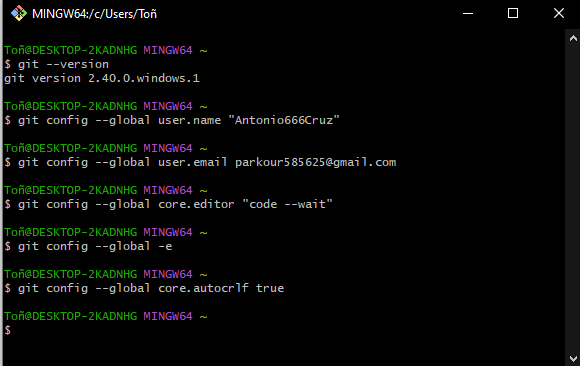
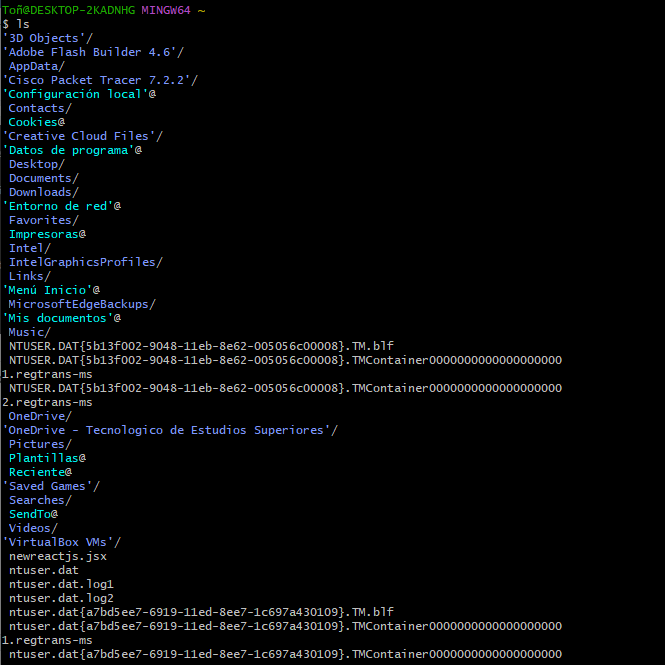
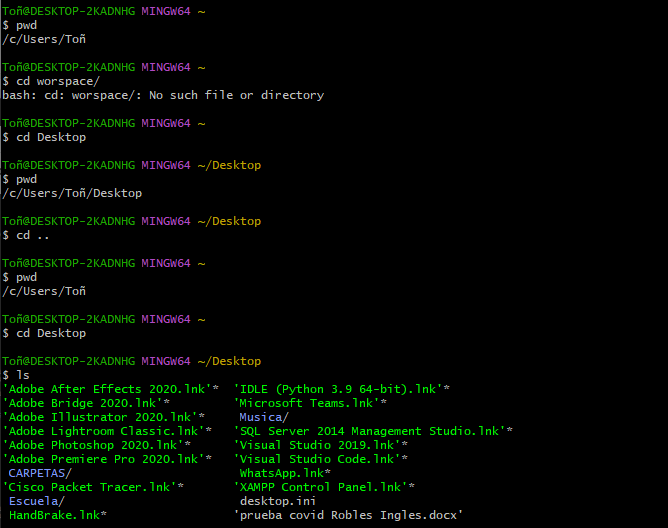
**Curso GIT**

Para configurar GIT primero hacemos la instalación, una vez instalado abrimos git bash y colocamos el comando para la versión que es **git –version**, después colocamos el nombre de un usuario para eso colocamos el comando **git config –global user.name “nombre del usuario”,** después hacemos la vinculación con nuestro correo electrónico con el comando **git –global user.email correo electrónico**, después hacemos la configuración con nuestro editar de código, en este caso es visual code con el comando **git config –global core.editor “code –wait”,** ahora para ver que se halla echo correctamente la vinculación con el editor de código colocamos **git config –global -e**, ahora con la configuración **git config –global core.autocrlf** true colocamos true por que se hizo la instalación en Windows.

