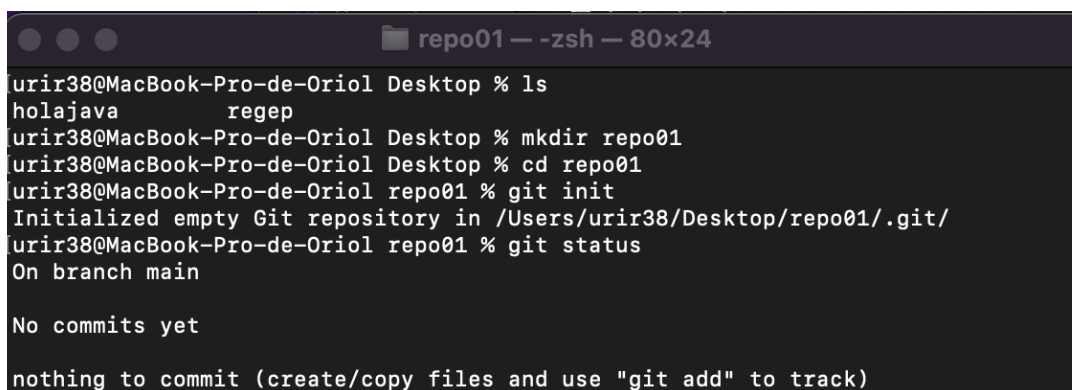


1-1 - Crea un directorio llamado `repo01` en local (desde tu máquina) e ejecuta el comando pertinente para que dicho directorio se transforme en repositorio en local ¿Cómo podemos identificar que el repositorio se ha inicializado?

- Primero de todo nos pondremos donde queremos crear el repositorio, en mi caso en el escritorio.
- Después crearemos nuestro repositorio con `mkdir` + el nombre del repositorio.
- Luego accedemos al repositorio con `cd` + nombre del repositorio.
- Y luego le decimos con `git init` que inicie el repositorio.
- Con el comando "`git init`" veremos si está iniciado o iniciaremos nuestro repositorio.



```
urir38@MacBook-Pro-de-Oriol Desktop % ls
holajava      regep
urir38@MacBook-Pro-de-Oriol Desktop % mkdir repo01
urir38@MacBook-Pro-de-Oriol Desktop % cd repo01
urir38@MacBook-Pro-de-Oriol repo01 % git init
Initialized empty Git repository in /Users/urir38/Desktop/repo01/.git/
urir38@MacBook-Pro-de-Oriol repo01 % git status
On branch main

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

1.2 - Añade un documento llamado `readme.md` dentro del repositorio (recuerda que MD es la extensión de los ficheros Markdown) y documenta en su interior todos los pasos que vas a realizando para crear un repositorio, etc. Puedes añadir fotos o lo que crea conveniente.

Ahora nos iremos al Visual Studio Code y dentro crearemos un archivo "`readme.md`", en el que escribiremos básicamente lo que ya estamos.

Seguidamente nos iremos a github y crearemos el repositorio, una vez creado le diremos que ya tenemos uno creado, y nos dará un link a nuestro repositorio virtual para que se sincronicen con el local que hemos creado.

Para poder subir lo que tenemos, iremos a nuestro terminal dentro del VSC y le añadiremos lo siguiente:

```
git remote add origin https://github.com/UriR38/repo01.git
git branch -M main
git push -u origin main
```

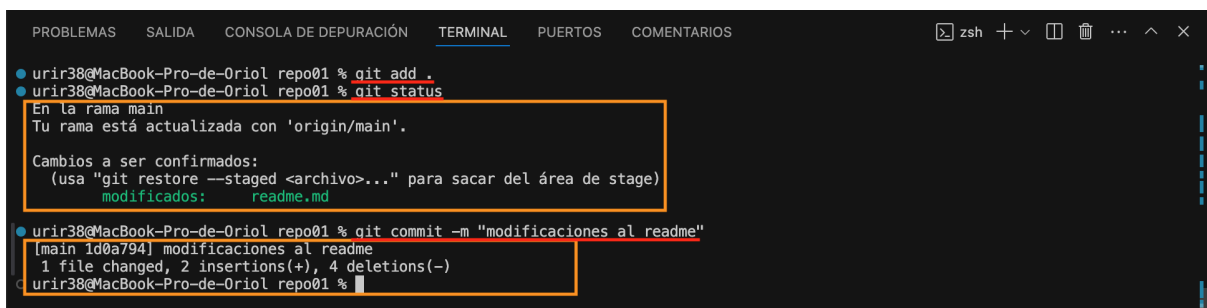
Una vez hecho esto, si nos vamos al repositorio en github veremos que se han sync.

1.3 - Añade el fichero que acabamos de añadir al repositorio al **staging area**, visualiza el estado del repositorio (con `git status`) y haz un **snapshot** (commit) del fichero hacia nuestro repositorio local. ¿En qué “file status lifecycle” se encuentra el fichero?

En la staging área es donde los cambios que hemos subido están esperando a que les pongamos un **commit** para subirlo o cancelar su subida, como podemos ver en el primer recuadro naranja.

Una vez hacemos el **commit**, como vemos en el segundo recuadro naranja, nos muestra los archivos que se han modificado, y si se ha añadido o borrado cosas.

Nuestro fichero sigue en la **staging area**, pues está esperando a que le hagas el push, o que canceles el commit.



```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  COMENTARIOS  zsh + v [ ] [ ] ... ^ x

• urir38@MacBook-Pro-de-Oriol repo01 % git add .
• urir38@MacBook-Pro-de-Oriol repo01 % git status

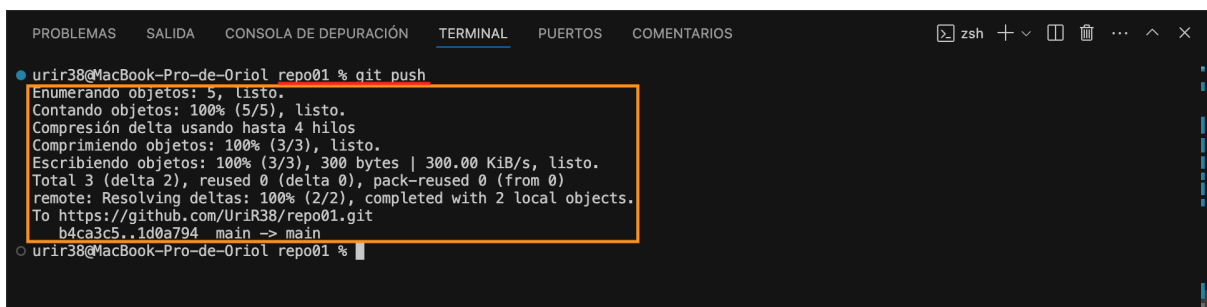
En la rama main
Tu rama está actualizada con 'origin/main'.

Cambios a ser confirmados:
(usa "git restore --staged <archivo>..." para sacar del área de stage)
modificados:   readme.md

• urir38@MacBook-Pro-de-Oriol repo01 % git commit -m "modificaciones al readme"
[main 1d0a794] modificaciones al readme
1 file changed, 2 insertions(+), 4 deletions(-)
• urir38@MacBook-Pro-de-Oriol repo01 %
```

1.4 - Intenta subir los ficheros al repositorio remoto mediante al comando **git push** ¿Se te ocurre qué está pasando? (si no lo sabes aún no te preocupes)

Lo que está haciendo es subir los archivos que hemos enviado con el commit, y reemplazar los cambios que hemos hecho a los ficheros que tenemos en el Github, y en este caso se están subiendo y modificando a la rama central (**main**).



```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  COMENTARIOS  zsh + v [ ] [ ] ... ^ x

• urir38@MacBook-Pro-de-Oriol repo01 % git push
Enumerando objetos: 5, listo.
Contando objetos: 100% (5/5), listo.
Compresión delta usando hasta 4 hilos
Comprimiendo objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 300 bytes | 300.00 KiB/s, listo.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/UriR38/repo01.git
   b4ca3c5..1d0a794  main -> main
• urir38@MacBook-Pro-de-Oriol repo01 %
```

1.5 - Ejecuta el comando **git remote -v** e investiga porque no nos aparece nada.

No funciona porque no está viendo/conectado al repositorio remoto.

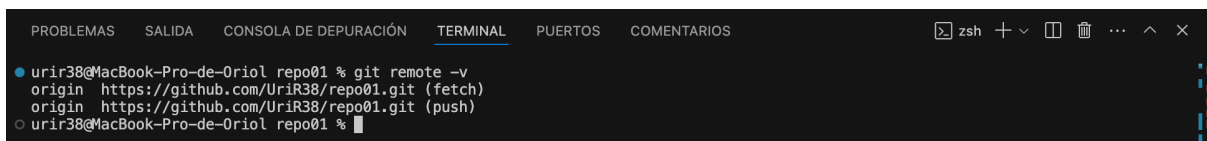
1.6 - Crea un repositorio remoto llamado **repo01**, asócialo a tu repositorio local.

Desde github creamos el repositorio y luego lo vinculamos con el nuestro con el siguiente comando:

```
git remote add origin https://github.com/UriR38/repo01.git
git branch -M main
git push -u origin main
```

1.7 - Vuelve a ejecutar el comando **git remote -v** nuevamente y explica el porque ahora sí que aparece.

Ahora ya se ve porque ya están conectados tanto el repositorio local, como el remoto.



```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  COMENTARIOS
urir38@MacBook-Pro-de-Oriol repo01 % git remote -v
origin  https://github.com/UriR38/repo01.git (fetch)
origin  https://github.com/UriR38/repo01.git (push)
urir38@MacBook-Pro-de-Oriol repo01 %
```