

# Git Cheat Sheet

## Básico

<code>git init &lt;carpeta&gt;</code>	Crea un repositorio vacío en <code>&lt;carpeta&gt;</code> . Sin argumentos permite inicializar la carpeta actual como un repositorio.
<code>git clone &lt;repositorio&gt;</code>	Clona un <code>&lt;repositorio&gt;</code> en nuestra máquina local. El <code>&lt;repositorio&gt;</code> puede ser local o remoto mediante <code>HTTP</code> or <code>SSH</code> .
<code>git config user.name &lt;autor&gt;</code>	Define al <code>&lt;autor&gt;</code> para todas las confirmaciones en el repositorio actual.
<code>git add &lt;carpeta archivo&gt;</code>	Agrega los cambios realizados en dicha <code>&lt;carpeta/archivo&gt;</code> para la siguiente confirmación. Acepta expresiones regulares.
<code>git commit -m "&lt;mensaje&gt;"</code>	Confirma los cambios en esa instancia, pero sin abrir o usar un editor de texto, usa el <code>&lt;mensaje&gt;</code> como mensaje de confirmación.
<code>git status</code>	Muestra la lista de archivos de los cambios montados, no montados y sin seguimiento.
<code>git log</code>	Muestra el historial de confirmaciones (formato por defecto). Para personalizar ver la sección de opciones adicionales.
<code>git diff</code>	Muestra los cambios no marcados entre el índice y la carpeta de trabajo.

## Deshaciendo cambios

<code>git revert &lt;confirmacion&gt;</code>	Crea una confirmación que deshace todos los cambios realizado en la <code>&lt;confirmacion&gt;</code> y lo aplica a la rama actual.
<code>git reset &lt;archivo&gt;</code>	Elimina el <code>&lt;archivo&gt;</code> del área de montaje, dejando el directorio de trabajo sin cambios por montar, sin sobreescribir ningún cambio.
<code>git clean -n</code>	Muestra los archivos que van a ser eliminados de la carpeta de trabajo. Para eliminar dichos archivos se usa <code>-f</code> en vez de <code>-n</code> .

## git config

<code>git config --global user.name &lt;autor&gt;</code>	Define el nombre del <code>&lt;autor&gt;</code> de todas las confirmaciones para el usuario actual.
<code>git config --global user.email &lt;correo&gt;</code>	Define el <code>&lt;correo&gt;</code> del autor de todas las confirmaciones para el usuario actual.
<code>git config --global alias.&lt;alias&gt; &lt;comando&gt;</code>	Crea un atajo de <code>&lt;comando&gt;</code> en Git que se usará mediante el <code>&lt;alias&gt;</code> . Ejm. alias.glog log --graph equivalente. glog -> git log --graph.
<code>git config --system core.editor &lt;editor&gt;</code>	Define el <code>&lt;editor&gt;</code> usado por los comandos para todos los usuarios. Por ejemplo <code>&lt;editor&gt;</code> puede ser vim.
<code>git config --global --edit</code>	Abre el archivo de configuración global para que pueda ser editado manualmente. Usa el editor por defecto de la máquina. Ejm. nano.

## git log

<code>git log --&lt;limite&gt;</code>	Límite el número de logs usando el <code>&lt;limite&gt;</code> . Ejemplo: git log -5 muestra los últimos 5 logs.
<code>git log --online</code>	Condensa cada una de las confirmaciones a una sola línea.
<code>git log -p</code>	Muestra todos los cambios de cada confirmación.
<code>git log --stat</code>	Incluye que archivos se modificaron y el número relativo de líneas que fueron añadidas o borradas de cada uno de ellos.
<code>git log --author="<autor>"</autor></code>	Busca las confirmaciones por un <code>&lt;autor&gt;</code> en particular.
<code>git log --grep="<patron>"</patron></code>	Busca confirmaciones cuyo mensaje de confirmación coincide con el <code>&lt;patron&gt;</code> asignado.
<code>git log --since..&lt;hasta&gt;</code>	Muestra las confirmaciones entre <code>&lt;desde&gt;</code> y <code>&lt;hasta&gt;</code> . Se puede usar hash de confirmación, rama, <code>HEAD</code> o cualquier punto de revisión.
<code>git log --&lt;archivo&gt;</code>	Muestra solo las confirmaciones del <code>&lt;archivo&gt;</code> especificado.
<code>git log --graph --decorate</code>	El <code>--graph</code> muestra un gráfico en texto al lado izquierdo de los mensajes de confirmación y <code>--decorate</code> el nombre de la rama o tag.

