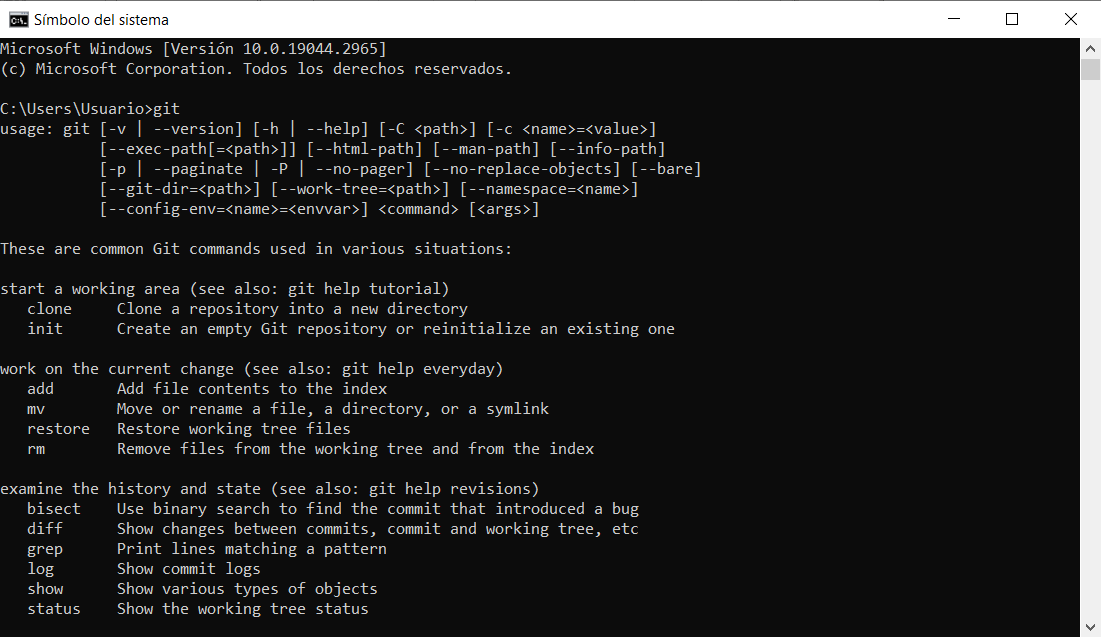
**GIT** es un sistema de control versiones, creado por **Linus Torvalds** (figura importante del mundo de la programación, creador de Linux).

Es de código abierto, tiene un libro gratis y en español en la misma página web de Git. <https://git-scm.com/> Desde esta web podemos descargar GIT.

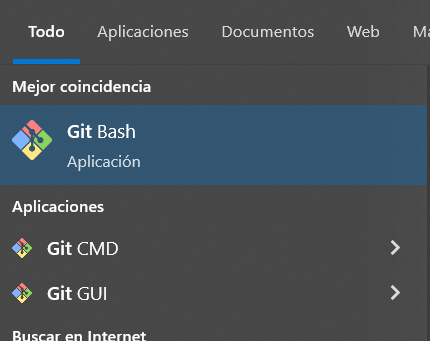
Tiene una versión de comandos y unas versiones gráficas. Para entender mejor el proceso es mejor usar la versión de comandos. Los comandos funcionan igual en Windows, MAC, Linux…

Una vez instalado, ya podemos trabajar con GIT. si abrimos el CMD y ponemos git saldrá algo como esto.



Esto significa que ya tenemos instalado el software.

Ahora para utilizarlo, según entiendo puedes usarlo desde una ventana CMD, pero como la idea es moverse por el terminal, para Windows tenemos GIT Bash, que es como una terminal Linux para poder movernos por todos los directorios.



Git tiene muchos comandos, pero con unos 10 seremos capaces de realizar el 90% de lo que necesitemos hacer.

