Comandos Git

Ezequiel Remus — Email: ezequielremus@gmail.com

ÍNDICE

1. INICIALES

Tener en cuenta que el doble guión debe ser consecutivo.

Todos los comandos git comienzan con la palabra **git**.

1. *git - -version*:
   Para conocer la version de git que tenemos.

2. *git config - -global user.name "nombreDeUsuario"*:
   Lo primero que tenes que hacer después de instalar **git** es establecer tu nombre de usuario (y también después de correo electronico). Este comando sirve para establecer tu nombre de usuario.

3. *git config - -global user.email "TuCorreo"*:
   Lo primero que tenes que hacer después de instalar **git** es establecer tu nombre de usuario y correo electronico. Este comando sirve para establecer tu correo.

   Para mas información sobre el comando *config* utilizar el comando: git - -help config

2. REPOSITORIOS

Hay varias formas para trabajar con repositorios, en particular la que explico primero sin haber creado el repositorio desde git y otra habiendo primero creado un repositorio (hablando de repos propios, ya voy a explicar como laburar con repos en grupo). Yo en particular les recomiendo la segunda que es mas rapida (para mi).

Empecemos con la primera:

Creamos una carpeta, abrimos git bash y corremos el comando *git init*. Este comando lo que hace es decirle a git que inicie el seguimiento de la carpeta sobre la cual corrimos el comando. El comando crea una carpeta oculta llamada *.git* donde git almacena su base de datos. Para ver esta carpeta, corremos el comando *ls -a* este muestra todos los directorios, incluso los ocultos.

Luego, dentro de nuestra carpeta repositorio creamos el archivo **README.md**. Este archivo es un archivo de texto markdown el cual se utiliza para realizar la descripción del repositorio e inclusive dar una idea de lo que esta documentado. Dentro de este pondremos el titulo del repositorio de la siguiente forma

$$\#NombredelRepositorio$$

Este va a ser nuestro titulo del *README.md*.

Luego de guardar el *README.md* corremos el comando *git status* que nos sirve para conocer en que estado esta el proyecto. Los estados pueden ser 2, todo rojo o todo verde. Si esta en rojo, significa que el archivo esta modificado y no esta subido (todabia esta en el working directory) a la base de datos git, en cambio si esta en verde significa que lo subis al *staging area*, que es el area de preparacion. Para mover el archivo de la zona de trabajo a el area de preparacion se utiliza el comando *git add .*, al pasarle el punto le decis que suba todos los modificados, si queres solo subir uno de esos archivos le pasas el nombre del archivo con su extencion.

Luego, para que el archivo forme parte de nuestro historial de proyecto es necesario hacer un **commit**, esto es subir el archivo al directorio de git. el comando completo es: *git commit -m "Comentario que represente al commit"*. (*No necesariamente se abre vim para agregar el mensaje, que es lo que dice en el tutorial, lo mas rapido es hacer el comando como se los pase yo sin tener que entrar a ningun editor de texto para escribir el mensaje*).

Si corremos *git status* despues de este paso, vamos a ver que nos dice que ya esta todo listo para subir. Para poder subirlo tenemos que asociar la carpeta a un **repositorido github** para luego realizar un **push**.

Para hacer un repositorio, entramos a nuestra cuenta github y nos logeamos.

