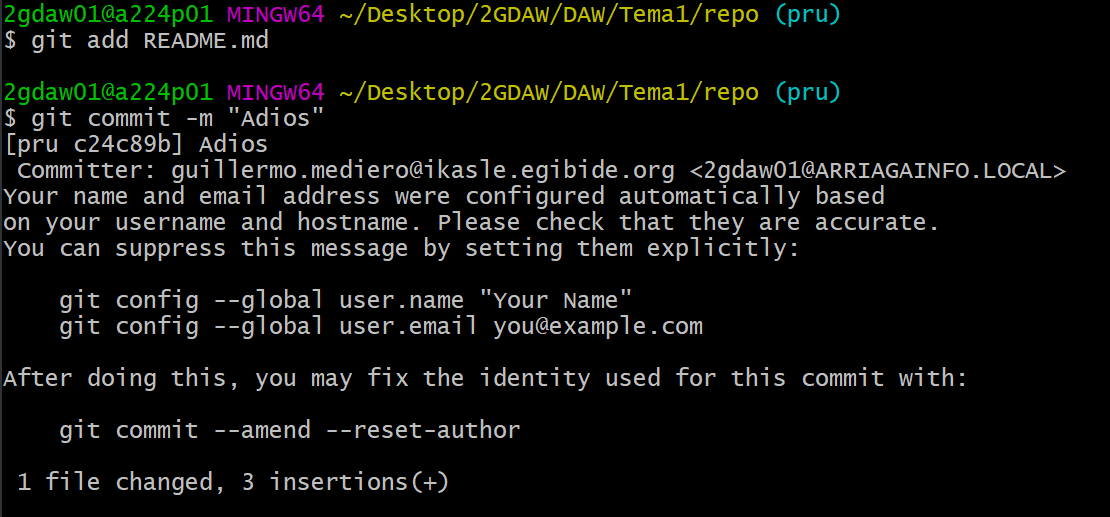
El comando git checkout “id de la rama” vuelve para atrás sin borrar lo commits posteriores, en cambio el git reset --hard HEAD~1 borra un commit

5- UNA RAMA DE UNA RAMA

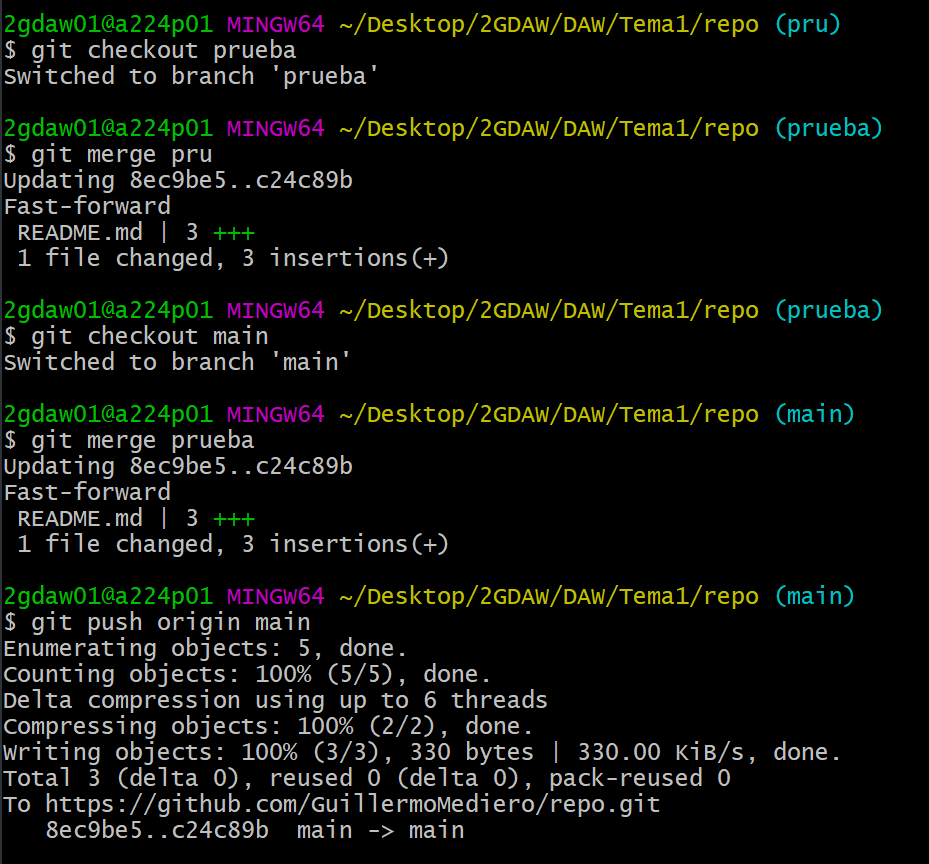
* En esta fase vamos a crear una ramas dentro de una rama o subrama, para ello vamos a crear una rama dentro de la rama de antes



* Una vez listo vamos a modificar un archivo y haremos el add y el commit .

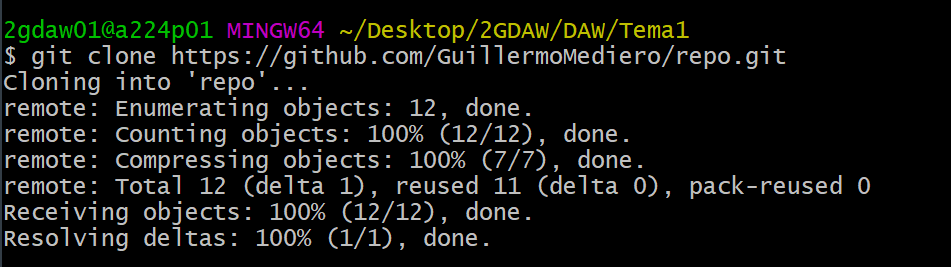


* Listo el commit vamos a realizar git merge hasta llegar a la main



6- HEMOS CAMBIADO DE EQUIPO, QUEREMOS IMPORTAR EL PROYECTO

* Después de todo lo visto ahora queremos que otro equipo tenga nuestro proyecto como lo tenemos aquí. Para ello usaremos el comando git clone “Url de nuestro proyecto de github”