## Reescribiendo el historial

<code>git commit --ammend</code>	Reemplaza la última confirmación y/o cambios montados. Usarlo sin cambios montados edita el mensaje de la ultima confirmación.
<code>git rebase &lt;base&gt;</code>	Elimina el <code>&lt;archivo&gt;</code> del área de montaje, dejando el directorio de trabajo sin cambios sin sobreescribir ningún cambio.
<code>git reflog</code>	Muestra un log de los cambios en el <code>HEAD</code> del repositorio local. Agrega <code>--relative-date</code> para mostrar la fecha o <code>--all</code> para todos.

## Ramas

<code>git branch &lt;rama&gt;</code>	Crea una nueva <code>&lt;rama&gt;</code> . Sin argumentos muestra la lista de ramas del repositorio.
<code>git checkout -b &lt;rama&gt;</code>	Crea una nueva <code>&lt;rama&gt;</code> en base a la rama actual. Si quitamos <code>-b</code> permite posicionarse en la <code>&lt;rama&gt;</code> .
<code>git merge &lt;rama&gt;</code>	Mezcla la <code>&lt;rama&gt;</code> con la rama actual.

## Repositorios remotos

<code>git remote add &lt;nombre&gt; &lt;enlace&gt;</code>	Crea una nueva conexión a un repositorio remoto. El <code>&lt;nombre&gt;</code> es el alias del <code>&lt;enlace&gt;</code> al cual se comunica.
<code>git fetch &lt;remoto&gt; &lt;rama&gt;</code>	Obtiene la <code>&lt;rama&gt;</code> del repositorio <code>&lt;remoto&gt;</code> . Sino se especifica la rama trae toda todas las ramas.
<code>git pull &lt;remoto&gt;</code>	Obtiene una copia del repositorio <code>&lt;remoto&gt;</code> y la mezcla al instante con el repositorio local.
<code>git push &lt;remoto&gt; &lt;rama&gt;</code>	Envía la <code>&lt;rama&gt;</code> al repositorio <code>&lt;remoto&gt;</code> actualizándola si existe y sino creandola.

## git diff

<code>git diff HEAD</code>	Muestra diferencias de la ult. confirmación y la carpeta trabajo.
<code>git diff --cached</code>	Muestra diferencias de la ult. confirmación y los cambios marcados.

## git reset

<code>git reset</code>	Reinicia el área de marcado para coincidir con la última confirmación, pero deja la carpeta de trabajo sin cambios.
<code>git reset --hard</code>	Reinicia el área de marcado y carpeta de trabajo para coincidir con la ultima confirmación, sobreescribiendo todos los cambios.
<code>git reset &lt;confirmacion&gt;</code>	Mueve la rama actual hacia la <code>&lt;confirmacion&gt;</code> reiniciando el área de marcado, pero deja la carpeta de trabajo.
<code>git reset --hard &lt;confirmacion&gt;</code>	Lo mismo que <code>git reset --hard</code> pero todos los cambios no confirmados y todas las confirmaciones después de la <code>&lt;confirmacion&gt;</code> .

## git rebase

<code>git rebase -i &lt;base&gt;</code>	Un rebase interactivo que abre un editor para ingresar los comandos para cada confirmación que sera transferida a la nueva <code>&lt;base&gt;</code> .
---	--

## git pull

<code>git pull --rebase &lt;remoto&gt;</code>	Obtiene una copia del <code>&lt;remoto&gt;</code> de la rama actual pero en vez de realizar una mezcla realiza un rebase de las ramas.
---	--

## git push

<code>git push --force &lt;remoto&gt;</code>	Fuerza a Git a enviar la rama al <code>&lt;remoto&gt;</code> ignorando si hay cambios e incompatibilidades. Solo usar cuando estas seguro de lo que haces.
<code>git push --all &lt;remoto&gt;</code>	Envia todas las ramas del repositorio local al repositorio <code>&lt;remoto&gt;</code> .
<code>git push --tags &lt;remoto&gt;</code>	Las etiquetas no son enviadas automáticamente por lo que es necesario usar <code>--tags</code> para enviarlas al repositorio <code>&lt;remoto&gt;</code> .