

# CheatSheet de Git y GitHub

## Configurar el entorno de trabajo

\$ git config user.name "Nombre de Usuario en GitHub u otra plataforma"	Determina el nombre del autor con el cual se identificarán los commits que se realicen en la actual sesión en la respectiva terminal que se este utilizando.
\$ git config --global user.name "Nombre de Usuario en GitHub u otra plataforma"	Al ser agregado la "palabra o flag --global", se modifica el comando para que se determine el nombre del autor con el cual se identificarán los commits que se realicen en cualquier sesión en la respectiva terminal que se este utilizando.
\$ git config user.email "Email que corresponde al Usuario en GitHub u otra plataforma"	Asocia el email del autor con el cual se identificarán los commits que se realicen en la actual sesión en la respectiva terminal que se este utilizando.
\$ git config --global user.email "Email que corresponde al Usuario en GitHub u otra plataforma"	Al ser agregado la "palabra o flag --global", se modifica el comando para que se asocie el email del autor identificado en los commits que se realicen en cualquier sesión en la respectiva terminal que se este utilizando.

## Creación del Repositorio

\$ git init "Nombre a asignar a la carpeta .git si es que no se desea el nombre por defecto .git"	Crea un nuevo repositorio local dentro de la carpeta en que la terminal este actualmente interactuando.
\$ git clone "url"	<p>Clona(es decir, replica) en el directorio local de trabajo, el repositorio remoto indicado en el comando por la url.</p> <p>Es la forma de escribir el comando cuando se utiliza una dirección web https y suelen estar escritas del siguiente modo:</p> <p><a href="https://github.com/usuario_en_Github/nombre-del-repositorio-remoto.git">https://github.com/usuario_en_Github/nombre-del-repositorio-remoto.git</a></p> <p>El repositorio local queda vinculado al repositorio remoto clonado.</p>

## Realizar Cambios

Revisar versiones y ajustar los commits

\$ git add "nombre_del_archivo.extensión"	Envia al archivo al staging area(sala de espera), lo que significa que procede a esperar a ser enviado a un commit(o versión empaquetada del proyecto en cuestión).
\$ git status	Comando para informarse sobre que archivos del directorio de trabajo no están agregados a la actual versión de trabajo(dicho de otro modo, no estan en el staging area, por tanto no están en la fila esperando a ser agregados a la versión empaquetada o commit).
\$ git commit -m "mensaje descriptivo"	Comando para utilizando los archivos encontrados en la sala de espera(staging area), crear un commit(o empaquetamiento de la actual versión de trabajo en curso en la línea de trabajo{o línea de tiempo de trabajo o historial de trabajo}). El mensaje descriptivo entrecomillas es una frase necesaria para ayudar a identificar ante los ojos de cualquiera que revise el historial del proyecto. Aludiendo al contenido y función en forma individual de cada commit realizado.

## Navegando el Historial del Proyecto

\$ git log	Comando para informarse del historial de trabajo de commits(o versiones empaquetadas o versiones establecidas) relativo a la actual rama de trabajo en el proyecto.
\$ git status	Comando para informarse sobre que archivos del directorio de trabajo no están agregados a la actual versión de trabajo(dicho de otro modo, no estan en el staging area, por tanto no están en la fila esperando a ser agregados a la versión empaquetada o commit). Y para que conocer en que rama se esta trabajando en la actualidad.

## Rehacer Commits

\$ git reset "Primeros términos del nombre alfanumerico asignado al commit"	D
\$ git reset --hard "Primeros términos del nombre alfanumerico asignado al commit"	A

## Lineas de trabajo Paralelas y Alternativas

Las líneas de trabajo paralelas o historias de trabajo paralelas o ramas inclusive. Son un conjunto de commits con una trayectoria propia e independiente en el proyecto(es decir, tampoco son criticas para la concreción del proyecto) y distinta de la trayectoria principal. La existencia de las ramas sirve para experimentar cambios y a la vez protegerse de errores catastróficos, gracias a que los cambios no se están realizando en la trayectoria principal(o rama principal). Si un cambio en alguna rama paralela lleva a buen puerto, pasa a ser una alternativa viable para concretar el proyecto y por tanto, tal rama alternativa( es alternativa al ser viable) puede ser fusionada con la rama principal para contribuir en avanzar hacia la culminación del proyecto.

**Importante:** La rama principal o “main”, solo comienza a existir(se establece automáticamente) luego de haber creado el primer commit con el comando “\$ git commit -m “mensaje descriptivo”. Por lo cuál, tampoco se pueden crear otras ramas distintas a main, sin antes de haber creado la rama “main” con el primer commit. Es decir, solo luego del primer commit, es útil el comando \$ git branch “nombre\_asignado\_a\_la\_rama”.

