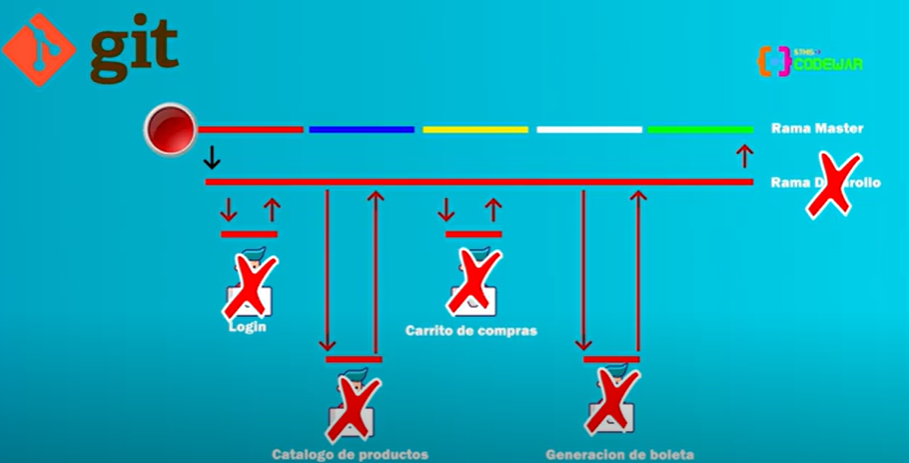
RESUMEN DE GIT

Hay una rama principal (main o master) que es la del proyecto y otras accesorias por medio de las cuales se va desarrollando el trabajo y la que soporta los cambios que se le van haciendo al proyecto y cada una de estas se van uniendo cuando todo vaya bien.



Para crear el repositorio en git utilizamos el comando “git init”

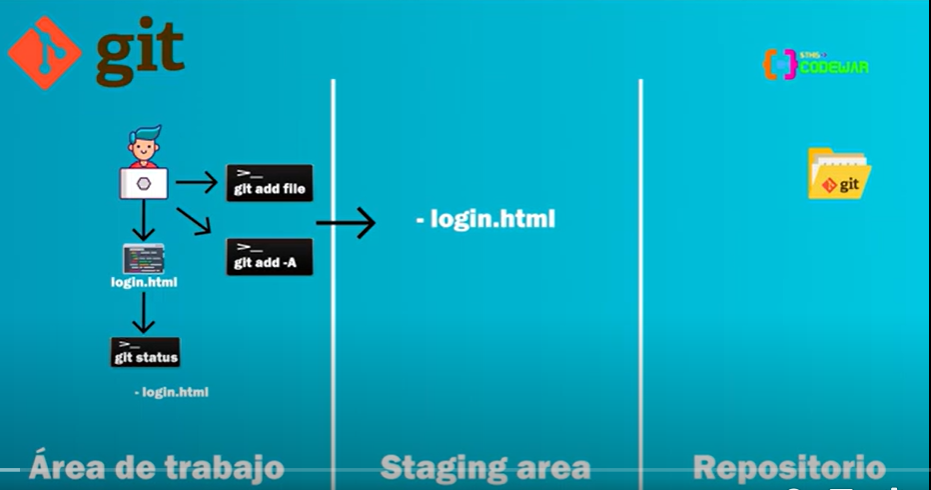


Creado nuestro proyecto, nos ubicamos en el área de trabajo que se compone de tres partes

Área de trabajo

Área de preparación

Repositorio



Con git status -🡪 para saber que archivos hay en el área de trabajo (nuevos y modificados)

