

Conexión de un Cliente con un Repositorio Remoto (Febrero 2014)

Lina Marcela Díaz Calderón 112501.

Índice de términos—omitido

I. INTRODUCCIÓN

Este documento describe en que consiste la conexión de un cliente con un repositorio remoto, dentro de este proceso podemos evidenciar el manejo de los datos utilizando Push y Fetch.

II. REPOSITORIOS REMOTOS

Al trabajar con proyectos Git, es necesario saber cómo funciona los repositorios remotos. Los repositorios remotos son versiones del proyecto que se encuentran alojados en Internet o en algún punto de la red. Se Pueden tener varios, cada uno de los cuales puede ser de sólo lectura, o de lectura/escritura, depende de los permisos que se tengan asignados, compartir con otros usuarios implica gestionar estos repositorios remotos, y mandar (push) y recibir (pull) datos ellos [1].

De acuerdo a lo anterior una vez se decide trabajar con repositorios remotos su implicación es conocer los procedimientos que de este se desprenden como añadir repositorios nuevos, eliminar aquellos que ya no son válidos, gestionar ramas remotas e indicar si están bajo seguimiento o no, y de mas que son necesarios para su correcto funcionamiento.

III. TRABAJANDO CON REPOSITORIOS REMOTOS

Para iniciar a trabajar con repositorios remotos es necesario tener presente las siguientes operaciones:

A. Mostrar los Repositorios Remotos

Para ver qué repositorios remotos que están configurados, se ejecuta el comando `git remote`. Mostrará una lista con los nombres de los remotos que hayas especificado. Si se ha clonado el repositorio, se deberá ver por lo menos "origin" — es el nombre predeterminado que le da Git al servidor del que clonaste—;

```
$ 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
```

B. Listar Repositorios Remotos

Para listar los remotos configurados en un repositorio git, utilizamos el comando `remote`. Si le agregamos la opción `-v` nos da la información completa con los urls de búsqueda y empuje de información.

```
$ git remote
$ git remote -v
```

Este comando lista todos los remotos que se tienen.

C. Añadiendo Repositorios Remotos

Para añadir un nuevo repositorio Git remoto, el comando que se debe ejecutar es `git remote add [nombre] [url]`:

```
$git remote add nombre_remoto url_remoto
```

D. Recibiendo en los Repositorios Remotos

Para recuperar datos los repositorios remotos se ejecuta:

```
$ git fetch [remote-name]
```

Este comando recupera todos los datos del proyecto remoto que no tengas todavía. Después de hacer esto, deberías tener referencias a todas las ramas del repositorio remoto, que puedes unir o inspeccionar en cualquier momento.

Si clonas un repositorio, el comando añade automáticamente ese repositorio remoto con el nombre de "origin". Por tanto, `git fetch origin` recupera toda la información enviada a ese servidor desde que lo clonaste (o desde la última vez que

ejecutaste fetch). 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 tu trabajo ni modifica aquello en lo que estás trabajando.

Si has configurado una rama para seguir otra rama remota, puedes usar el comando git pull para recuperar y unir automáticamente la rama remota con tu rama actual. Éste puede resultarte un flujo de trabajo más sencillo y más cómodo; y por defecto, el comando git clone automáticamente configura tu rama local maestra para que siga la rama remota maestra del servidor del cual clonaste (asumiendo que el repositorio remoto tiene una rama maestra). Al ejecutar git pull, por lo general se recupera la información del servidor del que clonaste, y automáticamente se intenta unir con el código con el que estás trabajando actualmente.

E. Buscando una Rama del Remoto

Para buscar los cambios realizados en una rama del repositorio remoto, utilizamos el comando fetch. Esto actualizará la rama que estemos buscando en nuestra copia local con las actualizaciones realizadas en el repositorio remoto.

```
$git fetch nombre_remoto nombre_rama
```

F. Enviando a los Repositorios Remotos

Cuando se va a compartir un proyecto y se desea enviar a un repositorio remoto el comando que se debe utilizar es :

```
git push [nombre-remoto] [nombre-rama].
```

Si se quiere enviar una rama maestra (master) a un servidor de origen (origin), se ejecuta esto para enviar el trabajo al servidor:

```
$ git push origin master
```

G. Eliminando una rama remota

Para eliminar una rama del repositorio remoto, utilizamos el comando push antecediendo el nombre de la rama con dos puntos (:).

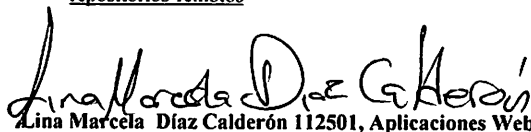
```
$git push nombre_remoto :nombre_rama
```

IV. CONCLUSIÓN

Al trabajar con un repositorio remoto se debe tener presente los comandos que se utilizan para lo que se quiera hacer.

REFERENCIAS

- [1] Git. "Fundamentos de Git - Trabajando con repositorios remotos"
<http://git-scm.com/book/es/Fundamentos-de-Git-Trabajando-con-repositorios-remotos>



Lina Marcela Díaz Calderón 112501, Aplicaciones Web Avanzadas