**Paso a paso del documento COMANDOS GIT**

**git init**

Con este código podemos iniciar un repositorio, en un directorio especificado

**git config –global user.email “**[**email@gmail.com**](mailto:email@gmail.com)**”**

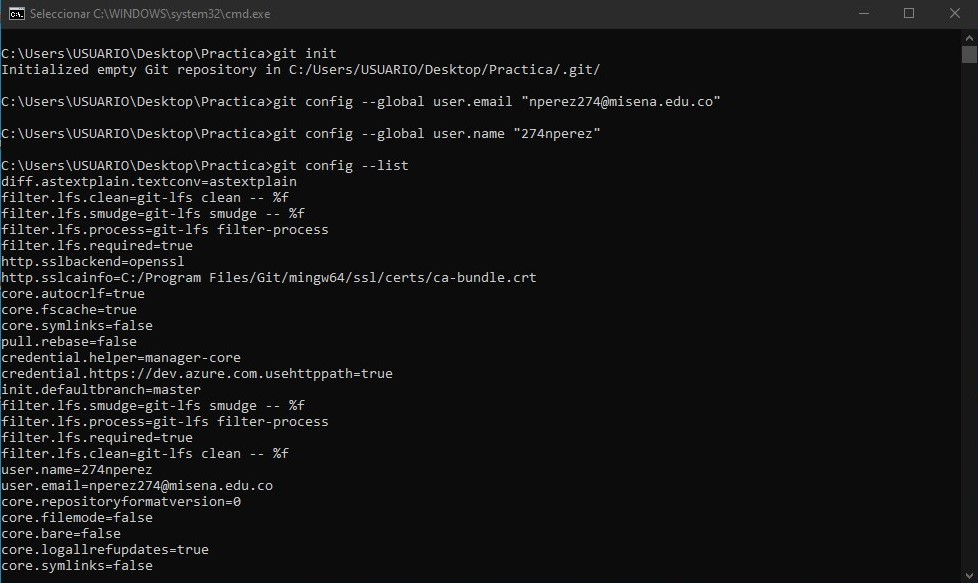
Con este código podemos configurar nuestro email de manera global en mi maquina

**git config –global user.name “User Name”**

Con este código podemos configurar nuestro nombre de usuario de manera global en mi maquina

**git config –list**

Con este comando podemos conocer los valores de configuración global



**git add nombrearchivo.extensión** / ejm index.html o **git add .**

Con este código podemos agregar un nuevo archivo, especificando que archivo queremos subir o subiendo todos a la vez a nuestro staging area (área de preparación) en el que se encuentra la información que está en el próximo commit.

**git status**

Con este código puedo ver el estado actual de mis archivos.

**git commit** O **git commit -m “mensaje del usuario sobre le commit”**

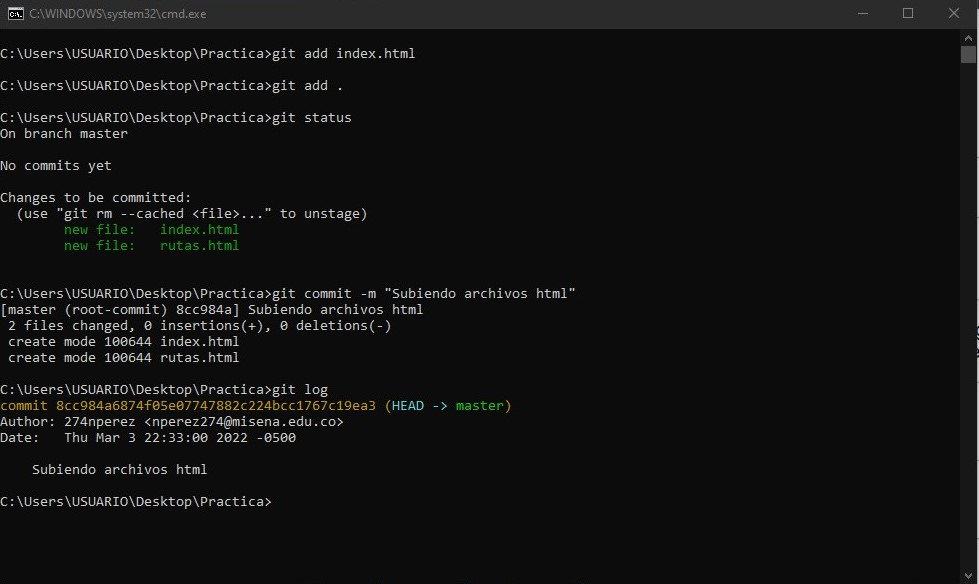
Con este código podemos guardar todos los cambios hechos en el staging area (área de preparación)

**git log**

Con este código podemos ver el historial de commits es uso con su autor, fecha-hora y código hash

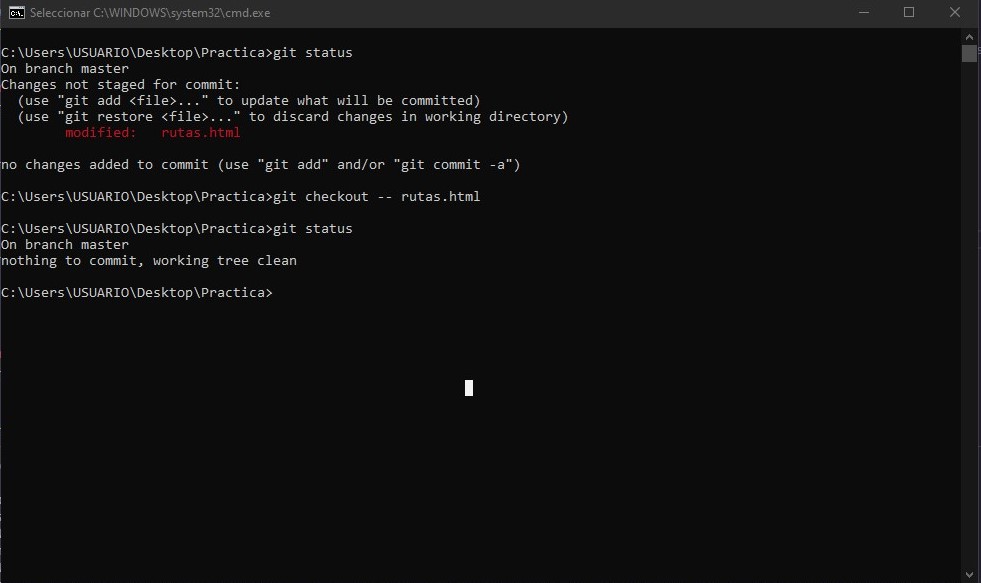
**git log –reflog**

Con este código podemos ver el historial de commits con su autor, fecha-hora y código hash



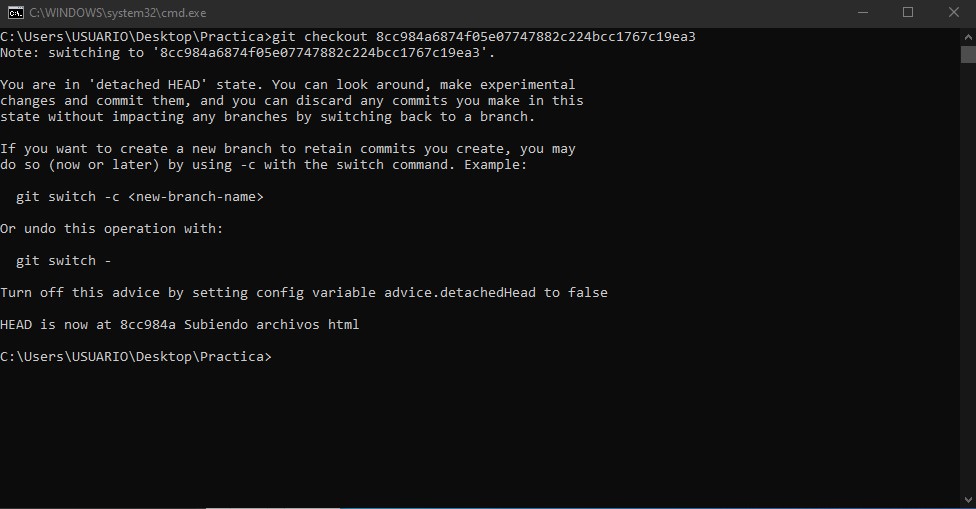
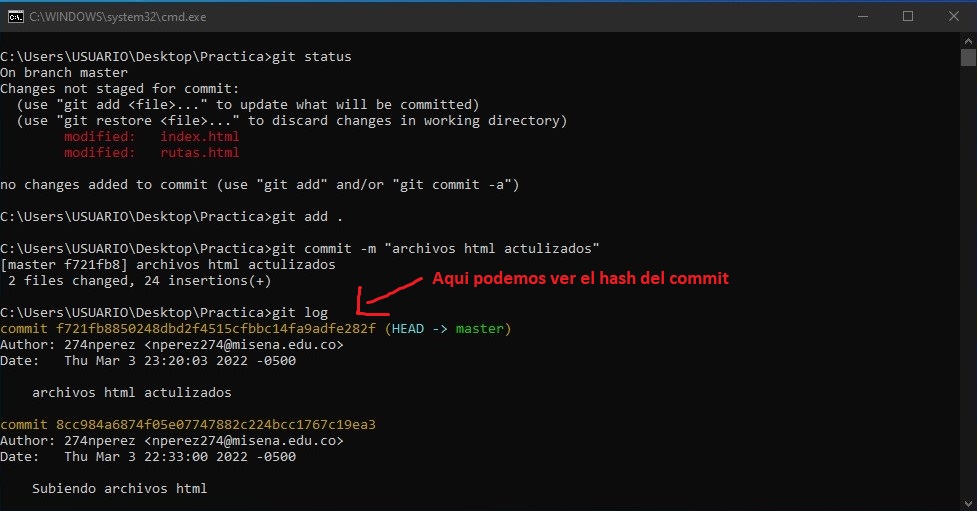
**git checkout – nombrearchivo.extencio** / ejm rutas.html

Con este código podemos devolver un archivo a la última versión guardada, como podemos ver en la siguiente imagen con el comando **git status** el archivo se le había hecho un cambio pero con comando dicho anteriormente volvió a su versión anteriormente guardada.



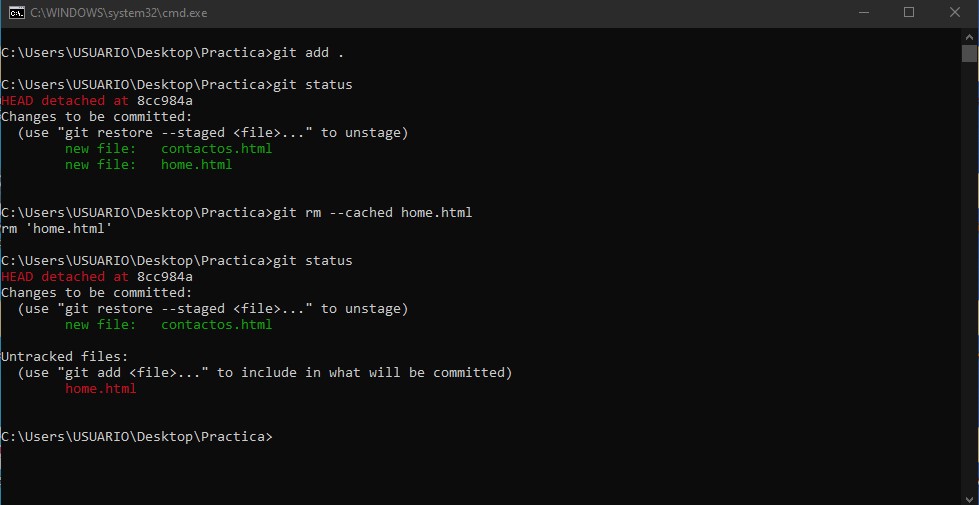
**git checkout hash\_commit**

Con este código podemos devolver un archivo o en este caso un proyecto a una versión en particular, como vemos en las siguientes imágenes hice un nuevo commit actualizando los archivos html y utilizando el código anteriormente mencionado lo devolví al commit anterior.



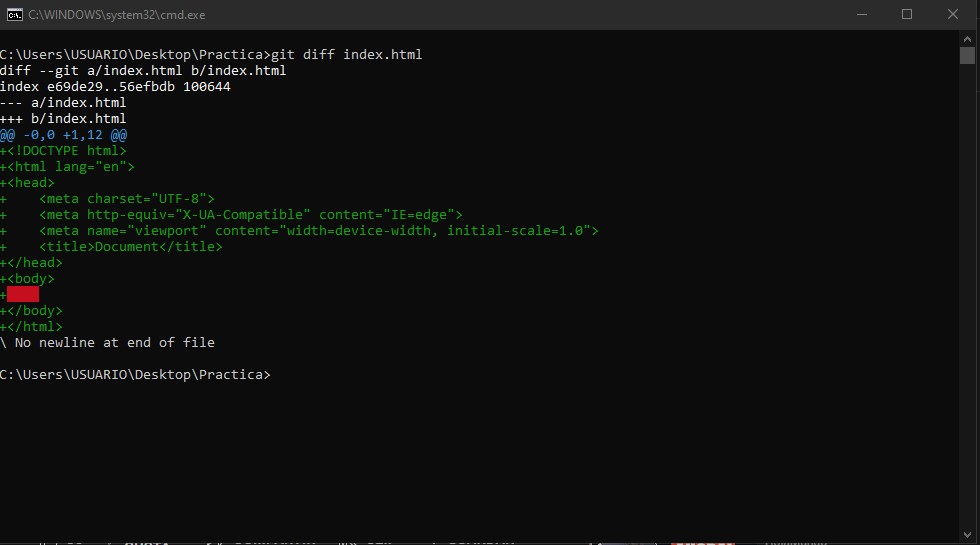
**git rm –cached nombrearchivo.extencion** / ejm index.html

Con este código podemos eliminar un archivo del staging area (área de preparación) y dejar de rastrearlo



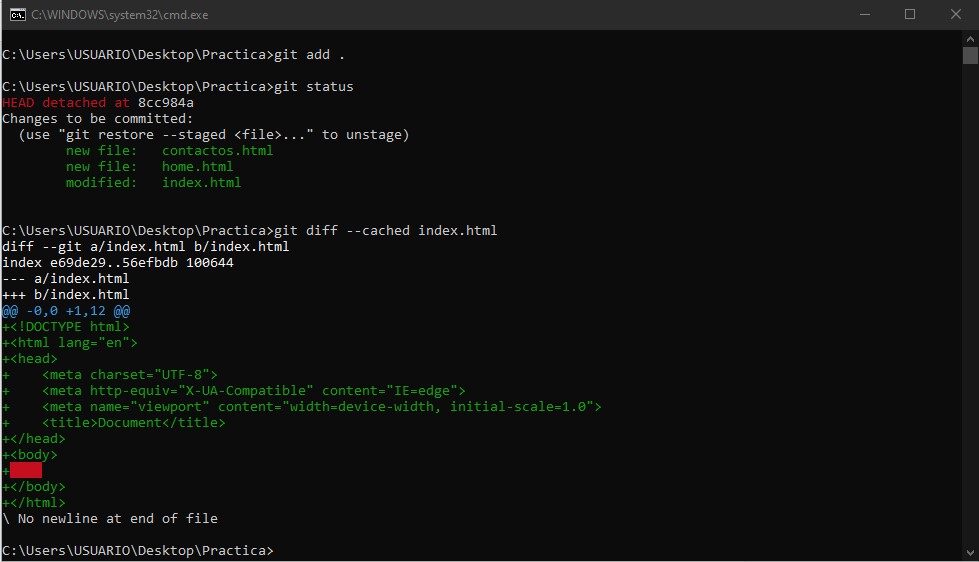
**git diff nombrearchivo.extencion** / ejm index.html

Con este código podemos ver la diferencia entre en directorio de trabajo y lo que se podría agregar al area de intercambio.



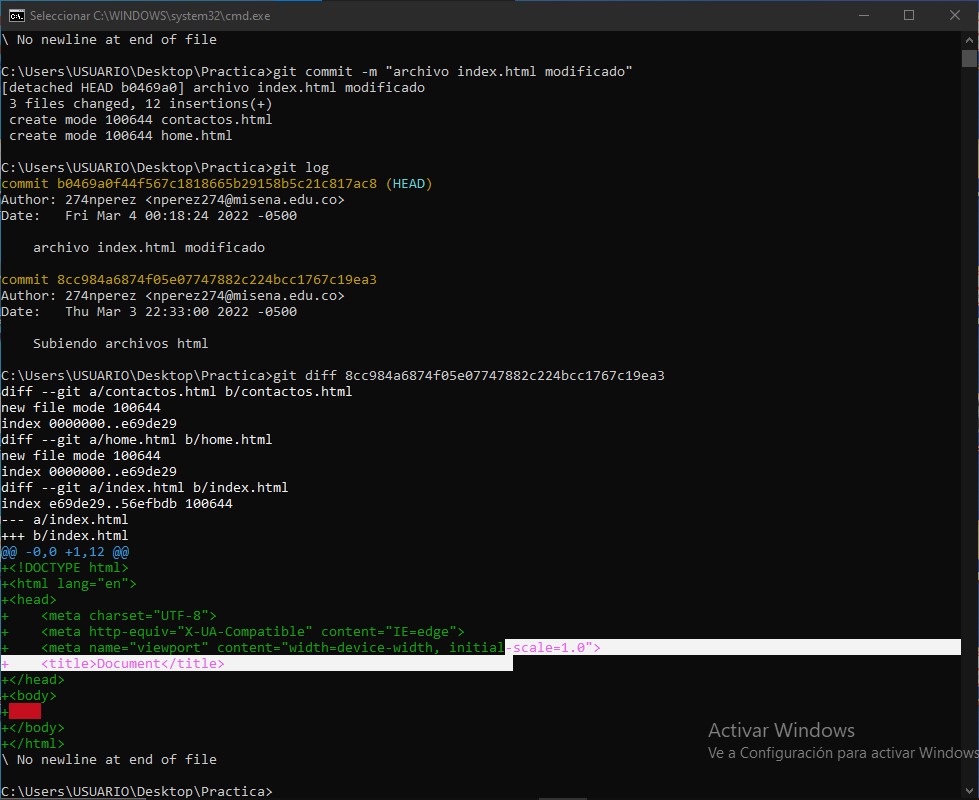
**git diff –cached** **nombrearchivo.extencion** / ejm index.html

Con este comando podemos ver la diferencia entre el directorio de trabajo y lo que se ha agregado al área de intercambio.



**git diff hashdelcommit**

Con este código podemos ver la diferencia entre el último commit y un commit en particular.



**git branch nombrenuevarama**

Con este código podemos crear una nueva rama.

**git branch**

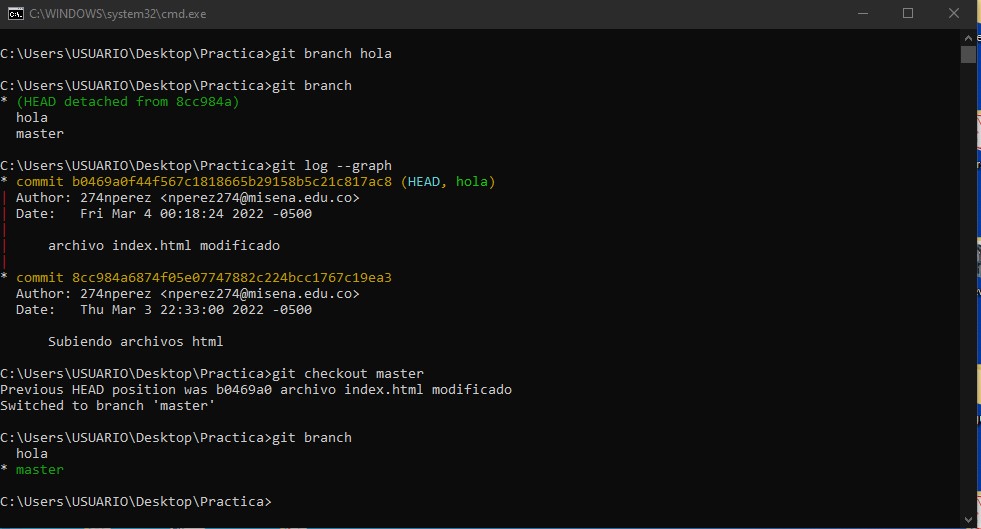
Con este código podemos consultar todas nuestras ramas.

**git log –graph**

Con este código podemos mostrar el historial del repositorio en forma de grafo incluyendo todas las ramas.

**git checkout nombrerama**

Con este código podemos cambiarnos a una rama.



**git merge nombrerama**

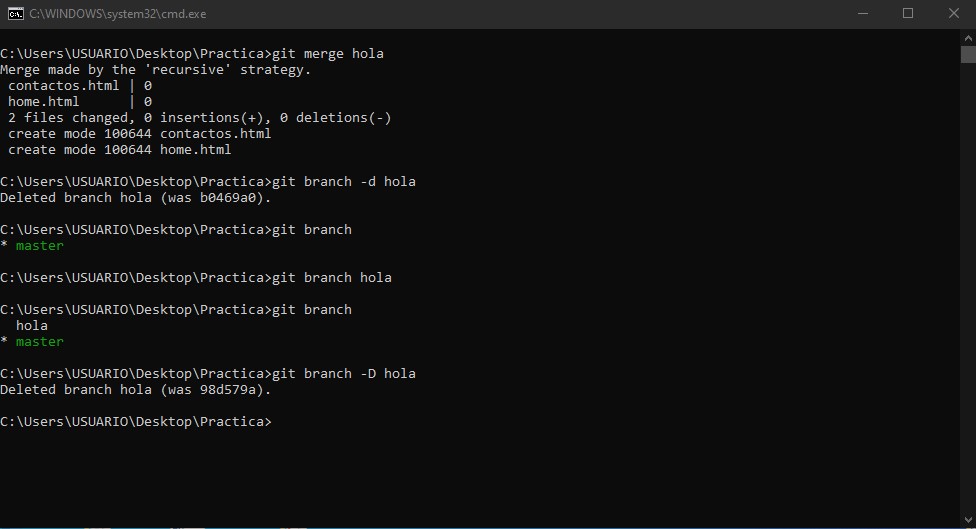
Con este código podemos fusionar las ramas.

**git branch -d nombrerama**

Con este código podemos eliminar una rama que ya haya sido fusionada.

**git branch -D nombrerama**

Con este código podemos eliminar una rama sin importar que ya haya sido fusionada o no.



**git add .gitignore**

Con este código podemos agregar archivos o carpetas al proyecto que serán ignorados por git, se crea un archivo llamado .gitignore en él, se ponen los nombres de los archivos con su extensión o los nombres de las carpetas, los cuales no harán parte del control de cambios.

REMOTO

**git remote add name url**

crear una conexión a un repositorio remoto

**git remote -v**

verificar si existe un repositorio remoto

**git push origin --delete rama\_a\_borrar**

eliminar una rama remota

**git remote remove nombrereporemoto**

remover un repositorio remoto de nuestra configuración local de git

**git push nombrereporemoto nombrerama**

enviar cambios a un repositorio remoto

**git pull nombrereporemoto nombrerama**

fusionar cambios remotos a nivel local