**Configuración de GIT.**

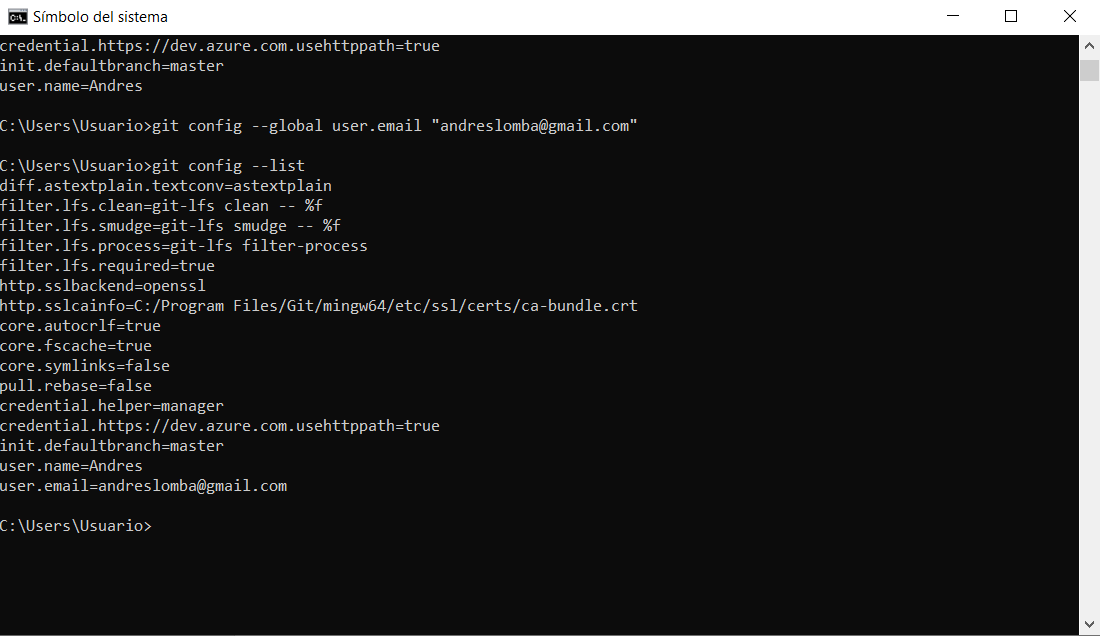
Lo principal e imprescindible que necesitas configurar para poder empezar a usar GIT es un nombre y un email. Para esto iremos a la consola y pondremos los siguientes comandos.

git config –-global user.name “nombreUsuario”

git config –-global user.email “email@email.com”

si vamos a la carpeta base de Git y de ahí al archivo de configuración. Nos saldrán el nombre de usuario y el email que hemos puesto. También podemos ver si están desde la consola con el comando.

git config --list



**INICIALIZAR GIT**

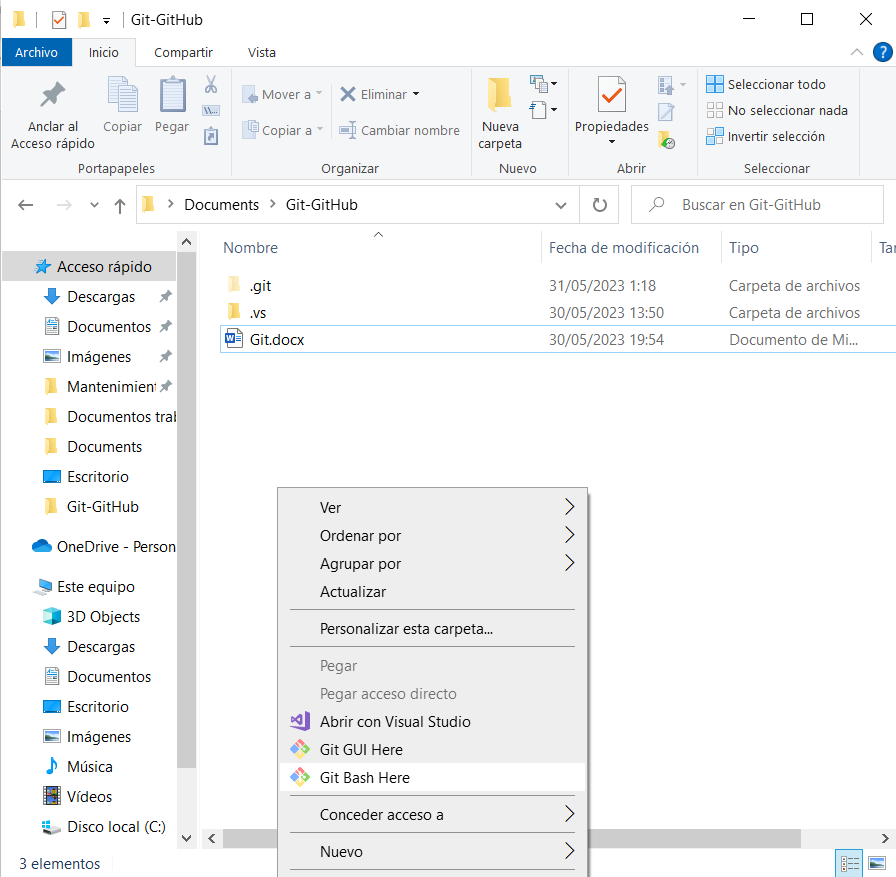
¿Cómo funciona GIT? ¿En qué carpeta estaría el repositorio donde tendremos todas las versiones de nuestro proyecto?

