Guía Git

1. Descargar Git del siguiente link <https://git-scm.com/downloads>.
2. Instalar Git.
3. Ya teniendo instalado Git en la máquina, se procede a configurarlo, para ello abrimos la consola de Git Bash y ejecutamos el comando git config --global user.name “nombre”, cambiando la palabra nombre que está entre comillas dobles por su nombre.
4. Ahora hay que configurar el email, en la consola de Git Bash ejecutan el siguiente comando git config --global user.email “email”, cambiando la palabra email que está entre comillas dobles por su email.
5. Para saber si su nombre se guardó correctamente ejecutan el comando git config --global user.name y en la consola debe aparecer su nombre. Para saber si su email se guardó correctamente ejecutan el comando git config --global user.email y en la consola debe aparecer su email.
6. Para ver todas las configuraciones realizadas se utiliza el comando git config --global --list.
7. Para decirle a Git que monitoree los cambios que ocurren en una carpeta se utiliza el comando git init, con ello Git va a detectar los cambios que se le hagan a los archivos de la carpeta en la cual se ejecutó el comando.
8. Para saber el estado de nuestro proyecto usamos el comando git status, la consola va a mostrar los archivos que se modificaron (de color rojo).
9. Para decirle a Git que archivos ya están listos para agregarlos al commit utilizamos el comando git add nombre\_del\_archivo. Pero si quieren agregar todos los archivos al commit utilizamos el comando git add -A.
10. Después de agregar archivos al commit, ejecutamos el comando git status y la consola muestra los archivos que se agregaron al commit (de color verde) y los que aún no se agregan (de color rojo).
11. Para realizar un commit, usamos el comando git commit -m “mensaje”, cambiamos la palabra mensaje por un mensaje que especifique que se hizo en el commit.
12. Para ver todos los commits que se han realizado, utilizamos el comando git log. Para salir se presiona la tecla q.

Esto es lo que muestra la consola:



1. Para viajar en el tiempo y devolvernos a commits anteriores se utiliza el comando git checkout codigo\_sha, cambiamos la palabra codigo\_sha por el código sha del commit al que queremos volver.
2. Para movernos al último commit utilizamos el comando git checkout master.
3. Con el comando git log > commits.txt Git nos genera un archivo de texto con todos los commits que hayamos realizado.
4. Con el comando git reset --soft codigo\_sha, cambiamos la palabra codigo\_sha por el código sha del commit que queremos eliminar. Cabe destacar que sólo se elimina el commit, el código queda completamente igual.
5. Con el comando git reser --hard código\_sha, cambiamos la palabra codigo\_sha por el código sha del commit que queremos eliminar. Cabe destacar que se elimina el commit y el código.
6. NOTA: Hay que tener en claro que el HEAD es el commit en el que nos encontramos.
7. Para ver todas las ramas de nuestro proyecto utilizamos el comando git branch.
8. Para crear una rama utilizamos el comando git branch nombre\_rama, cambiamos la palabra nombre\_rama por el nombre que queremos que tenga nuestra rama.
9. Para cambiarnos de rama utilizamos el comando git checkout nombre\_rama.
10. Para fusionar dos ramas, primeramente debemos movernos a la rama que va a absorber, luego ejecutamos el comando git merge nombre\_rama, cambiamos la palabra nombre\_rama por el nombre de la rama que queremos absorber.
11. Para eliminar una rama usamos el comando git branch -D nombre\_rama.
12. Con el comando git checkout -b nombre\_rama creamos una rama y nos movemos inmediatamente a ella.
13. Con el comando git clone link\_proyecto\_git\_hub, clonamos un proyecto a nuestra máquina.
14. El comando git remote add origin link\_proyecto\_git\_hub sirve para enlazar el repositorio local con el repositorio remoto.
15. Con git remote –v podemos ver los repositorios enlazados.
16. Con git remote remove origin desenlazamos los repositorios.
17. Para subir nuestros commits a GitHub usamos el comando git push origin nombre\_rama, nos va a pedir nuestro nombre de usuario y contraseña.
18. Con el comando git commit --amend --m “mensaje” se edita el mensaje del último commit que se hizo.
19. Para forzar a Git a que suba a GitHub cambios que se realizan a los detalles de un commit se utiliza el comando git push origin nombre\_rama -f.
20. Cuando se trabaja en equipo, todos los integrantes del equipo van a estar realizando cambios a la vez, es por esto que cuando alguno suba sus cambios a GitHub, los demás deben actualizar su repositorio local, para ello se utiliza el comando git fetch origin para que los cambios se almacenen en origin/master, luego se ejecuta git merge origin/master para fusionar los cambios que hizo dicha persona con los nuestros.

NOTA: Es lo mismo que hacer git pull.