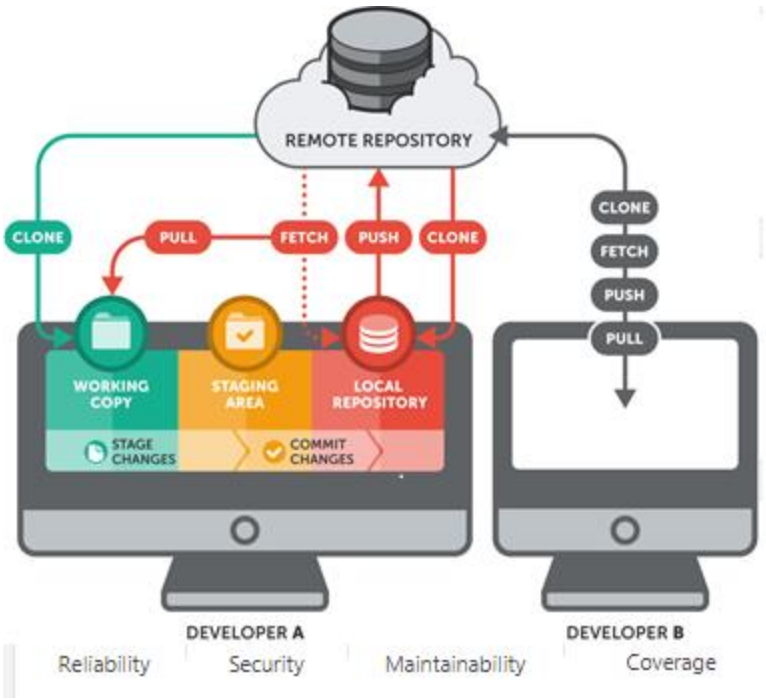
COMANDOS GIT



$ git config

Se usa para configurar GIT por primera vez. Se debe hacer una única vez y se puede actualizar en cualquier momento usando los comandos correctos.
Establecer nombre de usuario y dirección de correo electrónico es importante porque los commits de Git usan esta información y es introducida en los commits que se envían.

- $ git config user.name "Tu nombre"
- $ git config user.email "Tu correo"
- $ git config --list -> Para comprobar las configuraciones.

$ git init

Este comando marca el inicio de nuestro proyecto. Aquí decimos a Git que inicie a monitorear todos los cambios.
Al crear un directorio nuevo se abre una consola y se ejecuta el comando para crear un nuevo repositorio de GIT.

$ git status

Ver el estado de los archivos del proyecto.

$ git add

Se decide que archivos están listos para el siguiente paso y se registran los cambios añadiéndolos al *Staging Area.*

- $ git add <file name> -> añade archivos especificos
- $ git add . ->Añade todos los archivos.

$ git commit -m "Mensaje”

Guardar cambios con un mensaje para identificarlos.
Los cambios quedan en el Local Repository, aún no están en el repositorio remoto.

$ git push origin < nombre_rama >

Envía nuestros cambios (commits) al repositorio remoto.

$ git remote

Para ver los repositorios que hay configurados

$ git remote add origin <url>

Vincula nuestro proyecto local, con nuestro proyecto remoto.

$ git checkout

Con este comando viajamos a través de nuestras ramas o commits.

- $ git checkout <nombre_rama> -> Cambiar de rama.
- $ git checkout <id_commit> ->Permite ver el estado del proyecto en el commimt indicado.

$ git branch

Este comando se usa para listar ramas.

- $ git checkout –b <nombre_rama> ->Crea una rama
- $ git branch –d <nombre_rama> -> Elimina una rama

Una rama no estará disponible para las otras personas a menos que se suba la rama al repositorio remoto

- $ git push origin <nombre_rama>

$ git pull

Actualiza el repositorio local al commit más nuevo.
Actualiza rama actual con los cambios en origin

- Este comando ejecuta un fetch con los argumentos dados, y después realiza un merge en la rama actual con los datos descargados.

$ git reset

Es similar a **checkout** a diferencia que este elimina los commits.

- ### $ git reset –soft

  El **git reset** más simple y que no toca nuestro "**Working Area**" (No se mete con nuestro código).

- ### $ git reset –mixed

  Este **git reset** borra el "**Staging Area**", sin tocar el "**Working Area**".

- ### $ git reset –hard

  Este **git reset** borra absolutamente todo lo que hay en el commit.

$ git fetch

Actualiza las referencias remotas. Localiza en que servidor está el origen Recupera cualquier dato presente en el servidor que no se tenga localmente y actualiza la base de datos local moviendo la rama origin/master para que apunte a la nueva y más reciente posición.

$ git clone <url>

Crea una copia local del repositorio.

$ git pull <nombre_del_remote>

Si la rama actual es una tracking branch: El comando git pull [nombre_del_remote], actualiza la rama actual con los cambios realizados en la rama asociada del repositorio remoto.

$ git tag -a v1.0 -m 'Mensaje' 612d406

Los tags (etiquetas) se crean para cada nueva versión publicada de un software. 612d406 se refiere a los caracteres del commit id al cual se quiere referir la etiqueta. Se puede usar menos caracteres que el commit id, pero debe ser un numero único.

$ git log

Lista todos los commits con su respectiva información. De este comando se puede obtener el commit id para la creación del tag.

$ git help

Este comando nos ayuda a saber cómo funciona git o alguno de sus comandos.