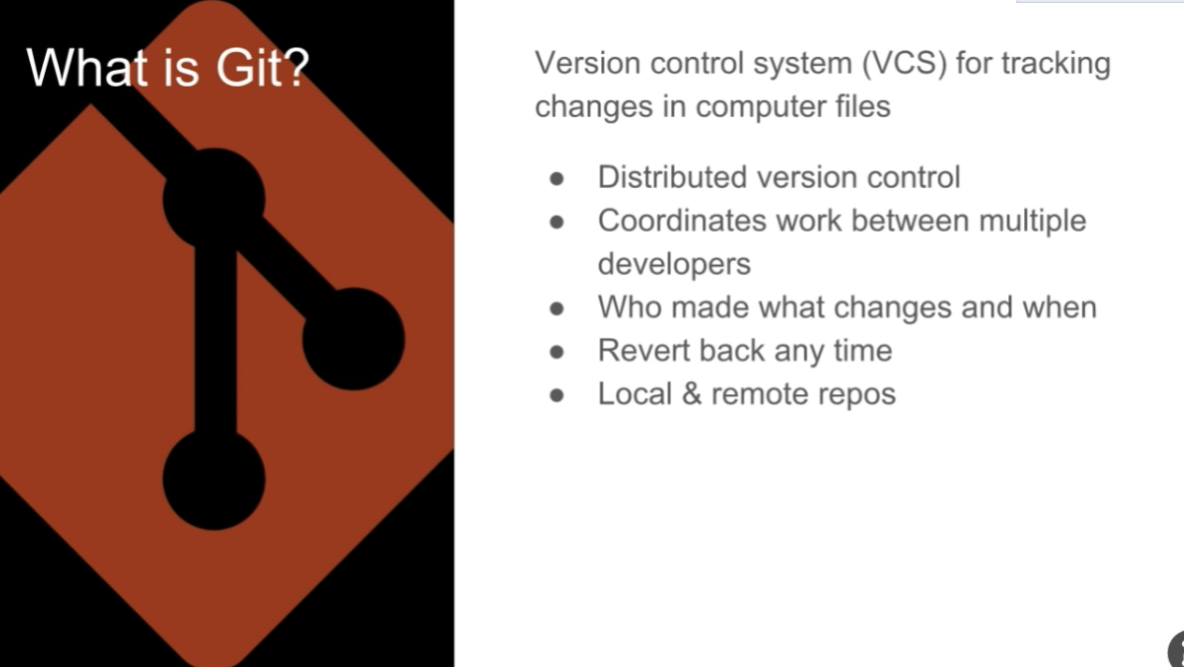
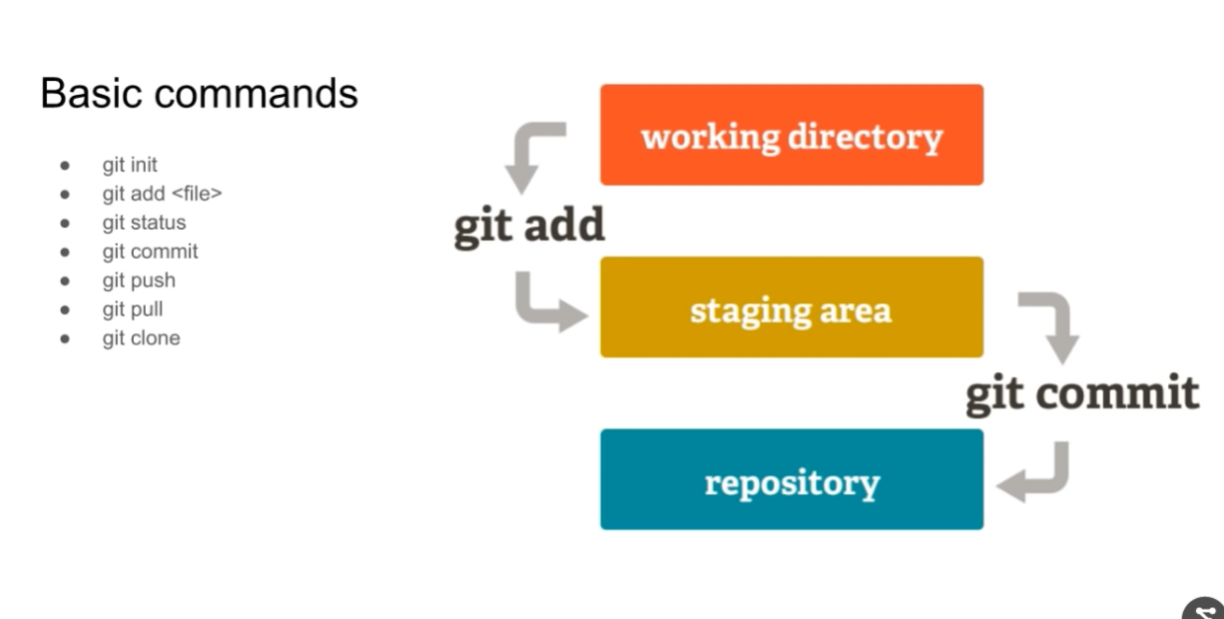
**GIT**





3 estados:

Working -> Area donde yo estare trabajando con todos mis archivos. En local , en mi pc

Stating área-> Es donde se agregaran todos los archivos donde se preparan para el guardado, para la lista de versiones

Repository -> y cuando ya estamos listos ya pasamos todo a el repositorio de lo que hay en el stating área

COMANDOS BASICOS

\* git init Oye Git, voy a usar estos documentos contigo ¿vale?

\* git add <file> Pasa los docs a staging area

\* git add . Pasa todos los archivos

\* git commit Pasa los docs de staging area a repository (Después de esto se te va a abrir el editor de código VIN en donde tendrás que escribir un comentario, si te quieres evitar abrir VIN entonces utiliza los siguientes comandos:)

\* git commit -m "comentario" Lo mismo que el commit regular, pero ahora no necesitas entrar a VIN

\* git push Subir los docs a un server (Github)

\* git pull Traer los docs de un server, traer los cambios de tus compañeros

\* git status Ver en que status (wd, sa, r) están los docs

\* git clone Hacerte una copia de lo que está en el server a tu PC

\* git checkout -- <file> Para revertir los cambios de los archivos

\* git diff <file> Para ver las diferencias hechas en los archivos

\* git branch Ver las ramas que hay ("master" es la rama default)

\* git branch "nombre" Crear una nueva rama

\* git checkout "nombre" Ir a una rama en especifico

\* git config -- global user.email "email" Para configurar email del usuario

\* git config -- global user.name "nombre" Para configurar nombre del usuario

vin Es el editor de código de git desde la consola, ahí escribes un comentario para la nueva versión que estés versionando (si no te deja escribir presiona a letra i). Cuando termines presiona esc y luego :wq (write & quit)pasa salir.

.gitignore Es un archivo reservado de git que tenemos que crear si queremos decirle a git los archivos que no vamos a utilizar y así decida ignorarlos.

Escribe dentro de el archivo .gitignore los nombres de los archivos que desees ignorar.

Nota: Las carpetas se escriben solas y los archivos con su terminación.

yo añadiria

Git chekout –b nombre de la rama: nos permite crear una nueva rama y cambiar de rama a la misma vez con un solo comando

Git branch –m nombreActualDeLaRama NombreNuevaDeLaRama: para cambiar el nombre de la rama

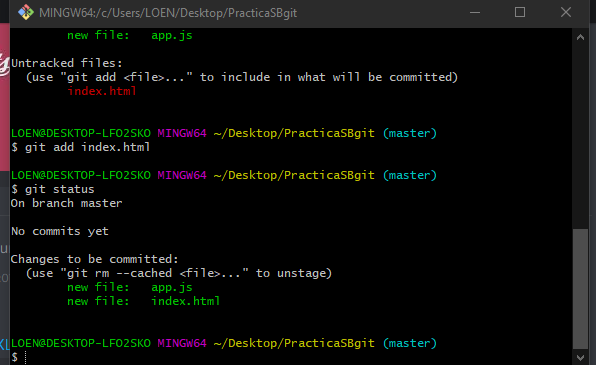
Git branch –d NombreDeLaRamaQueQuieroEliminar: eliminar rama

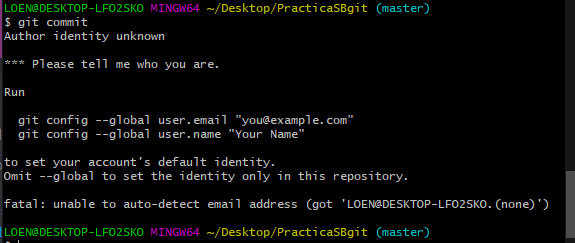
Git chekout –b nombre de la rama: nos permite crear una nueva rama y cambiar de rama a la misma vez con un solo comando

Git branch –m nombreActualDeLaRama NombreNuevaDeLaRama: para cambiar el nombre de la rama

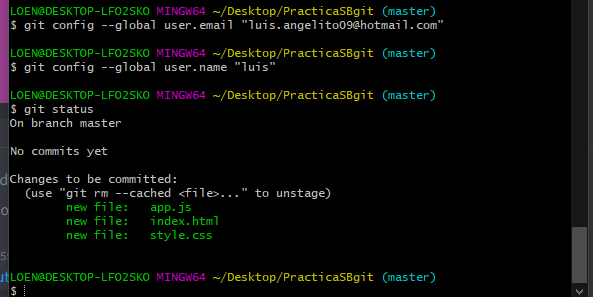
Git branch –d NombreDeLaRamaQueQuieroEliminar: eliminar rama

pd: no se si en este video lo especifica :v, te robo lo de config de usuario y email me acuerdo que me puse como loco buscando en internet xD





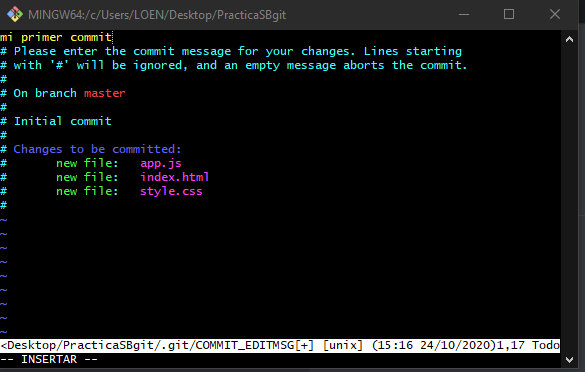
Esto pasa cuando no le decimos quien ha realizado los cambios



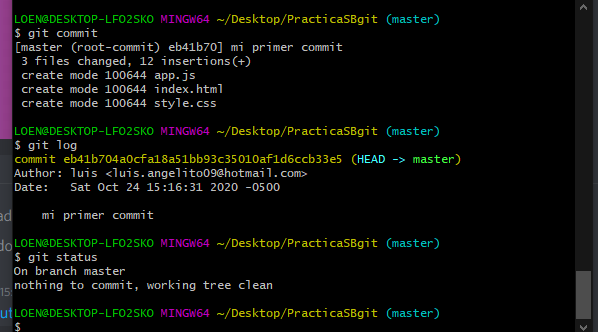
Con esto ya le configuramos las cosas , quien es la persona que esta editando

Con i

Si no nos deja escribir en el VIM



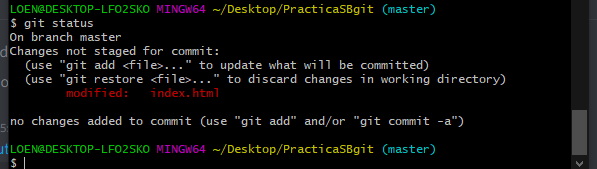
Esc y :wq



Git add

Git Commit

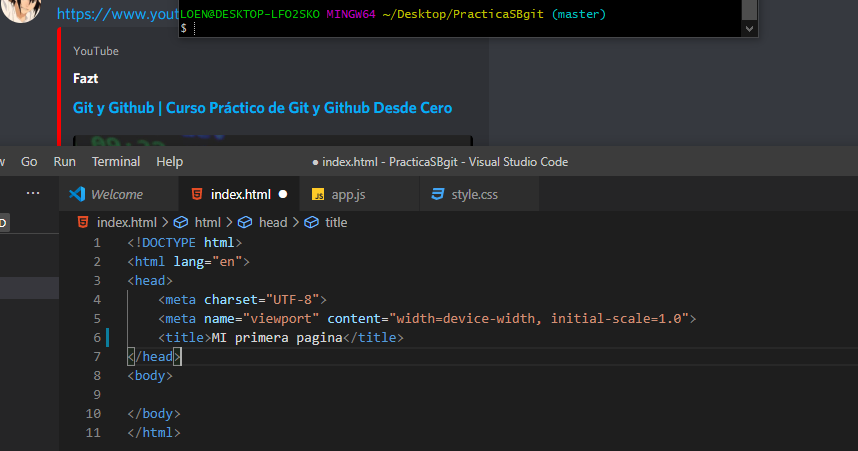
Git status



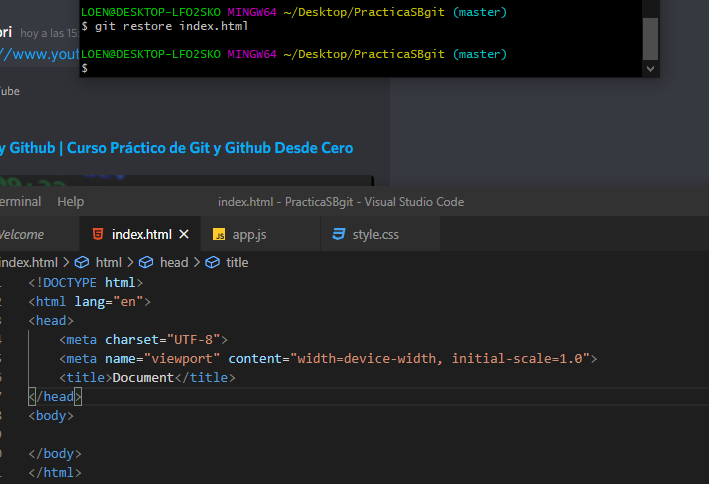
Con el git restore lo que hace es que si queremos devolvernos a como lo teniamos antes, automaticamente con ese codigo

Se nos devuelve

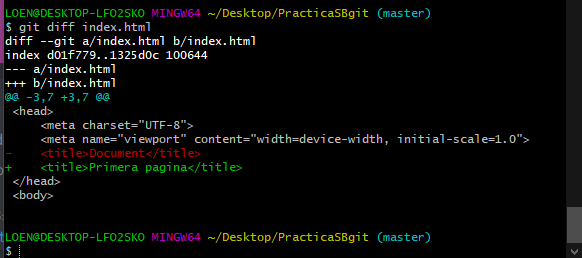
Ej



Yo cambie y puse mi primer pagina, pero si quiero devolver a como estaba antes, como lo tenia guardado en git seria asi

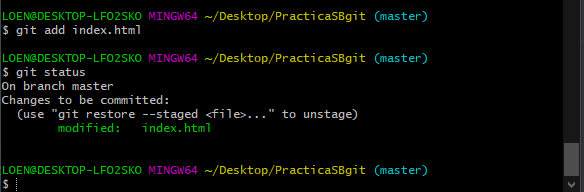


Con git diff archivo.extension

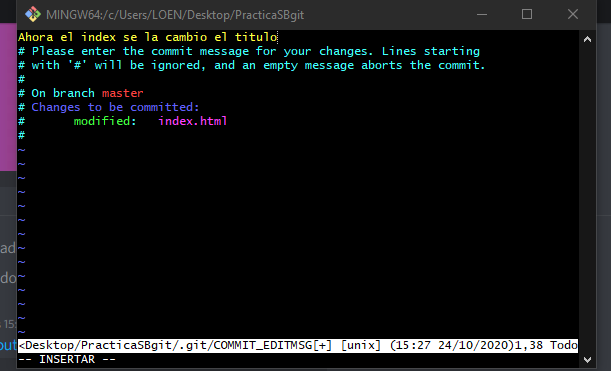


Nos dira cuales son los cambios3

Con el git add también se usa para cuando el mismo archivo tiene cambios y lo vamos a subir



Y por ultimo usamos git commit



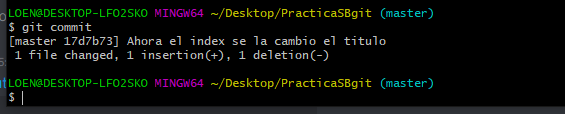
Lo de amarrillo es un comentario que le debemos hacer cuando subimos las cosas

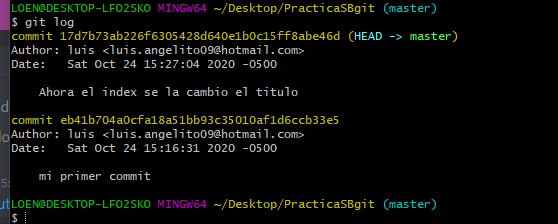
:wq

W write

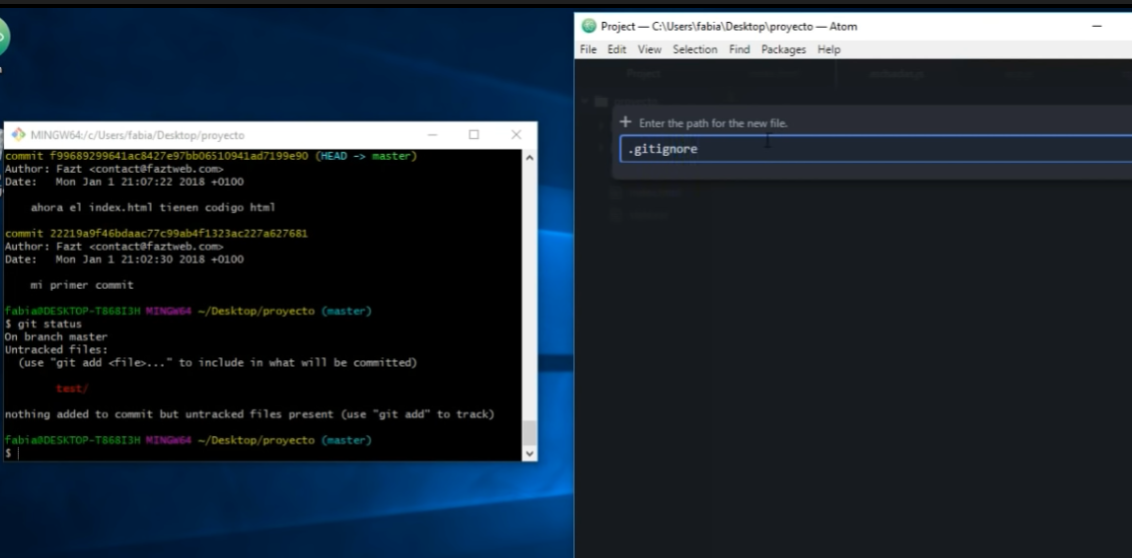
Q quit

W para escribir y q para salir

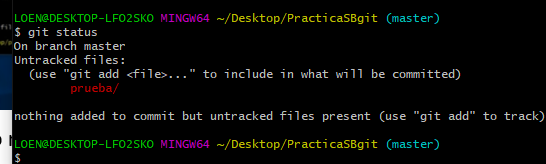




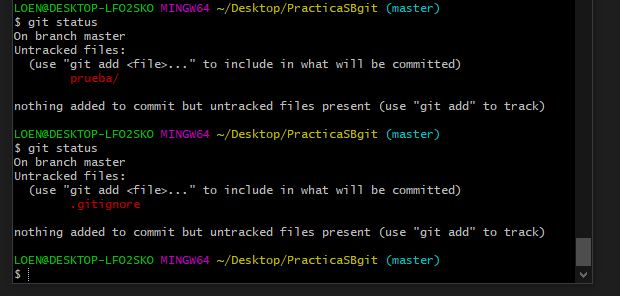
Esas son como las fotos del código anterior y la de ahora asi se van viendo



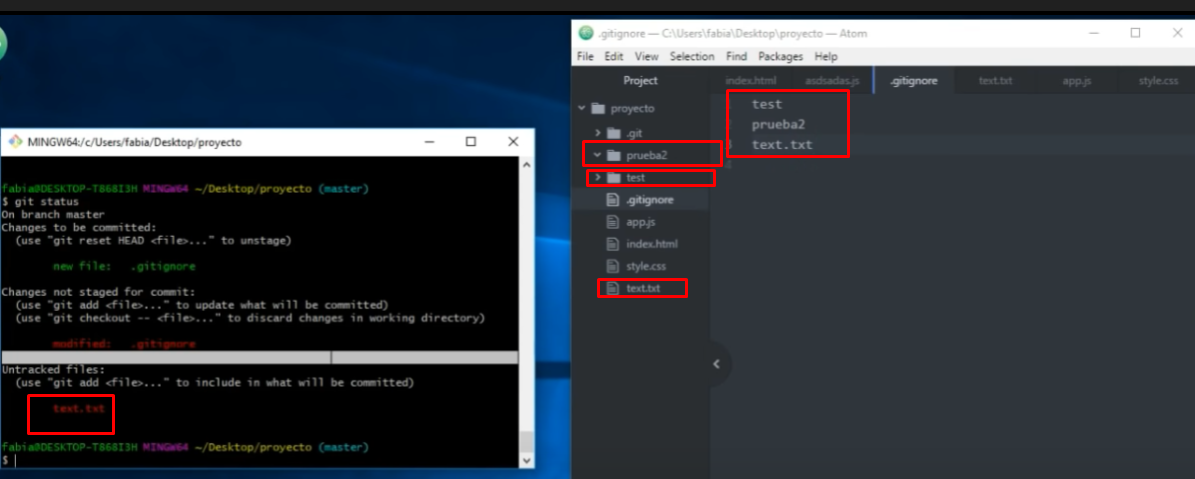
En este archivo muy importante para cuando queremos describir que cosas ignorar de nuestro proyecto



Vamos a hacer que ignore eso

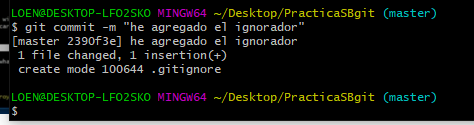


No aparece el otro pero si debemos incluir el .gitignore a nuestro commit





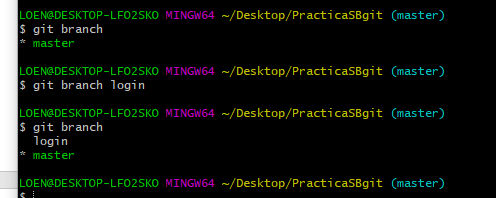
Con el git commit –m , nos evitamos que se abra el editor vim y   
aquí hacemos el commit y le pasamos el comentario de una vez

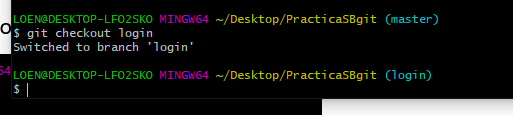


Con esto hacemos versiones alternativas de la master, osea creamos ramas alternativas

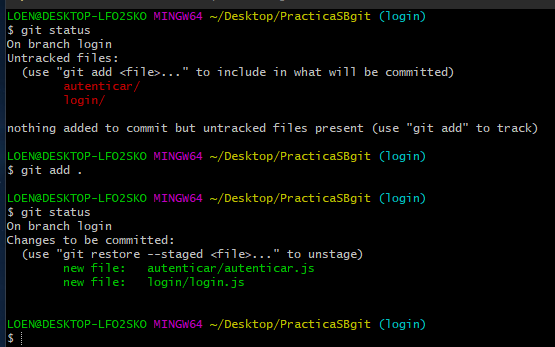
Con el git branch nombre   
creamos ramas

Con git checkout cambiamos entre ramas

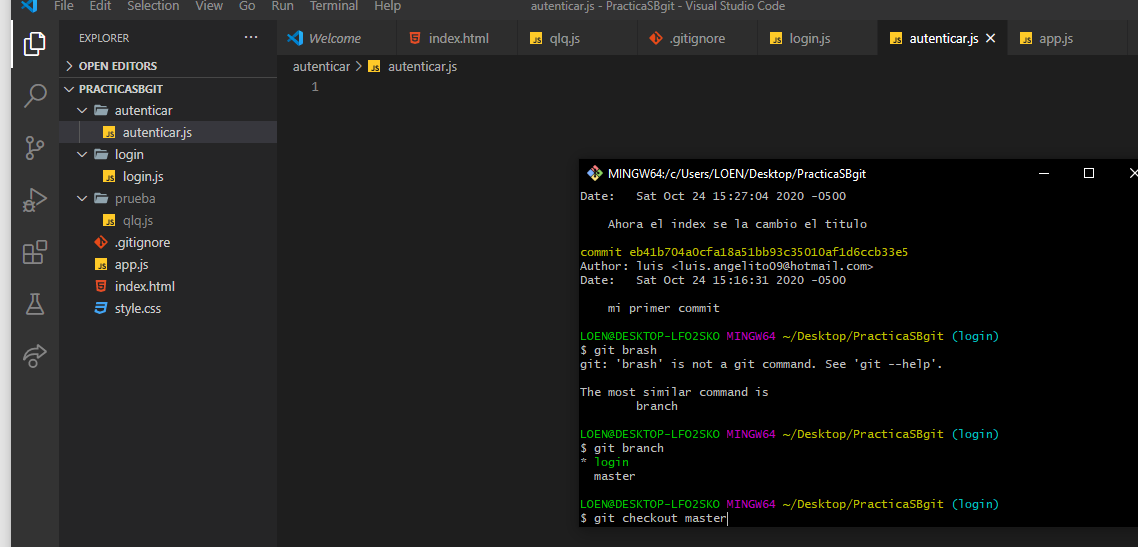




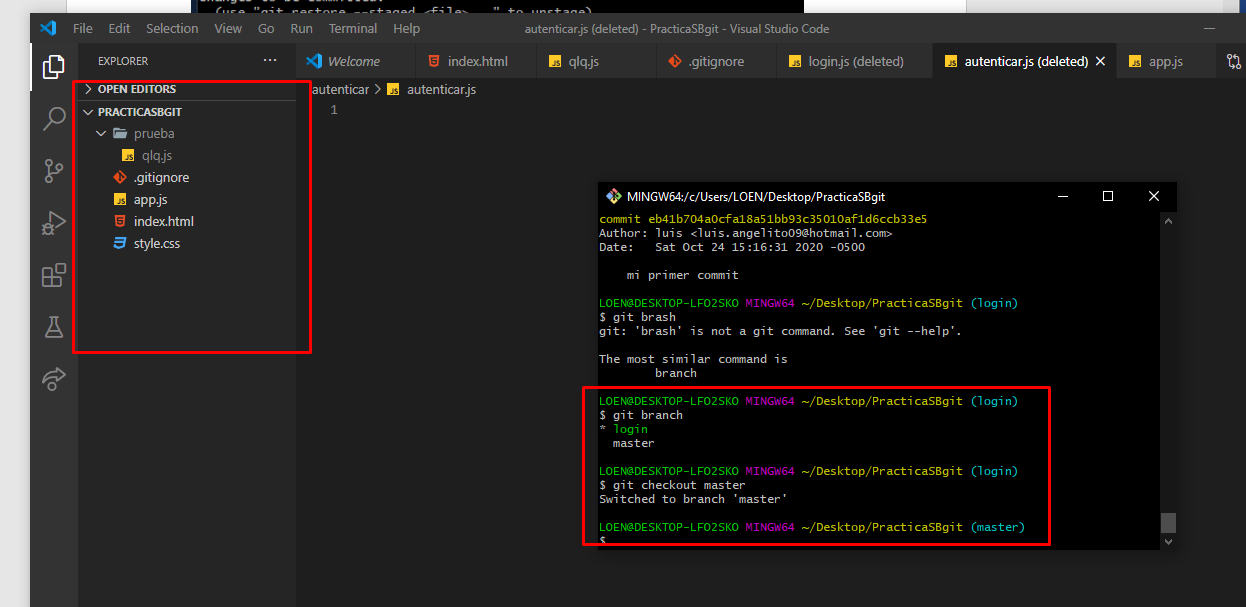
Colocando el git add .   
agregamos todo lo que este disponible o tenga cambios



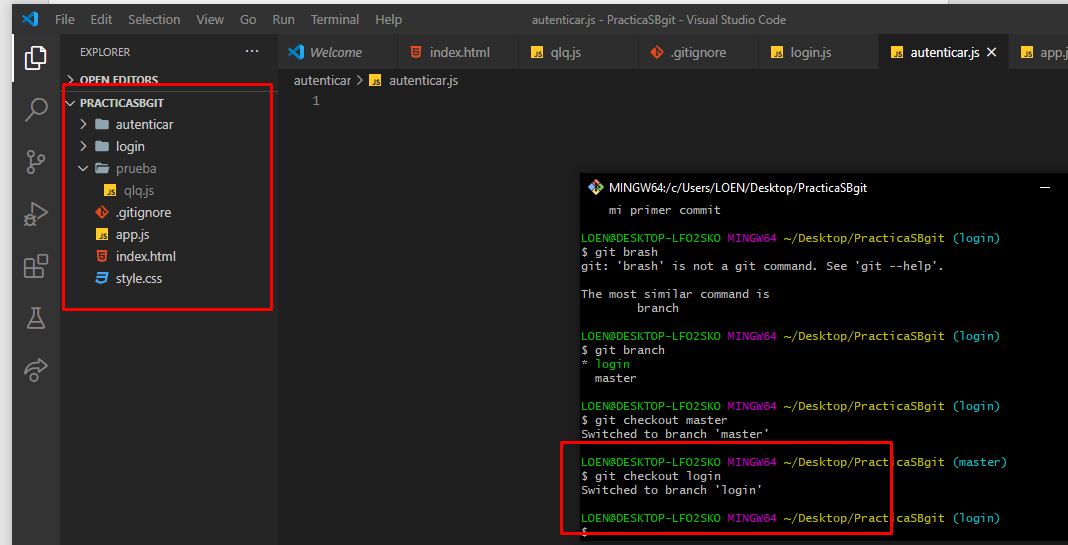
Como ahora mismo estamos en la rama login, entonces   
aquí están todos los cambios, pero apenas cambiemos de rama, eso ya no se vera   
lo notaremos en el editor



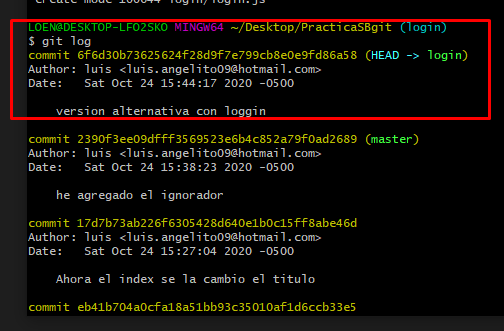
Rama master

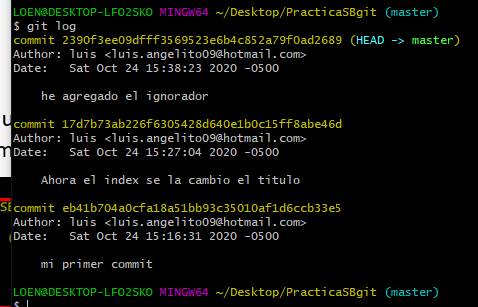


Rama Login

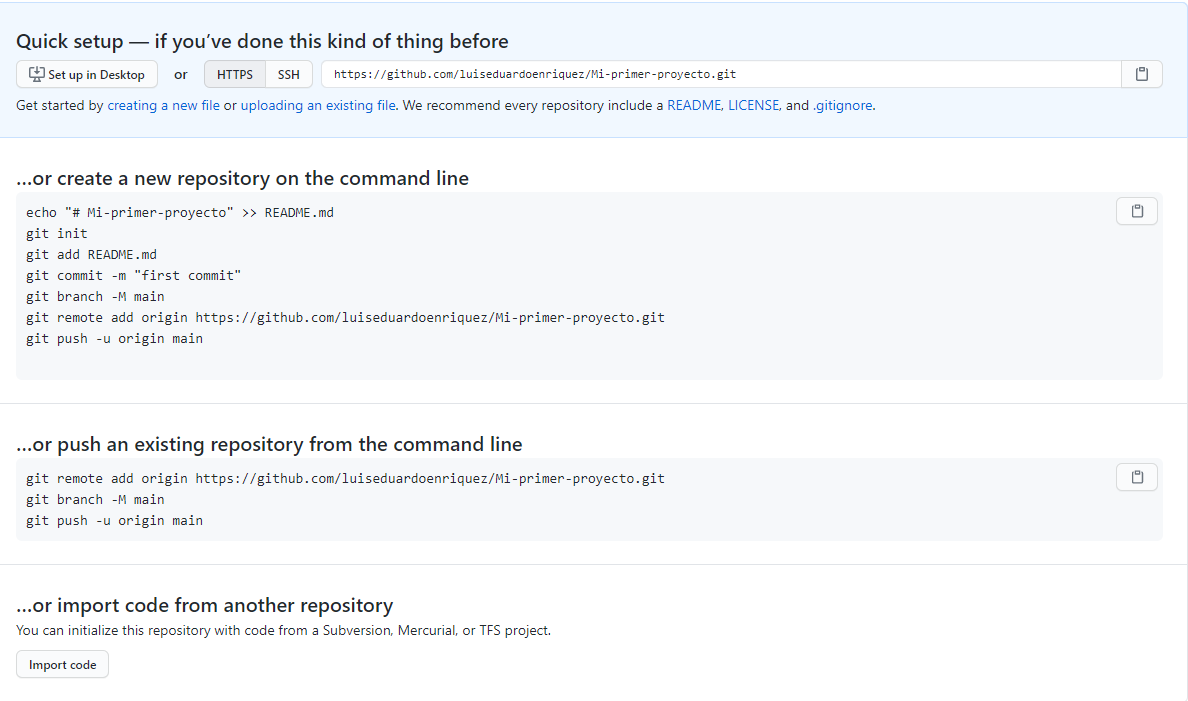


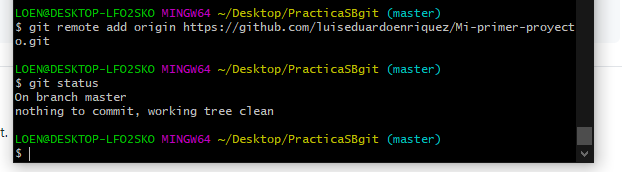
Y si revisamos en el git log   
tmabien visualizaremos que los commit que hacemos alla no se mostraran aca por que no es la misma rama



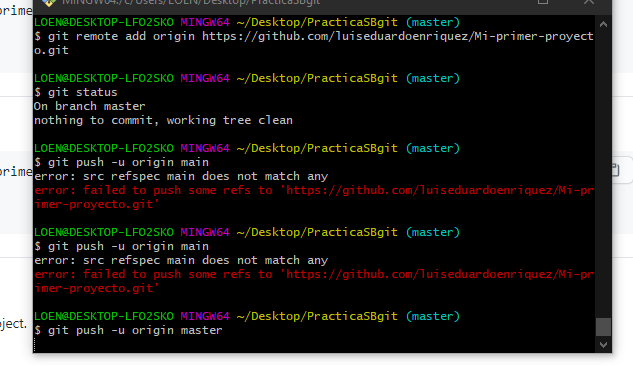


SUBIR LO NUESTRO A UN REPOSITORIO EN LA NUBE

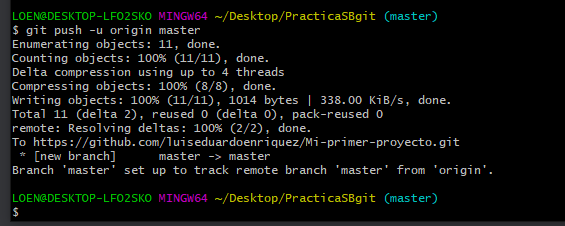




Y todo sigue funcionando normal



Error era porque le estaba colocando main en vez de master



Y asi clonamos un repositorio

