

# GIT y GitHub

- Añadir solo los cambios

## Terminal de comandos

ls -al → ocultos, lista.  
touch vacio.txt → archivo vacío.  
history → historia de comandos.  
!72 → invocar el comando.

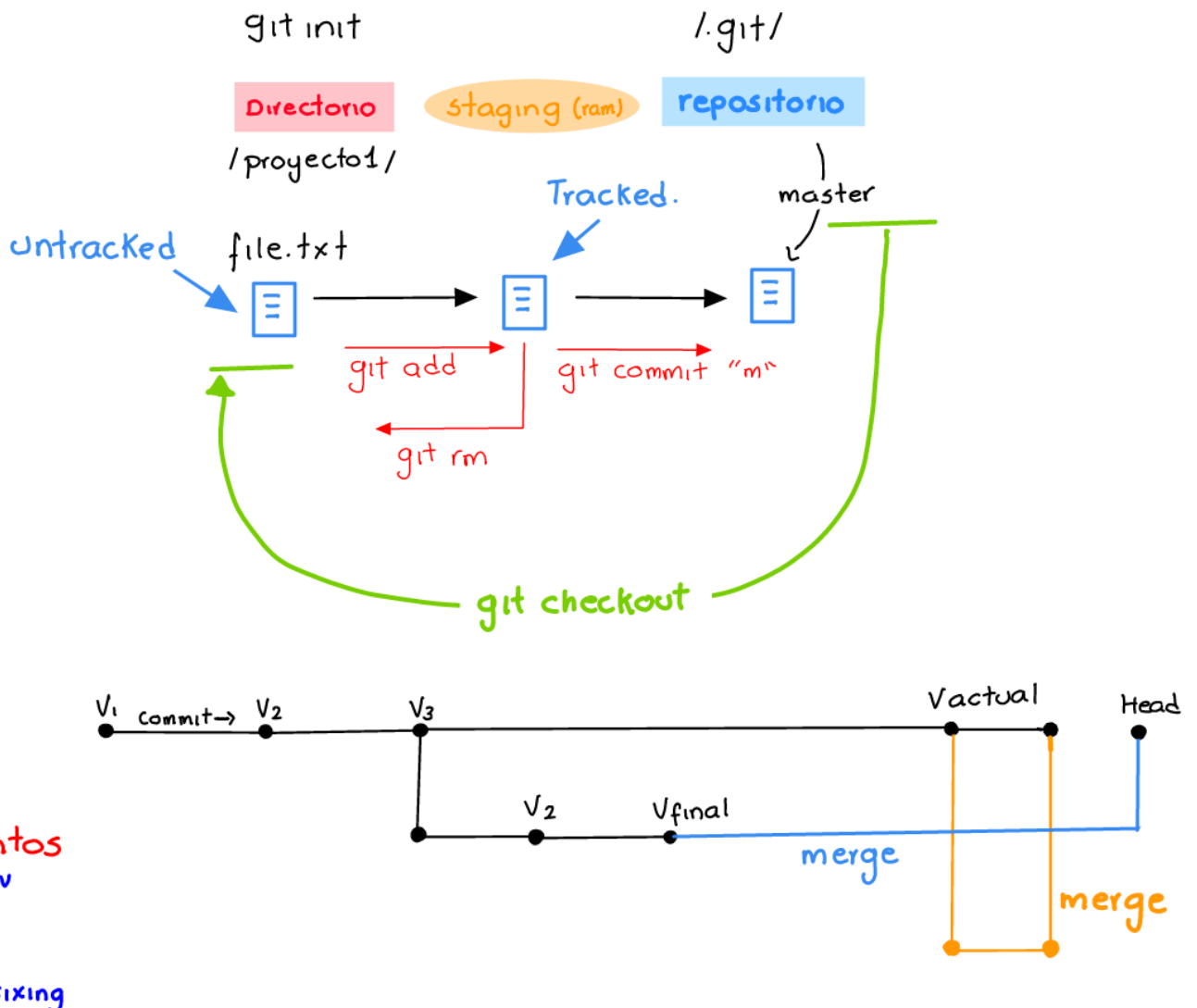
git init  
git add "biografia.txt" → arranca archivo.  
git commit -m "versión 1" → manda los cambios.  
git status → Estado de la base de datos.

git show → Cambios históricos hechos.  
git log "biografia.txt" → Cambios en un archivo.

git push → enviar archivos a un repositorio remoto

- git solo modifica texto plano.

## Estado del archivo.



git config → Configurar usuario y email.

git config --global user.name "Sergio Gonzalez"  
user.mail "..."

- acceder a una letra (rápido)
- - acceder a la palabra.

git config --list → ¿Cuál es la configuración que existe?

git log → Historia de un archivo. (Commits)

git show → Cambio mas reciente en un archivo.

git diff "address commit 1" "address commit 2" → Comparación entre commits

- Cambios eliminados.
- Cambios hechos.

git reset "address"

git reset "address" -- hard

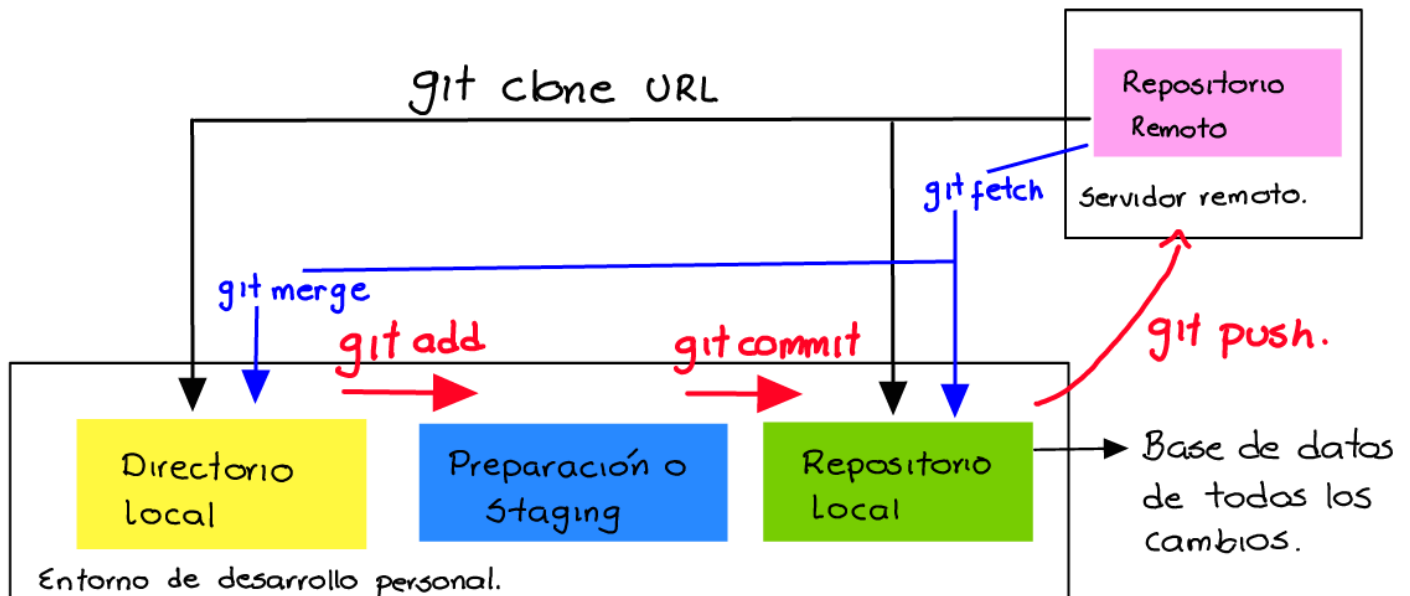
Todo queda en una versión anterior.

-- soft

Lo que esta en staging se queda, se recupera en el directorio el commit.

git diff → diferencia entre el disco duro y RAM  
(Directorio) (staging)

git checkout "address" file.txt → Trae archivo del commit que quiero!



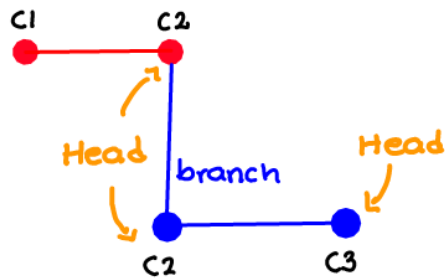
git init → Crea el staging y el repositorio remoto.

git add → Tracking de los cambios (staging).

git fetch + git merge = git pull.

git commit -a = commit + add.

master :



Cabecera :

Checkout Nombre  
branch  
→ Cambiar de rama  
a último commit.

git branch nombre → Crear rama.

## MERGE

- Un merge es un commit.
- Se debe hacer checkout a la rama a la que se quiere hilar la rama.
- Si en las dos ramas se hicieron cambios sobre la misma línea, git lanza un error y no deja hacer el merge.

→ On branch master

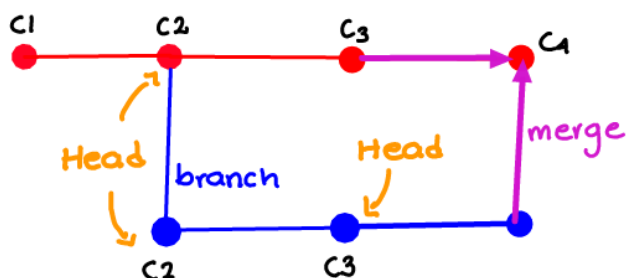
git merge rama a unir



Trae a la rama master los cambios hechos en ■

- Con git log puedo ver que git merge trae a la base de datos los dos últimos commits de las dos ramas y su fusión (merge).

master :



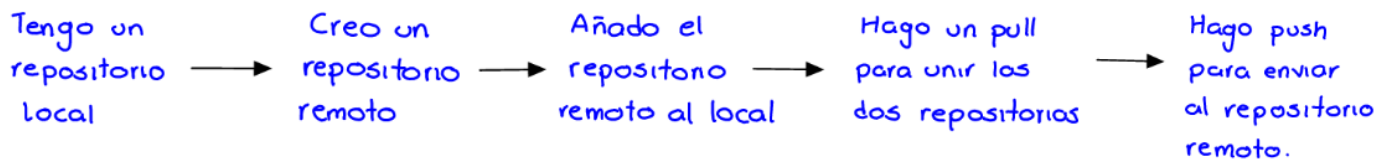
Cabecera :

## Solución a conflictos cuando se realiza Merge.

- Los merge se pueden hacer en los dos sentidos.
- Los conflictos son naturales.
- Comunicación entre el equipo.
- En Git nunca se borra nada.

## GitHub

- Crear repositorio con Readme.md en GitHub.
  - git remote add origin **url** → añadir espacio remoto.
  - git remote -v → ver si hay espacios remotos.
  - git push origin master. → enviar rama master.
  - git pull origin master → traer rama master.
  - git pull origin master --allow-unrelated-histories.



## ¿Cómo funcionan las llaves públicas y las llaves privadas?

- ssh-keygen -t rsa -b 4096 -C
- eval \$(ssh-agent -s)

Configurar para proyectos delicados.

## Tag y versiones en Git y GitHub.

- Es una buena práctica hacer pull siempre.

- git tag -a **Nombretag id.**  
-d → para borrar.

git push origin :refs/tag/nombre  
↓  
Borrar tag en r. remoto.

- git push origin tags.