# **Práctica**

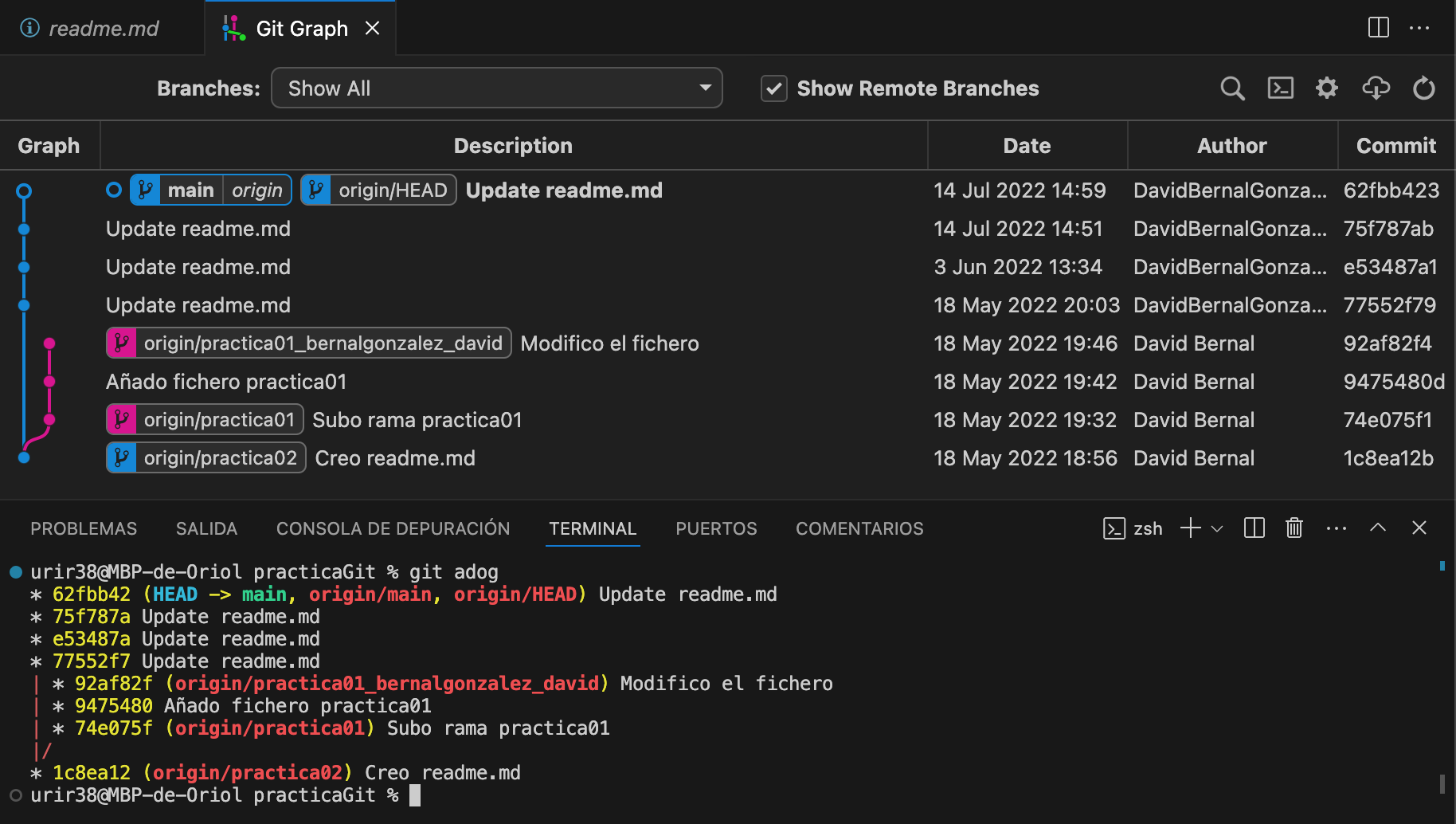
## **Ejercicio de Git bash**

1. **Clona el repositorio** [**https://github.com/DavidBernalGonzalez/practicaGit.git**](https://github.com/DavidBernalGonzalez/practicaGit.git)

Clonaremos el repositorio con git clone <https://github.com/DavidBernalGonzalez/practicaGit.git>.

