[[1]](#footnote-2)

Guía para conectarse a un servidor remoto (Febrero 2014)

Raúl Andrés Ortiz Fuentes, 112508.

*Índice de términos*—Estándar IEEE 754 ((*Institute of Electrical and Electronics Engineers*, IEEE) y establece dos formatos básicos para representar a los números reales en la computadora digital: *precisión simple* y *precisión doble*).

Exceso o Sesgo

Representación en exceso

normalización

# INTRODUCCIÓN

E

n el siguiente articulo se relación de forma detallada y simple os pasos para la conexión con un repositorio remoto Git.

# Representación de punto flotante simple y Doble

Git promueve la colaboración y la participación en los diferentes tipos de proyectos, por lo cual da la facilidad de crear y de poner a disposición repositorios remotos, para que los participantes puedan realizar sus aportes y/o modificaciones. Entrando en detalle los repositorios remotos son versiones de un proyecto que se encuentran alojados en Internet o en algún punto de la red, por lo que se pueden tener varios, cada uno de los cuales puede ser de sólo lectura, o de lectura/escritura, según los permisos que se le sean designados. Por tal motivo es importante conocer como se realiza la conexión a los diferentes repositorios remotos, por lo se describen a continuación los comandos básicos que permiten dicha acción:

1. *git remote*

Para ver qué repositorios remotos se tiene configurados, se ejecuta el comando git remote, que mostrará una lista con los nombres de los remotos que sean especificado. Si se ha clonado el repositorio local, en este listado aparecerá seguido de la palabra "origin".

Ejemplo:

*$ git clone git://github.com/schacon/ticgit.git  
Initialized empty Git repository in /private/tmp/ticgit/.git/  
remote: Counting objects: 595, done.  
remote: Compressing objects: 100% (269/269), done.  
remote: Total 595 (delta 255), reused 589 (delta 253)  
Receiving objects: 100% (595/595), 73.31 KiB | 1 KiB/s, done.  
Resolving deltas: 100% (255/255), done.*

*$ cd ticgit  
$ git remote  
origin*

A este comando se le puede añadir la opción -v, que muestra la URL asociada a cada repositorio remoto.

Ejemplo:

*$ git remote –v  
origin git://github.com/schacon/ticgit.git*

1. *git remote*

Este comando permite añadir un nuevo repositorio Git remoto, asignándole un nombre para identificarlos fácilmente.

Ejemplo:  
*$ git remote  
origin  
$ git remote add pb git://github.com/paulboone/ticgit.git  
$ git remote –v  
origin git://github.com/schacon/ticgit.git  
pb git://github.com/paulboone/ticgit.git*

Adicional a lo anterior se puede ejecutar el comando “*git fetch [nombre repositorio]”*, con el cual se puede recuperar toda la información que todavía no se tiene en el repositorio local.

Ejemplo:

*$ git fetch pb  
remote: Counting objects: 58, done.  
remote: Compressing objects: 100% (41/41), done.  
remote: Total 44 (delta 24), reused 1 (delta 0)  
Unpacking objects: 100% (44/44), done.  
From git://github.com/paulboone/ticgit*

*\* [new branch] master -> pb/master  
 \* [new branch] ticgit -> pb/ticgit*

Es importante tener en cuenta que el comando “*fetch”* sólo recupera la información y la pone en tu repositorio local no la une automáticamente con lo que actualmente se está trabajando.

1. *git pull*

Como se evidencio anteriormente el comando “*$ git fetch [remote-name]”* recupera todos los datos del proyecto remoto que todavía no se tengan, por lo que después de haber realizado dicho proceso se deberá tener referencias a todas las ramas del repositorio remoto, que se puede unir o inspeccionar en cualquier momento.

Ahora bien si se ha configurado una rama para seguir a otra rama remota se puede utilizar el comando git pull para recuperar y unir automáticamente la rama remota con tu rama actual, en otras palabras realizar el Merge entre el repositorio remoto y local. Al ejecutar git pull, por lo general se recupera la información del repositorio clonado, y automáticamente se intenta unir con el código con el que está trabajando actualmente.

