



# HOJA DE AYUDA GIT

Git es el sistema de control de versiones distribuido gratuito y de código abierto que es responsable de todo lo relacionado con GitHub que sucede localmente en su computadora. Esta hoja de trucos presenta los comandos Git más importantes y más utilizados para una fácil referencia.

## INSTALACIÓN Y GUIS Con

instaladores específicos de la plataforma para Git, GitHub también brinda la facilidad de mantenerse actualizado con las últimas versiones de la herramienta de línea de comandos al tiempo que proporciona una interfaz gráfica de usuario para la interacción, la revisión y la sincronización del repositorio del día a día. .

**GitHub para Windows** [https://](https://windows.github.com)

[windows.github.com](https://windows.github.com)

**GitHub para Mac**

<https://mac.github.com>

Para las plataformas Linux y Solaris, la última versión está disponible en el sitio web oficial de Git.

**Git para todas las plataformas**

<http://git-scm.com>

## CONFIGURACIÓN

Configuración de la información del usuario utilizada en todos los repositorios locales

**git config --usuario global.nombre "[nombre apellido]"**

establezca un nombre que sea identificable para el crédito cuando revise el historial de versiones

**git config --usuario global.email "[correo electrónico válido]"**

establecer una dirección de correo electrónico que se asociará con cada marcador de historial

**git config --color global.ui automático**

establezca el coloreado automático de la línea de comando para Git para facilitar la revisión

## CONFIGURACIÓN E INICIO

Configuración de información de usuario, inicialización y clonación de repositorios

**iniciar git**

inicializar un directorio existente como un repositorio de Git

**clon de git [url]**

recuperar un repositorio completo desde una ubicación alojada a través de URL

## ETAPA E INSTANTÁNEA

Trabajar con instantáneas y el área de preparación de Git

**estado de Git**

mostrar archivos modificados en el directorio de trabajo, preparado para su próxima confirmación

**git agregar [archivo]**

agregue un archivo como se ve ahora para su próxima confirmación (etapa)

**git restablecer [archivo]**

eliminar un archivo conservando los cambios en el directorio de trabajo

**diferencia de git**

diff de lo que se cambia pero no se escenifica

**git diff --por etapas**

diff de lo que está organizado pero aún no se ha comprometido

**git commit -m "[mensaje descriptivo]"**

confirme su contenido en etapas como una nueva instantánea de confirmación

## SUCURSAL Y FUSIÓN

Aislar el trabajo en las sucursales, cambiar el contexto e integrar los cambios

**rama git**

enumere sus sucursales. aparecerá un \* junto a la rama actualmente activa

**rama git [nombre-rama]**

crear una nueva rama en la confirmación actual

**pago git**

cambie a otra sucursal y compruébelo en su directorio de trabajo

**git merge [rama]**

fusionar el historial de la rama especificada con el actual

**registro de git**

mostrar todas las confirmaciones en el historial de la rama actual



INSPECCIONAR Y COMPARAR

Examen de registros, diferencias e información de objetos

<b>registro de git</b>
mostrar el historial de confirmaciones para la rama actualmente activa
<b>git registro ramaB..ramaA</b>
mostrar las confirmaciones en branchA que no están en branchB
<b>git log --sigue [archivo]</b>
mostrar las confirmaciones que cambiaron el archivo, incluso a través de cambios de nombre
<b>git diff ramaB...ramaA</b>
mostrar la diferencia de lo que está en branchA que no está en branchB
<b>espectáculo git [SHA]</b>
mostrar cualquier objeto en Git en formato legible por humanos

SEGUIMIENTO DE CAMBIOS DE RUTA

Eliminación de versiones de archivos y cambios de ruta

<b>git rm [archivo]</b>
elimine el archivo del proyecto y organice la eliminación para confirmar
<b>git mv [ruta-existente] [nueva-ruta]</b>
cambiar una ruta de archivo existente y organizar el movimiento
<b>registro de git --stat -M</b>
mostrar todos los registros de confirmación con indicación de las rutas que se movieron

PATRONES IGNORANTES

Evitar la puesta en escena o la confirmación no intencionada de archivos

<b>registros/ *.patrón de notas*</b>
Guarde un archivo con los patrones deseados como .gitignore con coincidencias directas de cadenas o comodines globales.
<b>git config --global core.excludesfile [archivo]</b>
todo el sistema ignora el patrón para todos los repositorios locales

COMPARTE Y ACTUALIZA

Recuperar actualizaciones de otro repositorio y actualizar repositorios locales

<b>git remoto agregar [alias] [url]</b>
agregar una URL de git como alias
<b>git buscar [alias]</b>
baje todas las ramas de ese control remoto de Git
<b>git merge [alias]/[rama]</b>
fusionar una sucursal remota en su sucursal actual para actualizarla
<b>git push [alias] [rama]</b>
Transmitir compromisos de la rama local a la rama del repositorio remoto
<b>tirar de git</b>
obtener y fusionar cualquier confirmación de la rama remota de seguimiento

REESCRIBIR LA HISTORIA

Reescritura de ramas, actualización de confirmaciones y limpieza del historial

<b>git rebase [rama]</b>
aplicar cualquier confirmación de la rama actual antes de la especificada
<b>git reset --hard [confirmar]</b>
borrar el área de preparación, reescribir el árbol de trabajo desde la confirmación especificada

COMISIONES TEMPORALES

Almacene temporalmente archivos modificados y rastreados para cambiar de rama

<b>alijo de git</b>
Guardar cambios modificados y preparados
<b>lista de alijo de git</b>
enumerar el orden de pila de los cambios de archivos ocultos
<b>git escondite pop</b>
escribir trabajando desde la parte superior de la pila de almacenamiento
<b>caída de alijo de git</b>
descartar los cambios desde la parte superior de la pila de almacenamiento