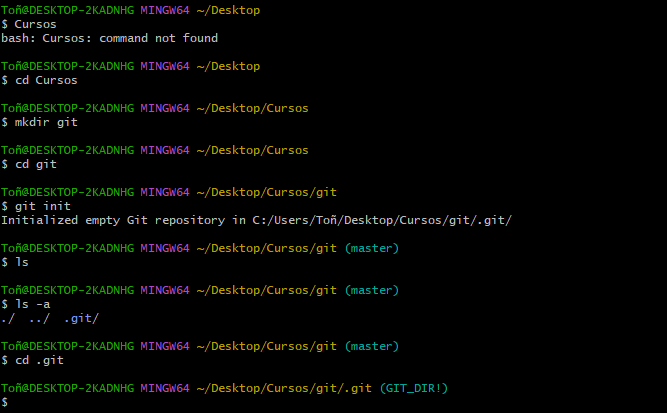


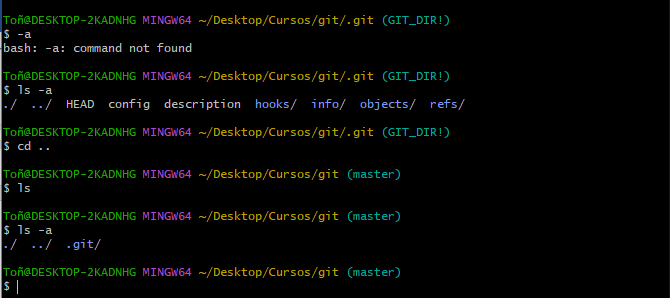


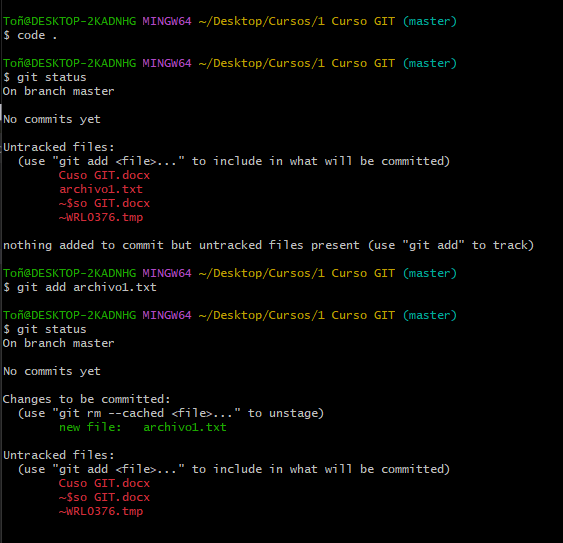
Con el comando **ls** nos mostrara todas las carpetas y archivos de nuestro equipo, para saber dónde nos encontramos ubicados colocamos el comando **pwd,** ahora con el comando **cd Desktop** ya que es el nombre de la carpera a donde nos queremos direccionar, ahora colocamos **pwd** para ver que en verdad nos encontremos en la carpeta Desktop, colocando **ls** nos mostrara las carperas y archivos dentro de esta, nos direccionaremos a la carpeta Cursos con el comando **cd Cursos**. Nota si queremos regresar de carpeta colocamos el comando **cd ..**

Ahora colocaremos el comando **mkdir git** para crear una nueva carpeta por medio de la consola, colocamos el comando **cd git** para entrar a la nueva carpeta creada, ahora colocamos **git init** y nos mostrara un mensaje con un .git creado dentro de la carpeta si colocamos el comando **ls -a** nos mostrara el .git creado, ahora colocamos **cd .git** para entrar dentro del archivo.