\$ git branch	Comando para informarse tanto de todas las ramas existentes en el proyecto, como también de en cuál rama se esta trabajando actualmente. Esto gracias a que la rama de trabajo aparece señalizada y resaltada.
\$ git branch “nombre_asignado_a_la_rama”	Comando para crear una nueva rama del proyecto.
\$ git checkout “nombre_Rama_a_la_que_se_quiere_ir”	Comando para cambiar(viajar) desde la actual rama de trabajo a la rama mencionada en el comando.
\$ git branch -d “nombre_asignado_a_la_rama”	Comando para borrar la Rama especificada por nombre en el mismo comando.
\$ git branch -df “nombre_asignado_a_la_rama”	Comando para borrar de manera forzada, la Rama especificada por nombre en el mismo comando.
\$ git checkout -b “nombre_Nva_Rama_a_la_que_se_quiere_ir”	Comando para crear y cambiar(viajar) desde la actual rama de trabajo a la nueva rama creada, mencionada en el comando.
\$ git merge “nombre_asignado_a_la_rama_B”	Comando para combinar la rama mencionada en el comando, con la actual rama de trabajo.
\$ git diff “nombre_RamaA” .. “nombre_RamaB”	
\$ git branch --merged	

## Actualizar y Sincronizar Repo local con Repo en la Nube

\$ git remote add origin "url"	Comando para vincular nuestro repositorio local(creado previamente) a un repositorio remoto origin alojado en GitHub, cuya dirección de alojamiento es del tipo <a href="https://github.com/usuario_en_Github/nombre-del-repositorio-remoto.git">"https://github.com/usuario en Github/nombre-del-repositorio-remoto.git"</a> .
\$ git remote -v	Comando que se utiliza para confirmar la correcta vinculación de un repositorio local con algún repositorio remoto en GitHub.
\$ git push	Subir el último estado del proyecto(definido por el último commit hecho con el comando <b>\$ git commit -m "mensaje descriptivo"</b> ) a la última Rama indicada(o señalizada). O dicho de otro modo: Comando para subir(agregar) los últimos cambios hechos en la actual Rama Local de trabajo(la última señalizada para trabajar localmente) del repositorio local. Al repositorio remoto(identificado como "origin" en la terminal) y que no existan en la <b>Rama remota principal</b> de trabajo en el repo remoto.(la indicada como principal en la configuración de GitHub para trabajar remotamente)
\$ git push origin "main"	Subir el último estado del proyecto(definido por el último commit hecho con el comando <b>\$ git commit -m "mensaje descriptivo"</b> ), pero indicando que actualmente y en el futuro(si solo se hace <b>\$ git push</b> ) se subirá a la Rama llamada <b>"main"</b> . O dicho de otro modo: Comando para subir(agregar) los últimos cambios hechos en la actual Rama Local de trabajo(la última señalizada para trabajar localmente) del repositorio local. Al repositorio remoto(identificado como "origin" en la terminal) y que no existan en la <b>Rama remota "main"</b> en el repo remoto. Para alojarlos en la <b>Rama "main"</b> .
\$ git push origin "Rama n"	Subir el último estado del proyecto(definido por el último commit hecho con el comando <b>\$ git commit -m "mensaje descriptivo"</b> ), pero indicando que

	<p>actualmente y en el futuro(si solo se hace \$ git push ) se subirá a la Rama llamada “n”.</p> <p>O dicho de otro modo:</p> <p>Comando para subir(agregar) los últimos cambios hechos en la actual Rama Local de trabajo(la última señalizada para trabajar localmente) del repositorio local. Al repositorio remoto(identificado como “origin” en la terminal) y que no existan en la <b>Rama remota “n”</b> en el repo remoto, para alojarlos en la <b>Rama “n”</b>.</p>
\$ git pull	Comando para descargar los últimos cambios hechos al repositorio remoto”origin”, por tanto que hayan sido agregados a la actual Rama principal del repo remoto y que no estén en la actual Rama Local de trabajo(la última señalizada para trabajar localmente)
\$ git pull origin “main”	Comando para descargar los cambios hechos al repositorio remoto”origin”, que hayan sido agregados a la Rama “ <b>main</b> ” del repo remoto y que no estén en la actual Rama Local de trabajo(la última señalizada para trabajar localmente).
\$ git pull origin “Rama n”	Comando para descargar los cambios hechos al repositorio remoto”origin”, que hayan sido agregados a la Rama “ <b>n</b> ” del repo remoto y que no estén en la actual Rama Local de trabajo(la última señalizada para trabajar localmente).