

Comandos básicos de git

git config --global -e -> muestra la configuración del git, como nombre de autor correo, la rama y mas

Configuración inicial recomendada, esto para cuando se trabaje de forma colaborativa, se pueda determinar quien hizo los cambios

git config --global user.name "Juan Alvarenga"

git config --global user.email "juanenrique94@gmail.com"

git branch -m master main -> permite cambiar el nombre de la rama principal (solo para la configuración actual)

git config --global init.defaultBranch main -> establece de forma global la rama principal, como main

git init -> Inicia un repositorio en un proyecto

git status -> muestra todos los archivos del repositorio con información de los archivos que ya están en el stage o que han sufrido modificaciones o los nuevos archivos que no han sido agregados

git add {nombre_archivo} -> agrega el archivo seleccionado a la lista de seguimiento para el próximo commit

git reset {nombre_archivo} -> quita del stage un archivo que no queremos darle seguimiento

git add . -> Agrega todos los archivos y directorios a la lista de seguimiento para el próximo commit

git commit -m "Mensaje a colocar" -> realiza un snapshot del repositorio para poder guardar los cambios y devolverse a ellos si es necesario

git commit -am "Readme actualizado" -> hace el commit de un archivo y lo agrega al stage, pero, solo si previamente se le hace seguimiento

git checkout -- . -> revierte todos los cambios al ultimo commit, incluyendo recuperación de directorios y archivos, solo de aquellos a los que se les hace seguimiento.

git log -> muestra el historial de commits realizados

git reflog -> muestra el historial de commits a lo largo de toda la historia (incluyendo destrucciones)

git config --global alias.{nombre_alias} "{comando a ejecutar}" -> crea un alias para ejecutar un comando de forma mas fácil

——— alias utiles ———

Log:

```
git config --global alias.lg "log --graph --abbrev-commit --decorate --
format=format:'%C(bold blue)%h%C(reset) - %C(bold green)(%ar)%C(reset)
%C(white)%s%C(reset) %C(dim white)- %an%C(reset)%C(bold yellow)%d%C(reset)' -
-all"
```

Status

```
git config --global alias.s status -sb
```

——— fin alias utiles ———

git commit --amend -> me permite modificar el mensaje de un commit

Comandos para restaurar puntos anteriores en el repositorio

git reset --soft HEAD^ -> Permite regresar de forma "segura" el HEAD, sin modificar el índice, esto deja todos los archivos modificados a "cambios por confirmar" (en el stage), preparado para hacer un commit.

git reset --mixed {hash} -> Restablece el índice (stage) pero no el árbol de trabajo (es decir, los archivos modificados se conservan pero no se marcan para confirmación) e informa lo que no se ha modificado (se debe agregar todos los archivos para agregar en el próximo commit).

git reset --hard {hash} -> Restablece el índice y el árbol de trabajo, todo cambio en los archivos que fueron agregados posteriormente al hash al que se hace referencia se descarta, cualquier archivo o directorio sin seguimiento se elimina.

Creando Ramas

git branch -> muestra todas las ramas que hay creadas

git branch {nombre_rama} -> crea una nueva rama, sustituyendo "{nombre_rama}" por el nombre que queramos asignar.

git branch -d {nombre_rama} -> Elimina una rama

git checkout {nombre_rama} -> hace el cambio a la rama seleccionada

git checkout -b {nombre_rama} -> crea una rama y hace el cambio a la misma en una sola línea

git merge {nombre_rama} -> hace la unión con la rama a la que se hace referencia

Creación de tags

git tag {nombre_tag} -> Crea un tag para asociarlo al último commit (HEAD)

git tag -a {nombre_tag} {hash} -m "{una descripcióndel tag}" -> Crea un tag para asociarlo a un hash específico siendo una versión "anotada", esta permite visualizar más información.

git show {nombre_tag} -> Visualiza información completa de un tag anotado

git config --global pull.ff only -> Permite al hacer un clone de un repositorio guardar solo cuando no hay conflictos en los repositorios (sería lo ideal)

git config --global pull.rebase true -> Permite al hacer un clone de un repositorio en github, poder resolver los conflictos de forma manual.