GIT

# ***Que es?***

Es un sistema de control de versiones.

Repositorio:

Todo proyecto que está siendo seguido por GIT, que ya tiene un historial de GIT en que se están registrando los cambios.

Commit:

Es cada uno de los cambios registrados en GIT.

Ramas:

Copia exacta del proyecto en un “lugar aislado” (por si algo se rompe, no se compromete el proyecto). La rama Master es la que se presenta al público.

Clon:

Copia exacta del repositorio. Una para cada desarrollador en su equipo local.

Fork:

Es un proyecto distinto pero basado en otro.

GIT es **distribuído**, cada desarrollador tiene una copia en su ordenador (por lo tanto cada quien tiene un backup del proyecto), no hace falta internet para trabajar.

Ramas y fusiones (BRANCH y MERGE):

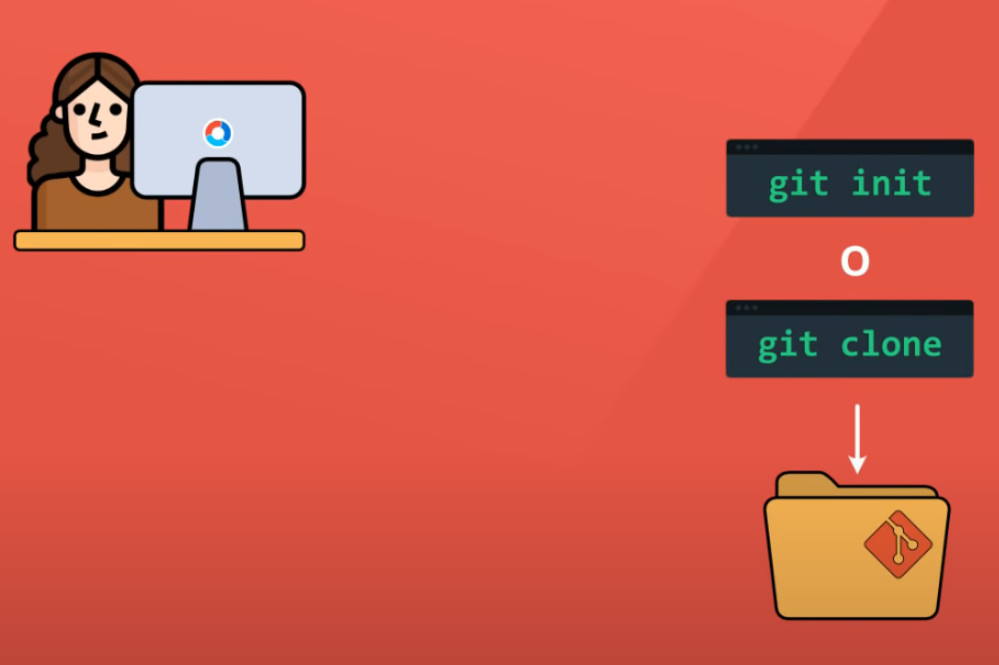
Se hacen para implementar nuevas funciones o para solucionar algún error. Luego se fusiona con la rama master.

Integridad de datos:

GIT agrega una suma de comprobación a cada uno de los archivos y de los commits, para que cada uno de los desarrolladores tenga los mismos datos que los demás (para que no haya datos corruptos).

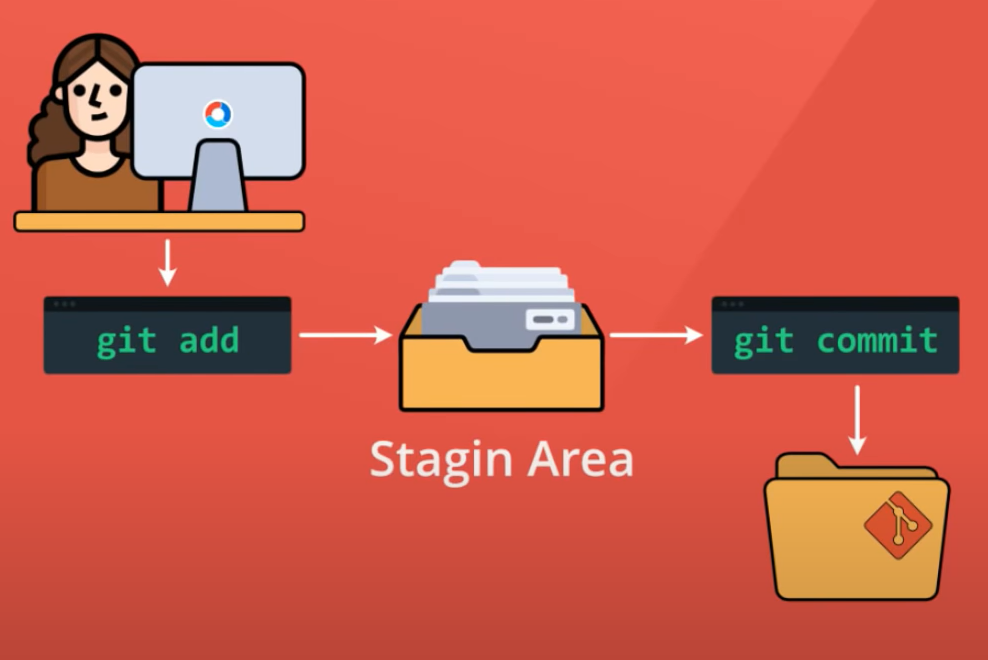
# 

# ***Flujo de trabajo***



Si es un repositorio nuevo se hace con git init

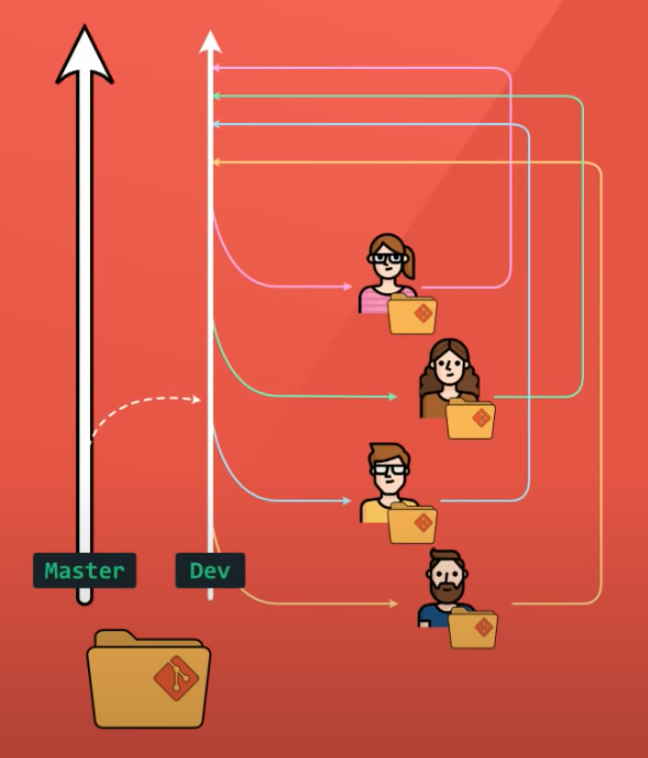
Si hay que clonar un repositorio ya hecho se utiliza git clone



Los commit no se mandan directo al repositorio, se mandan al stagin area primero (la cantidad de commits que sean necesario) para luego cuando se esté seguro de los cambios realizados enviarlos al repositorio con sus respectivos comandos:

* git add: Enviar a stagin area.
* git commit: Enviar al repositorio.

(Los commits se envían al repositorio MANUALMENTE).



Nunca se trabaja con el Branch Master.

Se hace un Branch Dev del cual los desarrolladores sacan sus propias Branches para trabajar en lo que le toque a cada uno.

Luego esas ramas (Branches) deben volver a integrarse (Merge) a la Branch Dev.

Finalmente al comprobar que todo funcione correctamente se integra al Branch Master.

# ***Comandos***

git init : iniciar un repositorio

git add (nombre): para añadir al stagin area

git diff (nombre): saber la diferencia entre el archivo que está en el directorio y el que está en el stagin area.

git diff --staged (nombre): saber la diferencia entre el archivo que está en el stagin area y el que está en el repositorio.

git resert “(nombre)”:para sacarlo del stagin area.

git log: mostrar todos los commits del repositorio.

git commit -m “descripción”: crea el snapshot en el repositorio con la descripción.

git checkout “nombre del commit” : te manda lo que está en el repositorio al ordenador.

git branch: mostrarte la rama en que estás parado

git checkout -b “(nombre)”: crea ramificación desde donde estas parado

git checkout (nombre de la rama): me para en ese commit.

git clone (link): copia en el directorio lo del repositorio online.

git push: mandar del repositorio local al remoto.

git pull: traer del repositorio remoto al local.

### ***Fuente:***

<https://git-scm.com/downloads>

<https://www.youtube.com/watch?v=jGehuhFhtnE>

<https://app.slack.com/client/T025V950XKN/C025L2E3NJH>

<https://developer.ibm.com/technologies/web-development/tutorials/d-learn-workings-git/>

<https://rogerdudler.github.io/git-guide/>

<https://www.youtube.com/watch?v=8YgCTcTdIWI>