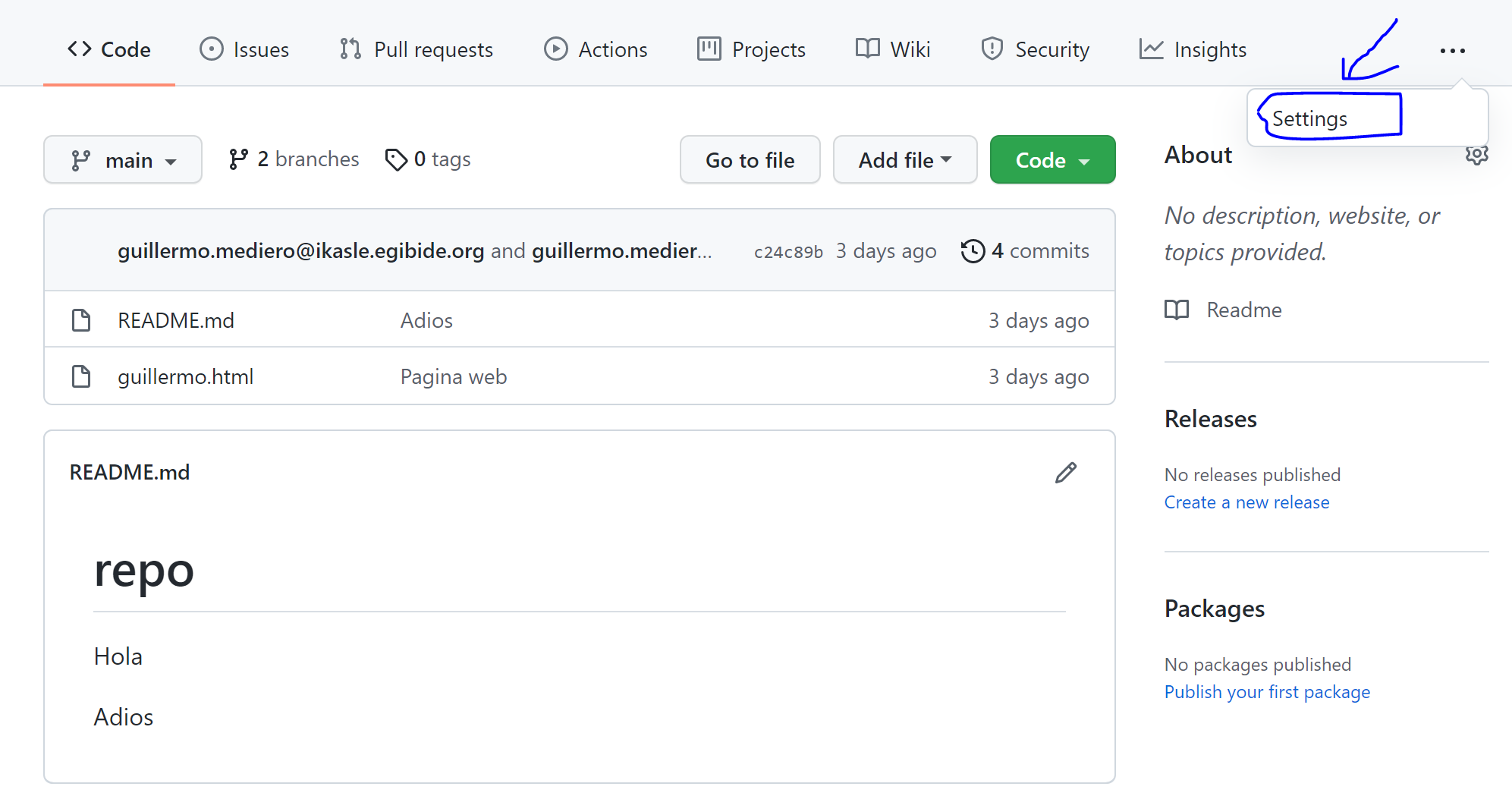


* Para comprobar las ramas siguen estando lo miramos con git branch -a

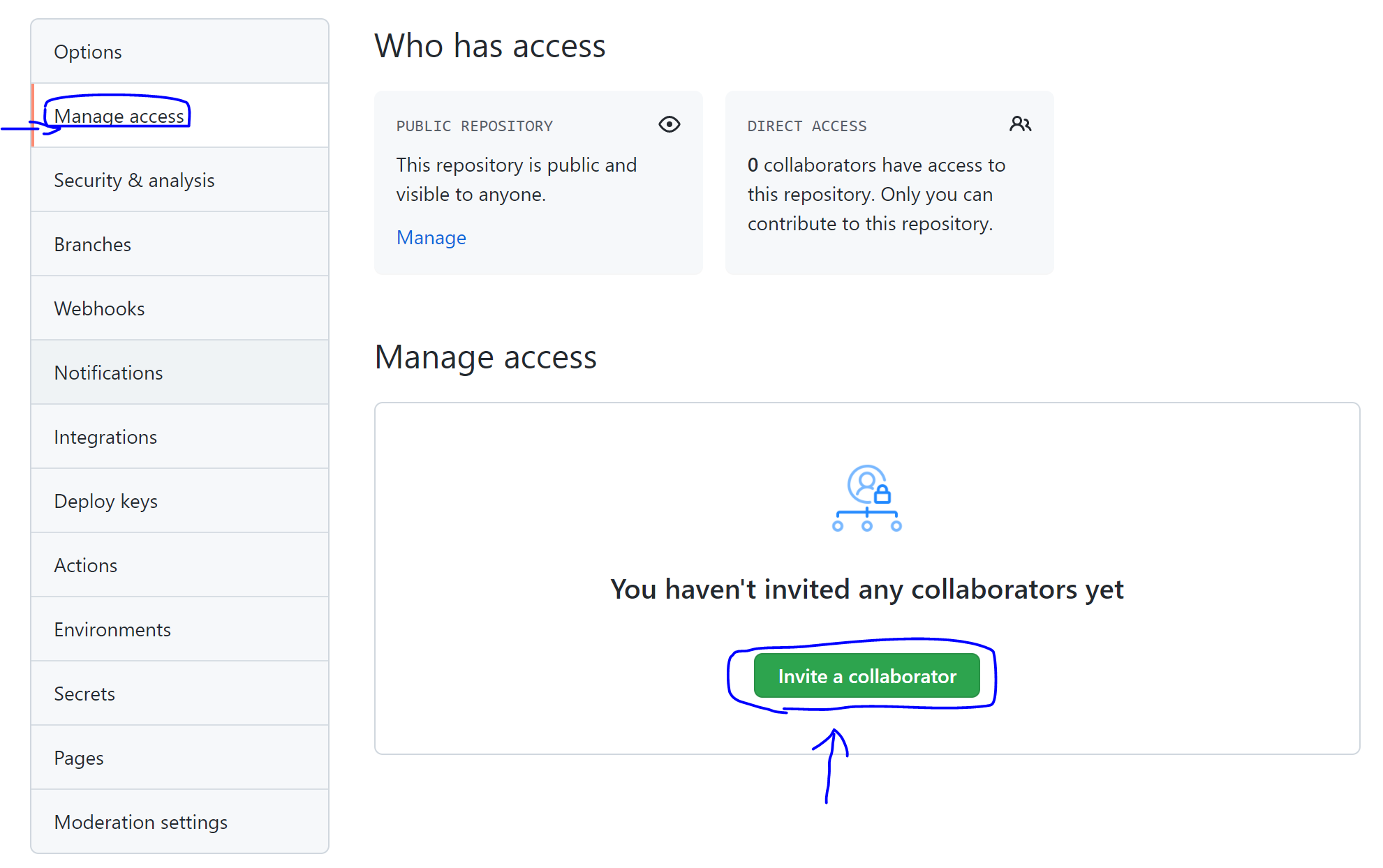


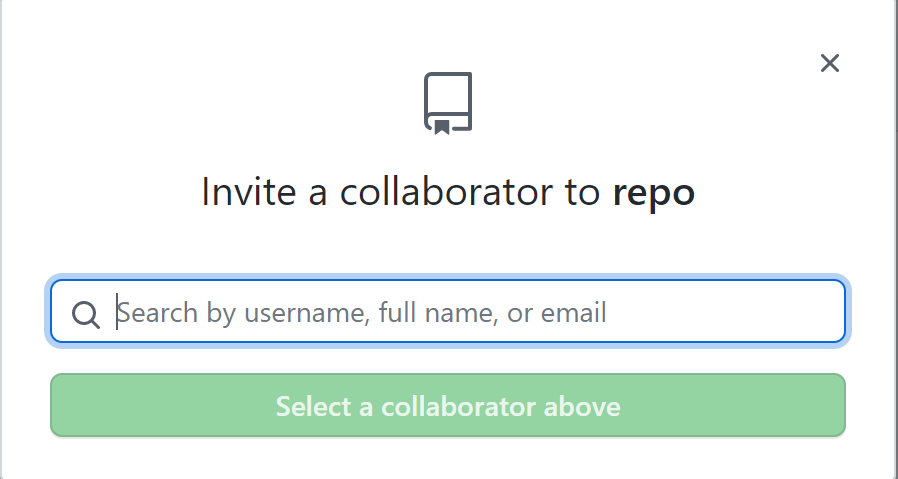
7- QUEREMOS COLABORAR

* Ahora queremos que gente de nuestro ambiente puede entrar a nuestro repositorio y modificar. Para ello iremos a GitHub y dentro de nuestro repositorio damos click en los tres puntos que aparecen arriba en la esquina derecha y daremos a settings.



* Dentro de los setting iremos al apartado de manage access y damos a “invite a collaborator”, donde nosotros pondremos o el nombre del usuario o su correo de la cuenta de GitHub del que queremos que colabore con nosotros.





* Listo este proceso si todo esta bien le llegara un correo al colaborador para poder entrar al repo y poder clonarlo y modificarlo.

