Ver ejemplo git en carpeta dcuemntos. /git

Git init: ubicado desde la carpeta de trabajo en la terminal, empezar a usar git en esa carpeta:

Git add myproyecto : añadir el proyecto a la fase de iniciar.

Git commit –m “version nueva”: envia el proyecto y lo guarda en el control de versiones.

Hasta este punto esta guardado en el disco duro

envia todos los archivos que estan en la carpete myproyecto. Luego hago el commit, esta como otra forma de hacerlo el commit.

Git status para evaluar como estan el control de versiones de manera local, si esta todo comitiado.

Git show: muestra loc cambios y como y quien hizo.

Git log myarchivo, muestra cambios.

Hit push envia el repositoio de myproyecto a otro repositorio remoto. Y con pull se trae.

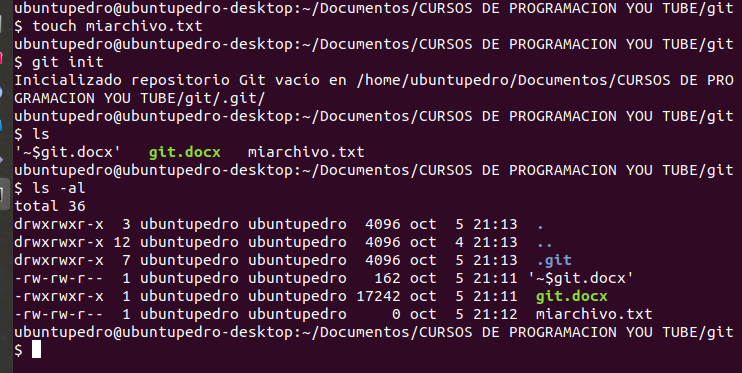
Git sirve para control de versiones en texto plano, es decir no archivos binarios, como texto que tiene negritas, letras grandes, fuentes, o un archivo Word con imágenes, solo es control de versiones para texto plano, como codigo.

Despues que abri la terminal desde donde esta mi archivos principales

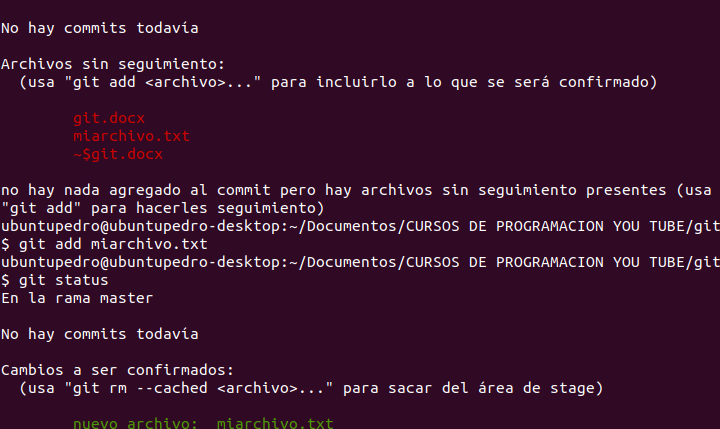
Voy a git init, y ahí me inicia el repositorio con los archivos de esa carpeta.

Me crea un archivo ocuilto .git que lo puedo ver desde la terminal con el comando ls –al:

Y es alli donde git guarda los cambios quie se agregan al repositorio local.



Con git status verifico si hay cambios que he hayan sido añadidos : en rojo si hay cambios y no estan añadidos, y en verde si ya estan añadidos, para añadir:



Para sacarlo antes de comitiarlo una vez lo he agregado:



Sale de giit add, para volverlo a añadir mas adelante si deseo. Git add ., para añadir todo.

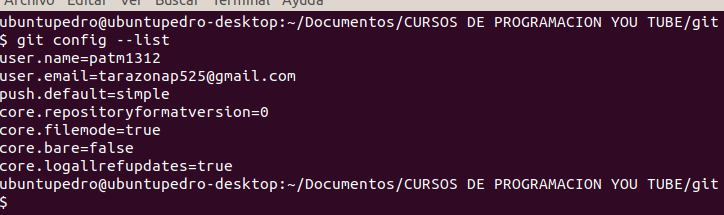
Una vez agregado de nuevo con git add, los archivos estan en estado de stage, es decir falta comitiarlos: o enviarlos al repositorio con :

Git commit –m “mi coimmit uno ”;

Pero antes debo configurar mi git para saber como identificarse y quien hace los cambios:

Git config🡺 muestra lista de como funciona git

Git config –list🡺 ver las confuguraciones de que esta hecho y que hace falta, y lo q falta el nombre de email dl iusuario



Para cofigiurarlo como crear nombre de usuario y correo:

Git config –global user.name “mi nombre”🡺cambiar todos los usuarios globales., cambiar la configuración del usuario de git.

Git config –global user.email “mi correo ”🡺cambiar todos los usuarios globales., cambiar la configuración del correo del usuario de git.

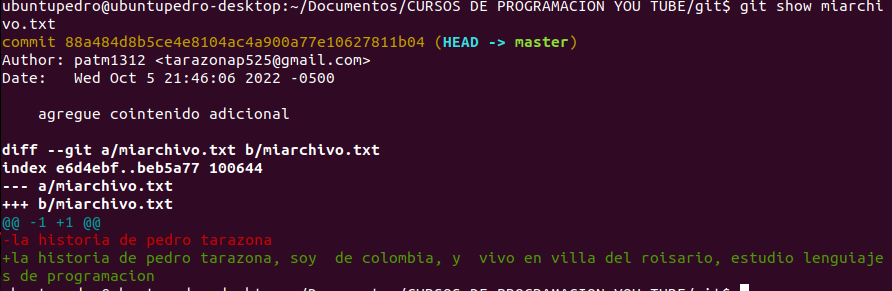
Luego de crear el usuario correctamente hago mi primer commit: en este punto todavía esta en local,

Suponiendo que ahora modifique el codigo o arhivos, entonces puedo volver a modificar y agregar con add ay liuego commitiar los cambios para subiirlos al repositorio local de git: todavía no los he subido al rpositiorio de hithiub:

Puedo agregar con git add . 🡺 para subir todo.

Git log 🡺 muestra cambios, hhead master es el archivo mas reciente de la rama trabajada.

Git show miarchivo 🡺 muestra los cambios que han xistido en el archivo coimmitiado:



Diff es que toma la version anterior y la nueva y me muestra la diferencia

Index es un indicador de la bd de git de donde estan los cambios

--a ++b las versiones que van apareciendo

Los bits donde cambiaron

Y muestra informacion de lo quie quite agregeue.

Añadiendo un tercer cambio:

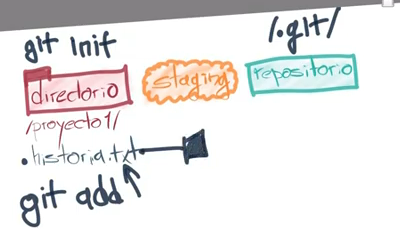
Verfificar con git satatus, añadir con add, y luego commit pero sin escribir mensaje:

Git commit🡺 me abre una ventana para escribir el mensaje obloigatorio, lo escriibio y luego me salgo:

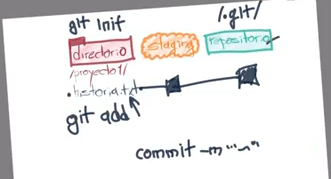
Git diff version1comimit version2coimiit🡺 para ver las diferencias de cambios qiue se han hecho e n los cambios en l bd de git de mi pc, copio y pego el codigo de commit a comprara que se obitnene con git log:

$ git diff 0a01b51e547d66fa7f4ae15107739a3de4cbf2fc 5717e90514c010491afea864b9b27db6b2f2c663

En este punto los archivos cuando puls git add, pasan al estado stagin, que es un directorio en la memoria ram del pc.

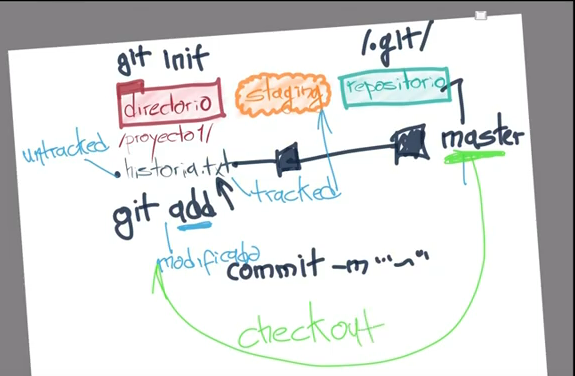


Liuego cuando hago el commit del archivo, pasa a el estado en la carpeta creada como .git, es ese repostorio con el nombre de master

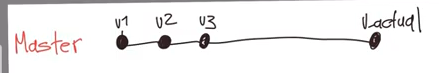


Cuando un archivo recien se modifica puezs este esta en estado untrked, es decir que esta por hacer add, y cuiando hago add pasa a hacer traked o ya esta trakeado, para luego poder hacer el commit y pasar al rpositoirio.

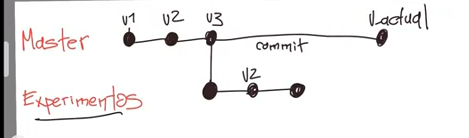
Cuando yo quiero traerme ls cambios que estan en el repositorio a mi archivo local es decir a mi estado traked, entonces: uso l comando checkout:



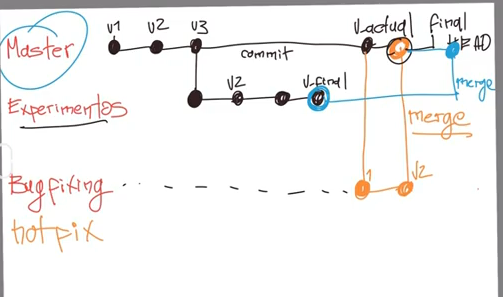
Por lo general se trabaja en la rama master, entonces cada vez que hago un commit aparece una nueva version llamada versio 1, luego version dos…



A partir de la version x de la rama master decido crear una nueva forma de crear versiones pero a partir de otra rama que se llama experimental, para hacer o trabajar con diversas formas en el proyecto, y se empezaran a crear versiones a partir de la version de duplicado de la rama master, pero quie cuando pase el tiempo se ira trtanasformando y sera diferente a de la rama master actual:



Cuando por ejemplo tengo un bug o error en la rama master y e la version actual, en ese caso debo creqar un a nueva rama que se llama bugfixing o hotfix que es para empezar a corregir errores y construir versiones en esa rama , y luego cuando ya tengoi corregido esos errores pues fusiono esa version con la ultima verison de la rama aster con el comando 🡺 merge



Tambien se puede fusionar con la rama experimental con la rama master y experimental, y tambien la fucnion se hace con el comando merge.

Csi quiero retroceder a la version anterior, debo mostrar los commits hechos con

Git log🡺

Git reset🡺 es para volver a la version anterior, se pone luiego el numero largo de commit para ir a esa version.

Git reset hard🡺 vuelve a la version antiigua todo el repositorio, es pelioigroso usarlo.



Cuando le de git log de nuevo, solo aparece la version a la que actualicey boirra todo los commits , excepto la version elegida.

Git reset soft 🡺igual que el anterior pero lo que est aen stagging oen add, sigue ahí.

Teniendo como archivo una carpeta con uin documento css y por fuera un archivo html. Lo agrego a git y lo agrego con add, lueg ogit crea carpetas como rutas y las crea.

Si por ejemplo quiero ver coimo era una version anterior a el actual archivo o a el primer commit, entonces copio el numero de commit a el siguiente comado

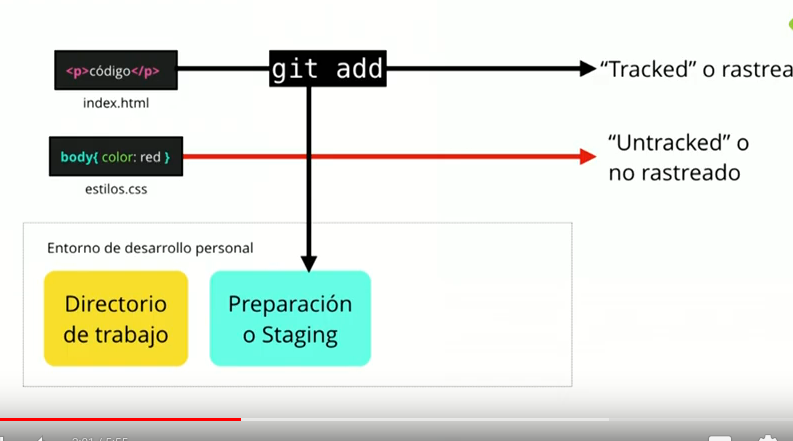
🡺git checkout numerocommit + nombredearchivoderegreso:



Luego intento abrir el archivo y aparce como la version que especifique en el comando, con esto lo puedo ver, y si quiero guardarlo asi debo hacer el commit, pero si queria solo verlo y volver a la version que tenia antes, es decir la mas actualizada entonces:



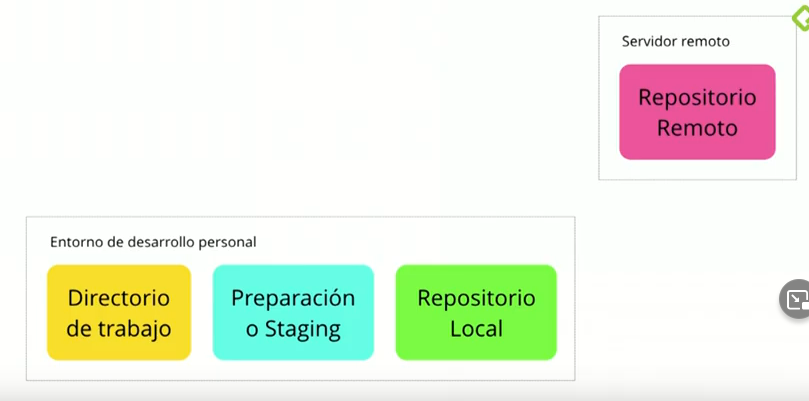
Resumen:



Si los archivos css no se ha hecho add. Pues aparecera como untracked luiego:

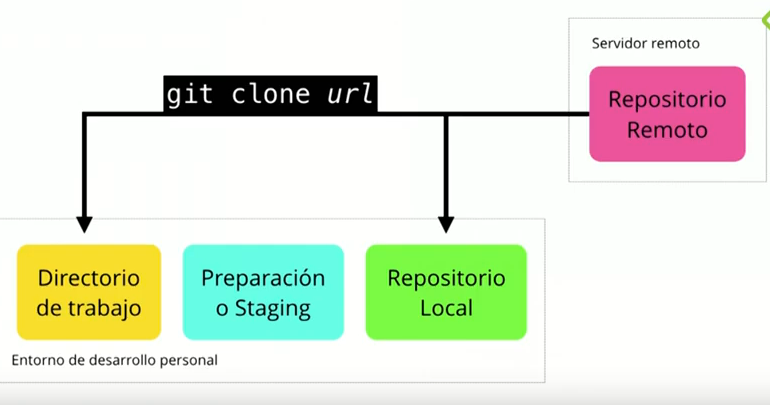


Unidad 2

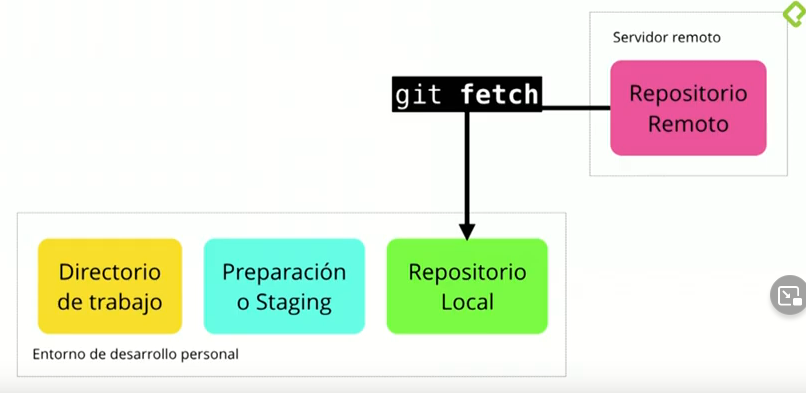


Se necesita tener todo lo comitiad en un espacio de trabajo remoto para que muchos usuarios puedan ver el codigo.

Git clone url🡺 es para traer los datos de iun servidor remoto como github, l oque hacees clonar losdatos y alamcena una copia del master en el directorio de trabajo y crea una base de datos historicos en el repositoirio local:

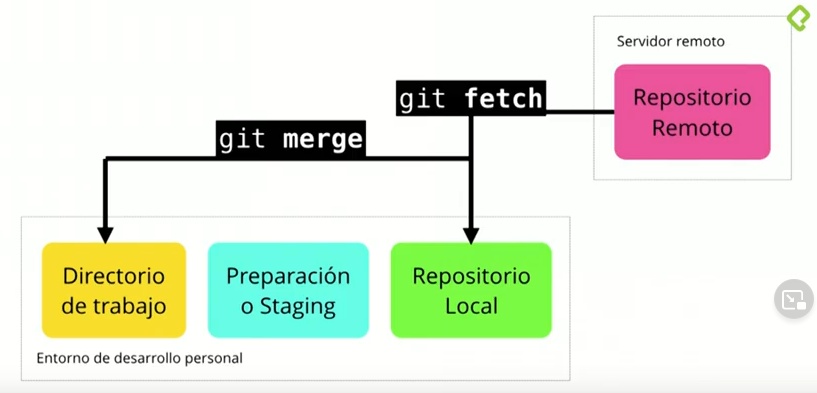


Cuando ya tengo la ultima version lista la pudo subir al repositorio a travez de git push🡺 qiue suibe todo al repositorio remoto.

Git fetch🡺 sirve para traer datos cuando alguien los ha cambiado en el repositorio remoto y

los quiero bajar a mi repositorio local:

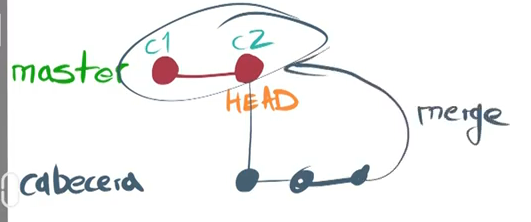
pero no los copio a misarchivos locales, para copiarlos a mis archivos locales debo:

git merge🡺 quie es fuisionar la ultima version que esta en el repositorio local con mi version actual 

es decir quie es lo mismo que hacer git clone, pero con git pull🡺 copio el repositorio local y la base de datos de cambios y el directorio , para tener coipia actualizada de loque paso en le repositorio.

Ramas

Si por ejemplo tengo el archivo articulo.html, que lo tengo en el repositorio d master, entonces, lo que quiero hacer es que apartir de alli crear otra rama llamada por ejemplo hedaerr que creee el header de la pagina html a partir de ese master, y luiego fusionar el master y la rama header.



1. Entonces le hago camcios y agrego codigo a mi html y css.
2. 2. Git commit –am “mensaje”🡺 $ git commit -am "cambios a html y css, nuevo contenido"

cion este comando hago el git add y el commit junto, pero solo funciona a archivos que habia hecho git add, previamente.

1. Con git show aseguirar que mi head esta apuntando amii area de trabajo de master:

$ git status

En la rama master

nada para hacer commit, el árbol de trabajo está limpio.

$ git show

commit d213287996e6ae7fee67011c55454d20de7aac29 (HEAD -> master)

1. Git branch nombrederama🡺 de esta forma creo la rama a tranajar.
2. Vuielvo a dar git show para verificar:

$ git show

commit d213287996e6ae7fee67011c55454d20de7aac29 (HEAD -> master, cabecera)

Author: patm1312 <tarazonap525@gmail.com>

Date: Fri Oct 14 21:28:01 2022 -0500

Lo que me dice es que el ultimo ocmmit esta pegado a la rama master y la cabecera, a dos ramas distintas.

1. Git checkout nombrederamacreado 🡺 con este comando me cambio a la rama para trabajar alli:

ubuntupedro@ubuntupedro-desktop:~/Documentos/CURSOS DE PROGRAMACION YOU TUBE/git$ git checkout cabecera

Cambiado a rama 'cabecera'

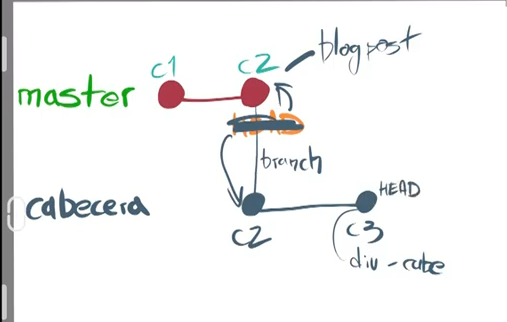
1. Verificao con git status:

$ git status

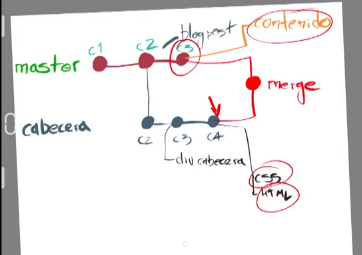
En la rama cabecera

nada para hacer commit, el árbol de trabajo está limpio.

1. Ahora puiedo empezar a trabajar en esa rama cabecera. Grego cosas o modifico el codigo y asi…
2. Hago los add y los commits a esa rama cabecera
3. Puedo cambiar de rama con git checkout y automáticamente se cambia el codigo en vcode, porque el head se pasa para el ultimo commit de la rama master o maestra.

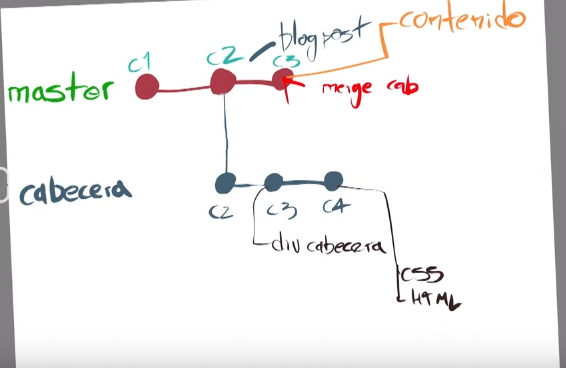


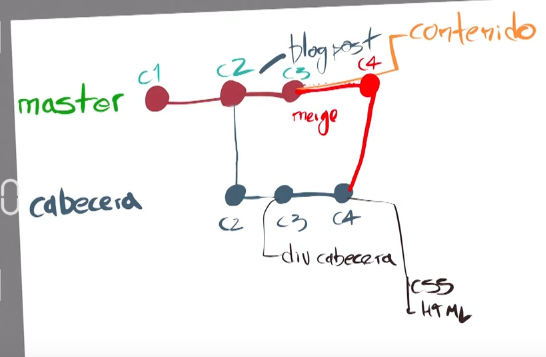
Hacer cambios o agregar cosas en la rama cabecera en la parte de css y cabecera html, y en larama master solo agrego nuevo contenido a ahtml, es decir que voy a tener dos archivos distintos cuando intente fusionarlos conmerge cuando tenga listos los commit de cada uno respectivamente:



Cuando se hace la fusion con merge el head o arama de trabajo creada desaparece para unirse a la rema master, pero hay forma de que se mantenga la rama cabecera para utiliazarla mas adelante.

1. Estoy en la rama que quiero hacer el cambio o agregar, en este caso la rama cabecera, y agrego cosas a css.
2. Hago el comit con git commit –am “mensaje”;
3. Me cambio al aotra rama de trabajo con git checkout “master”, y los cambios desaparecen.
4. Liego hago los cambios y agrego cosas diferentes a mi trabajo, y lo comiteo.
5. Luego fiusiono todos los cambios con merge pero me posiciono en master con checckout para que este siga siendo el principal y no se borren sus commits, pero si hay doslineas de codigo iguales qque fueron afectadas pues se haraun conflicto y nose puede fucionar, en este caso los cambios fuieron en distintas partes del codigo:



1. Git merge nombre de rrama🡺 para fusionarlas, para saber la rama puedo usar el comadn ogit branch. Git merge es un nuevocommit a la rama master para este caso por ello me obliga a escribir unmensaje de codigo en vim:
2. Desde la rama master se crea otro comit numero4 que se trae el commit numero 4 de la rama cabecera y lo fusiona en ese nuevo commit
3. 
4. Hacer otro ejemplo de fusion para entender: