configuracion global

git --version

git config --global user.name "Nelson Rincon"

git config --global user.email nelsonrincon@usantotomas.edu.co

#####configurar editor vscode…

$ git config --global core.editor "code --wait"

######ver archivo de configuracion en code

git config --global -e

#####salto de linea

git config --global core.autocrlf true

#####ver los demas comandos de configuracion

git config -h

#####comandos

pwd ### en que carpeta nos encontramos

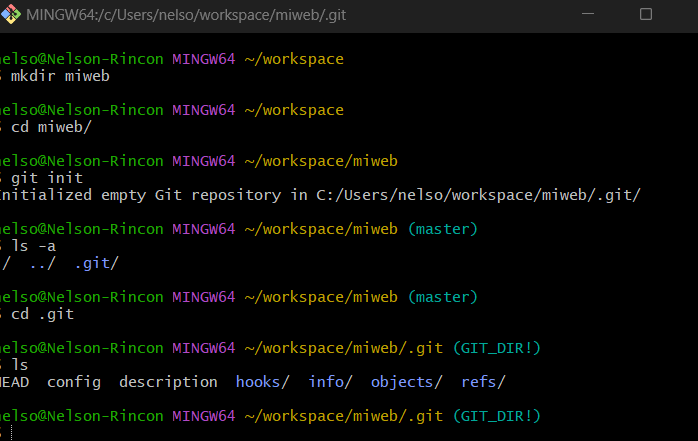
cd y cd .. #### igual a win

######PARA INICIAR UN REPOSITORIO VACIO

git init ### esto grea una carpeta .git que no se puede ver

ls -a ####para ver la carpeta

cd .git ####para trabajar en el espacio creado en git



Trabajando en la carpeta de desarrollo dentro del workspace

code .

crear archivo nuevo y revisarlo en git con

git status

para que almacene este archivo en git debemos ejecutar

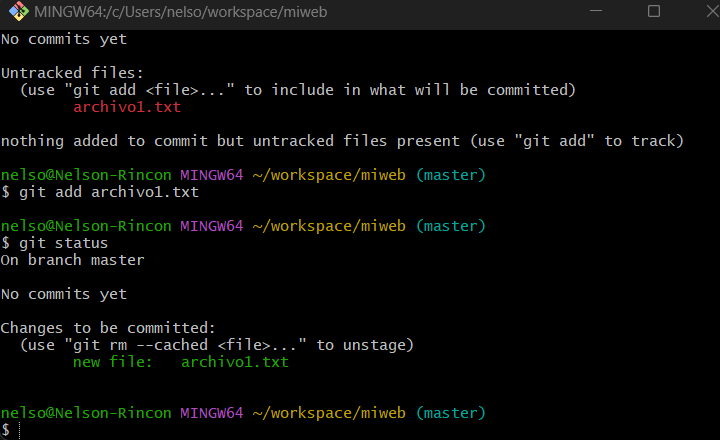
git add archivo.ext el archivo con su extension y guarda solo ese archivo

git add \*.ext guarda solo los archivos de esa extension

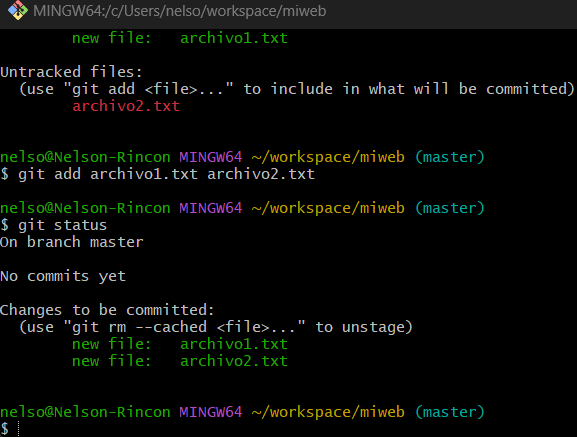
git . guarda toda la carpeta \*\*\*NO USAR ESTE\*\*\*

SE REVISA el almacenado con

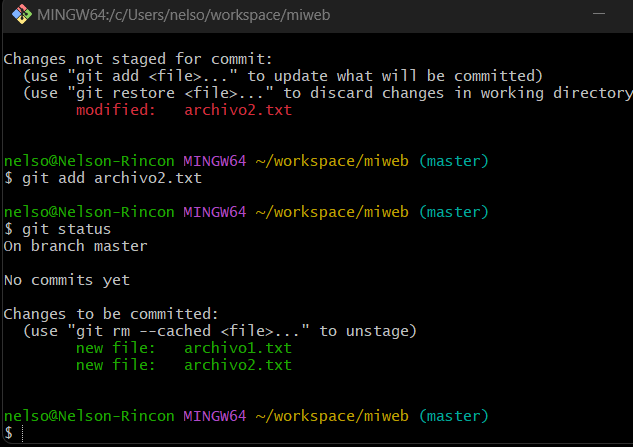
git status



git add separando con espacio para almacenar mas de uno

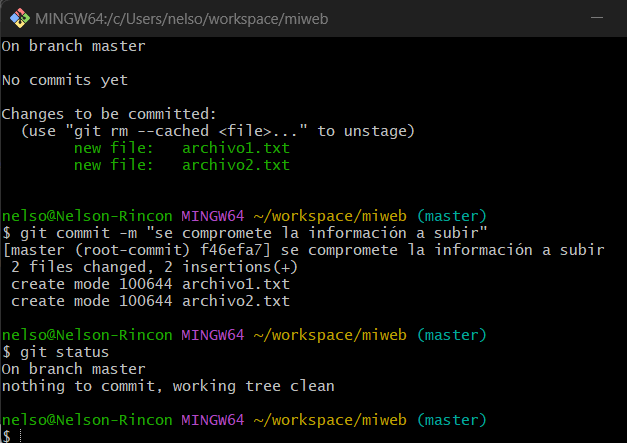


cada que se edite un archivo aparecera en rojo y hay que volverlo a agregar si se quiere comprometer

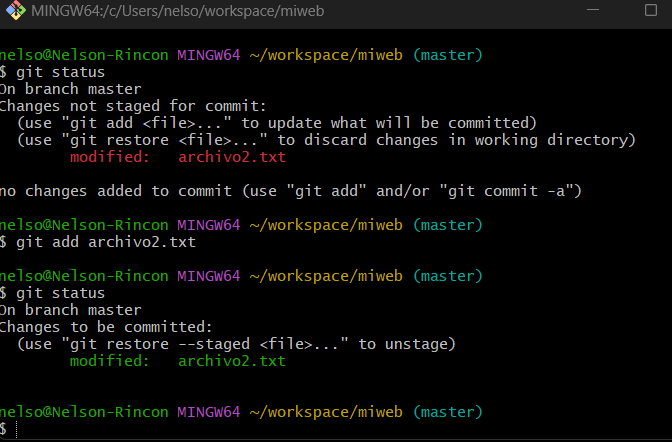


una vez actualizado se puede comprometer

git commit -m “aclarar que se esta haciendo aqui”



ya no apreceran como subidos pues estan en etapa intermedia



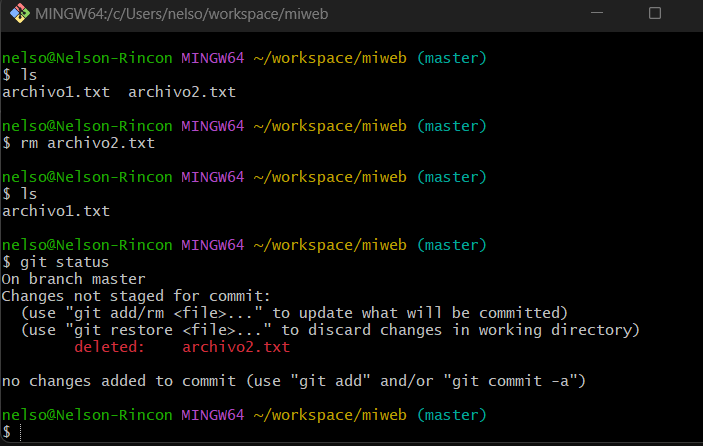
aun no aparecen en commit aunque lo he vuelto a add agregar

si solo hago

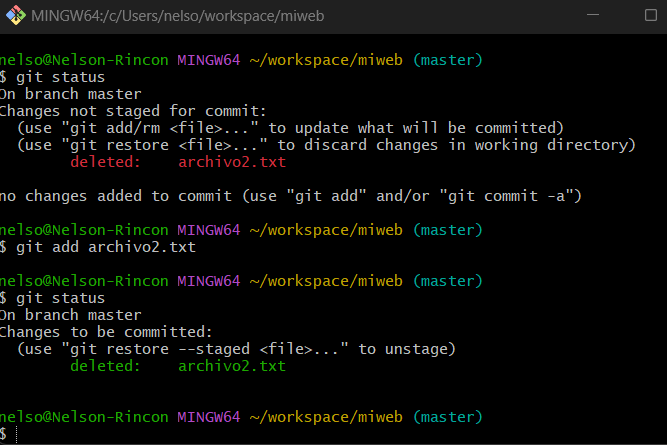
git commit

me abre vscode para modificar un archivo de comentarios

###BORRANDO CORRECTAMENTE



No basta con solo borrar se tiene que volver a agregar para que se observe el cambio



cuando lo add grego ya se ve el status en verde

git add archivo.ext --- removido con su extension

tenemos también que comprometer el borrado

git commit -m “borrado el archivo 2” ## la palabra eliminado presenta problemas

##RECUPERAR ARCHIVO

primero lo borro con

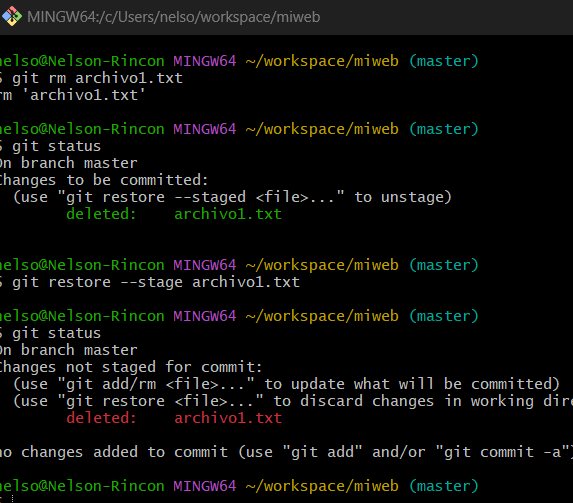
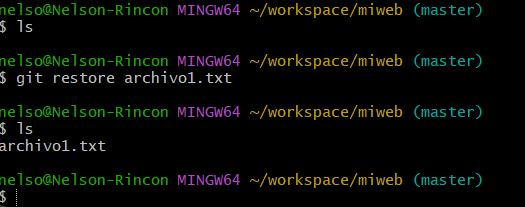
git rmarchivo.ext

luego ejeccuta restaurar

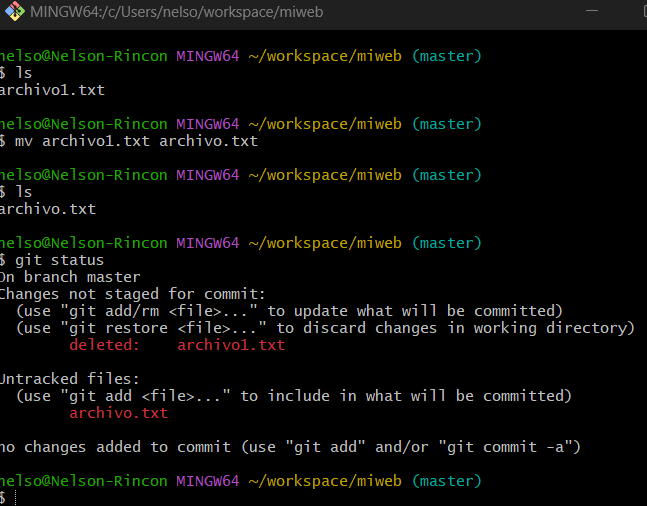
git restore --staged archivo.ext -----> sigue abajo

luego finalmente restauras en el archivo de desarrollo

git restore archivo.ext

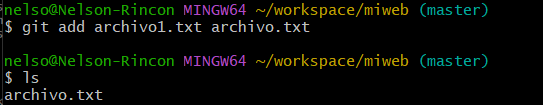
Mover y renombrar archivos

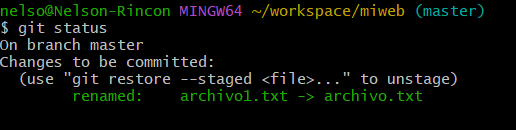


git mv archivo.ext archioALGO.ext cambie el nombre y agrege a stage

no olvidar agregar los cambios

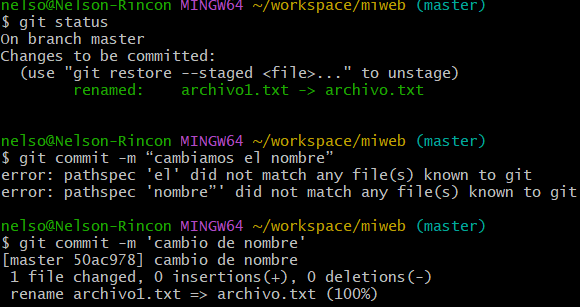
git add archivo1.txt archivo.txt



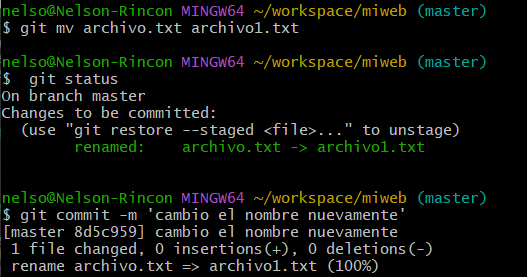


también hay que comprometer el cambio

git commit -m “cambiamos el nombre” ### MEJOR USAR LA MILLA SIMPLE XQ LA DOBLE ME ESTA MOLESTANDO

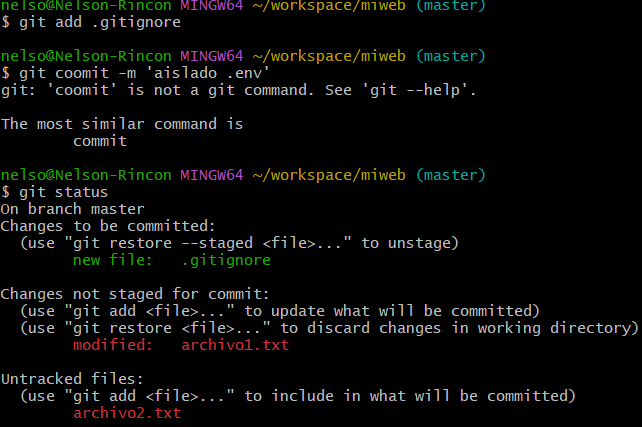


git mv archivo.txt archivo1.txt

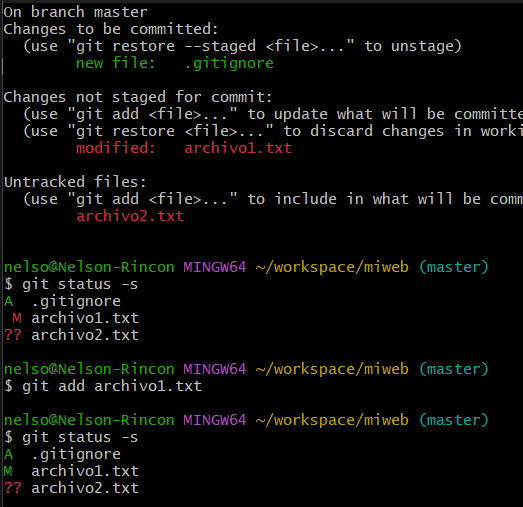


aislar archivo para que no se suba nunca, lo creo con .archivo y los agrego a gitignore(igual carpetas como node\_modules)





MEJORANDO COMANDO STATUS

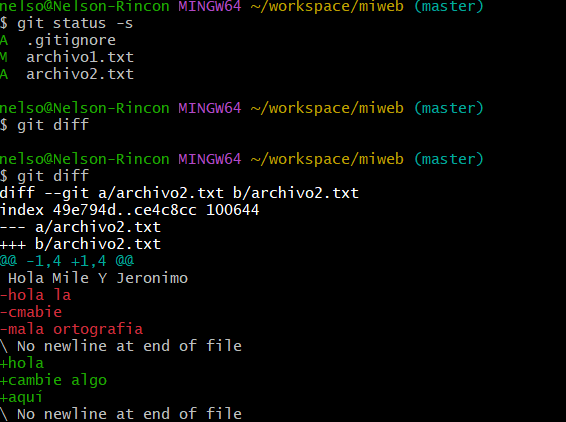


A de add y commit

M de add

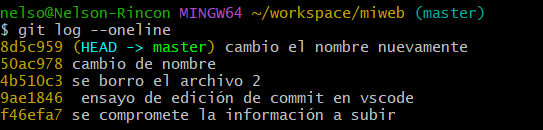
?? sin stage or commit

para poder obervar solo los cambios específicos se debe add y luego indicadesde donde y hasta donde se cmabio

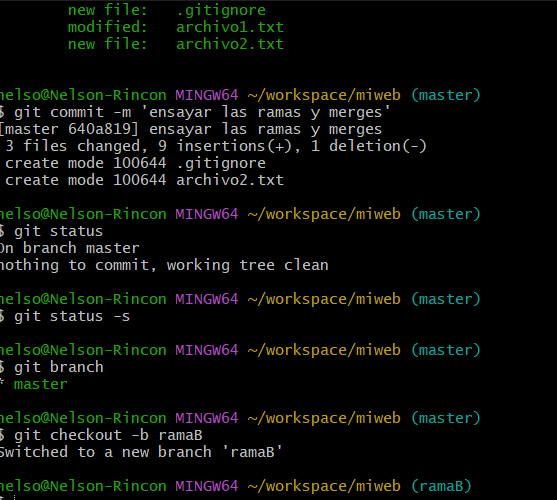


para ver todos los cambios a lo largo del desarrollo se usa

git log --oneline

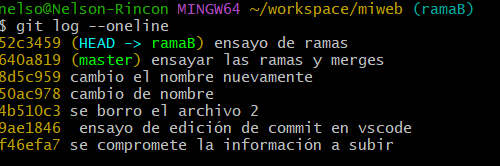


TRABAJANDO EN RAMAS PARALELAS



git checkout -b ‘ramaB’ ### crea una rama nueva llamadaramaB

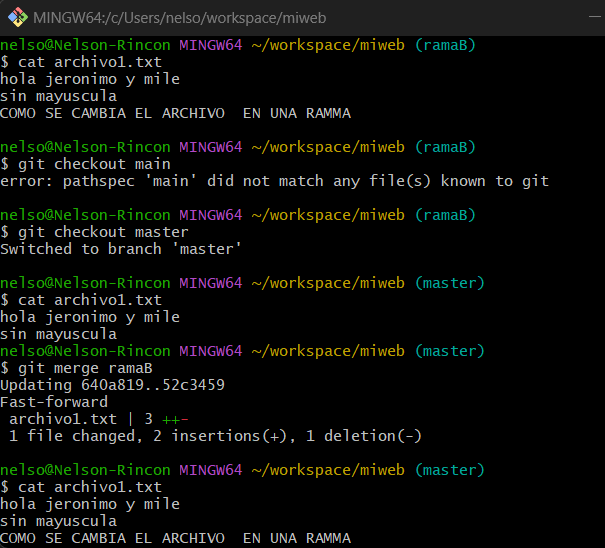
estos nuevos cambios son en una rama alternativa y paralela



retornando a main

git checkout main en windows es master

los cambios se traen con merge o quedan solo en la version de la rama



git merge ramaB # traer la rama paralela B