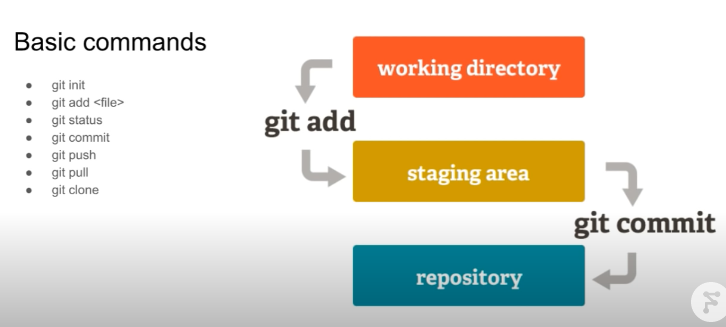
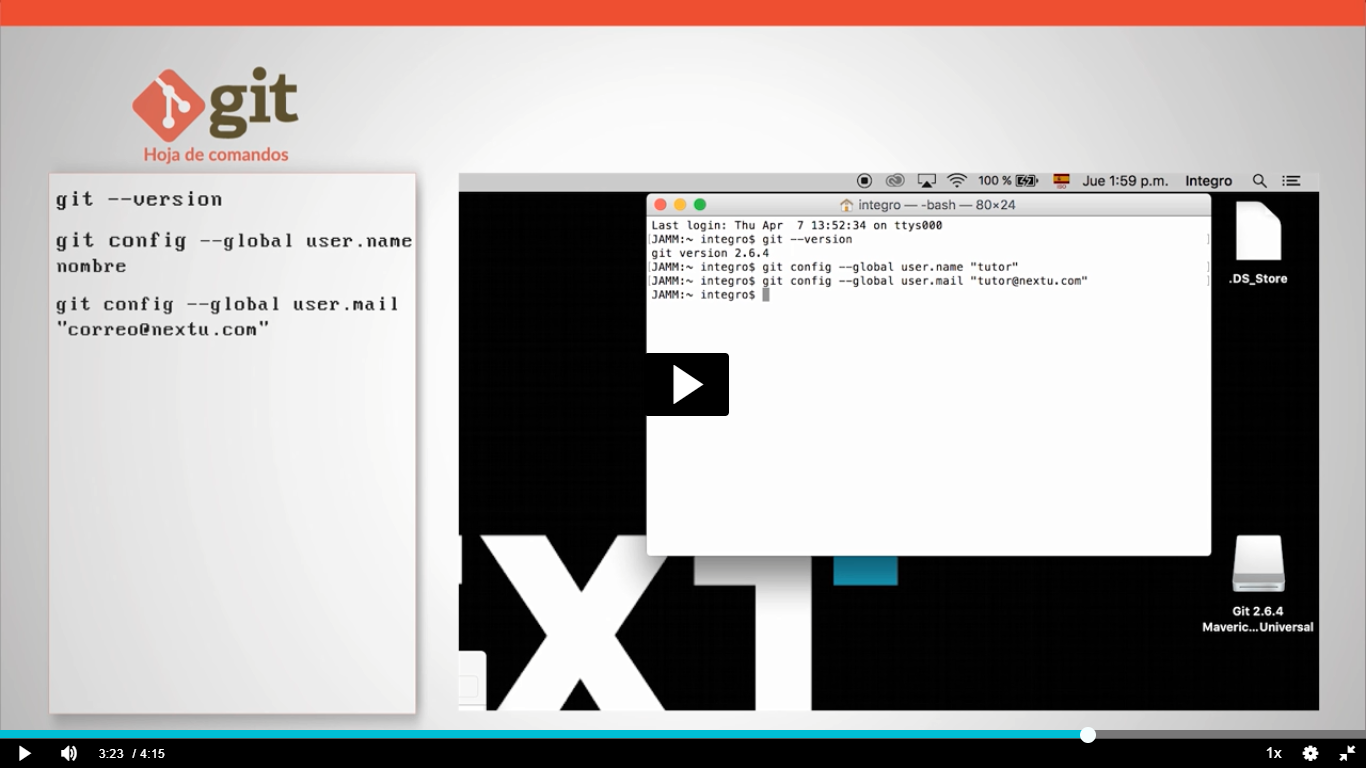
GIT