Con el comando **clear** limpiamos la terminal

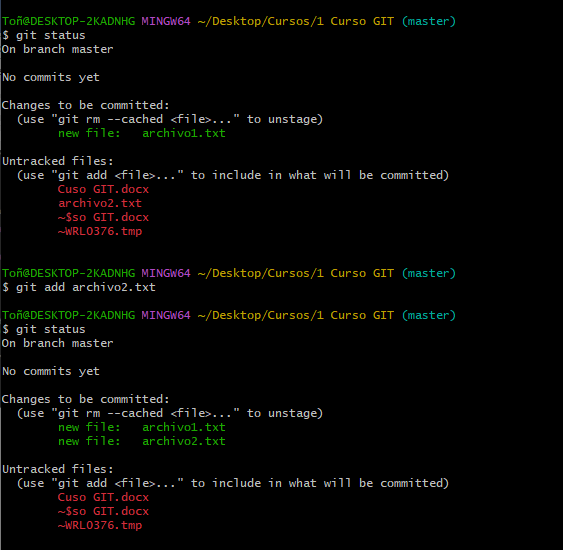
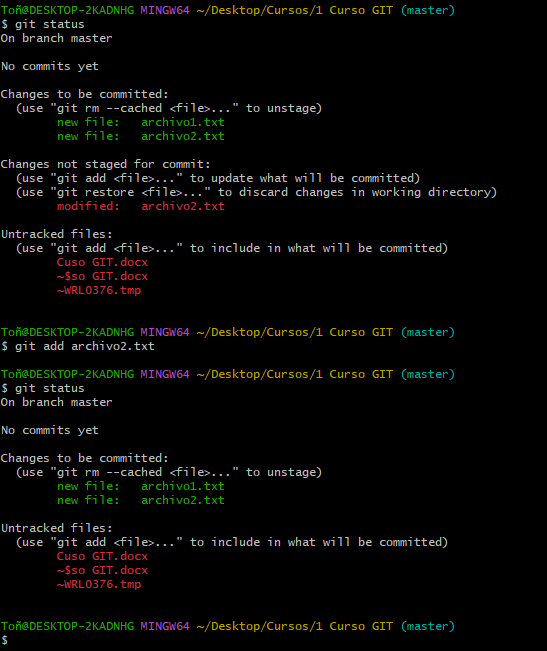


Ahora con el comando **ls -a** podemos observar todos los supositorios con los que trabaja git, salimos del archivo con **cd ..** y colocamos **ls -a** para observar que se encuentre el archivo .git

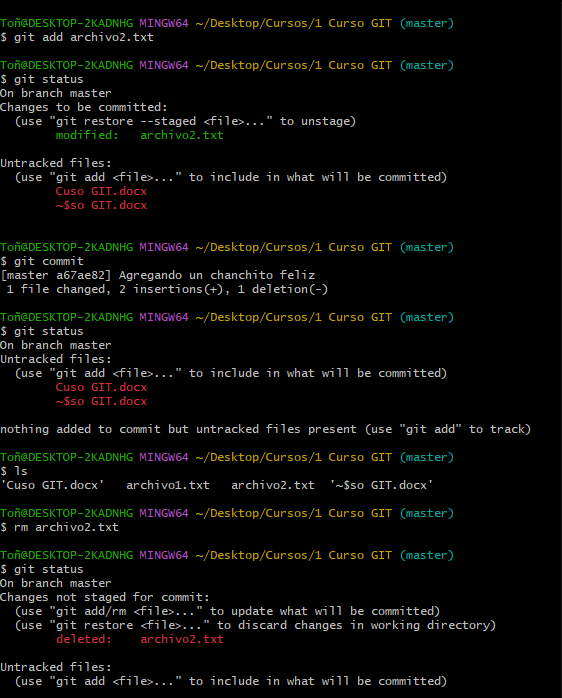
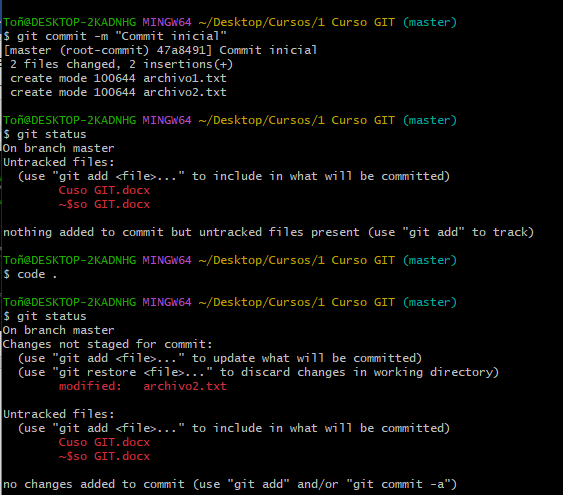


* **code . :** nos abrirá nuestro editor de código dentro de la carpera en donde estemos. Se creo un archivo llamado archivo1.txt con un texto.
* **git status:** nos muestra el estatus dentro de la carpeta.
* **git add nombredelarchivo:** nos permite agregarlo al estatus. Podemos observar que cuando colocamos git status nos muestra el archivo dentro del estatus y los que no están dentro.
* Si observamos en la imagen cuando se colocó otra vez git status nos mostro otro archivo llamado archivo2.txt, ahora si colocas git add archivo2.txt y mostramos el estatus nos muestra que los archivos archivo1.txt y archivo2.txt ya están dentro del estatus.
* Nota si se modifica la información dentro del cualquier archivo se debe de volver agregar al estatus, esa modificación puede observar en la siguiente imagen.