La carpeta la eliges tú. Iremos moviéndonos a través de la consola hasta la carpeta donde queramos alojar nuestro proyecto, y una vez dentro de la carpeta, ejecutaremos el comando.

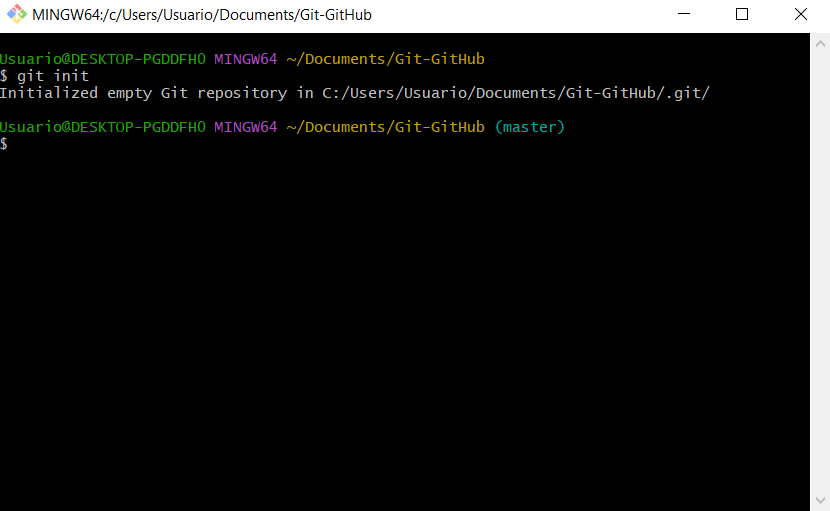
git init

Este comando nos creará, dentro de nuestra carpeta donde alojamos el proyecto, una carpeta oculta con el nombre de .git.

En este caso, lo que hice yo fue ir a la carpeta en cuestión, dar click derecho una vez dentro de la carpeta y seleccionar la opción Git Bash Here.

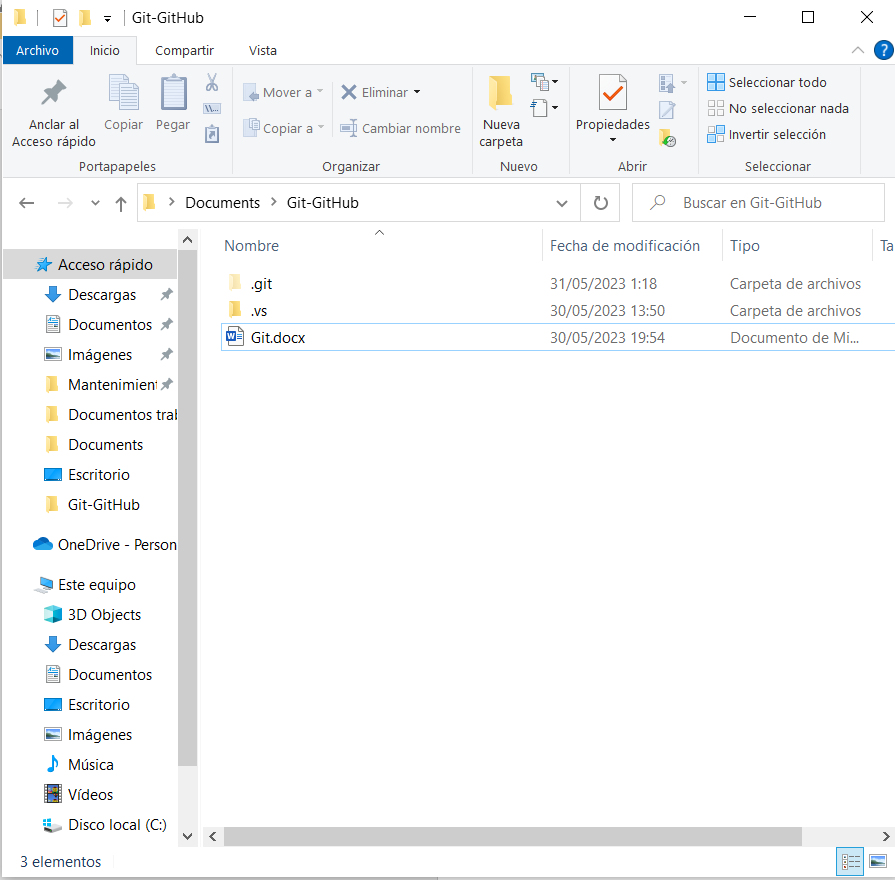


Se me abre una consola y en ella pongo el comando git init.

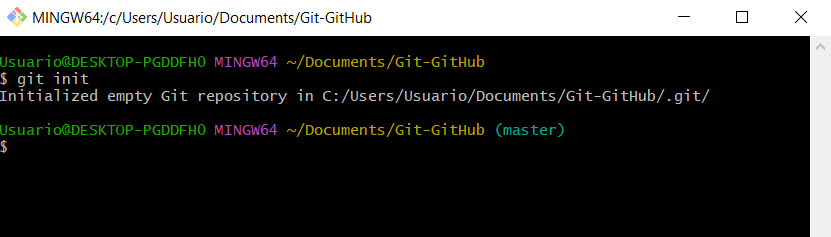


Ocurrieron dos cosas:

* Me generó una carpeta .git dentro de la carpeta donde tengo el proyecto.



* Ahora en el bash, si nos fijamos, en el directorio que estamos sale la palabra (master)



Git se entiende como una ramificación. Donde su rama principal sería esta carpeta que dice master.

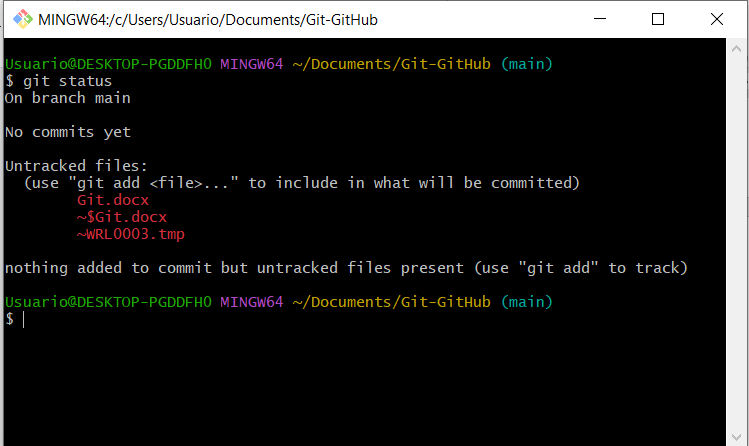
Si preferimos llamar de otra manera a nuestra rama principal, podemos utilizar el comando

Git Branch -m Nombre

Nota: Por lo visto, es común denominar a la rama principal como main.

Una vez tenemos nuestra rama principal podemos mirar el estado en el que está.

Git status



Nos indica

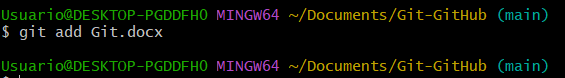
* Que estamos en la rama main.
* Que no hay commits (más adelante en el documento explicaremos que es esto).
* Y que hay un archivo Git.docx. que no se ha subido como versión.

El archivo WRL0003.tmp es un archivo temporal que crea Windows aparentemente y no es relevante para la explicación.

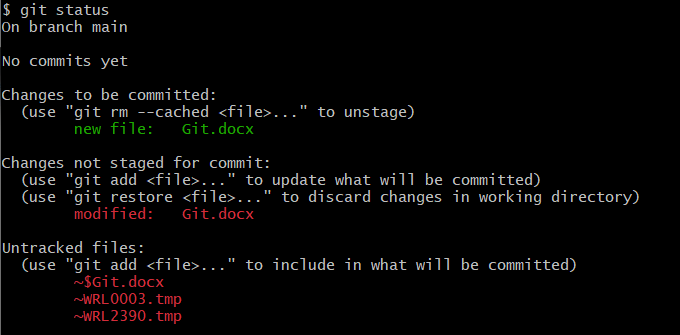
Hasta ahora se entendería que todo lo estamos trabajando de manera local. De momento nuestro Git no tiene ninguna versión de ningún archivo. Entonces lo primero que habría que hacer es añadir ese archivo, el cual guardará como si de una foto única se tratase y cada vez que subamos otra versión de ese mismo archivo será como otra foto diferente.

Para realizar esa “fotografía” de nuestros archivos realizaremos dos comandos.

git add Nombrearchivo o también podemos usar git add. Para añadir todos los archivos de la carpeta a la vez.



Si hacemos git status vemos como ha añadido el fichero, pero aún falta realizar el commit para hacer esa “fotografía”.



El comando que utilizaremos ahora será

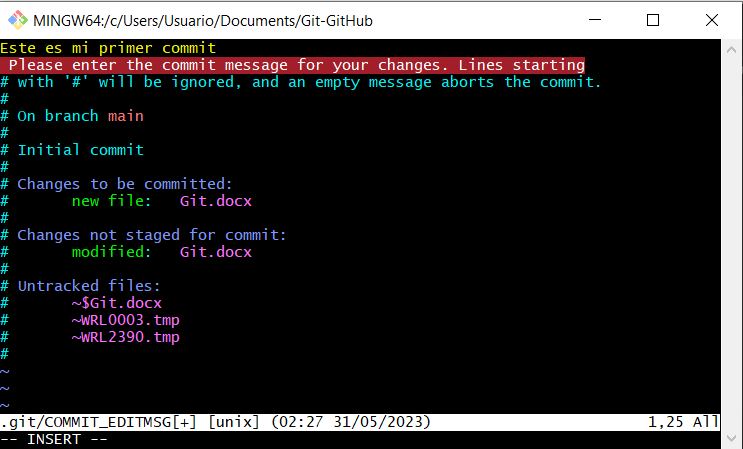
Git commit

Git commit -m “Este es mi primer commit”

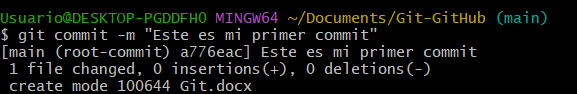
Si simplemente ponemos git commit nos saldrá el siguiente mensaje.



Esto nos dice que no podemos terminar el proceso sin poner un comentario. Con lo cual pondríamos en la primera línea un comentario.



En este caso, voy a abandonar el proceso anterior escribiendo qa! y dando enter para realizar el commit de la forma sencilla.



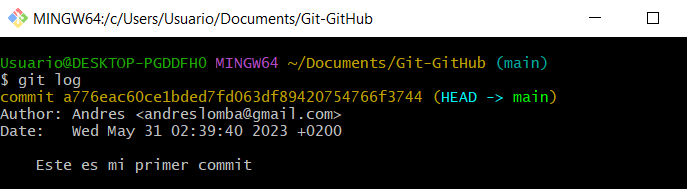
Esto nos crea la instantánea de nuestro archivo, generándole un identificador único que en este caso sería a776eac (a este identificador único se le denomina hash).

Con esto ya tendríamos la primera versión de nuestro archivo. Si hacemos status veremos como ha agregado la primera versión de nuestro archivo.

**Comando log**

git log

Este comando lo usaremos para verificar que efectivamente ha guardado la versión de nuestro archivo.



Nos dice el hash que tiene esa instantánea, el autor, la fecha y el comentario.

Por eso es muy importante tener configurado el nombre y el email, para poder identificar quien ha guardado esa versión.

**Volver a una versión anterior.**

git checkout nombrearchivo

git reset

con estos comandos volveríamos a la última versión que tengamos guardada. Con checkout aparentemente especificamos un archivo en concreto y con reset se aplica a los archivos que el sistema encuentre que están modificados.