

Guía de comandos Git

CONFIGURACIÓN INICIAL

- **\$ git config --global user.name "user"** # Asigna nombre del usuario de git.
- **\$ git config --global user.email "user@email.com"** # Asignando el email del usuario git (usando las comillas).
- **\$ git config --list** # visualiza datos como el usuario y el email.
- **\$ git config --global http.sslVerify false** # Utilizado para resolver problemas de certificado ssl no válidos.
- **\$ git init** # Inicia un nuevo repositorio en la carpeta actual.
- **\$ git clone <URL>** # Descarga un repositorio de GitLab, GitHub o bitbucket(sin < >)

AÑADIR ARCHIVOS AL INDEX

- **\$ git add .** # Añade todos los archivos al commit.
- **\$ git add <archive>** # Añade un solo archivo especificado.
- **\$ git add --all** # Añade todos los archivos para el commit omitiendo los nuevos.
- **\$ git add *.txt** # Añade los archivos con la extensión específica.

AÑADIR ARCHIVOS AL HEAD

- **\$ git commit -m "Texto que identifique porque se hizo el commit"**
Carga al HEAD los cambios realizados.
- **\$ git commit -a -m "mensaje"** # Automáticamente pone en *index* los archivos modificados sin necesidad de usar el comando *git add*.
- **\$ git commit --amend -m "Nuevo mensaje del commit"** # Sobrescribe el último commit con un nuevo mensaje.

BRANCH

- **\$ git branch <nombre-rama>** # Únicamente crea rama.
- **\$ git checkout <nombre_rama>** # Cambiar a la rama indicada en *nombre_rama*.
- **\$ git checkout -b <nombre_rama>** # Crea y cambia a la rama creada.
- **\$ git merge <rama>** # Mezcla la rama actual con <rama>
- **\$ git branch -d <rama>** # Elimina la *<rama>*.
- **\$ git branch** # lista las ramas y muestra la actual.
- **\$ git branch -v** # lista ramas mostrando su respectivo último commit.



PUSH

- **\$ git push -u origin <branch>** # Sube al repositorio, en el branch especificado.
- **\$ git push -tags** # Sube al repositorio los cambios con un tag.

PULL

- **\$ git pull origin <nameBranch>** # Te actualiza el repositorio local.

DESCARTAR CAMBIOS Y MOVER

- **\$ git rm <file>** # Borrar el archivo con Git para siempre.
- **\$ git rm -f <file>** # Para cuando el archivo ya está modificado y en el index.
- **\$ git mv <file> <renamed_file>** # Renombra el archivo.
- **\$ git checkout -- <file>** # Descartar los cambios en el directorio de trabajo.

GIT HEAD

- **\$ git reset HEAD <file>** # Remueve el archivo del commit.
- **\$ git reset --hard HEAD^** # Devuelve el ultimo commit y todos los cambios.
- **\$ git reset --hard HEAD** # Limpia todo el hilo de trabajo contenido en HEAD.

REPOSITORIOS REMOTOS

- **git remote add origin <URL>** # Sube un repositorio local a uno remoto (URL).

OTROS Y UTILIDADES

- **\$ git status** # Lista un estado actual del repositorio con lista de archivos modificados o agregados.
- **\$ gitk** # herramienta grafica para git.
- **\$ git fetch** # Verifica cambios en el repositorio online con el local.
- **\$ git log** # muestra los últimos commits realizados.
- **\$ git log -stat** # Commits realizados con más detalle.