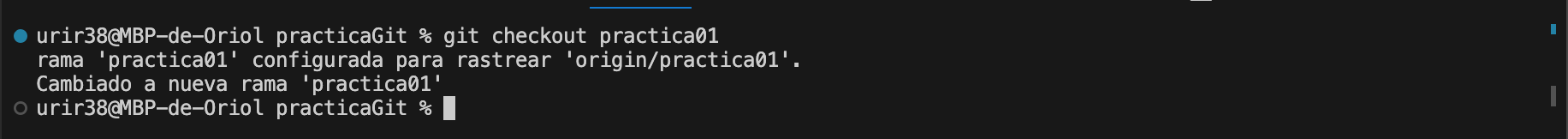
1. **Visualiza las distintas ramas que hay en el repositorio.**

Podemos ver las ramas con el plugin Git Grap, o con git adog (adog es un alias de log --all --decorate --oneline --graph que creamos en la practica anterior.

****

1. **Dentro del repositorio hay una carpeta llamada practica01, la encontrarás en la rama “practica01”. Sitúate en rama "practica01" para ver la carpeta practica01.**

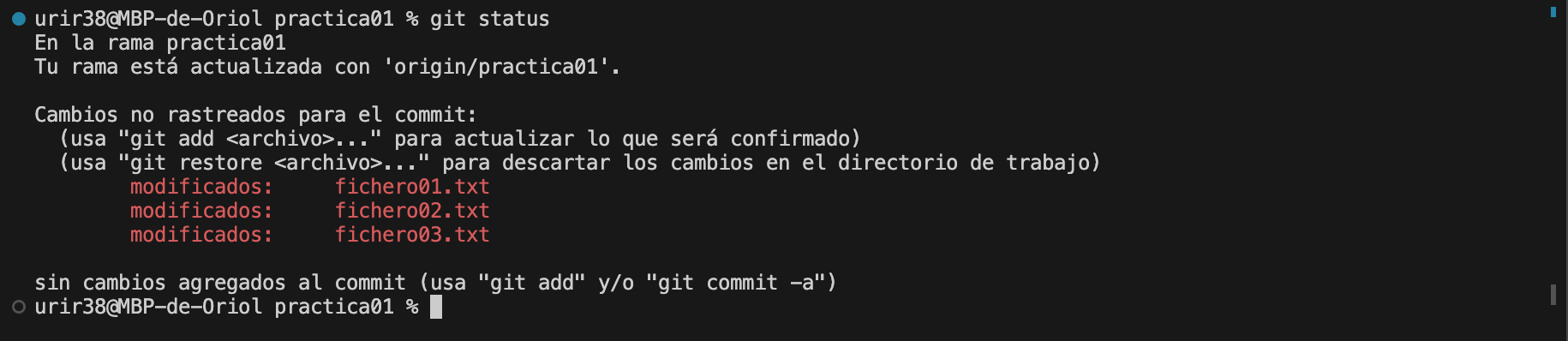
Para cambiar de rama usamos git checkout practica01, y luego podremos listar el interior de la rama con ls y veremos que ahí se encuentra la carpeta practica01.

****

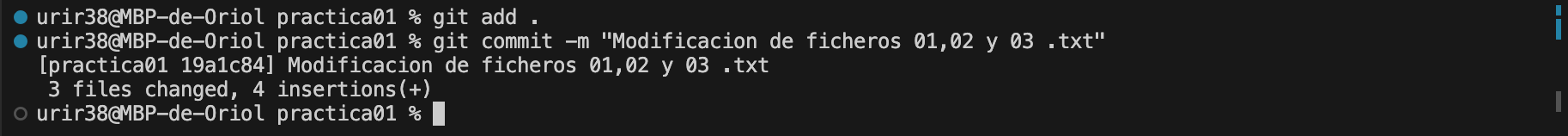
****

1. **Sitúate en el directorio practica1 y edita los archivos a tu gusto. Una vez editados los ficheros, haz un git status para ver qué ha pasado con los ficheros. Finalmente, haz un commit para subir los cambios al repositorio local.**

Al hacer un git status, vemos que nos indica que hemos modificado los ficheros 01,02 y 03. Y que podemos usar git restore <archivo> para volver a dejarlos como estaban antes de cambiarlos.

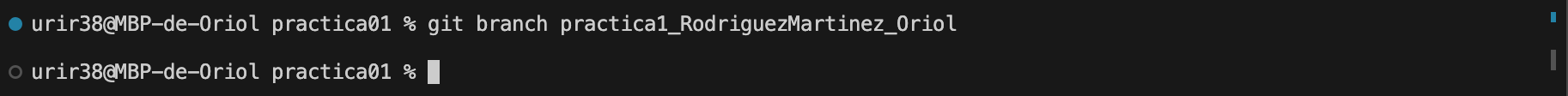
****

Finalmente le hacemos un git add . y un git commit “”.



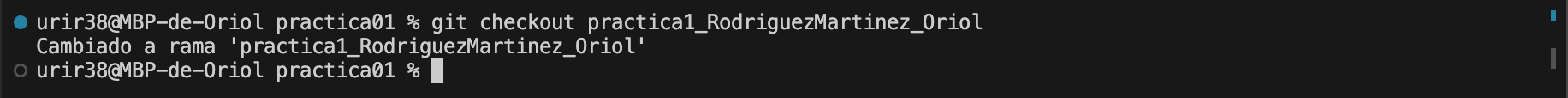
1. **Crea una rama nueva, llámala practica1\_APELLIDOS\_NOMBRE**

Utilizaremos el comando git branch practica1\_RodriguezMartinez\_Oriol.



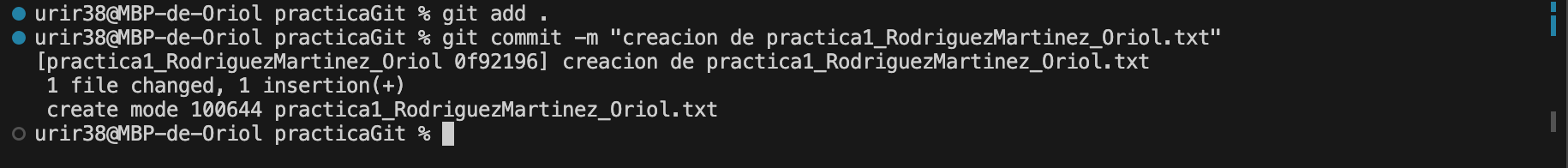
1. **Muévete a la rama que has creado(practica1\_APELLIDOS\_NOMBRE)**

Cambiamos de rama con git checkout practica1\_RodriguezMartinez\_Oriol.

****

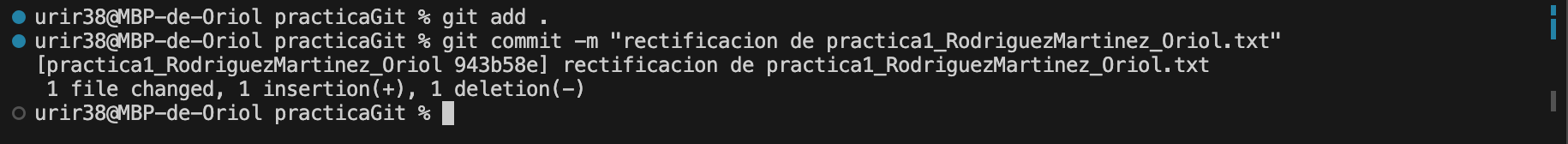
1. **En el directorio raíz, crea el fichero practica1\_APELLIDOS\_NOMBRE.txt y cualquier cosa en su interior. Y haz un commit de los cambios.**

Volvemos al directorio raíz, hacemos un git add ., y seguidamente el git commit “”.

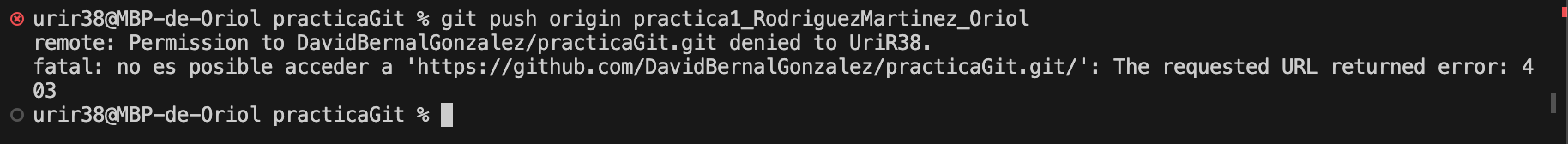
****

1. **¡Nos hemos equivocado! Queremos que en el interior del fichero practica1\_APELLIDOS\_NOMBRE.txt tendremos que poner nuestro nombre y apellidos. Vuelve a hacer otro commit revirtiendo los cambios (haciendo un nuevo commit) y sube los cambios al repositorio remoto ¿Qué está pasando? ¿Cómo lo solucionarías? ¿Tienes los permisos para escribir en dicho repositorio? ¿Existe posibilidad de arreglarlo? TIP: Elimina el remote actual y añade el de un remote de un repositorio nuevo en tu cuenta ya así si que tendrás permisos para escribirlo.**

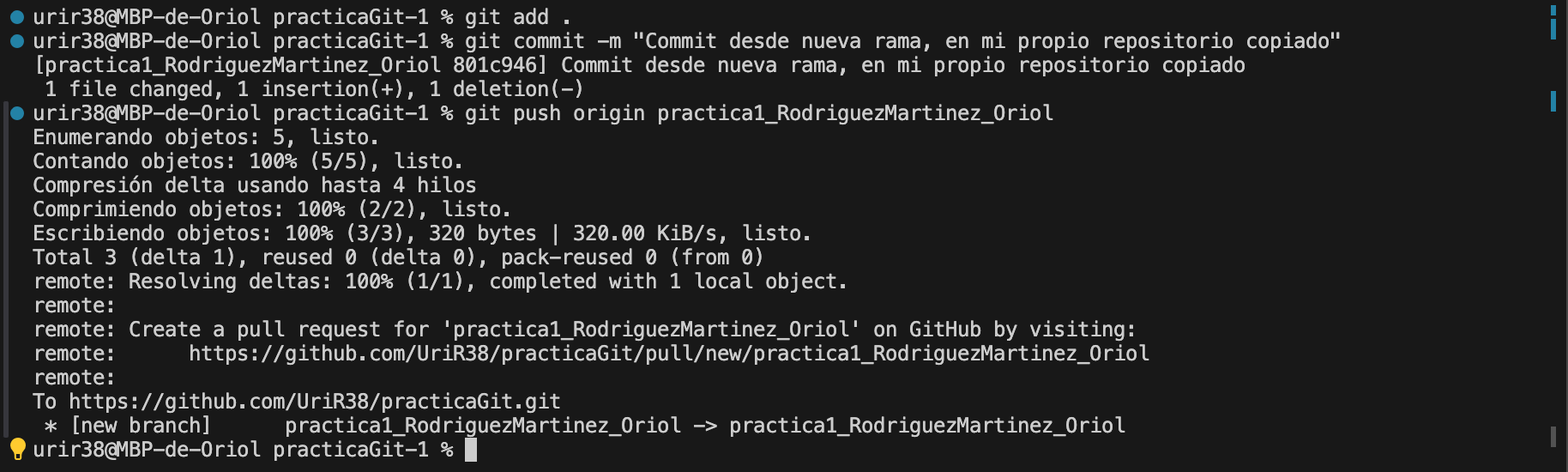
Primero de todo rectificamos el error del archivo mal escrito, y lo volvemos a commitear.

****

Al intentar hacer un git push origin practica1\_RodriguezMartinez\_Oriol nos lanzara un error diciendonos que no tenemos permisos para subir cambios al repositorio remoto. Para poder subirlo necesitamos pedirle permisos al admin del repositorio.

****

Otra opción es copiar el contenido del repositorio y crear nosotros nuestro propio repositorio con dicho contenido, así ya nos dejará subir los cambios al Github sin problemas.

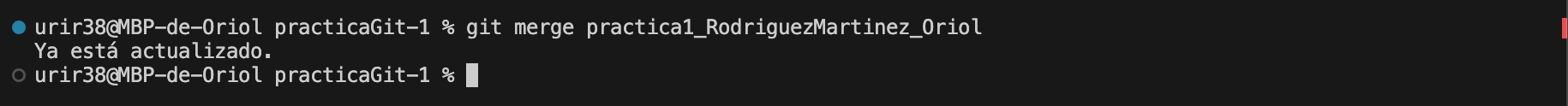


1. **Vuelve a la rama “practica1”.**

Cambiamos de rama con git checkout practica1.

1. **Haz un merge de tu rama (practica1\_APELLIDOS\_NOMBRE) con practica1**

Hacemos desde la rama que queremos hacer un merge un git merge practica1\_RodriguezMartinez\_Oriol.

****

Y podemos chequearlo con un git branch conforme ya no esta la rama pues la hemos fusionado con la actual.

****

1. **Añade dentro de la rama de tu carpeta todas los comandos que has utilizado. Puedes crear un documento por ejemplo de Word.**

## **Ejercicio de como trabajar con Git desde un IDE**

Probablemente en tu proyecto te den libertad para usar el cliente git y el IDE que prefieras, por lo que está muy bien que te vayas acostumbrando a trabajar con alguno

1. Visual Studio Code: <https://code.visualstudio.com/>
2. Vamos a utilizar VSC para clonar el repositorio <https://github.com/DavidBernalGonzalez/practicaGit.git>
3. Abre el proyecto desde VSV e investiga sobre cómo realizar un commit, push, crear rama, “mergear”, borrar rama y resolver conflictos con los clientes.