

Algunos comando útiles

A ordenar commit en línea git log –oneline

B presionar la tecla “q” para salir de algún lugar o Ctrl c

C configuración de nombre 🡪git config - -global user.name “Fill Santillan”

D configuración de email 🡪git config - - global user.email [fi@hola.com](mailto:fi@hola.com)

E guardar commit en archivo txt git log > commits.txt

F descargar todo lo que se subio en git hub 🡪 git pull origin master

G regresar a un punto el merge iniciado git reflog

git reset --hard <commit id>

1. Primer paso inicia le monitoreo de los archivos modificados

git init

1. Segundo estatus de nuestros archivos

git status

1. Agregar archivos a nuestro repositorio

git add -A

1. Agregar commit en el archivo modificado

git commit -m “se ha modificado tal cosa”

1. Commit creados

git log

1. Verificar el commit que deseamos ver

git checkout as23as33asdsadf33

1. Borrar commit

Borrar commit consecutivos git rebase –onto master~1 master

Despues del código anterior ejecutar el siguiente : git push origin master -f

Borra commit pero no el código git reset –-soft CodigoDeCommit

Borra commit y el código o lo que se modifico git reset --hard CodigoDeCommit

1. Como crear una rama

git branch NombreDeRamaACrear

1. Ver ramas

git branch

1. Elegir rama para trabajar en ella

git checkout NombreDeRama

1. Borrar Ramas

git branch -D NombreDeRama

1. Fusionar Rama

git merge NombreDeRamaAFusionar

Fast-forward no pasa nada, o mas bien no pregunta nada

Manual Merge

1. Crear Rama movernos hacia esa rama

git checkout –b NombreRama

1. Clonar Rama de Hit-Hub

git clone UrlDelRepositorio

1. Repositorio en Git-Hub, hacer remote o controlar lo que se suba al git hub

git remote add origin UrlDeRepositorio

1. Remover remote origin,

git remote remove origin

1. Ver remotes en git

git remote –v

1. Subir archivos a repositorio en git-hub

git push origin NombreDeRama --> Verificar en que rama esta XD

1. Editar commit

git commit --amend -m “Nuevo mansaje”

ahora subir el nuevo commit

git push origin master –f

1. Tags anotadas “tag nos ayuda a identificar los commit y ver en que versión vamos ”

git tag –a v0.8 –m “versión 0.8 de nuestro proyecto”

Nota: en donde dice v0.8 , el número “8” indica el ultimo commit al cual se agregará el commit

1. Asiganando tag a commit

git tag -a v0.1 -m "version 0.1 de nuestro proyecto " 344a8861e0e1f19f7580dd54dfe54d48e8824acd 🡪commit

1. Subir tag o compartir a git-hub

git push origin v.4 🡪 al igual que el punto 20 tenemos que verificar que ultimo comentario se tiene

1. Subir todos los tags

git push origin –tags

1. Trabajando en una organización

git fetch origin -> Aquí esta bajando los commit

git merge origin/master -> Aquí esta fusionando lo que descargo

Nota: Si modifican en la misma línea de código el primero en subir gana y al segundo no lo dejará subir su modificación por que no tiene la ultima versión que le gano el primero, debe hacer los pasos de git fech y hablar con su equipo y ver que código es el que se queda.

1. Upstream

git fetch upstream