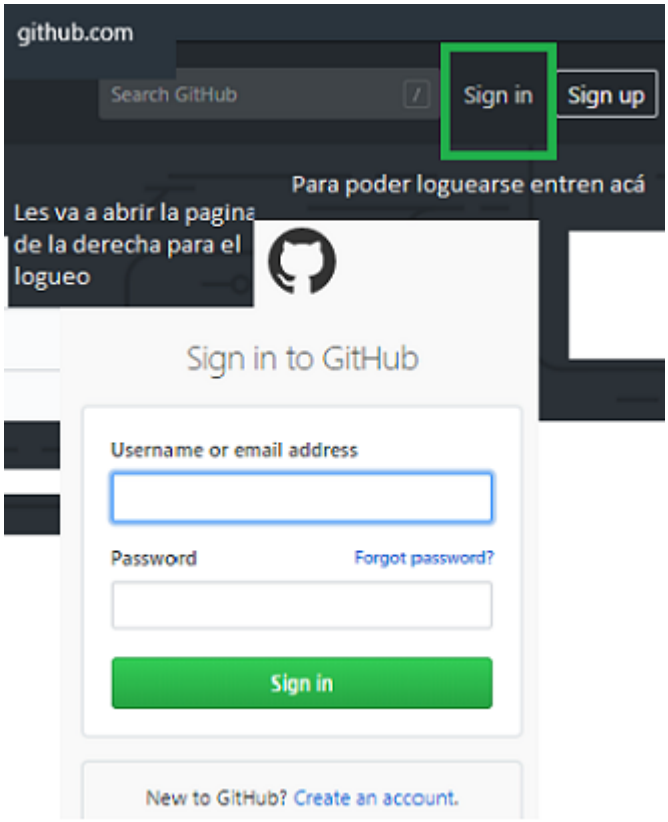

Figura 1. Pagina de github y logueo (aparecen en paginas diferentes)

Despues del logueo van a la solapita con un signo de suma y apretan en **New Repository**

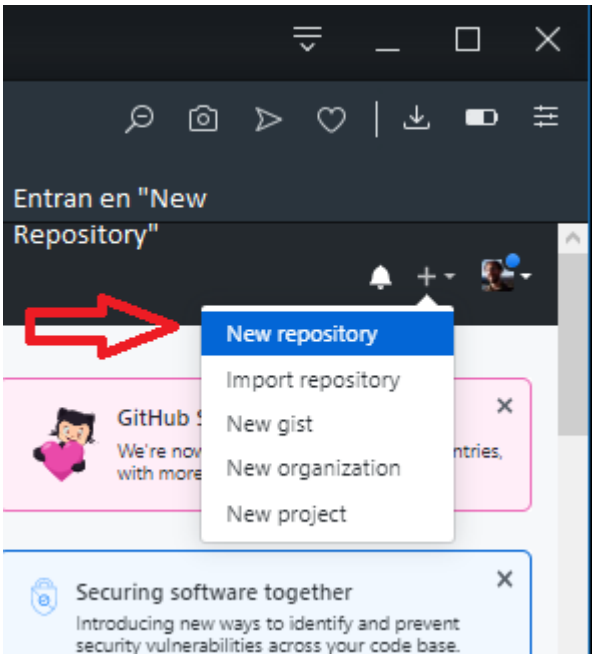

Figura 2. Una vez logueados podemos crear un repo nuevo entrando a ese Boton

Al entrar nos abre una ventana donde debemos tener en cuenta un par de cosas:

Primero que nada, tenemos que ponerle un nombre si o si al repo (esto es poque sino no sabe a que corno apuntan los datos).

Por otra parte, *yo prefiero en particular ahorrarme los pasos anteriores y crear primero el repositorio* con el *README.md* desde la pagina (estamos hablando ya de la otra forma de las que les hable), te ahorras un par de comandos y es basicamente lo mismo.

Los pasos anteriores sirven cuando ya tenes algo hecho y queres subirlo a un repo que ya existe o queres hacer un seguimiento de lo que fuiste haciendo a travez de los commits sin subirlo a un repositorio, en nuestro caso queremos trabajar con repos, por lo que les recomiendo primero crear el repositorio, clonar este en su pc (eso ya se los digo, es sencillo), poner los archivos que quieran subir y correr los comandos que venimos viendo de reconocimiento de estado, add y commit para poder subirlo y hacer el push directamente. Esto lo digo para que no se tengan que romper la cabeza para entrelazar la carpeta que creamos con las ramas del repositorio (origin y master) mediante comandos.

