CONFIGURAR GIT:

git - -version → para ver la version instalada

git config - -global user.name “Alvaro Lorente” para configurar el usuario

git config - -global user.email [alorentealman@gmail.com](mailto:alorentealman@gmail.com) para el correo

git config - -global core.editor “code - -wait” para configurar vsCode por defecto

git config - -global -e para ver el archivo de configuracion global en VSCode

git config -h para ver ayudas de config

Crear un repo (miweb): get init miweb2

Crear .gitignore

git init web2

cd a miweb2

code . (creo el gitignore)

# **ENTORNOS DE DESARROLLO - 1º DAM**

# **UD2. INTRODUCCIÓN A GIT**

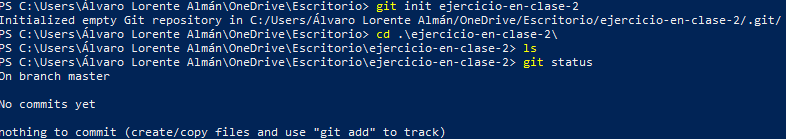
## 

## EJERCICIO EN CLASE 2

| **Resultado de Aprendizaje 2**: Utiliza sistemas de control de versiones como herramienta básica para la gestión colaborativa del código fuente. |
| --- |
| **Criterio de evaluación** |
| RA02.b Se han inicializado o clonado repositorios y se han realizado operaciones con ellos (añadir, confirmar o consultar) |
| RA02.c Se han realizado operaciones con copias remotas de un repositorio. |

1. Crea un nuevo repositorio llamado **ejercicio-en-clase-2** en tu ordenador.

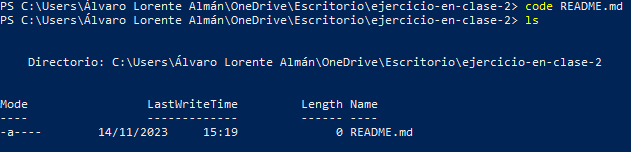
en el escritorio creo el repositorio



1. Añade un fichero README.md (por ahora vacío)

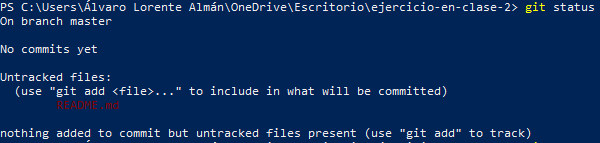
en la carpeta ejercicio-clase-2 creo el README.md

muestro que se ha creado

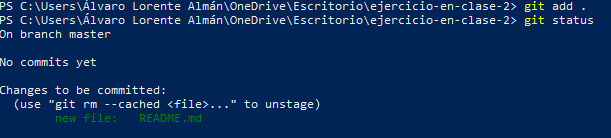


1. Añádelo a la *staging area*.

muestro que aun no esta en la staging area

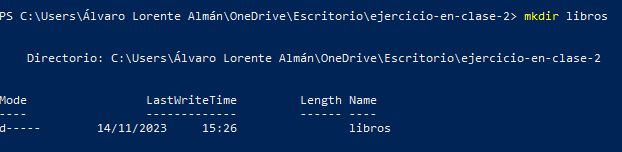


con git add . añado a la staging area todo lo que esté en untracked files

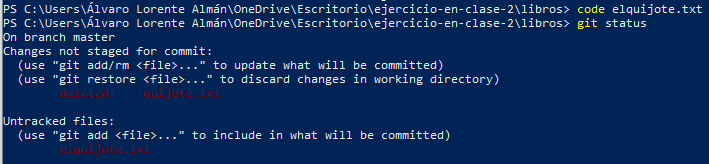


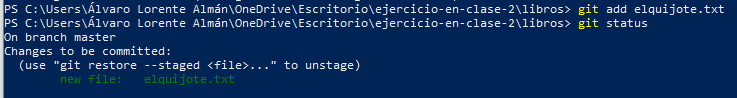
Se ha agregado a la staging area

1. Crea un nuevo directorio, que se llame libros

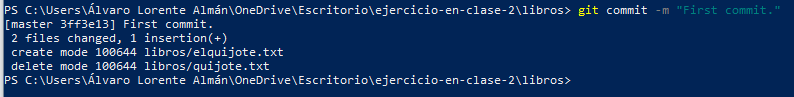


1. Crea un nuevo fichero que se llame elquijote.txt (por ahora también vacío).

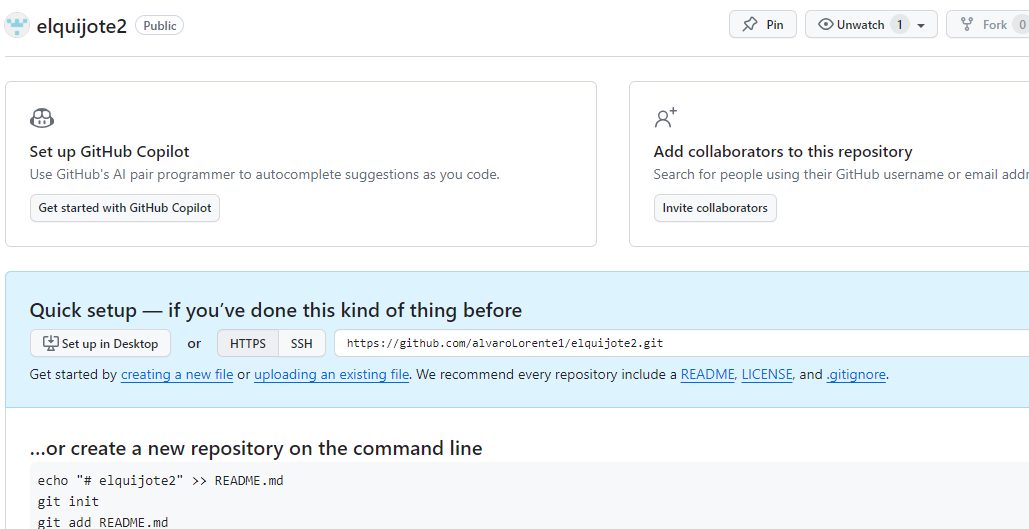




1. Añade un nuevo commit, con el mensaje “First commit”.

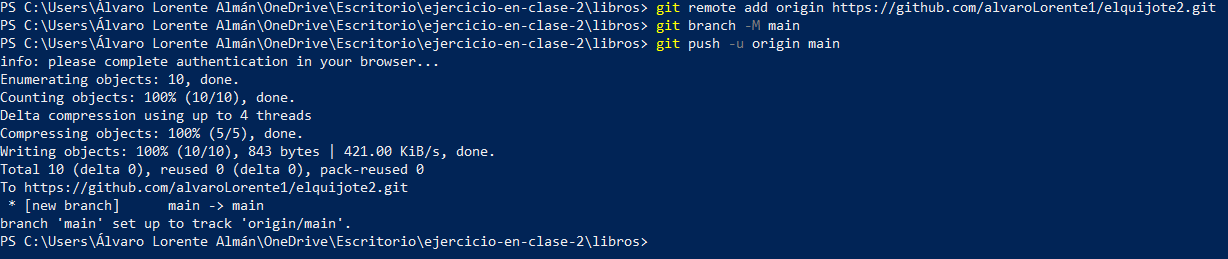


1. Crea un nuevo repositorio, vacío, en tu cuenta de Github. Se llamará *el-quijote*

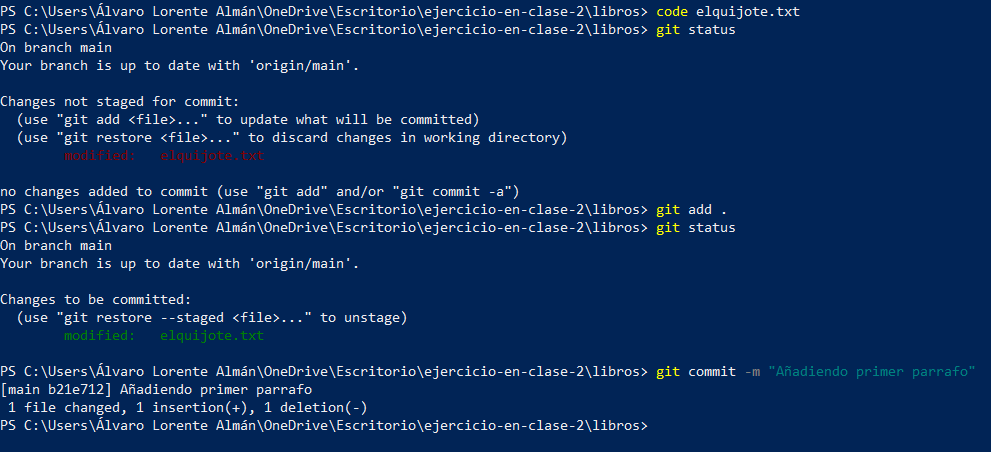
**

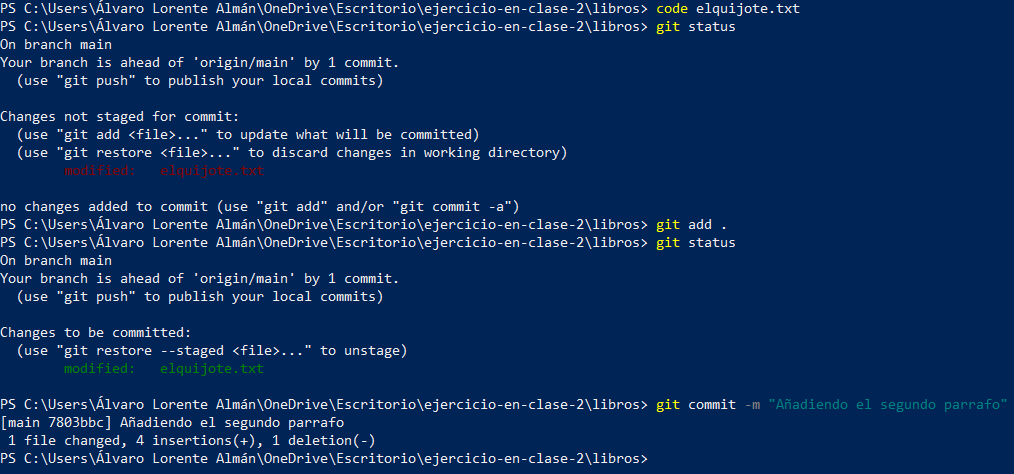
1. Conecta tu repositorio local y remoto y sube los cambios.

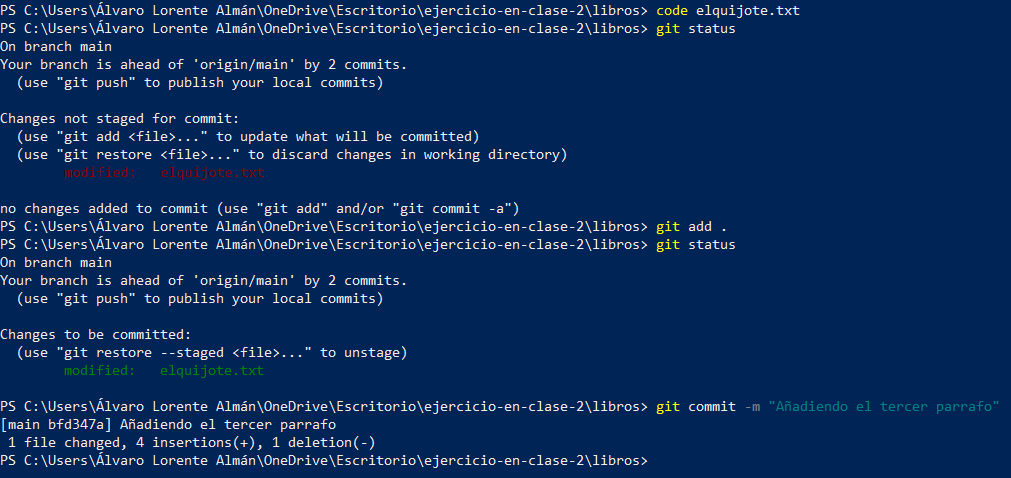
Creo el repositorio en git



1. Supongamos que tienes que copiar El Quijote entero (solo van a ser algunos párrafos nada más). Añade el primer párrafo, y confirma los cambios con un mensaje de commit adecuado. Después, repite los mismos pasos con el segundo, y posteriormente con el tercero. Puedes encontrar el texto de El Quijote aquí: <https://www.cervantesvirtual.com/obra-visor/el-ingenioso-hidalgo-don-quijote-de-la-mancha-6/html/>

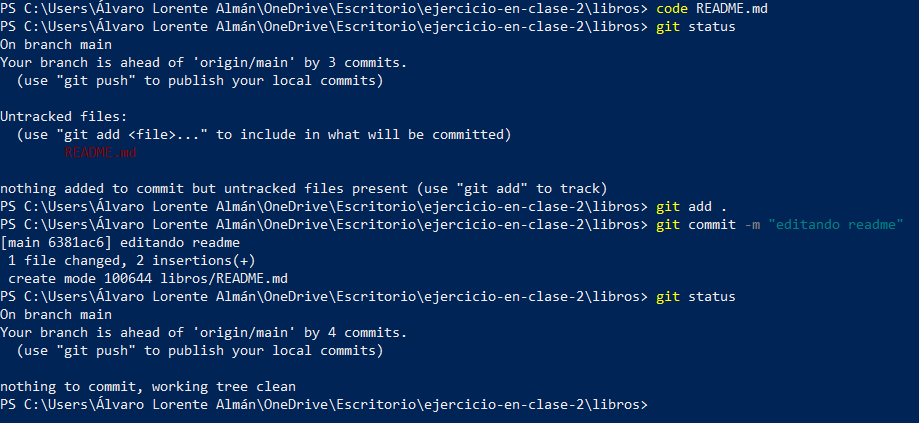




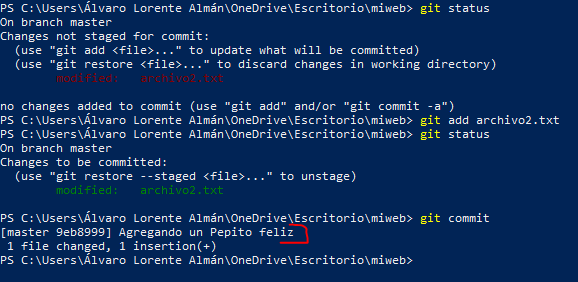


(despues de cada git add . tengo que hacer git push para que se suba el cambio que estoy haciendo porq sino al hacer varios solo guarda el ultimo que hago.)

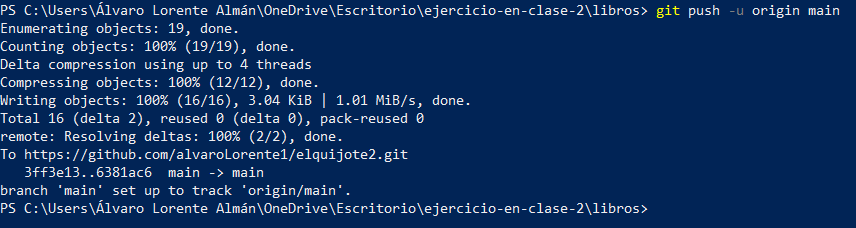
1. Edita el fichero README.md, añadiendo información sobre el repositorio. Si quieres saber más sobre el formato markdown, aquí tienes una hoja de trucos: <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>



1. Confirma los cambios finales con un mensaje de confirmación adecuado (sin utilizar la opción -m).



1. Sube los nuevos cambios al repositorio remoto.

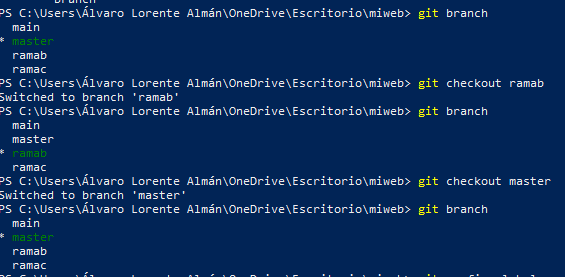


git brach nombrerama para crear una rama

git branch te dice en que rama estas

git checkout ramab para ir a ramab

git checkout master para ir a la rama master



git log para ver el estado actual de mi repositorio (para ver los commits que he hecho)

para referirme a un commit creo un tag y cuando lo construyo se crea un alias asociado a este commit

para crear tag:

git tag -a nombre del tag → ej: git tag -a v1.0

git tag veo el tag creado

git show v1.0

git tag -a v1.1

git log –oneline para ver los commits y poder referirme a uno de ellos

git tag -a v1.1 42343242

git tag para ver los dos tags

git push para subirlo