Sistema de control de versiones

COMANDOS:



Se ejecutan entrando a la consola de comandos; buscar en windows CMD, ejecutar -> Git CMD



Para la version que estamos usando

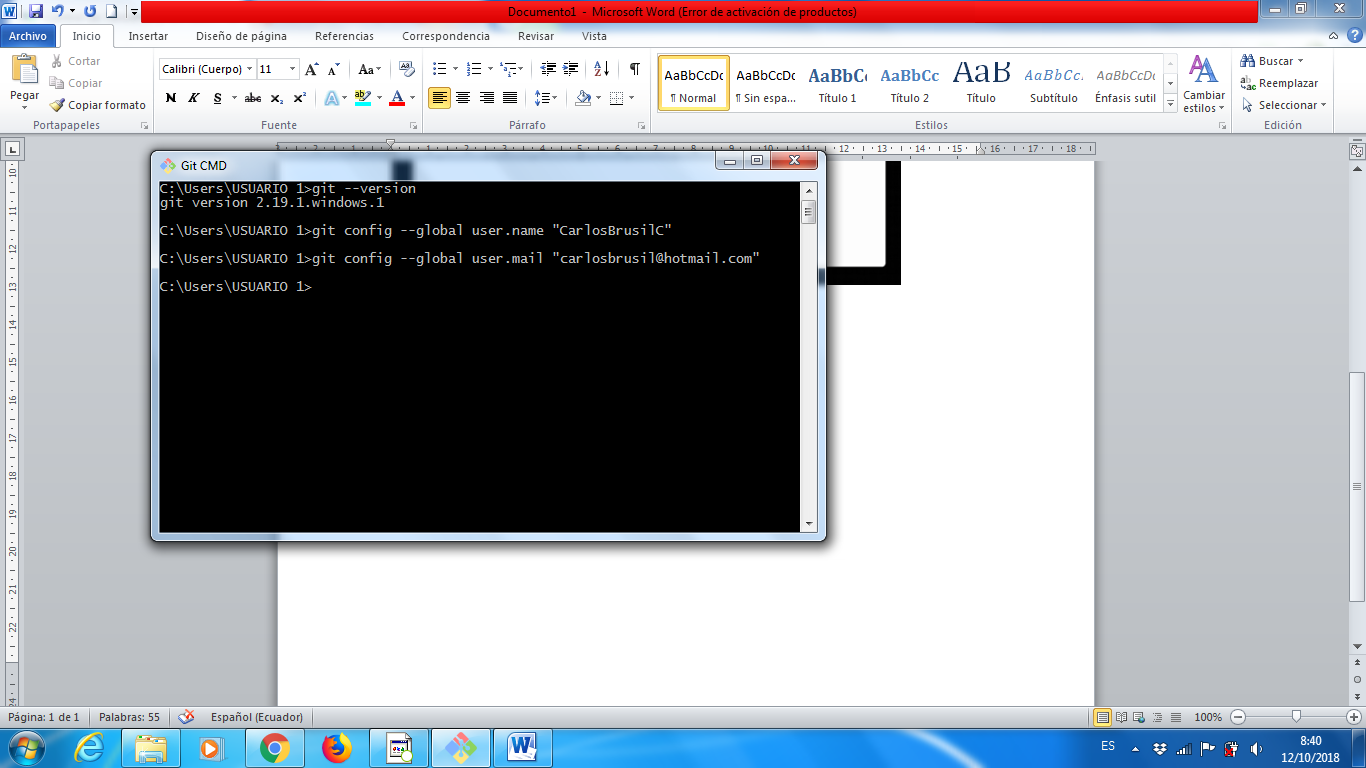
git --version

Para crear y configirar el usuario

git config –global user.name “nombreuser”

