

# **git**

## **GIT –HELP**

### **la guía sencilla**

**[https://rogerdudler.github.io/git-  
guide/index.es.html](https://rogerdudler.github.io/git-guide/index.es.html)**

# Contenido

GIT –HELP .....	4
----- start a working area ----- (see also: git help tutorial) .....	5
---- work on the current change --(see also: git help everyday).....	6
--examine the history and state -(see also: git help revisions) .....	7
LA GUÍA SENCILLA .....	11
crea un repositorio nuevo.....	12
hacer checkout a un repositorio .....	13
flujo de trabajo .....	14
add & commit .....	16

<b>envío de cambios.....</b>	<b>18</b>
<b>ramas .....</b>	<b>19</b>
<b>actualiza &amp; fusiona .....</b>	<b>22</b>
<b>etiquetas.....</b>	<b>25</b>
<b>reemplaza cambios locales .....</b>	<b>27</b>
<b>datos útiles .....</b>	<b>29</b>
<b>enlaces &amp; recursos.....</b>	<b>31</b>
<b>clientes gráficos .....</b>	<b>31</b>
<b>guías .....</b>	<b>31</b>

# **GIT -HELP**

**These are common Git commands used in various situations:**

----- **start a working area** -----

(see also: [git help tutorial](#))

**clone**      Clone a repository into a  
new directory

**init**      Create an empty Git  
repository or reinitialize an existing one

**--- work on the current change --**

**(see also: git help everyday)**

**add**

Add file contents to the

index

**mv**

Move or rename a file, a

directory, or a symlink

**restore**

Restore working tree files

**rm**

Remove files from the

working tree and from the index

**--examine the history and state -**

(see also: git help revisions)

**bisect**      Use binary search to

find the commit that introduced a bug

**rebase**      Reapply commits on

top of another base tip

**reset**      Reset current HEAD to

the specified state

**switch**      Switch branches

# **tag**

Create, list, delete or

verify a tag object signed with GPG



----- **collaborate** ----- (see  
also: git help workflows)

**fetch** Download objects and  
refs from another repository

**pull** Fetch from and integrate  
with another repository or a local branch

**push** Update remote refs  
along with associated objects

**'git help -a' and 'git help -g' list available subcommands and some**

**---concept guides. See 'git help <command>' or 'git help <concept>'**

**to read about a specific subcommand or concept.**

**See 'git help git' for an overview of the system.**

# **la guía sencilla**

**<https://rogerdudler.github.io/git-guide/index.es.html>**

## **crea un repositorio nuevo**

**Crea un directorio nuevo, ábrelo y ejecuta**

**git init**

**para crear un nuevo repositorio de git.**

## **hacer checkout a un repositorio**

**Crea una copia local del repositorio  
ejecutando**

**git clone /path/to/repository**

**Si utilizas un servidor remoto, ejecuta**

**git clone**

**username@host:/path/to/repository**

## **flujo de trabajo**

**Tu repositorio local esta compuesto por tres "árboles" administrados por git. El primero es tu Directorio de trabajo que contiene los archivos, el segundo es el Index que actua como una zona intermedia, y el último es el HEAD que apunta al último commit realizado.**



working  
dir

add  
→



Index  
(Stage)

commit  
→



HEAD

**add & commit**

**Puedes registrar cambios (añadirlos al Index) usando**

**git add <filename>**

**git add .**

**(con . añade todos)**

**Este es el primer paso en el flujo de trabajo básico. Para hacer commit a estos cambios usa**

**git commit -m "Commit message"**



**Ahora el archivo esta incluido en el HEAD, pero aún no en tu repositorio remoto.**

**envío de cambios**

**Tus cambios están ahora en el HEAD de tu copia local. Para enviar estos cambios a tu repositorio remoto ejecuta**

**git push origin master**

**Reemplaza *master* por la rama a la que quieres enviar tus cambios.**

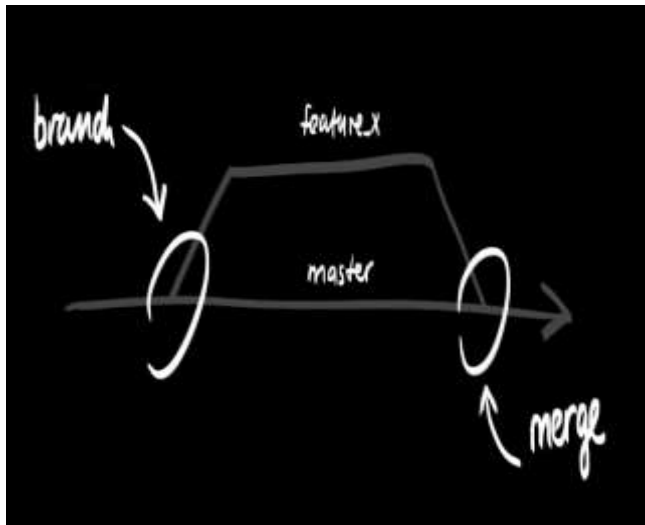
**Si no has clonado un repositorio ya existente y quieres conectar tu repositorio local a un repositorio remoto, usa**

**git remote add origin <server>**

**Ahora podrás subir tus cambios al repositorio remoto seleccionado.**

**ramas**

**Las ramas son utilizadas para desarrollar funcionalidades aisladas unas de otras. La rama *master* es la rama "por defecto" cuando creas un repositorio. Crea nuevas ramas durante el desarrollo y fusiónalas a la rama principal cuando termines.**



**Crea una nueva rama llamada  
"feature\_x" y cámbiate a ella usando**

**git checkout -b feature\_x**

**vuelve a la rama principal**

**git checkout master**

**y borra la rama**

**git branch -d feature\_x**

**Una rama nueva *no estará disponible para los demás* a menos que subas (push) la rama a tu repositorio remoto**

**git push origin <branch>**

**actualiza & fusiona**

**Para actualizar tu repositorio local al  
commit más nuevo, ejecuta  
git pull**

**en tu directorio de trabajo para *bajar y  
fusionar* los cambios remotos.**

**Para fusionar otra rama a tu rama activa  
(por ejemplo master), utiliza**

**git merge <branch>**

en ambos casos git intentará fusionar automáticamente los cambios.

Desafortunadamente, no siempre será posible y se podrán producir *conflictos*.

Tú eres responsable de fusionar esos *conflictos* manualmente al editar los archivos mostrados por git. Después de modificarlos, necesitas marcarlos como fusionados con

**git add <filename>**

**Antes de fusionar los cambios, puedes  
revisarlos usando**

**git diff <source\_branch>  
<target\_branch>**



## **etiquetas**

**Se recomienda crear etiquetas para cada nueva versión publicada de un software.**

**Este concepto no es nuevo, ya que estaba disponible en SVN. Puedes crear una nueva etiqueta llamada *1.0.0* ejecutando**

**git tag 1.0.0 1b2e1d63ff**

***1b2e1d63ff* se refiere a los 10 caracteres del commit id al cual quieres referirte con tu etiqueta. Puedes obtener el commit id con**

**git log**

**también puedes usar menos caracteres que el commit id, pero debe ser un valor único.**

**reemplaza cambios locales**

**En caso de que hagas algo mal (lo que seguramente nunca suceda ;) puedes reemplazar cambios locales usando el comando**

**git checkout -- <filename>**

**Este comando reemplaza los cambios en tu directorio de trabajo con el último contenido de HEAD. Los cambios que ya han sido agregados al Index, así como también los nuevos archivos, se mantendrán sin cambio.**

**Por otro lado, si quieres deshacer todos los cambios locales y commits, puedes traer la última versión del servidor y apuntar a tu copia local principal de esta forma**

**git fetch origin**

**git reset --hard origin/master**

**datos útiles**

**Interfaz gráfica por defecto**

**gitk**

**Colores especiales para la consola**

**git config color.ui true**

**Mostrar sólo una línea por cada commit  
en la traza**

**git config format.pretty oneline**

# Agregar archivos de forma interactiva

**git add -i**

## enlaces & recursos

### clientes gráficos

- [GitX \(L\) \(OSX, open source\)](#)
- [Tower \(OSX\)](#)
- [Source Tree \(OSX, free\)](#)
- [GitHub for Mac \(OSX, free\)](#)
- [GitBox \(OSX\)](#)

### guías

- [Git Community Book](#)
- [Pro Git](#)
- [Think like a git](#)
- [GitHub Help](#)
- [A Visual Git Guide](#)