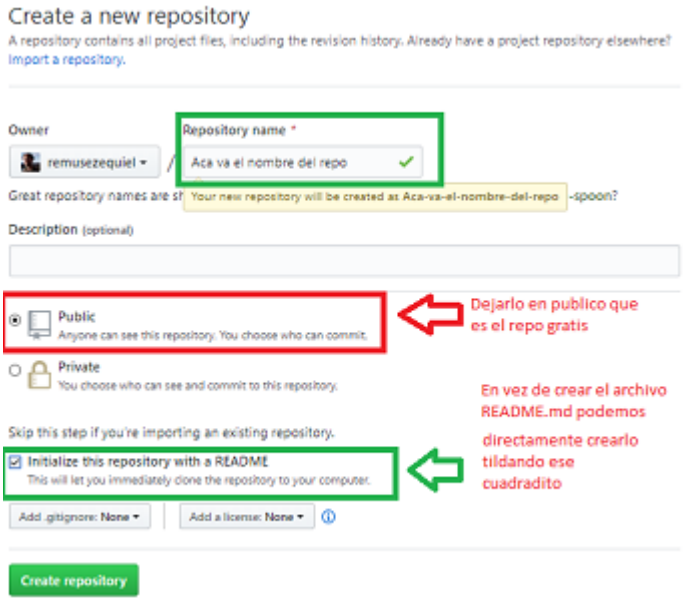

Figura 3. Creando el repo

Bueno, seguimos los pasos que aparecen en la *Figura 3* y listo, ponemos **create repository** y se crea el repositorio. Yo cree uno para poner este pdf y despues voy a usarlo para poner apuntes, les dejo el link a este repositorio en las referencias por si lo quieren clonar cuando suba mas cosas y sino para que sepan que tambien pueden clonar repositorios de otros sin modificarles nada a los demas y asi poder leer las cosas de un repoistorio x que les interese de una forma mas comodo.

Bueno, siguiendo con esto de los repositorios, una ves creado el repositorio nos manda a la pagina del repo, que van a ver algo como esto:

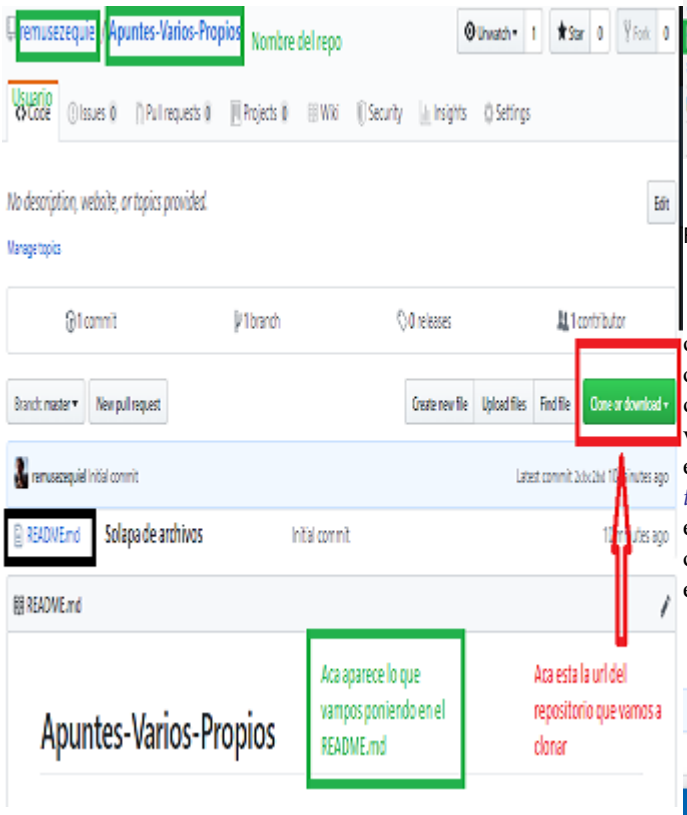

Figura 4. Repo nuevo

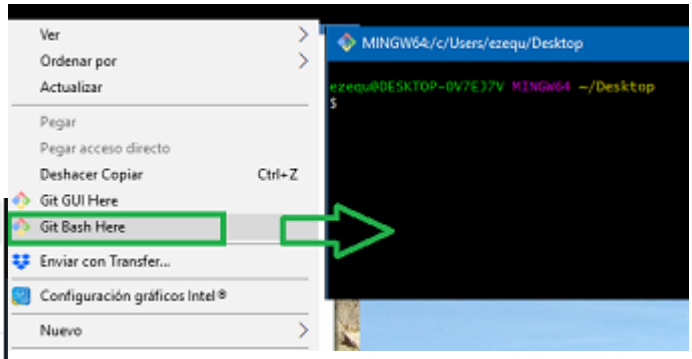

Figura 5. Terminal desde windows

Sino, si trabajan en linux y instalaron git con el comando *sudo apt-get install git* abren una terminal de linux y corren los mismos comandos que voy a poner ahora, solo que clonan el repositorio que quieran clonar. En este caso, voy a clonar el repo que hice para este apunte. Para clonar el repositorio se utiliza el comando *git clone urlDelRepositorio*. El url lo sacan de la solapa **clone Repository** que esta en el repositorio, lo copian y lo pegan despues del comando git clone. Verifiquen que esten parados sobre el escritorio, sino se los va a clonar en otra ruta.

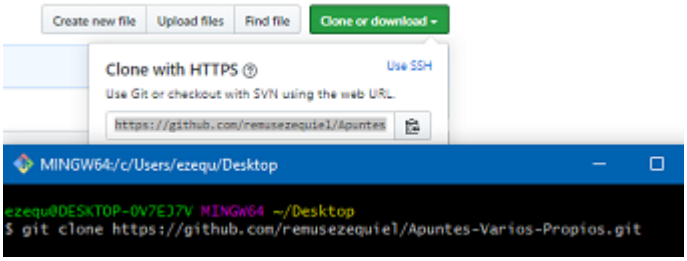

Figura 6. Clonando el repositorio en el escritorio

Ese es lo que se llama repositorio. Tenemos la parte de los archivos, donde ya de entrada vemos que aparece el archivo README.md, abajo de los archivos aparece lo que vamos escribiendo dentro del README (Si quieren no les ponen nada) este archivo es una guia sobre el repositorio, un resumen de lo que hay o muchas veces te explican como se usan o como se instalan los repositorios. Por ejemplo, estube viendo que el manejador de paquetes de Haskell *Cabal* tiene su repositorio de github (dejo el link en referencias).

Ahora, para trabajar con este repositorio, tenemos que clonarlo en nuesto escritorio. Para eso, abrimos la terminal desde nuestro escritorio. tenemos que tener en cuenta, que si instalamos git en linux abrimos una terminal comun y nos corremos al Escritorio y ya esta, en windows, si siguieron los pasos del tutorial que puse en referencias para instalarlo e instalaron esa terminal que dice el pibe seria asi:

Vamos a ver que nos clona la carpeta en el escritorio con el nombre del repo. Abrimos la carpeta y vemos que el contenido es el mismo mas la carpeta oculta *.git* de la que hablamos antes.

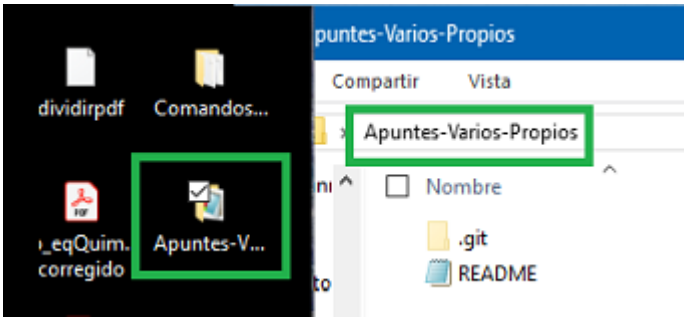

Figura 7. Clonando el repositorio en el escritorio

Ahora, lo que voy a hacer es poner la carpeta donde esta este pdf en el repositorio y voy a seguir los pasos de arriba:

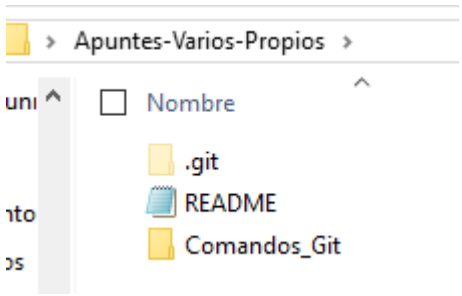

Figura 8. Pongo mi proyecto en el repo

Ahora, abro una terminal sobre el repositorio o me corro adentre del repositorio, pero asegurense de estar dentro de la carpeta del repositorio y corro *git status*.

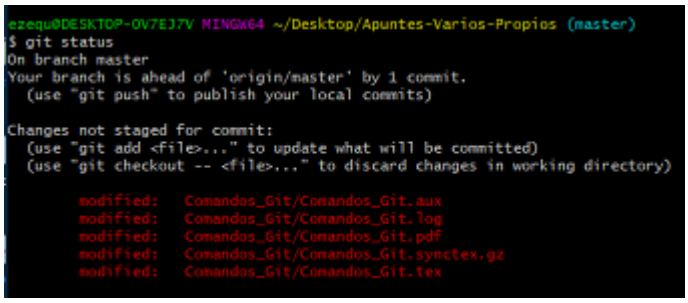

Figura 9. Realizo un status

Como vemos, esta en rojo y nos dice "a estas cosas les podes hacer el add . ". Hacemos el *add .* y corremos git status denuevo.

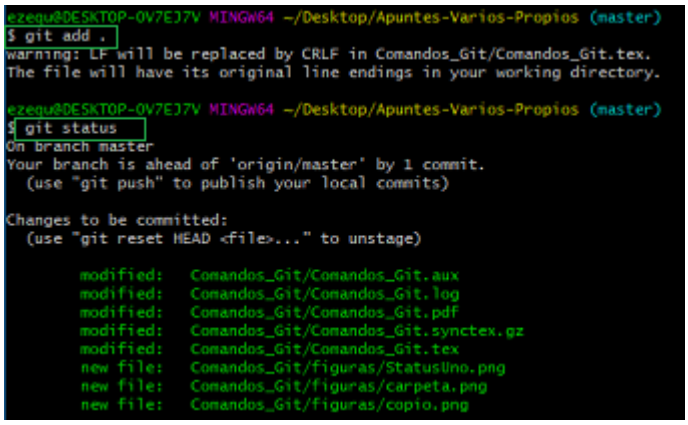

Figura 10. Realizo el add .

Ahora, hacemos el *commit*:

el comando *git push origin master*. Este comando lo voy a explicar mejor un poco mas adelante. Lo que es necesario entender hasta aca es que este comando lo sube a tu nube. Al ir a tu pagina en github, vas a ver que se renovo el repositorio de tu cuenta.

REFERENCIAS

[1] *StackOverflow*, Link: https://stackoverflow.com
[2] *Link a un cursito de git basico*,
Link: https://www.youtube.com/watch?v=m4wh8GhzcYg&list=PLmUnyBCRHkvUPkrsseI1SmMtYgfc-f8Kn&index=3
[3] Cuenta propia de github .
Link: https://github.com/remusezequiel?tab=repositories
[4] Repositorio de github donde se encuentra el codigo de este proyecto .
Link: https://github.com/remusezequiel/Apuntes-Varios-Propios
[5] Repositorio de github donde se encuentra el proyecto Cabal .
Link: https://github.com/haskell/cabal