Para asignar el correo electronico del usuario

git config –global user.mail “correo”



INICIAR UN REPOSITORIO EN GIT

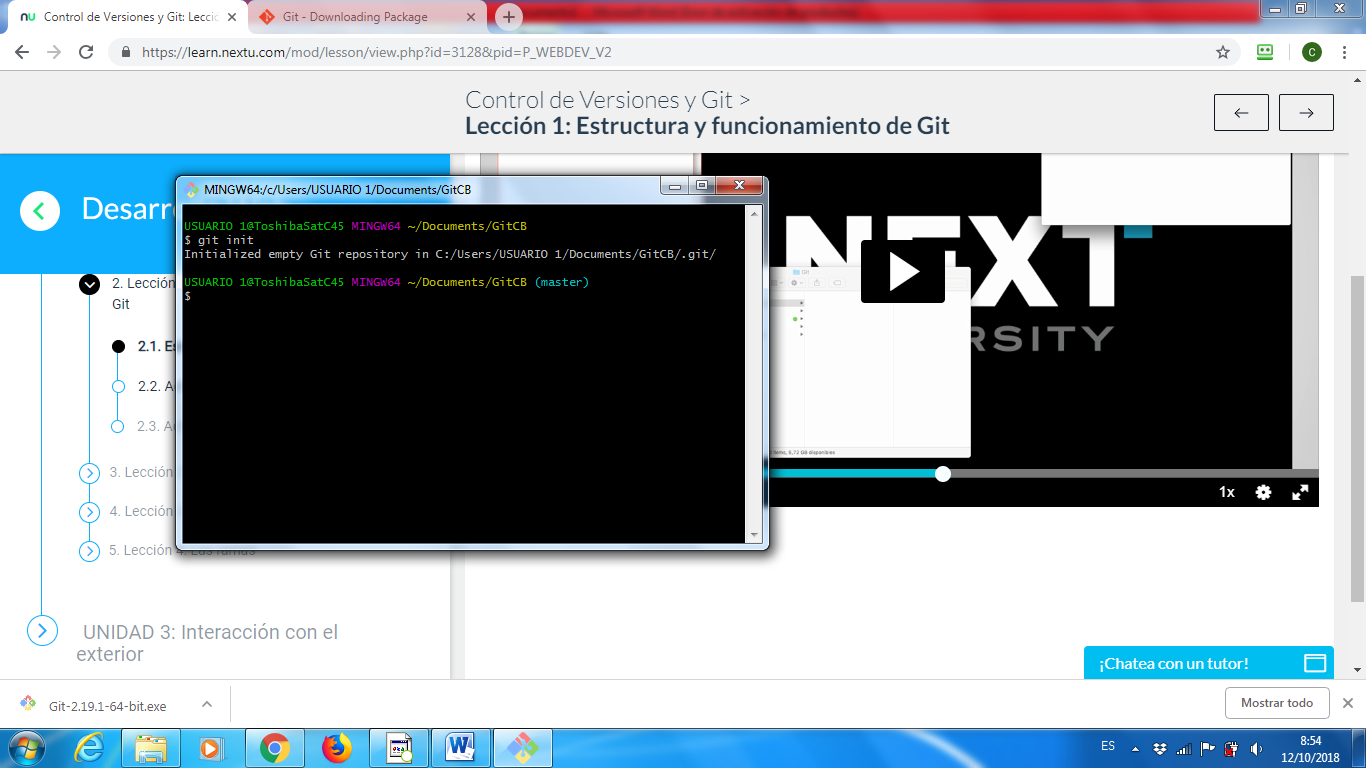
1.-Crear una carpeta en nuestra maquina donde se guardaran todos los documentos de GIT

