

\$ pwd : me da la ubicación donde estoy
/c/Users/viejo

\$ cd Escritorio : ingreso a una carpeta
\$ cd ../ : Devuelve un nivel anterior de carpeta
\$ cd ../../ : Devuelve 2 niveles anteriores de carpeta

\$ mkdir repaso_clase1y2 : crear una carpeta
\$ mkdir repaso_clase1y2/infraestructura/física : crea una carpeta a una ruta relativa
\$ mkdir ../Prueba1/ : Desde DH/Prueba/ se regresa un nivel quedando en DH/ y luego crea la carpeta Prueba1
\$ mkdir ../Prueba2/ : Desde DH/Prueba/Subprueba/ regresa dos niveles quedando en DH/ y luego crea la carpeta Prueba2
\$ mkdir -p carpeta/subcarpeta/{carpeta1,carpeta2,carpeta3,etc} : Crea una carpeta raíz dentro una subcarpeta y dentro crea varias carpetas
\$ mkdir -p carpetaSemana{1..9}/clase{1..3} : Crea 9 carpetas y dentro de cada una crea 3 carpetas mas
\$ mkdir historia_de_que_haremos_hoy servidores : crear una o varias carpetas

\$ ls : lista los archivos de la carpeta
\$ ls -t : me en lista los últimos archivos modificados
\$ ls -R : lista jerárquicamente ficheros y directorios desde el directorio raíz
\$ ls -a : Me muestra todos los archivos hasta los ocultos

\$ touch repaso.txt : crear un archivo
\$ touch {archivo1,archivo2,archivo3} : Crea varios archivos simultáneamente.
\$ touch carpeta/subcarpeta/{archivo1,archivo2,archivo3,etc} : Crea varios archivos en una subcarpeta con ruta relativa y absoluta(/c/users/...)
\$ touch /c/.../Películas/Comedia/nombreArchivo.txt : crea un archivo utilizando una ruta absoluta
\$ >prueba.txt : crear un archivo

\$ rm que_haremos_hoy : borrar una carpeta
\$ rmdir nombreCarpeta : borra una carpeta vacía
\$ rm -d nombreCarpeta : borra una carpeta vacía
\$ rm nombreArchivo.txt : borra un archivo
\$ rm Películas/Accion/nombreArchivo.txt : Borra un archivo con ruta relativa
\$ rm /c/Users/viejo/Escritorio/DH/camada.txt : Borra un archivo con ruta absoluta
\$ rm {archivo1,archivo2,archivo3} : Elimina varios archivos simultaneamente.
\$ rm -r nombreCarpeta : borra una carpeta, dentro subcarpetas y archivos
\$ rm -rf nombreCarpeta : La opción -f te permite eliminar a la fuerza todo el contenido de una carpeta.
\$ rm -rf carpeta1 carpeta2 carpeta3 : eliminará de forma recursiva y forzosa todo lo que resida en las carpetas
\$ rm -i nombreCarpeta : Te pedirá que confirmes individualmente los archivos que serán borrados. Así puedes evitar errores desagradables.

\$ rmdir ../carpeta : sube un nivel de carpeta y borra otra carpeta de su mismo nivel
\$ rmdir -p carpeta/subcarpeta1/subcarpeta2 : La opción -p te permite eliminar la carpeta y sus carpetas madre. Este comando eliminará subcarpeta2 y sus carpetas madre subcarpeta1 y carpeta con ruta relativa

\$ find -name nombreArchivo : busca archivos o carpetas
\$ find -size 1(tamaño archivo) : Busca archivos o carpetas por tamaño
\$ find . : Desde la posición actual busca toda jerarquía de archivos, carpetas

\$ mv repaso.png repaso1.png : cambiar el nombre de un archivo
\$ mv repaso1.jpg /c/Users/viejo/OneDrive/Escritorio/DH : mover un archivo a una ruta absoluta
\$ mv repaso2.png repaso3.doc /c/Users/viejo/OneDrive/Escritorio/DH : mueve dos archivos a una ruta absoluta
\$ mv Examen_Eder/Memorias/rom.txt cache.txt Examen_Eder/Memorias/MemoriaPrincipal : mueve dos archivos de una ruta relativa a otra ruta relativa.
\$ mv prueba1.txt repaso_clase1y2/infraestructura : mueve un archivo a una ruta relativa
\$ mv Terror/IT.xls Infantiles/ : Desde una carpeta mayor mueve un archivo de la carpeta Terror a Infantiles utilizando ruta relativa

GIT

\$ rm -rf .git : Fuerza Eliminar la carpeta .git del repositorio local
\$ rm -r .git/ : borra la carpeta y archivos de la carpeta .git del repositorio local

Crear repositorio local con Git

\$ mkdir nombreCarpeta/ : Crear carpeta del proyecto
\$ git init : para inicializar el repositorio con git
\$ git config user.name "EderRamos2020"
\$ git config user.name : me arroja el usuario
\$ git config user.email "viejoeder@hotmail.com"
\$ git config user.email : me arroja el email
\$ git config --global user.name "EderRamos2020"
\$ git config --global user.email "viejoeder@hotmail.com"
\$ git config user.password "token" : otra forma de ingresar el token
\$ nano .git/config : archivo de configuracion del user.name y user.email

\$ git add archivo : En seguimiento
\$ git status : Estados archivos -> en added-stage /sin seguimiento
\$ nano archivo : modificar el archivo
\$ git status : Estados archivos -> en added-stage /modificado /sin seguimiento
\$ git restore archivo : descarta cambios y lo agrega en added-stage
\$ git status : Estados archivos -> en added-stage /sin seguimiento
\$ git rm --cached archivo : saca el archivo del added-stage y queda sin seguimiento
\$ git status : Estados archivos -> sin seguimiento/modificados/added-stage
\$ git add . : Agrega todos los archivos sin seguimiento y modificados al added-stage
\$ git commit -m "mensaje" : confirmar cambios
\$ git log : Historial de cambios o commits

\$ git remote add origin urlRepo(<https://token@link>) : Vincula el repositorio local con el repo remoto
\$ git remote -v : Me muestra el repositorio remoto con el que se enlazo \$ git branch -M
nombreRama(main o otra) : Cambia el nombre de la rama master por main o otra.
\$ git push -u origin nombreRama(main o otra) : Cuando la rama está creada local pero no está creada en el remoto
entonces -u crea la rama en el remoto y se sincronice con la local, cuando es la primera vez (push) en el repo.
\$ git push origin nombreRama(main o otra) : Envío los archivos del repo local al repo remoto siempre debe coincidir el
nombre es decir la misma rama local-remoto.

\$ git push -f origin nombreRama(main) : ? \$ git rev-parse --show-toplevel : ?
\$ git amend : ? \$ git reset : ?
\$ git checkout -b : ?

\$ cd .git luego \$ rm index.lock : al hacer git add y genere error de procesos (git process) en el repositorio
\$ git config --list (:wq) : Datos configuracion (user,email,repo remoto,ramas) \$ git clone
[url\(https://token@link\)](https://token@link) : Baja el total proyecto de github parametrizando el token.
\$ git branch nombreRama: Crea una nueva rama \$ git branch : Enumera las ramas de tu repositorio
similar git branch --list \$ git branch -d nombreRama: Elimina la rama. Git evita que eliminemos la rama si tiene
cambios que aún no se han fusionado con la rama Main. \$ git branch -D nombreRama: Fuerza la
eliminación de la rama especificada, incluso si tiene cambios sin fusionar.

\$ git checkout nombreRama: se mueve de una rama a otra, sino tiene cambios.
\$ git pull origin nombreRama: Baja los cambios, nombreRama es desde donde queremos traer los cambios.

\$ git remote set-url origin url(<git://new.url.here>) : Sobrescribe el repositorio.