1. *git push*

Cuando se desee publicar los cambios o el proyecto generado en el repositorio remoto se debe ejecutar el comando *“git push [nombre-remoto][nombre-rama]”.* Este comando funciona únicamente si ha clonado de un servidor en que se tienen permisos de escritura, y nadie ha enviado información en el momento.

1. *git remote show*

Si se quiere ver más información acerca de un repositorio remoto en particular, se puede usar el comando “*git remote show [nombre]”*.

Ejemplo:

*$ git remote show origin  
\* remote origin  
 URL:* [*git@github.com:defunkt/github.git*](mailto:git@github.com:defunkt/github.git) *Remote branch merged with 'git pull' while on branch issues  
 issues  
 Remote branch merged with 'git pull' while on branch master  
 master  
 New remote branches (next fetch will store in remotes/origin)  
 caching  
 Stale tracking branches (use 'git remote prune')  
 libwalker  
 walker2  
 Tracked remote branches  
 acl  
 apiv2  
 dashboard2  
 issues  
 master  
 postgres  
 Local branch pushed with 'git push'  
 master:master*

1. *git remote rename*

Si se quiere renombrar una referencia a un repositorio remoto, en versiones recientes de Git se puede ejecutar “*git remote rename”*.

Ejemplo:

*$ git remote rename pb paul  
$ git remote  
origin  
paul*

# Pasos básicos para conectarse a un repositorio remoto

A continuación se mencionan siete pasos básicos para la conexión con repositorio remoto:

**Paso 0. Identificarse.**

No es que sea necesario, pero Git no dejará permitirá avanzar hasta que no se le indique el nombre y el email con los cuales se configurara el repositorio local:

*git config –global user.name "Pepito el de los Palotes"*

*git config –global user.email pepitopalotes.com*

**Paso 1. Crear el repositorio local.**

Partimos de un directorio con al menos un fichero dentro.

*cd Fotos/photo-scripts*

*git init*

*git add .*

*git commit -m 'Initial commit'*

Paso 2. Crear un repositorio remoto.

Se realiza la creación de la siguiente manera:

*mkdir -p ~/repos/photos-scripts.git*

*git --bare init ~/repos/photos-scripts.git*

**Paso 3. Vincular ambos repositorios.**

Git no tiene el concepto de un repositorio central del que se derivan todas las copias locales, como CVS o Subversion. Para Git, todas las copias son repositorios, así que todo son repositorios. Lo que tenemos ahora es un repositorio en la máquina local y un repositorio en un servidor, lo que queremos es vincularlos. Esto se hace añadiendo el remoto al local, así:

*git remote add github git@github.com:miguev/photo-scripts.git*

Esta operación sólo establece el vínculo entre ambos repositorios, pero no transmite nada entre ellos. Para enviar los contenidos al repositorio remoto, hay que empujarlos:

*git push github master*

**Paso 4. Crear una copia local nueva.**

O lo que es lo mismo, crear un nuevo repositorio clonando uno que sea accesible:

git clone http://github.com/miguev/photo-scripts.git

A partir de ahí tienes un repositorio completamente funcional en tu propio disco local. Puedes hacer cambios y guardarlos (commit) sin estar conectado a Internet.

**Paso 5. Integrar los cambios de vuelta al repositorio remoto.**

Tras hacer cambios en los ficheros, añadir ficheros, etc. viene el momento de guardar (commit) esos cambios, de forma que si algo va mal más adelante se puede volver atrás. Esta operación es igual que en CVS y Subversion:

*git add muestra.txt*

*git commit -a -m "Muestra gratuita." muestra.txt*

Una vez que los cambios guardados están listos para publicar (para algún valor de publicar), la operación necesaria es empujar el repositorio local hacia el remoto:

*git push github master*

**Paso 6. Recuperar esos cambios de vuelta al repositorio original.**

Esta operación no es sino la inversa de la anterior:

*git pull github*

**Paso 7. Push.**

Cuando se hacen cambios en varios repositorios locales, hay que tener cuidado de sincronizar cada uno (pull) con el repositorio remoto antes de enviarle a éste los cambios (push) ya que de lo contrario se producen conflictos al intentarlo.

*git push github*

**Primer autor** Raul Andres Ortiz Fuentes, 112508, Aplicaciones Web Avanzadas.

1. [↑](#footnote-ref-2)