Documentos, CLIC\_DERECHO, crear nueva carpeta ,asignar un nombre-> “GitCB”

2.- CLIC\_DERECHO, en GitCB, para iniciar el modo de comandos

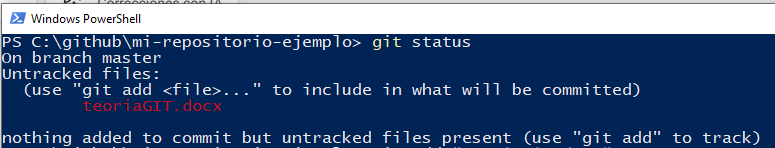
3.- Para iniciar un repositorio vacio

git init



4.- Para ver el estado de trabajo en git

git status

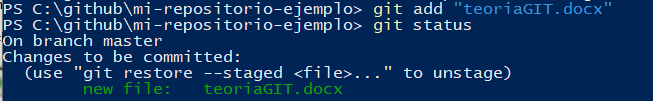


Untracked files: archivos sin seguimiento Nos informa que no estan seguidos algunos archivos en “On branch master” la rama principal

5.- Para añadir un archivo al seguimiento

Git add .

Git add “teoriaGIT.docx”

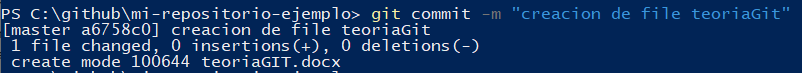


Al consultar el estado nos indica que esta en modo “Changes to be committed” cambiado para ser admitido

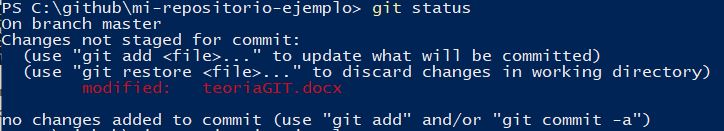
6.- Para realizar un commit

Git commit -m “mensaje”

Guarda el historial de las ultimas actualizaciones realizadas con fecha, hora, indicativo de lineas cambiadas

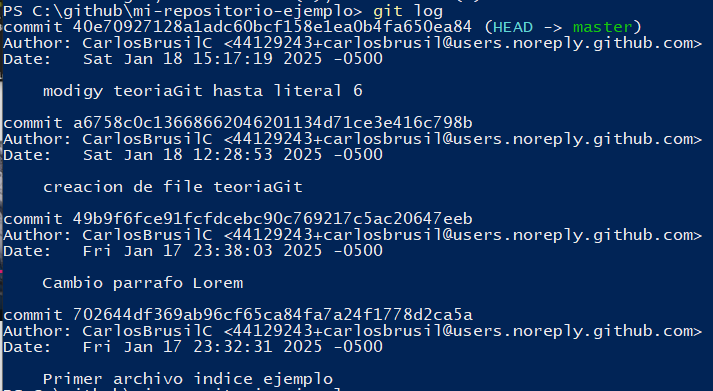


7.- Si realizamos un cambio en un archivo, en la consulta nos dira que se han hecho cambios y por tanto debe realizarse una nueva actualizacion “commit”.



8.- Para revisar el historial de modificaciones

Git log



9.- Para volver a una version anterior de un archivo, descartando cambio posterior

Git checkout #commit

Git checkout 702644df

10.- Cuando se hace un checkout se crea una nueva rama

Para ver las ramas:

Git branch



11.- Para crear una rama (cuando se ha realizado un checkout, se puede renombrar la rama dividida con un nuevo nombre)

Git switch -c “rama2”

OJO se presentar diferencias entre los archivos modificados en cada rama (si es Word no olvidarse de cerrar el archivo antes de cada paso de “git”)

12.- Para cambiarse a otra rama

Git switch nombrerama

13.- Para publicar (push) en un repositorio remoto de github (subir a un repositorio)

