

## Ejercicio Git

1. Crea una carpeta llamada `repasoexamen`.

```
mkdir repasoexamen
```

2. Entra en esa carpeta

```
cd repasoexamen
```

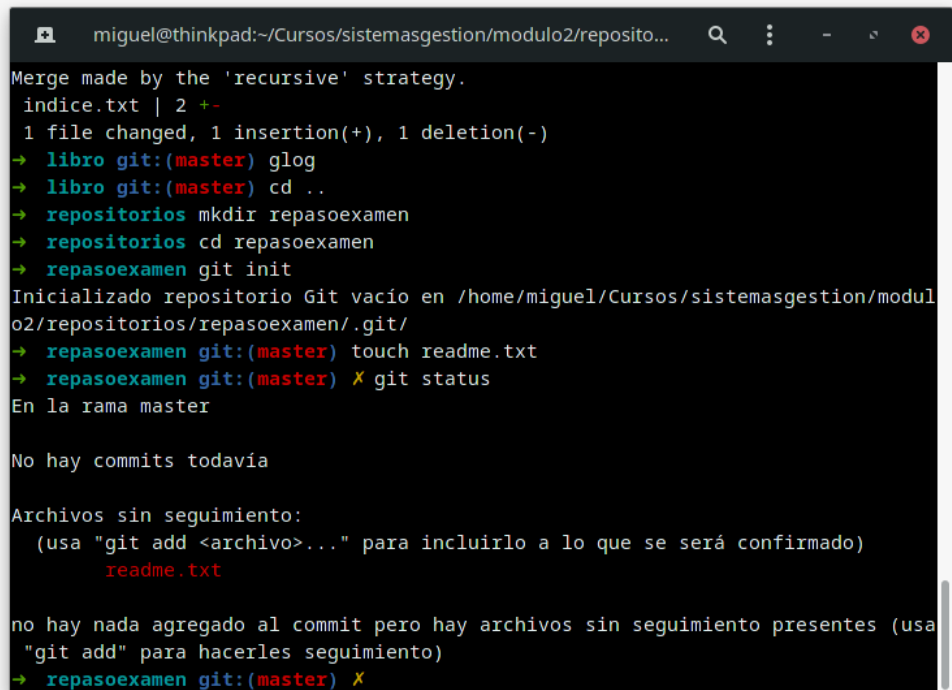
3. Inicializa un repositorio de git vacío.

```
git init
```

4. Crea un archivo llamado `readme.txt` y ejecuta `git status`

```
touch readme.txt
```

```
git status
```



```
miguel@thinkpad:~/Cursos/sistemasgestion/modulo2/reposito...
Merge made by the 'recursive' strategy.
 indice.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
→ libro git:(master) glog
→ libro git:(master) cd ..
→ repositorios mkdir repasoexamen
→ repositorios cd repasoexamen
→ repasoexamen git init
Iniciado repositorio Git vacío en /home/miguel/Cursos/sistemasgestion/modulo2/repositorios/repasoexamen/.git/
→ repasoexamen git:(master) touch readme.txt
→ repasoexamen git:(master) X git status
En la rama master

No hay commits todavía

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)
  readme.txt

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacerles seguimiento)
→ repasoexamen git:(master) X
```

5. Add & commit

```
git add .
```

```
git commit -m «Añadido readme.txt»
```

6. Modifica el archivo creado en el punto 4 añadiendo tu nombre y fecha de nacimiento

```
vim readme.txt
```

7. Ejecuta el comando necesario para deshacer esos cambios introducidos en el punto 6 y que el fichero se quede como estaba en el commit anterior.

```
git restore readme.txt
```

8. Modifica de nuevo el archivo del punto 4, esta vez escribe el texto "Ejercicio de repaso previo al examen"

```
vim readme.txt
```

9. Add & commit

```
git add .
```

```
git commit -m «Modificado readme.txt»
```

10. Ejecuta el comando necesario para eliminar el último commit y que los archivos vuelvan al staging área previo a dicho commit.

```
git reset --soft HEAD~1
```

11. Comprueba que efectivamente, los archivos están en el staging área y haz un commit con el mensaje que prefieras

```
git status
```

```
git commit -m «Modificado nuevamente readme.txt»
```

12. Crea un archivo llamado `estudia.txt`.

```
touch estudia.txt
```

13. Añadir `estudia.txt` al *staging área*

```
git add estudia.txt
```

14. Mueve `estudia.txt` desde el *staging area* al directorio de trabajo.

```
git restore --staged estudia.txt
```

15. Añadir `estudia.txt` al *staging area* y ejecuta `git status`

```
git add estudia.txt
```

```
git status
```

16. Eliminar `estudia.txt` del *staging área* y del directorio de trabajo y ejecuta `git status`

```
git rm -f estudia.txt
```

```
git status
```

17. Crea un nuevo archivo llamado `aprobaremos_todos.txt`.

```
touch aprobaremos_todos.txt
```

18. Añadir `aprobaremos_todos.txt` al *staging area*

```
git add .
```

19. Confirmar con el mensaje "añadiendo aprobaremos\_todos.txt".

```
git commit -m «Añadiendo aprobaremos_todos.txt»
```

20. Crea otro archivo llamado `index.html`.

```
touch index.html
```

21. Añade `index.html` al *staging area*

```
git add .
```

22. Haz commit con el mensaje "añadiendo index.html".

```
git commit -m «Añadiendo index.html»
```

23. Utilizando ramas y los comandos necesarios vuelve al commit del punto 11 y desde allí crea una rama en la que modificarás el archivo `readme.txt`,

añadiendo una línea “viaje al pasado a reparar un error”. Una vez hecho esto haz lo necesario para incorporar esos cambios de la rama recién creada a la rama máster (de la manera que prefieras)

```
git checkout 065afe6
```

```
git checkout -b fix
```

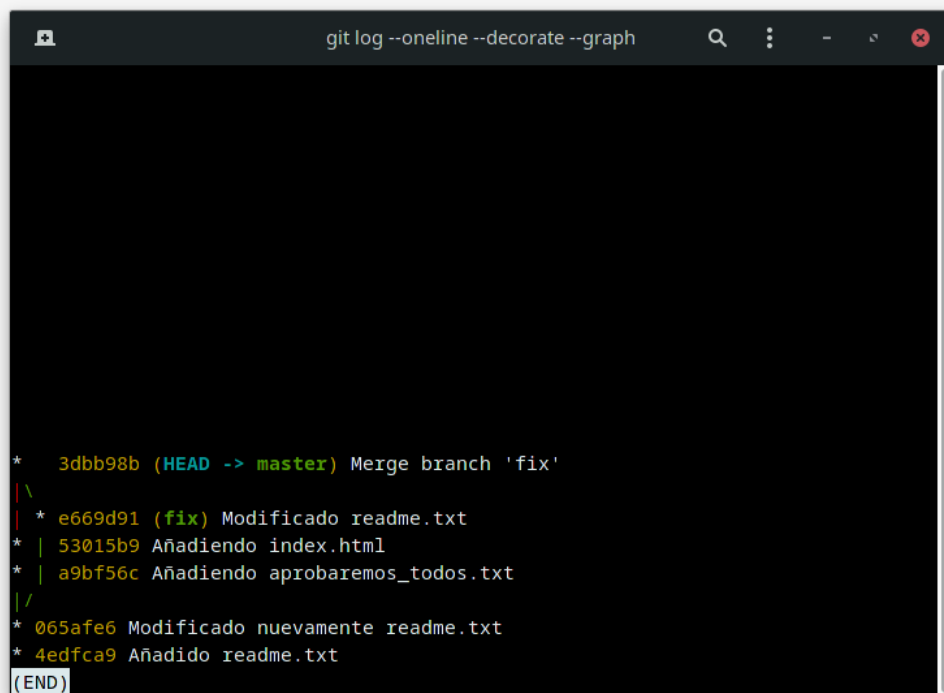
```
vim readme.txt
```

```
git add .
```

```
git commit -m «Modificado readme.txt»
```

```
git checkout master
```

```
git merge fix
```



```
git log --oneline --decorate --graph

* 3dbb98b (HEAD -> master) Merge branch 'fix'
| \
|  * e669d91 (fix) Modificado readme.txt
|  | 53015b9 Añadiendo index.html
|  | a9bf56c Añadiendo aprobaremos_todos.txt
|  | /
|  /
* 065afe6 Modificado nuevamente readme.txt
* 4edfca9 Añadido readme.txt
(END)
```