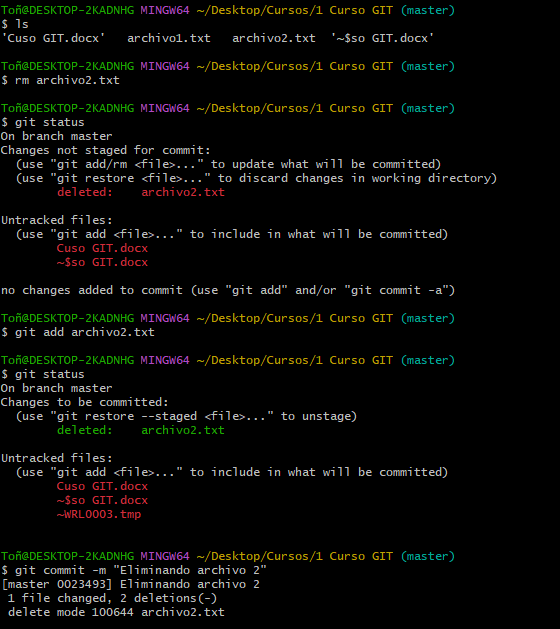
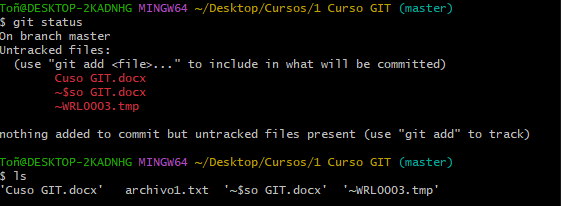
Modificación del archivo2.txt



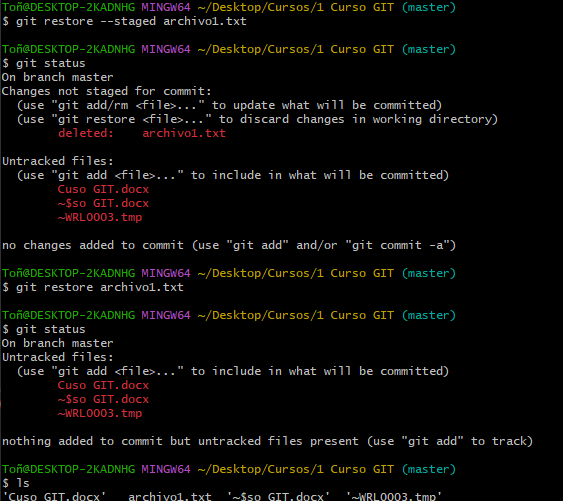
**Commit en GIT:** para poder comprometer un trabajado (commit), tenemos que colocar **git commit -m “Mensaje”,** después colocamos git status para ver que no hay archivos comprometidos. Ahora en el editor de texto realizamos un cambio dentro del archivo 2 y colocamos git status para ver la modificación del archivo, ahora lo volvemos agregar con el comando **git add archivo2** y colocamos git status para volver el estatus de la carpeta. Volvemos a realizar un commit, pero colocando el **comando git commit** esto hará que el editor de texto se abra y podamos agregar otro mensaje “Agregando un chanchito feliz”, podemos ver que en la terminal muestra el mensaje agregado