* Ingresar o (Log in) en [www.github.com](http://www.github.com) (email: [carlosbrusil@hotmail.com](mailto:carlosbrusil@hotmail.com) pass: 1958cbcb)
* Crear un nuevo repositorio en github
* Del repositorio creado copiar su link, para relacionarlos desde git local
* Git remote origin link

git remote add origin <https://github.com/carlosbrusil/mi-repositorio-ejemplo.git>

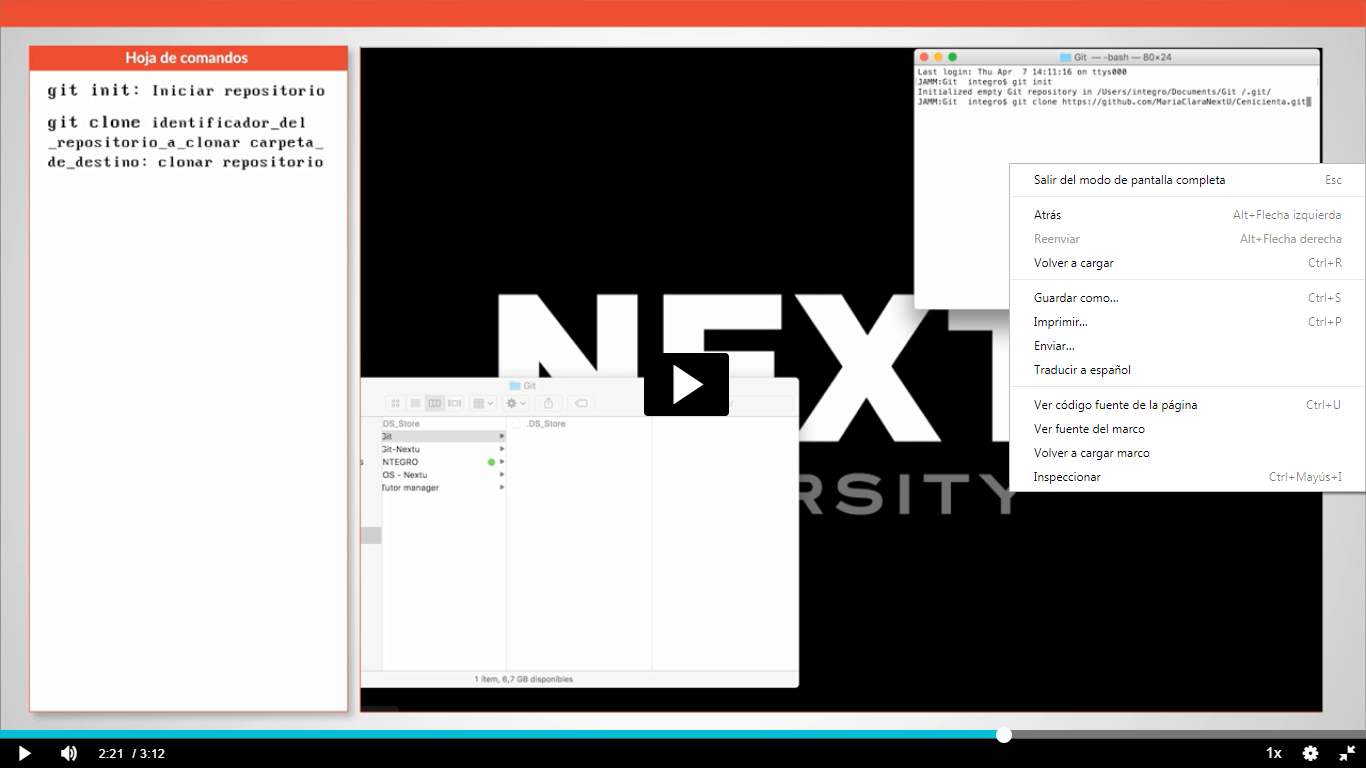
* Realizar el push de la rama

Git push origin master

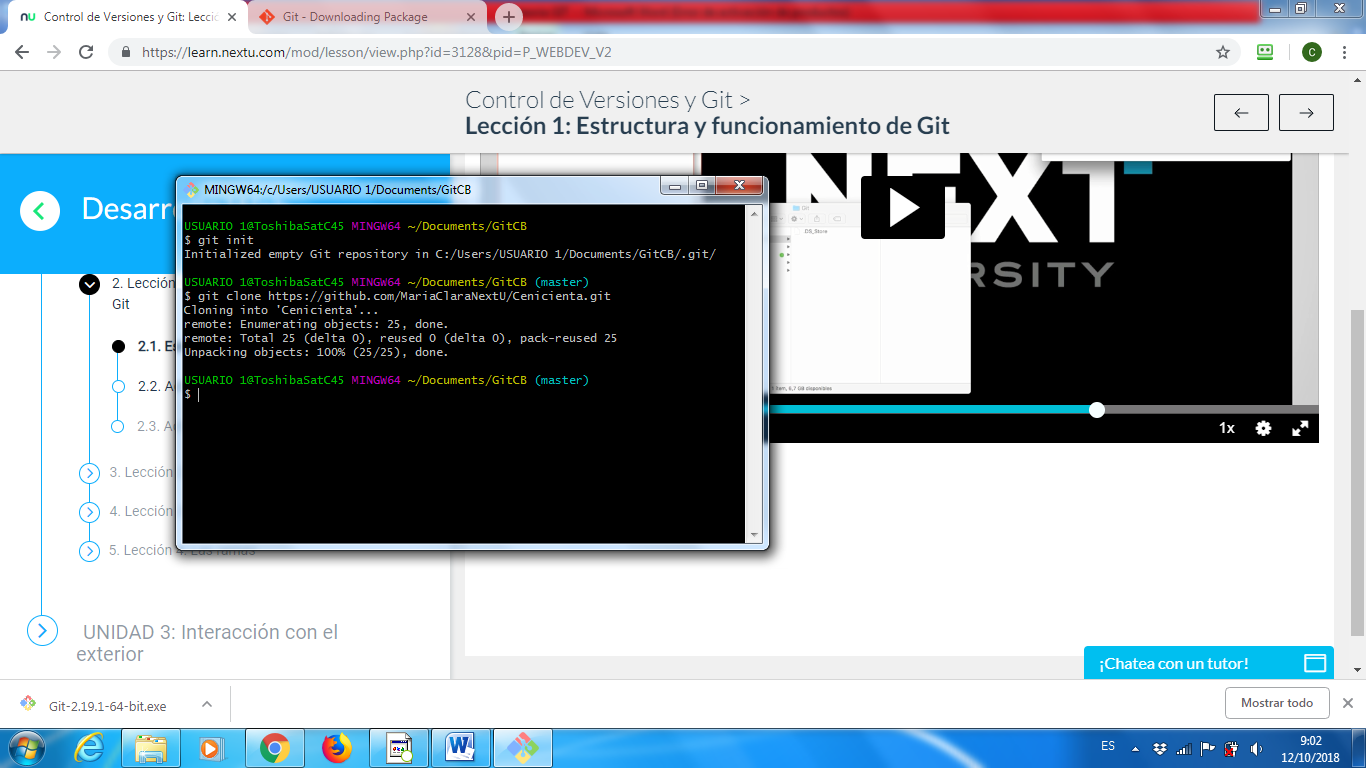
14.- Git pull Para acceder a los cambios que han hecho los otros desarrolladores que tienen permiso de trabajar en el proyecto.

15.- Para clonar o traer una copia de un repositorio desde el servidor central hasta la computadora local

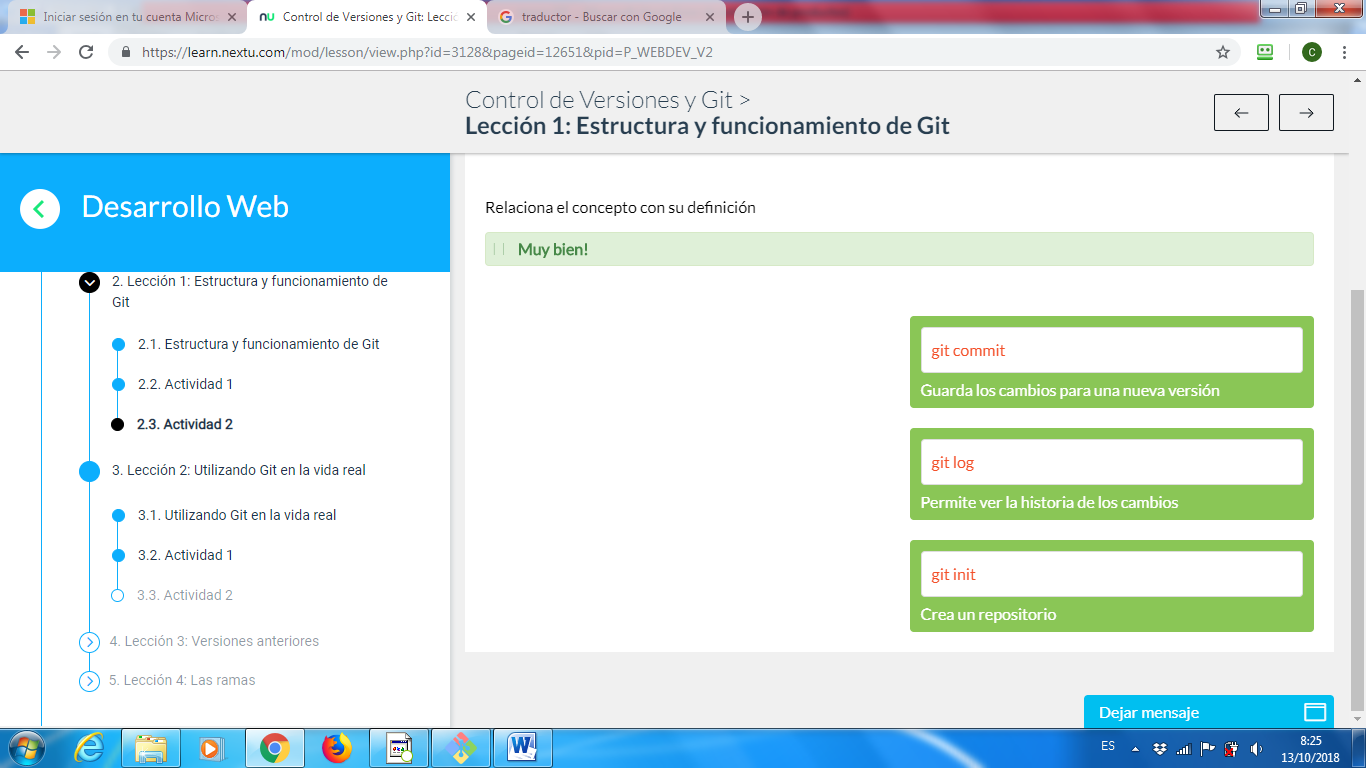
Git clone nombrerepositorioexistente



git clone <https://github.com/MariaClaraNextU/Cenicienta.git>



FUNCIONAMIENTO DE GIT



Git log

Permite ver el historial de los cambios realizados

Git init

Crear un repositorio

GIT ADD.- Sube el archivo al area de preparacion

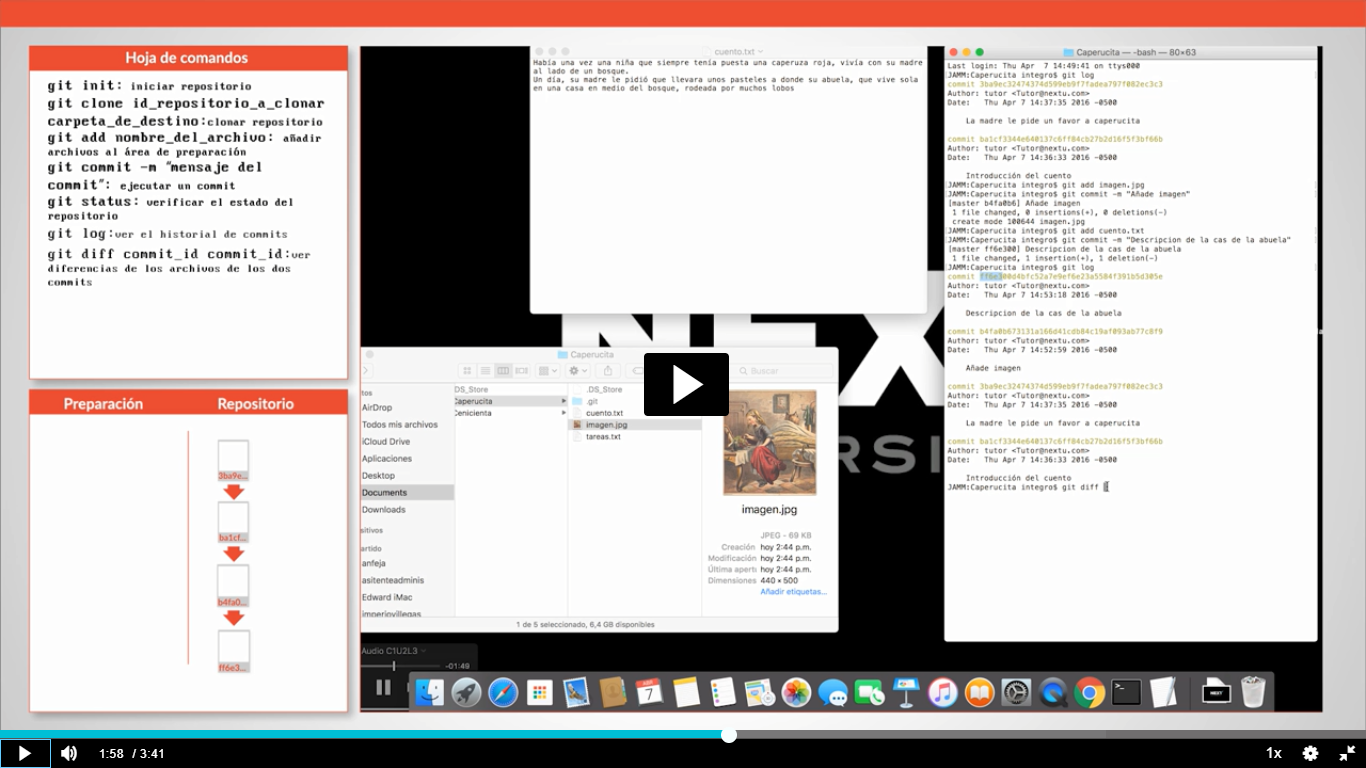
git add archivo.extension

Git commit .- Guardar los cambios realizados en una nueva version

git commit -m “mensaje”

GIT DIF.- Permite ver la diferencia entre los commit referenciados

Git dif conmit\_id1 commit\_id2



GIT CHECKOUT.- Para volver a una version de un commit id

Git checkout commit\_id

UTILIDADES PARA MANEJO DE GIT

Github.- descargar de

<https://desktop.github.com/>

configurar cuenta

user CarlosBrusilC

email [carlosbrusil@hotmail.com](mailto:carlosbrusil@hotmail.com)

clave 1958cbcb