



COMANDOS DE GIT

COMANDO	NOMBRE	DESCRIPCIÓN
CREAR REPOSITORIOS: Inicia un nuevo repositorio u obtiene uno de una URL existente		
git init	[project-name]	Buscar en todas partes del proyecto
git clone	[url]	Descarga un proyecto y toda su historia de versión (clono un proyecto en mi área local).
git ignore	[Dentro nombre archivos]	Es un archivo de texto que le dice a Git qué archivos o carpetas ignorar en un proyecto.
EFECTUAR CAMBIOS: Revisa las ediciones y elabora una transacción de commit		
git commit -m	"[descriptive message]"	Registra las instantáneas del archivo permanentemente en el historial de versión (repositorio local).
git status -s		Enumera todos los archivos nuevos o modificados que se deben confirmar (los que estan en el directorio).
git commit -am	"[descriptive message]"	Para realizar un add y un commit simultáneamente y en la misma línea de código.
git commit -amend		Agrega los cambios realizados al commit anterior (remendar commit).
git tag	[version] -m "message"	Agregamos la versión al local que llevamos del programa.
git add	[file]	Toma una instantánea del archivo para preparar la versión (de uno en uno según agreguemos el archivo).
git add .		Agrega todos los archivos que esten en el área de trabajo.
git status		Enumera todos los archivos nuevos o modificados que se deben confirmar.
git diff	[id id]	Muestra las diferencias de archivos que no se han enviado aún al área de espera
git diff --staged		Muestra las diferencias del archivo entre el área de espera y la última versión del archivo
git shortlog		Permite visualizar cuantos commits a realizado cada miembro del equipo.
git shortlog -sn		Permite visualizar la cantidad de commits que han realizado los miembros del equipo.
git shortlog -sn --all --no-merge		Los mismo que el git shortlog -sn pero incluyendo commits borrados y excluyendo los merge.
git shortlog		Permite visualizar cuantos commits a realizado cada miembro del equipo.
REPASAR HISTORIAL: Navega e inspecciona la evolución de los archivos de proyecto		
git log		Enumera el historial de la versión para la rama actual.
git log -s "<word>"		Permite buscar a través de cualquier commit por una cadena de texto.
git log --oneline		Enumera el historial de versión para el archivo, incluidos los cambios de nombre (commits realizados).
git log --follow	[file]	Enumera el historial de versión para el archivo, incluidos los cambios de nombre
git diff	[first-branch]...[second-branch]	Muestra las diferencias de contenido entre dos ramas
git show	[commit]	Produce metadatos y cambios de contenido del commit especificado
git grep <word>	[word]	Permite buscar a través de cualquier arbol o directorio de trabajo con commit por una cadena de texto.
git grep -c <word>	[word]	Busca en arbol o directorio por cadena de texto y muestra la cantidad de veces que se repite dicha palabra.
git grep -n <word>	[word]	Busca en arbol o directorio por cadena de texto y muestra la linea de código donde se referencia.
git blame <file name>	[name file]	Permite ver los cambios que hizo algun miembro del equipo en un archivo especifico.
git blame -c <file name>	[name file]	Funciona igual que git blame <name file> pero con un mejor formato de salida.
git blame <file name> -<#line, #line> -c	[file name] -[#line, #line]	Funciona igual que git blame <name file> pero desde un rango de lineas de código.
git log --oneline --graph --decorate		Vista gráfica del historial de commits, mostrando la estructura de la rama actual.
git log --oneline --graph --pretty=format:"%h %an <%ae> %s"		Muestra el historial de commits graficamente con el nombre y correo electrónico del autor.
git log --graph --oneline --all --decorate		Muestra el historial grafico de commits de todas las ramas (locales y remotas) en el repositorio.
git log --author="Saulolo" --oneline --graph --decorate	"\$(git config user.name)"	Vista gráfica del historial de commits realizados por un usuario en especifico.
SINCRONIZAR CAMBIOS: Registrar un marcador de repositorio e intercambiar historial de versión		
git push origin	[branch]	Carga todos los commits de la rama local al GitHub (sube al Git Hub los cambios realizados de local).
git tag [tag name]	[tag name]	Crea una nueva etiqueta en el commit actual.
git fetch	[branch]	Descarga ramas de otros repositorios.
git push --tags		Empuja todas las etiquetas locales al repositorio remoto. E
git pull origin	[branch]	Descarga el historial del marcador e incorpora cambios (descargo al local los cambios hechos en el Git Hub).
git pull origin	[branch] -- allow-unrelated-histories	Forzar a realizar la sincronización si no hay conflicto de archivos.

git remote add origin	[url repository Git Hub]	Permite agregar conexiones con otros repositorios en mi local.
git remote -v		Muestra las url que tengo conectadas con mi Git local.
git remote remove <name>	[name]	Elimina un remoto del repositorio local. Se usa para eliminar un remoto que ya no es necesario o ha cambiado.
git remote add <name> <url>	[name] [url]	Agrega un nuevo remoto con un nombre y una URL especificados
CAMBIOS GRUPALES: Nombra una serie de commits y combina esfuerzos ya culminados		
git branch <branch-name>	[branch-name]	Crea una nueva rama.
git branch -r		Muestra todas las ramas que estan en remoto.
git branch -a		Muestra todas las ramas que estan en remoto y en local.
git branch		Enumera todas las ramas en el repositorio actual y me muestra la rama en la cual me encuentro.
git checkout	[branch-name]	Cambia a la rama especificada y actualiza el directorio activo (muevo mi flujo de trabajo a la rama creada).
git chery-pick <hash commit>	[has-commit]	Elige una confirmación arbitraria de una rama y la aplica en otra por su referencia de commits.
git branch -m	[rama actual - rama renombrada]	Renombrar una rama por otra, ej: git branch -m master main (renombra la rama de master a main).
git checkout -b	[branch-name]	Crea una nueva rama e inmediatamente se posiciona en ella
git branch -d	[branch-name]	Borra la rama especificada.
git rebase	[branch-name]	Recopila uno a uno los cambios confirmados en una rama, y reaplicarlos sobre otra
git merge	[branch]	Combina el historial de la rama especificada con la rama actual (se debe estar ubicado en la rama main).
git mergetool		Abre una herramienta grafica para resolver los conflictos.
CONFIGURAR HERRAMIENTAS: Configura la información del usuario para todos los respositorios locales		
git config		Permite obtener y establecer variables de configuración que controlan el aspecto y funcionamiento de Git
git config --global user.name	[name]	Establece / muestra el nombre que desea esté anexado a sus transacciones de commit.
git config --global user.email	[email address]	Establece / muestra el e-mail que desea esté anexado a sus transacciones de commit.
git help <comando>	[command name]	Ayuda en git para buscar la funcionalidad de un comando.
git <comando> --help	[command name]	Ayuda en git para buscar la funcionalidad de un comando.
git config --list		Muestra en lista la configuración que tenemos en Git.
git config --global color.ui auto		
NOMBRES DEL ARCHIVO DE REFACTORIZACIÓN: Reubica y retira los archivos con versión		
git rm	[file]	Borra el archivo del directorio activo y pone en el área de espera el archivo borrado
git rm --cached	[file]	Retira el archivo del control de versiones, pero preserva el archivo a nivel local
git mv	[file-original] [file-renamed]	Cambia el nombre del archivo y lo prepara para commit
REHACER COMMITS: Borra errores y elabora historial de reemplazo		
git reset	[commit]	Deshace todos los commits después de [commit], preservando los cambios localmente.
git reset <hash commit>	[commit o hash]	Borra las modificaciones echas antes de commitear y regresa al commit especificado de manera suave.
git reset --hard <hash commit>	[hash commit]	Borra las modificaciones echas antes de commitear y regresa al commit especificado.
git reset --hard		Borra las modificaciones echas antes de commitear.
git clean --dry-run		Elimina archivos que se crearon por error en la raíz del proyecto y que se pueden indexar.
git reflog		Muestra un historial de todas las actualizaciones del puntero HEAD, permitiendo recuperar commits "perdidos".
GUARDAR FRAGMENTOS: Almacena y restaura cambios incompletos		
git stash save		Guarda cambios actuales en un stash temporal.
git stash pop		Aplica cambios de un stash y elimina el stash.
git stash list		Muestra la lista de stashes guardados.
git stash drop		Elimina un stash específico.
git stash apply		Aplica cambios de un stash al área de trabajo sin eliminar el stash.
git stash branch <branch name>	[branch-name]	Crea una rama nueva y aplica los cambios de un stash en ella.
COMANDOS CLÁSICOS DE CONSOLA		
pwd		Muestra la ruta (carpetas) en la que nos encontramos.
clear		Limpiar la consola.