Para enviar los archivos al área de preparación utilizamos

Git add A-

Git add file + nombreArchivo

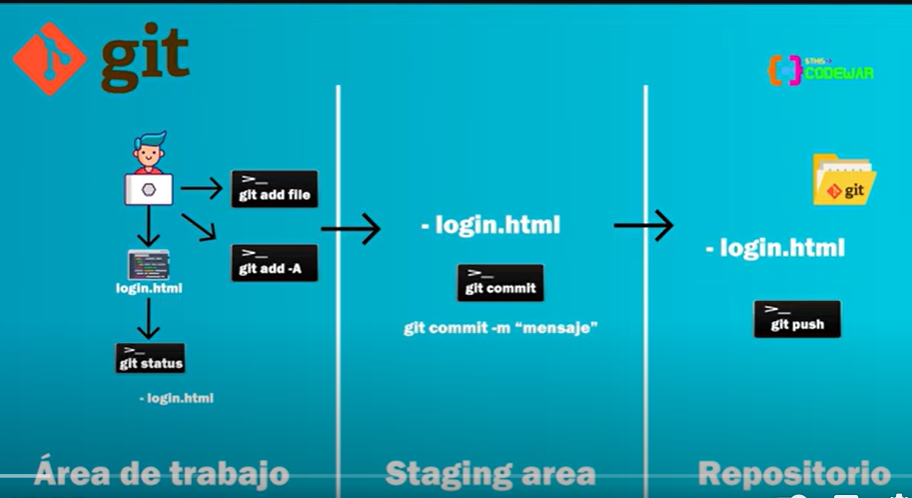
Git add . (para subir todos los archivos)

Para enviar los archivos al repositorio

Git commint –m “el comentario donde explique el cambio”

Para subir nuestros cambios al repositorio de la nube

Git push



Para comenzar a trabajar con git en un repositorio, los pasos son:

1. Descargar el git
2. Creamos nuestro proyecto de manera local
3. Abrimos el proyecto en un IDE
4. Creamos el proyecto local en git Bush
5. En la consola damos git init
6. Para ver los archivos del proyecto en el área de trabajo git status
7. Agregar el archivo a el área de preparación git add .
8. Informar los cambios realizados con git commit –m “comentario”
9. Creamos nuestro repositorio remoto con git remote add origin + url de github
10. Cargamos los archivos al repositorio de la nube con git push –u master --- $ git push origin main
11. Clonamos el repositorio con git clone + dirección url ti
12. Para ver en que rama estamos git branch
13. Para crear una rama git branch + nombre
14. Para cambiar de rama de trabajo git checkout
15. Para enviar la rama que hemos creado al repositorio original git push --set-upstream origin NombreRama
16. Para que el dueño del proyecto acepte nuestro trabajo y fusione las ramas, vamos al repositorio en la web y presionamos pull request y le pedimos que nos de el merge
17. Para eliminar la rama con que se trabajó git

Nota para Creau una clave ssh

$ ssh-keygen -t ed25519 -C "bumeran333@gmail.com"

Estos nos crea dos archivos

Your identification has been saved in /c/Users/Usuario/.ssh/id\_ed25519

Your public key has been saved in /c/Users/Usuario/.ssh/id\_ed25519.pub

Añadir las claves privadas al SSH agent de nuestro equipo

$ eval “$(ssh-agent -s)”

Con la clave vas a github /settings /key ssh/nombras la llave /pegas la clave ssh/aceptas

Cd ~

En la consola das git config - -list

Git config –global user.mail [bumeran333@gmail.com](mailto:bumeran333@gmail.com)

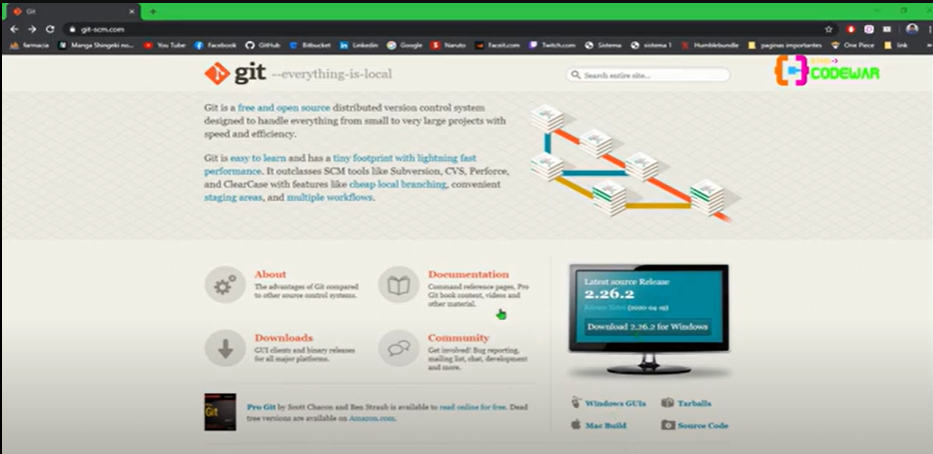
Git config –global user.name

Nota= Para bajar los cambios desde git hub remoto al repositorio local utilizamos

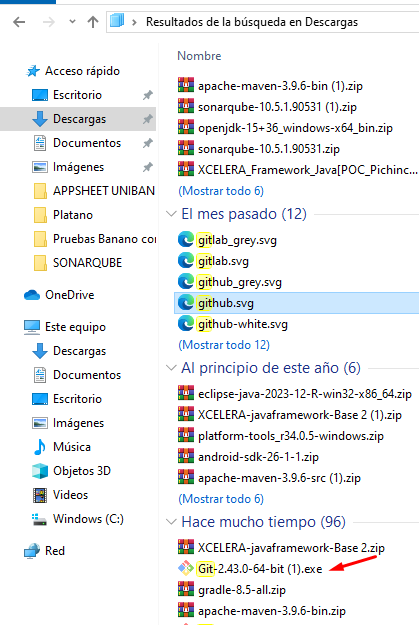
$git fetch para saber qué cambios se hicieron desde el repositorio remoto

$ git pull -para bajar los cambios y que se acoplen al repositorio local

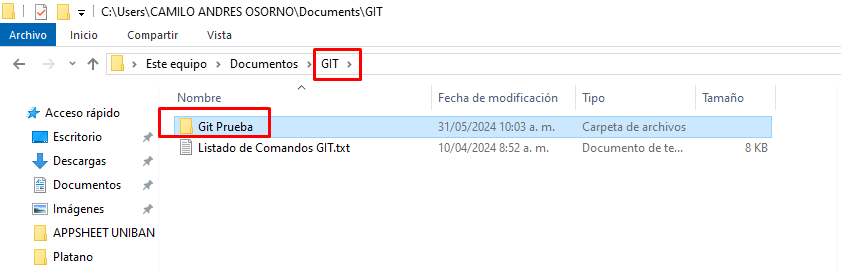
Descargar el git



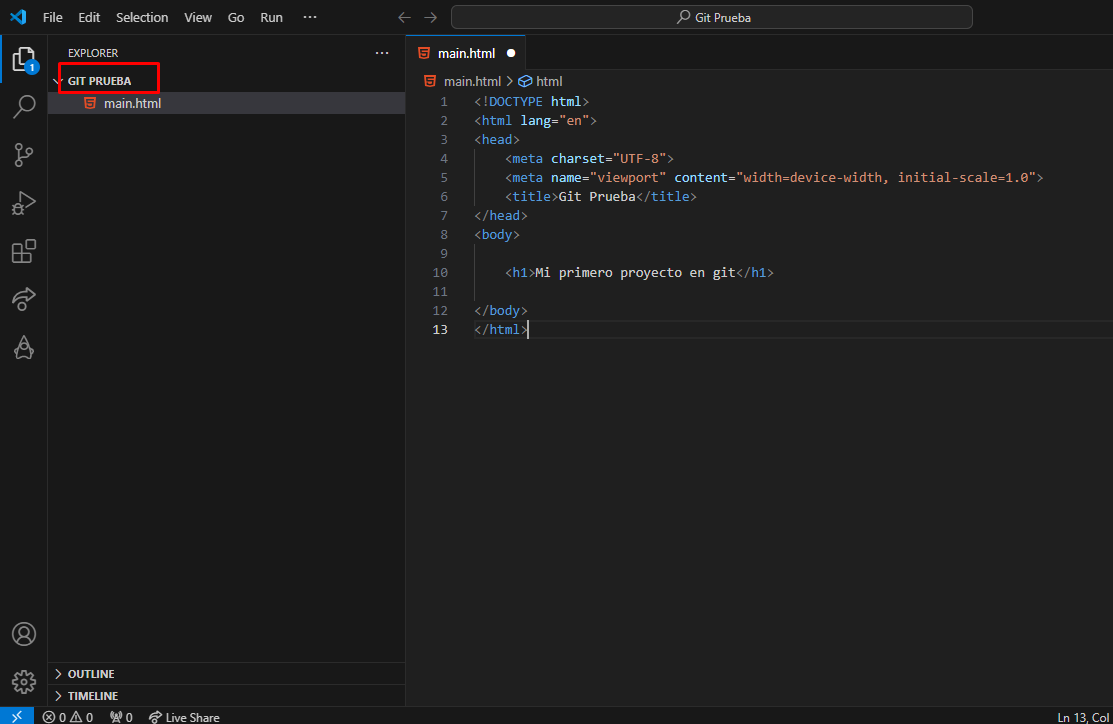
Se baja un ejecutable



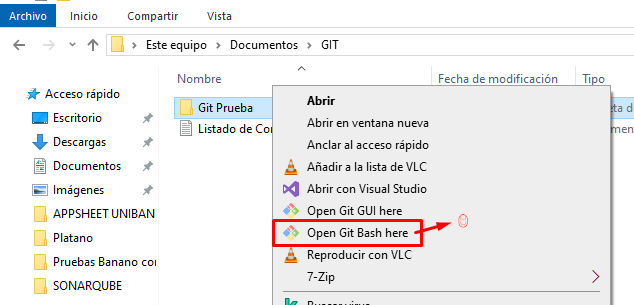
Creamos el proyecto de manera local

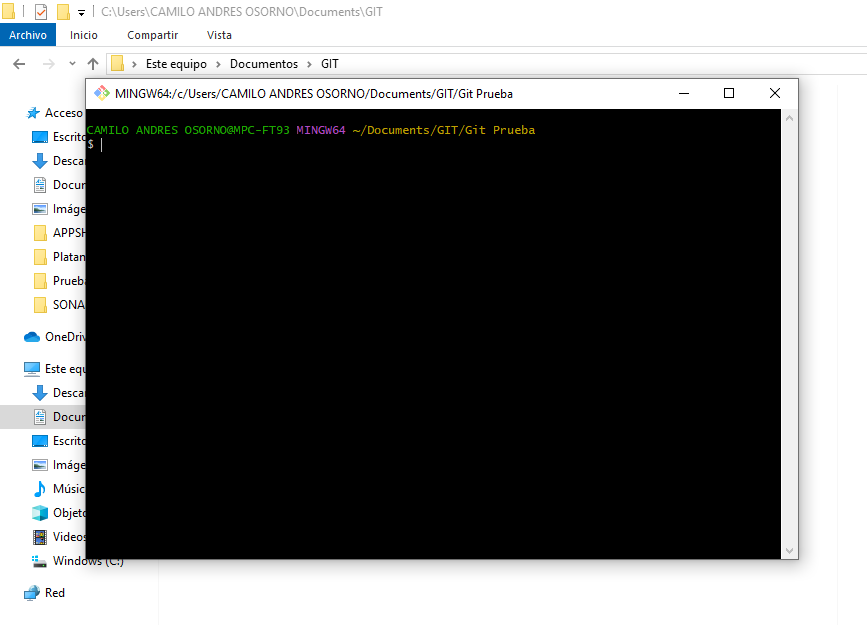


Abrimos el proyecto en un IDE

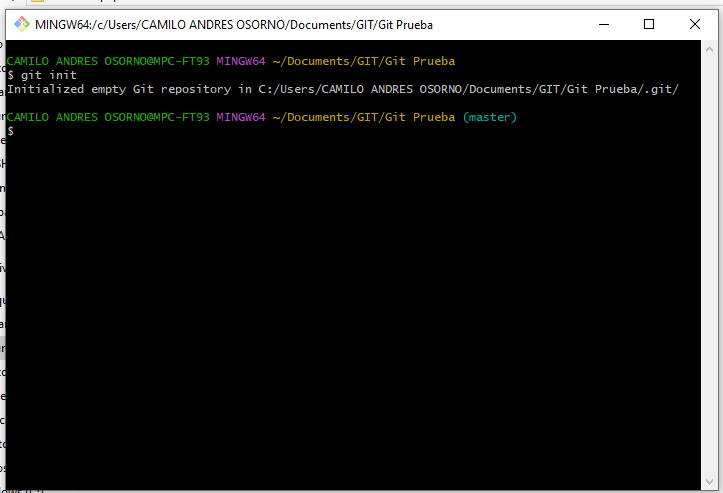


Creamos el proyecto en el git bash

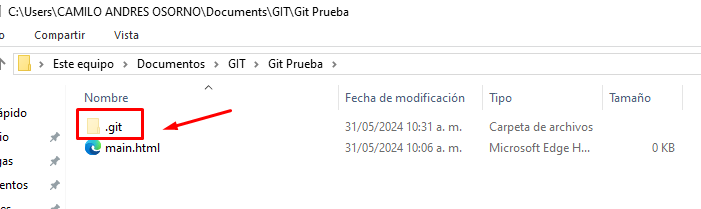




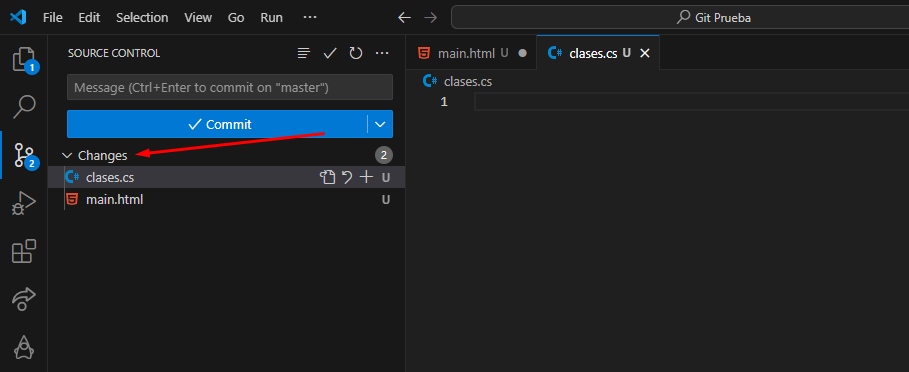
Iniciamos el proyecto con el comando git init



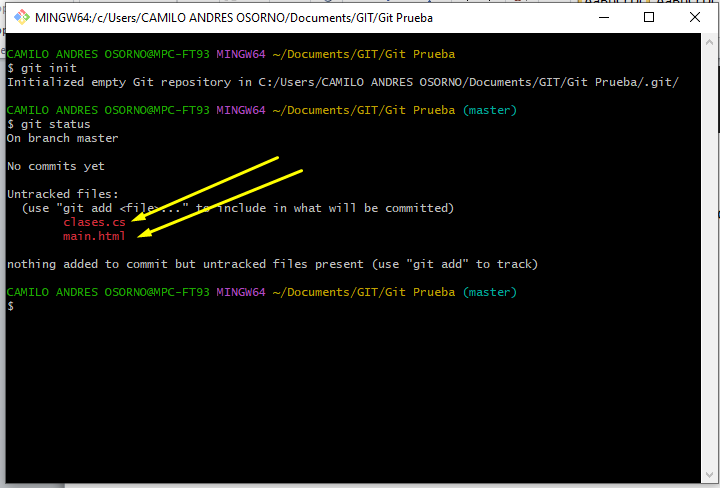
Se puede verificar en la carpeta del proyecto un archivo .git



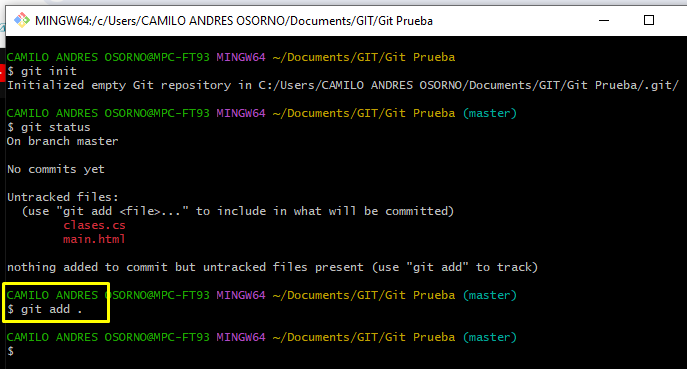
Realizamos cambios en el proyecto



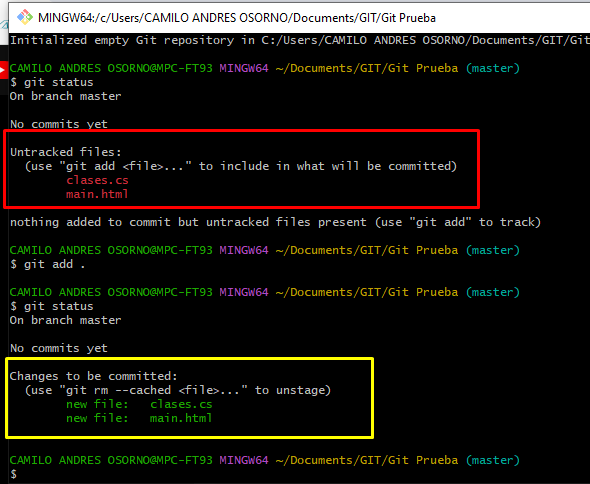
Para ver los archivos que hay en el área de trabajo utilizamos el comando git status



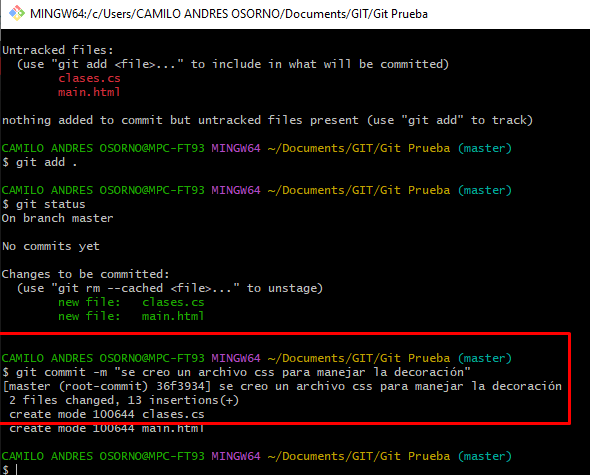
Agregar el archivo a el área de preparación con git add .



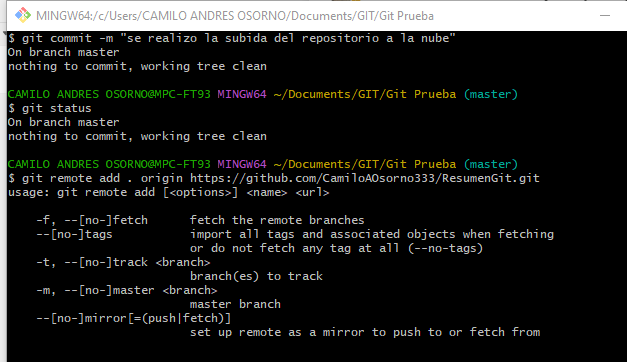
Para ver que se cambiaron de lugar los archivos, le damos de nuevo git status



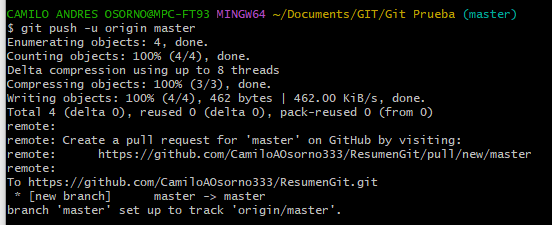
Informar los cambios realizados con git commit –m “comentario”



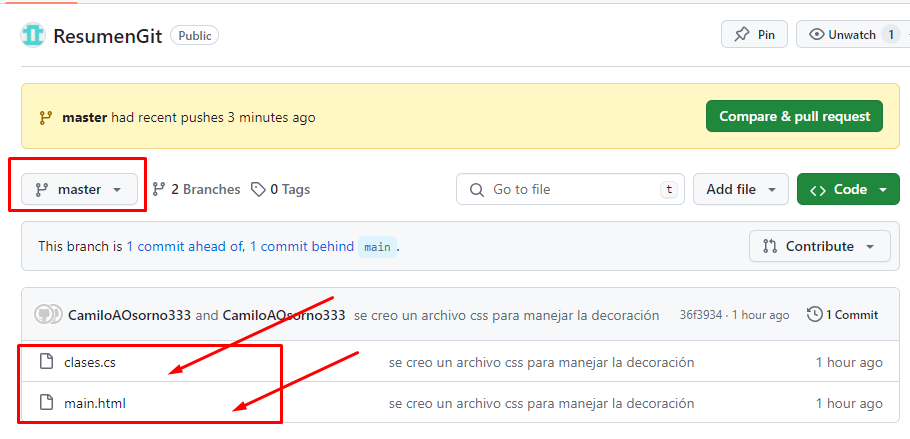
Creamos nuestro repositorio remoto con git remote add origin + url



Cargamos los archivos al repositorio de la nube con git push –u master

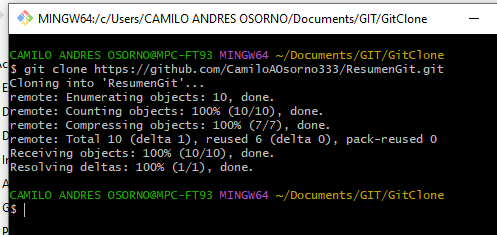


Verificamos en el repositorio en la nube

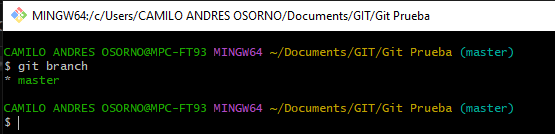


Clonamos el repositorio con git clone + dirección url

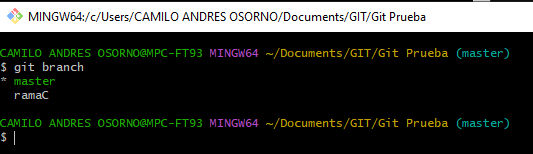
Para esto debemos crear una carpeta diferente a la que tenemos, lo hacemos para crear ramas dentro del repositorio



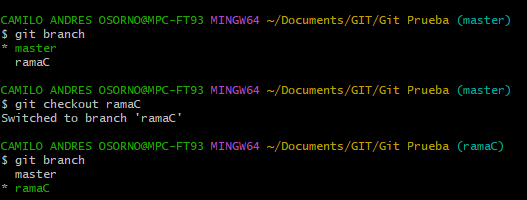
Para ver en que rama estamos git branch



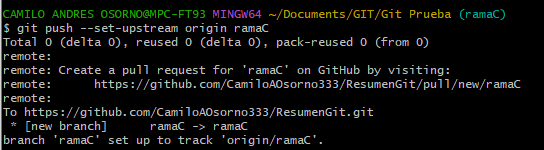
Para crear una rama nueva git branch + nombre

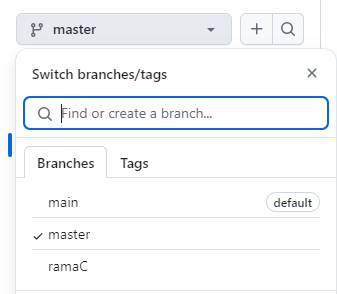


Para cambiar de rama de trabajo git checkout



Para enviar la rama que hemos creado al repositorio original git push --set-upstream origin NombreRama





Para cargar los cambios de nuevo

Nos ubicamos en la rama con git branch nombreRama

Git status para ver los cambios y los archivos

Git add . para agregar los archivos

Git commit –m “comentario” para hacer seguimiento y advertir lo que se hizo

Git push para subir los cambios git push

Para que el dueño del proyecto acepte nuestro trabajo y fusione las ramas, vamos al repositorio en la web y presionamos pull request y le pedimos que nos de el merge