** **

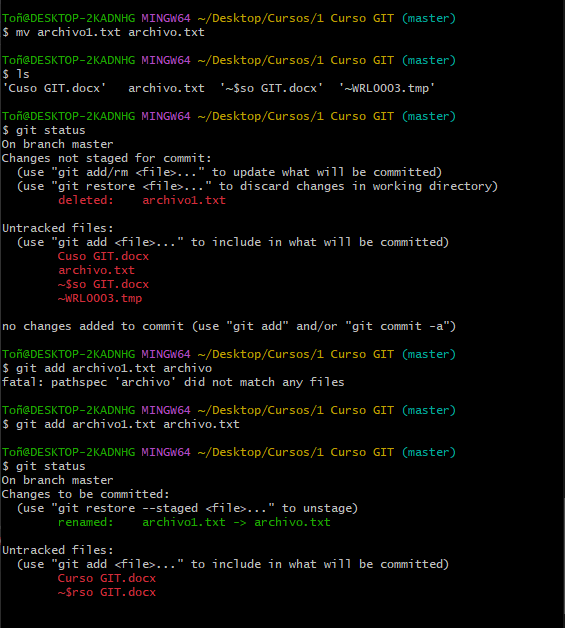
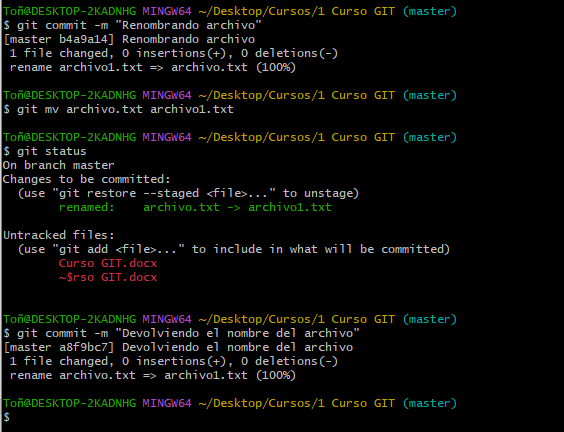
**Eliminar archivos:** para eliminar colocamos **ls** para observar los archivos y colocamos **rm y el nombre del archivo**, colocamos **git status** para observar que se elimino el archivo pero colocamos un **git add nombre del archivo** para que ya no nos aparezca archivos dentro de la carpeta, colocamos **git status** y nos muestra que ya se elimino el archivo, ahora colocamos un mensaje con **git commit -m “Eliminando archivo 2”** y nos muestra que se ha eliminado todo rastro ahora para corroborar colocamos git status y ya no nos aparece ningún registro.



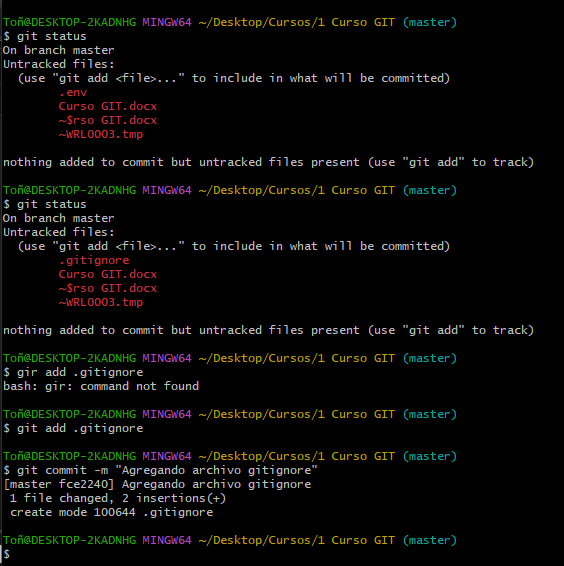
**Volviendo del state y descartar cambios:** colocamos **git restore –stage nombre del archivo y git status**, vemos que ya no se encuentra dentro del state, ahora colocamos **git restore nombre del archivo + git status**, después para corroborar que se recupero el archivo colocamos **ls** y nos parecerán los nombres de los archivos dentro de la carpeta.



Mover y renombrar archivos: colocamos el comando **mv y el nombre del archivo + el nuevo nombre**, colocamos **ls** y verificamos que se cambio el nombre de la carpeta, ahora colocamos **git status** y observamos que se borro el nombre del anterior nombre, ahora los añadimos con el comando **git add + el nombre del archivo anterior + el nombre del nuevo archivo**, colocamos el estatus con **git status**, con el comando **git commit -m “nota” hacemos el mensaje de renombramiento**, ahora volvemos a renombrar el archivo con el comando **git mv archivo.txt a archivo1.txtx**, comocamos el estatus con **git status y git commit -m “Devolvimos el nombre del archivo”** se volvió a renombrar.



**Ignorando archivos y directorios:** comenzamos por crear un archivo en el editor de código con la terminal **.env**, colocamos el estatus para ver si ya existe dentro de la carpeta con **git status**, ahora creamos otro archivo en el editor de código con terminal **.gitignore** pero dentro del archivo pones los archivos que queremos que ignore git en este caso se puso el **.env**, colocamos el estatus nuevamente y vemos que ya aparece el archivo .**gitignore**, ahora lo agregamos a git con el comando **git add .gitignore** y colocamos el comando **git commit -m “Agregamos archivo gitignore**”.



Un mejor git status: