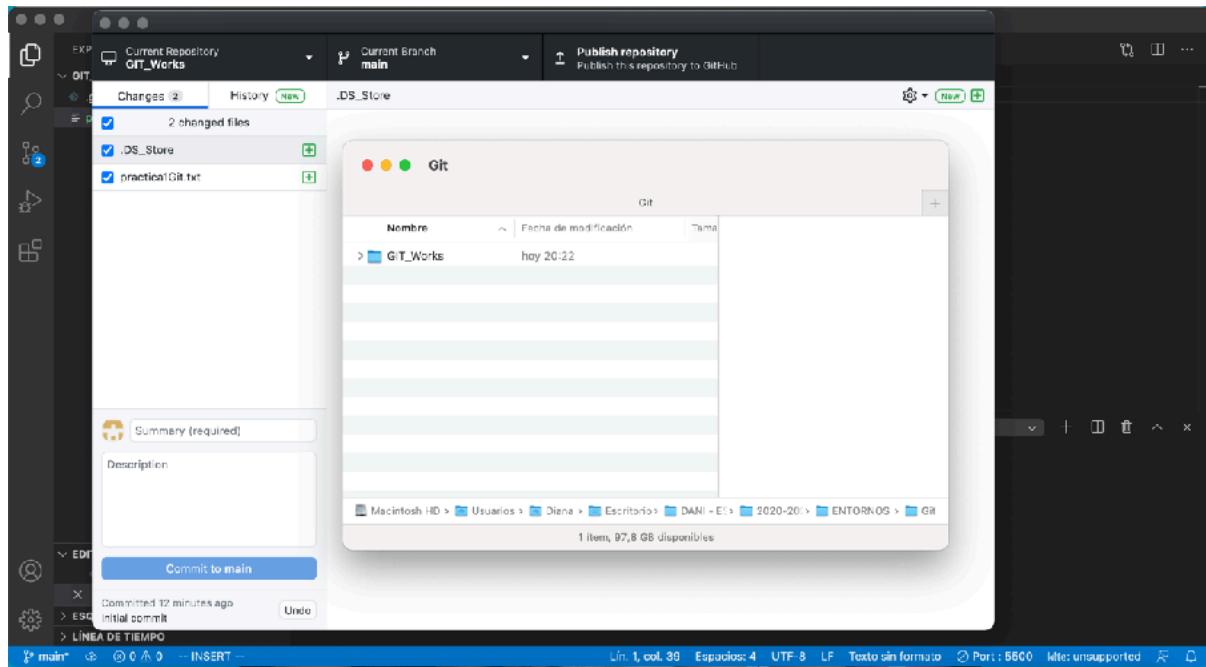


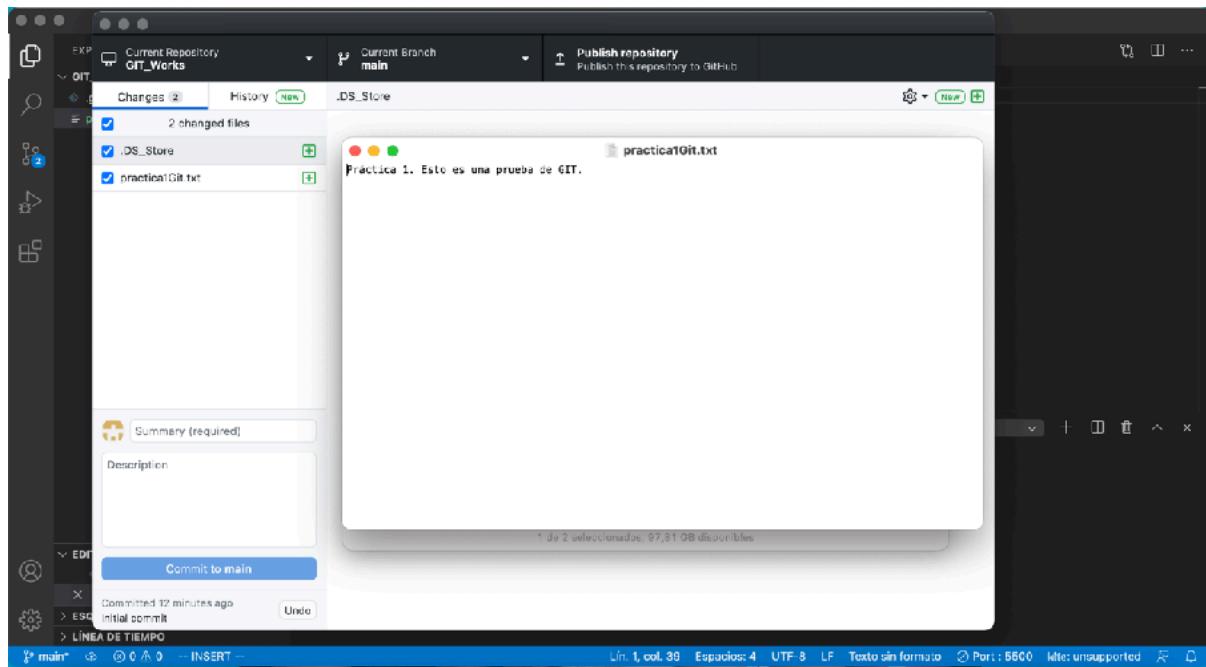
Practical: Introducción a VCS



1. Crea una carpeta llamada GIT_Works.



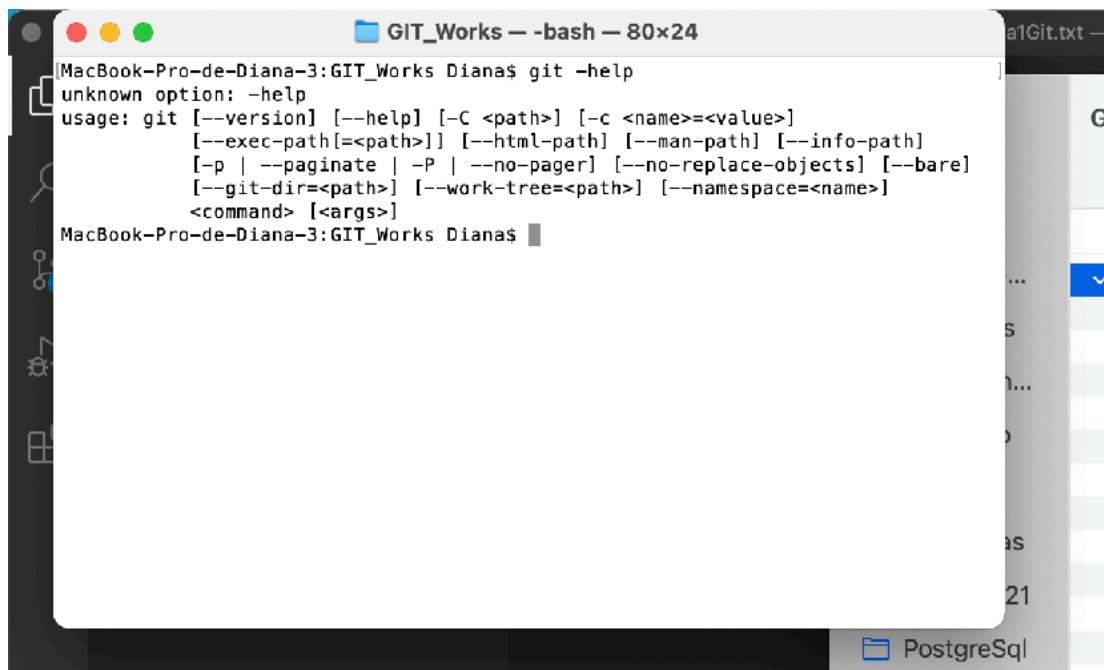
2. En ella crearemos un documento de texto e introduciremos el siguiente texto: Práctica 1. Esto es una prueba de GIT. Guárdalo con el nombre practical1Git.txt.



3. Inicia git cmd o git shell.

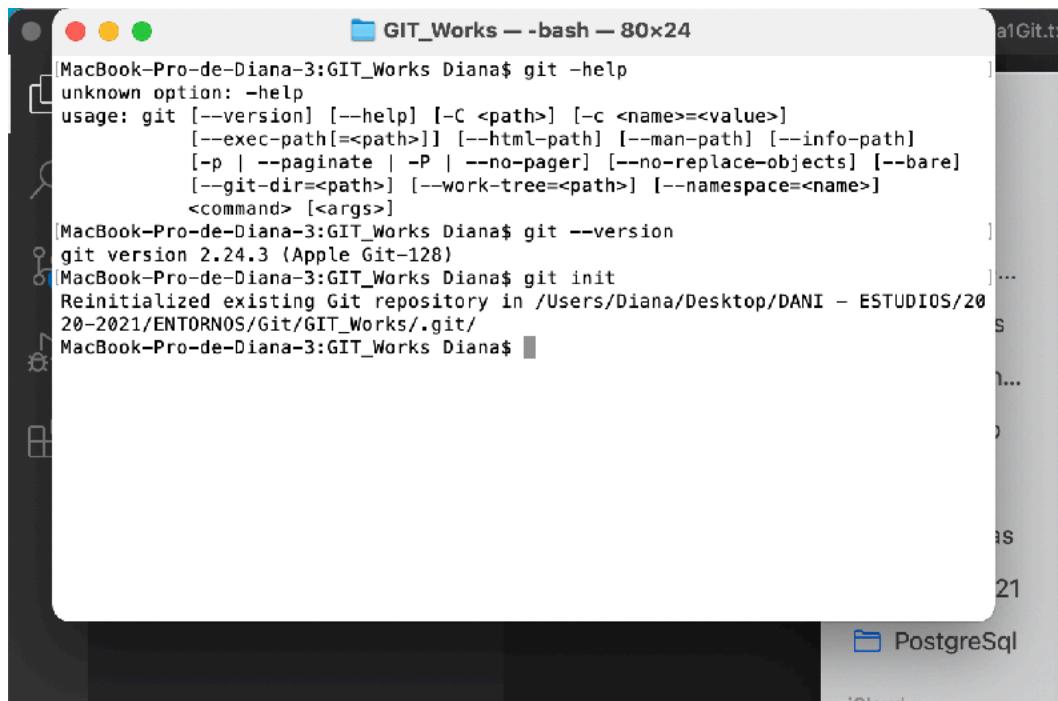
4. Asegúrate de estar ubicado en el directorio de trabajo GIT_Works cuando estemos en cmd o shell.

5. En la línea de comandos, ejecuta git --help para ver la lista de acciones disponibles en git.



```
MacBook-Pro-de-Diana-3:GIT_Works Diana$ git -help
unknown option: -help
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]
MacBook-Pro-de-Diana-3:GIT_Works Diana$
```

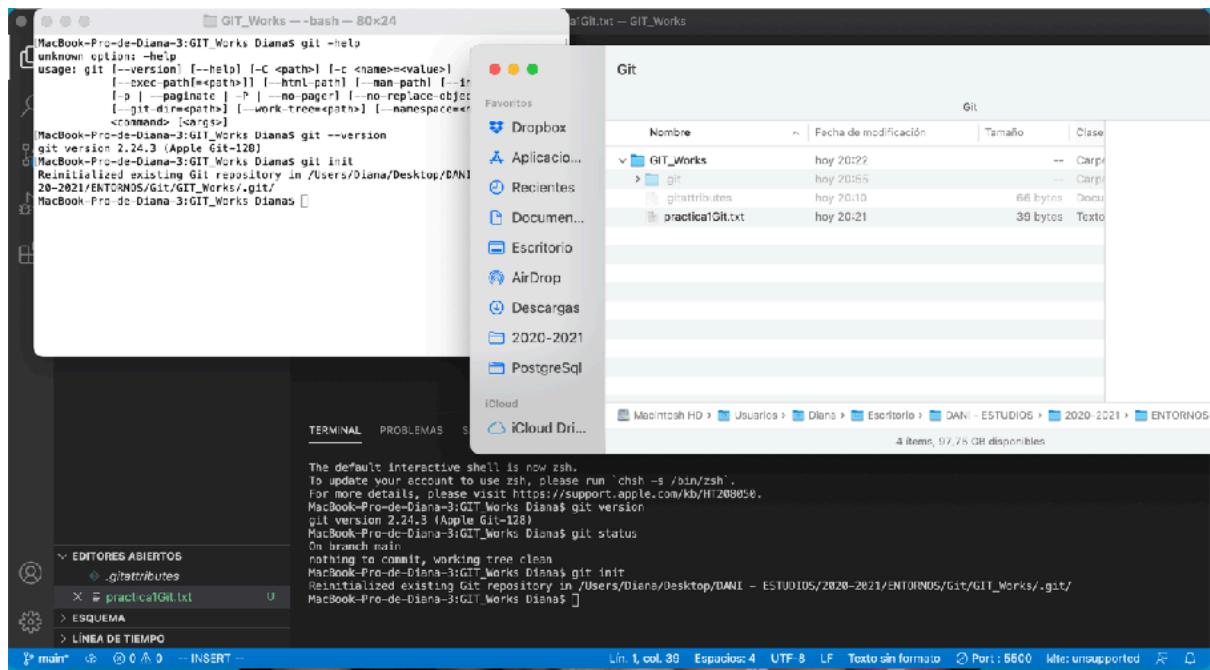
6. Inicializa nuestro repositorio.



```
MacBook-Pro-de-Diana-3:GIT_Works Diana$ git -help
unknown option: -help
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]
MacBook-Pro-de-Diana-3:GIT_Works Diana$ git --version
git version 2.24.3 (Apple Git-128)
MacBook-Pro-de-Diana-3:GIT_Works Diana$ git init
Reinitialized existing Git repository in /Users/Diana/Desktop/DANI - ESTUDIOS/20
20-2021/ENTORNOS/Git/GIT_Works/.git/
MacBook-Pro-de-Diana-3:GIT_Works Diana$
```

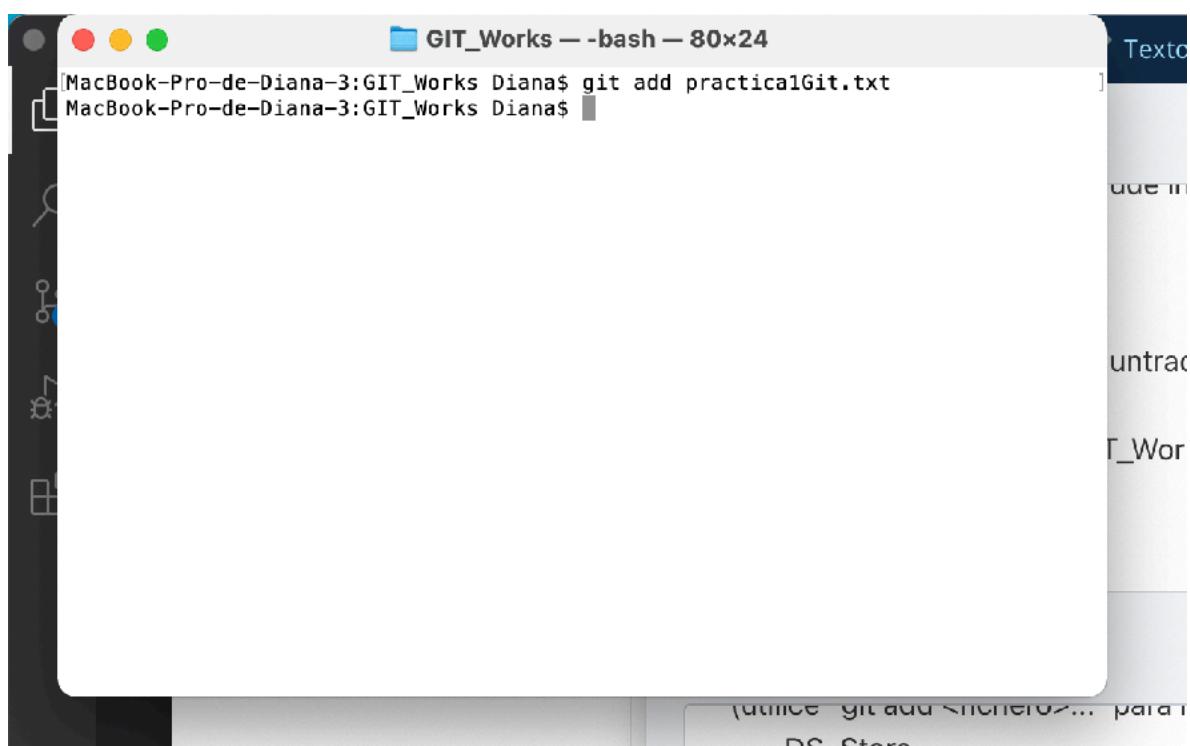
8.

Verificamos que en nuestra carpeta GIT_Works se crea una carpeta oculta llamada .git que será la carpeta de nuestro repositorio.

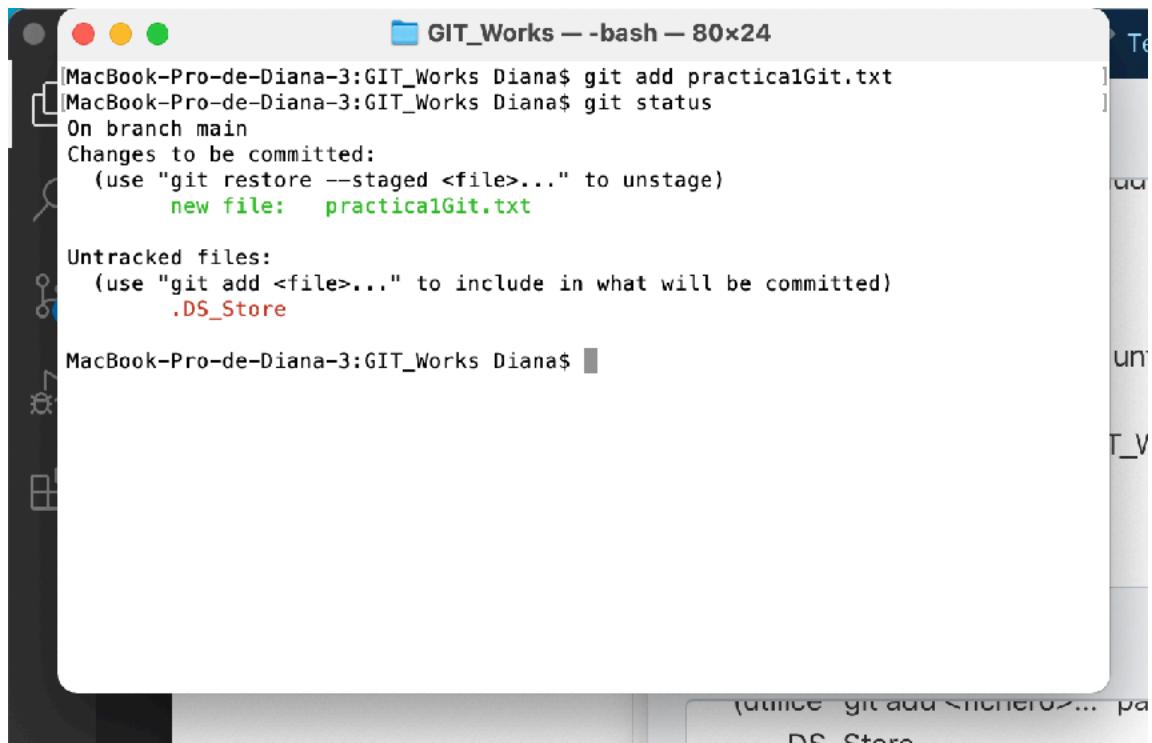


9. El archivo practical1.txt será el que tenga nuestro código, y queremos que git comience a seguirlo. Para indicar que está rastreando el archivo, utilizaremos la acción de agregar.

NOTA: no se mostrará ningún mensaje.



10. Ahora verifiquemos el estado en que se encuentra nuestro repositorio, para esto usaremos la acción de estado. NOTA: Si NO escribimos el nombre del archivo correctamente, nos dará un error.

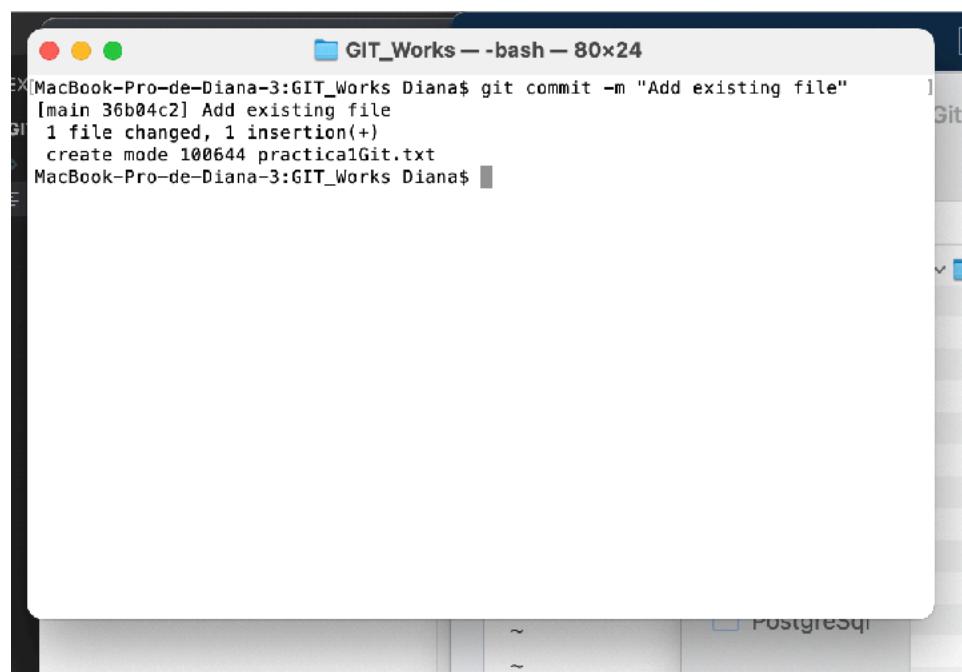


```
MacBook-Pro-de-Diana-3:GIT_Works Diana$ git add practicalGit.txt
MacBook-Pro-de-Diana-3:GIT_Works Diana$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   practicalGit.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .DS_Store

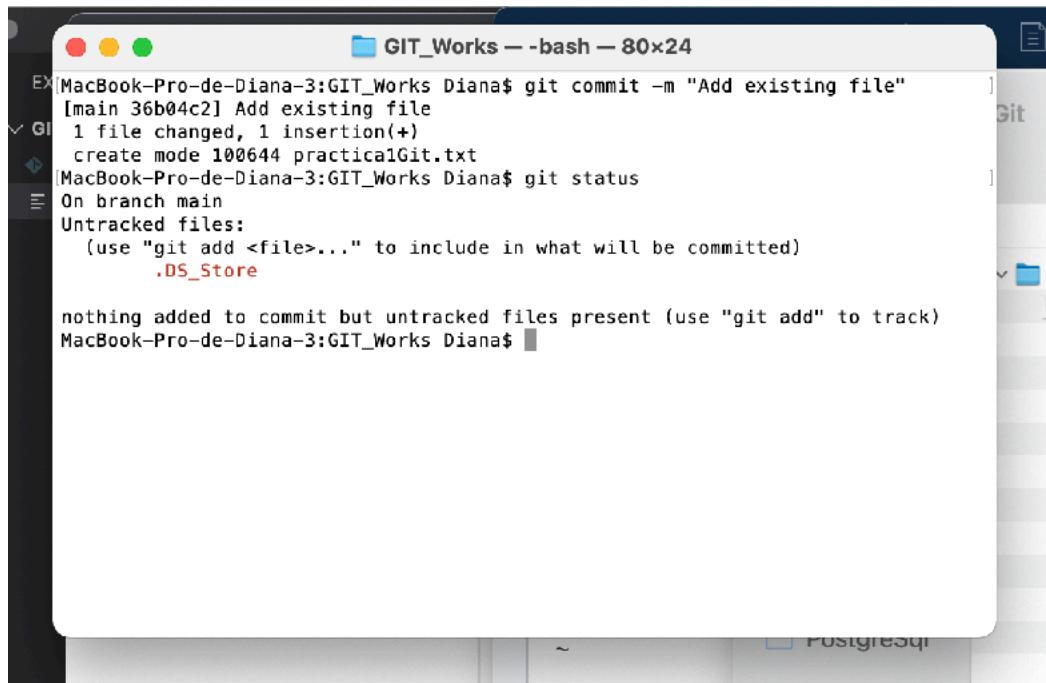
MacBook-Pro-de-Diana-3:GIT_Works Diana$
```

11. Finalmente podemos enviar los archivos al repositorio. Recordar que debemos escribir un mensaje que se refiera a esa confirmación. Por lo tanto, el commit aprueba y trae al repositorio un conjunto de modificaciones, es decir, una revisión.



```
MacBook-Pro-de-Diana-3:GIT_Works Diana$ git commit -m "Add existing file"
[main 36b04c2] Add existing file
 1 file changed, 1 insertion(+)
  create mode 100644 practicalGit.txt
MacBook-Pro-de-Diana-3:GIT_Works Diana$
```

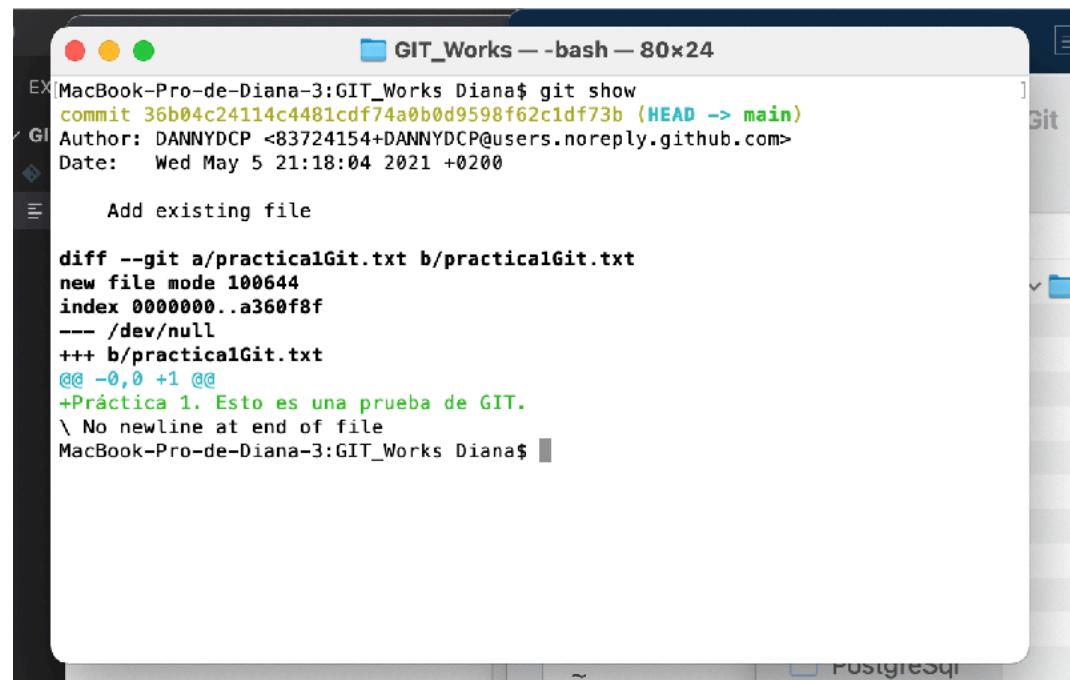
12. Verificamos el estado nuevamente, para esto usaremos la acción de estado. Ahora, ya no hay ningún archivo pendiente para confirmar.



```
MacBook-Pro-de-Diana-3:GIT_Works Diana$ git commit -m "Add existing file"
[main 36b04c2] Add existing file
  1 file changed, 1 insertion(+)
   create mode 100644 practicalGit.txt
MacBook-Pro-de-Diana-3:GIT_Works Diana$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .DS_Store

nothing added to commit but untracked files present (use "git add" to track)
MacBook-Pro-de-Diana-3:GIT_Works Diana$
```

13. Queremos mostrar los cambios que han tenido lugar en el repositorio. Para esto usamos la acción git show y git log. Comentad la diferencia entre las dos opciones.



```
MacBook-Pro-de-Diana-3:GIT_Works Diana$ git show
commit 36b04c24114c4481cdf74a0b0d9598f62c1df73b (HEAD -> main)
Author: DANNYDCP <83724154+DANNYDCP@users.noreply.github.com>
Date:   Wed May 5 21:18:04 2021 +0200

  Add existing file

diff --git a/practicalGit.txt b/practicalGit.txt
new file mode 100644
index 000000..a360f8f
--- /dev/null
+++ b/practicalGit.txt
@@ -0,0 +1 @@
+Práctica 1. Esto es una prueba de GIT.
\ No newline at end of file
MacBook-Pro-de-Diana-3:GIT_Works Diana$
```

```

[MacBook-Pro-de-Diana-3:GIT_Works Diana]$ git log
commit 36b04c2414c4481cdf74a0b0d9598f62c1df73b (HEAD -> main)
Author: DANNYDCP <83724154+DANNYDCP@users.noreply.github.com>
Date:   Wed May 5 21:18:04 2021 +0200

    Add existing file

commit f5686367b8bbcd5856946d8b7b52a993387bca3e
Author: DANNYDCP <83724154+DANNYDCP@users.noreply.github.com>
Date:   Wed May 5 20:10:51 2021 +0200

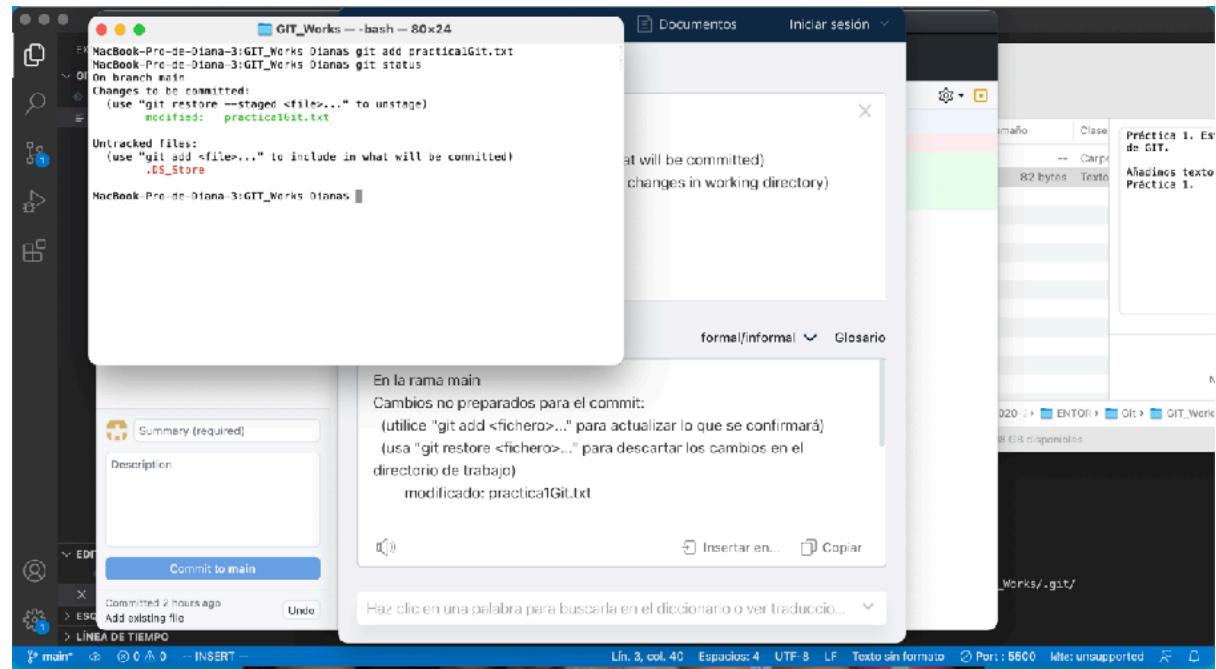
    Initial commit
MacBook-Pro-de-Diana-3:GIT_Works Diana$

```

14. Ahora vamos a nuestro archivo de texto y lo modificamos, agregamos las nuevas líneas y lo guardamos Se recomienda una vez que se modifique el archivo para cerrarlo, porque si el archivo está abierto, es posible que no se aplique correctamente ninguna acción.

15. Como hemos modificado el archivo inicial, envíelo de vuelta al repositorio

16. Verifica nuevamente el repositorio.



17. Modifica el archivo practical1.txt, elimina la primera línea y guarda el documento.
18. Envía el nuevo documento al repositorio git commit -a -m "Eliminé la primera línea".

```
MacBook-Pro-de-Diana-3:GIT_Works Diana$ git commit -a -m "Eliminé la primera linea"
[main 57a9a46] Eliminé la primera linea
 1 file changed, 5 insertions(+), 1 deletion(-)
MacBook-Pro-de-Diana-3:GIT_Works Diana$
```

19. Verificamos los cambios con git show. La línea que hemos eliminado aparece precedida por un carácter - y en color rojo.

```
ea"
[main 57a9a46] Eliminé la primera linea
 1 file changed, 5 insertions(+), 1 deletion(-)
MacBook-Pro-de-Diana-3:GIT_Works Diana$ git show
commit 57a9a46667c495ebef4eb5f8e8a77bc8d8b278b7 (HEAD -> main)
Author: DANNYDCP <83724154+DANNYDCP@users.noreply.github.com>
Date:   Wed May 5 23:50:08 2021 +0200

    Eliminé la primera linea

diff --git a/practical1Git.txt b/practical1Git.txt
index a360f8f..6c6a9ff 100644
--- a/practical1Git.txt
+++ b/practical1Git.txt
@@ -1 +1,5 @@
-Práctica 1. Esto es una prueba de GIT.
\ No newline at end of file
+
+
+Añadimos texto al documento Práctica 1.
+
+
+Hemos borrado la primera linea
\ No newline at end of file
MacBook-Pro-de-Diana-3:GIT_Works Diana$
```

20. Agregamos una nueva línea al final del documento practical1.txt y guardamos. Ahora, en nuestra carpeta de trabajo, tenemos un documento diferente que en el HEAD del repositorio (porque aún no hemos realizado el commit).

21. Comprueba todos los commits que hemos realizado, con git log. Git coloca un identificador para cada confirmación realizada. Ese identificador son los primeros 6 caracteres de la confirmación. El identificador de mi primer commit será 24a403. Este identificador nos servirá, entre otras cosas, para comparar las diferencias entre dos confirmaciones. Recuerda que tendrás 6 caracteres diferentes a los míos.

```
+ Añadimos texto al documento Práctica 1.
+ Hemos borrado la primera linea
\ No newline at end of file
[MacBook-Pro-de-Diana-3:GIT_Works Diana$ git log
commit 57a9a46667c495ebef4eb5f8e8a77bc8d8b278b7 (HEAD -> main)
Author: DANNYDCP <83724154+DANNYDCP@users.noreply.github.com>
Date:   Wed May 5 23:50:08 2021 +0200

    Eliminé la primera linea

commit 36b04c24114c4481cdf74a0b0d9598f62c1df73b
Author: DANNYDCP <83724154+DANNYDCP@users.noreply.github.com>
Date:   Wed May 5 21:18:04 2021 +0200

    Add existing file

commit f5686367b8bbcd5856946d8b7b52a993387bca3e
Author: DANNYDCP <83724154+DANNYDCP@users.noreply.github.com>
Date:   Wed May 5 20:10:51 2021 +0200

    Initial commit
MacBook-Pro-de-Diana-3:GIT_Works Diana$
```

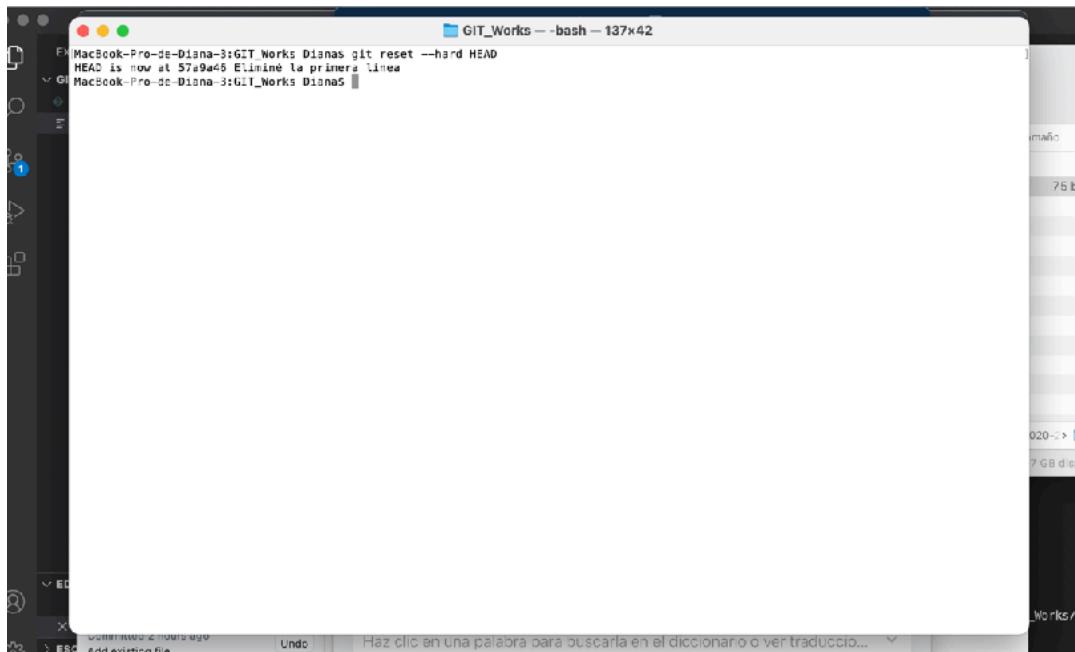
22. Veamos las diferencias entre 2 commits, usando git diff id_commit1 id_commit2. Verifica las diferencias entre el primer commit y el último.

```
Author: DANNYDCP <83724154+DANNYDCP@users.noreply.github.com>
Date:   Wed May 5 21:18:04 2021 +0200
Add existing file

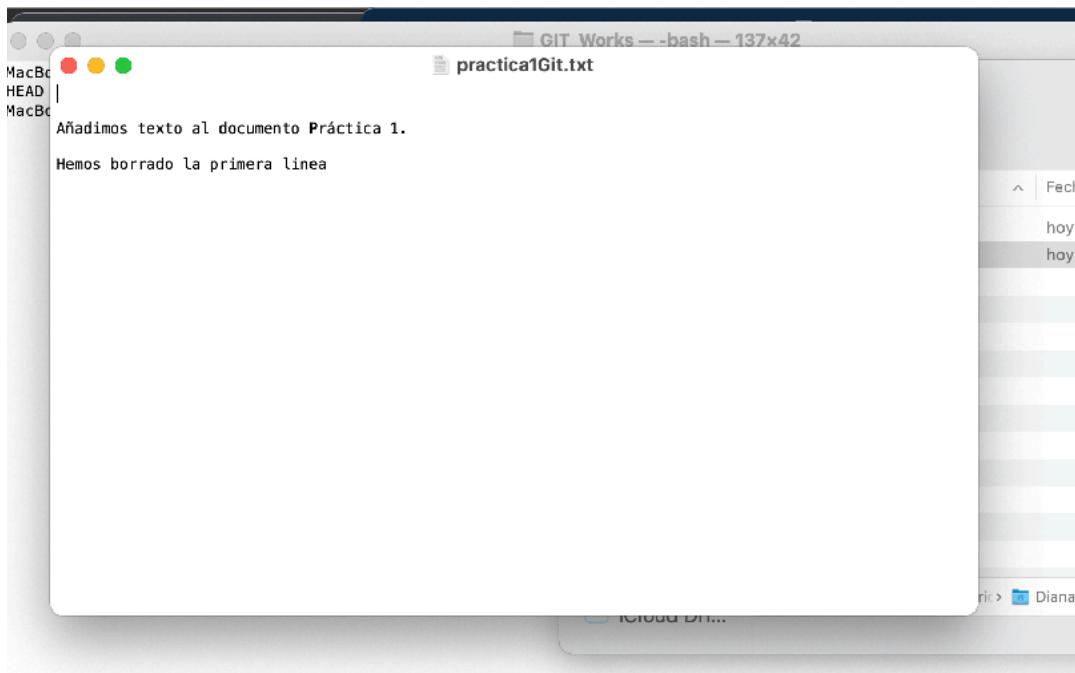
commit f5686367b8bbcd5856946d8b7b52a993387bca3e
Author: DANNYDCP <83724154+DANNYDCP@users.noreply.github.com>
Date:   Wed May 5 20:10:51 2021 +0200

Initial commit
MacBook-Pro-de-Diana-3:GIT_Works Diana$ git diff f5586 57a9a
diff --git a/practicalGit.txt b/practicalGit.txt
new file mode 100644
index 000000..6c6a9ff
--- /dev/null
+++ b/practicalGit.txt
@@ -0,0 +1,5 @@
+
+Añadimos texto al documento Práctica 1.
+
+Hemos borrado la primera linea
\ No newline at end of file
MacBook-Pro-de-Diana-3:GIT_Works Diana$
```

23. Ahora suponemos que no queremos la última modificación que hicimos (paso 20) y queremos recuperar la última versión del repositorio. Traed la versión que está en HEAD al directorio de trabajo. Ahora verifica que el archivo practical1.txt haya cambiado y que sea como la versión en HEAD.



```
MacBook-Pro-de-Diana-3:GIT_Works Diana$ git reset --hard HEAD
HEAD is now at 57e9ac6 Eliminé la primera linea
MacBook-Pro-de-Diana-3:GIT_Works Diana$
```



```
MacBook-Pro-de-Diana-3:GIT_Works Diana$ practical1Git.txt
practical1Git.txt
MacBook-Pro-de-Diana-3:GIT_Works Diana$ Añadimos texto al documento Práctica 1.
MacBook-Pro-de-Diana-3:GIT_Works Diana$ Hemos borrado la primera linea
```