

# **FUNDAMENTOS GIT TRABAJANDO CON REPOSITARIOS REMOTOS**

**Presentado por:**

**WILSONENRIQUE SALAZAR SIERRA**

**COD: 092502**

**Presentado a:**

**BRAYAN RODRIGUEZ SANCHEZ**

**APLICACIONES WEB AVANZADAS**

**BOGOTÁ 2014**

**FUNDACIÓN INVERISTARIA SAN MARTÍN**

**FACULTAD DE INGENIERIA**

## Remoto:

Un repositorio remoto no es más que una copia de los archivos, carpetas, fotos, etc. que se encuentran bajo la supervisión de git, y están respaldados en un sitio externo a nuestro computador (Internet, o una red local domestica/oficina, etc.) Esto nos permite poder colaborar en un proyecto con otros usuarios de manera distribuida; cabe destacar que un repositorio remoto se puede tener permisos de lectura y escritura o solo lectura

Cuando utilizamos el comando git remote podemos observar que nos retornó origin, y esto no es más que un seudónimo para la dirección <https://github.com/codeheroco/tutorial-git.git> que viene siendo nuestra dirección remota de donde efectuamos la clonación. Para observar si esto es cierto de manera detallada utilizamos el siguiente comando:

```
$ git remote -v
origin https://github.com/codeheroco/tutorial-git.git (fetch)
origin https://github.com/codeheroco/tutorial-git.git (push)
```

## Trabajando con repositorios remotos.

Para poder colaborar en cualquier proyecto Git, necesitas saber cómo gestionar tus repositorios remotos. Los repositorios remotos son versiones de tu proyecto que se encuentran alojados en Internet o en algún punto de la red. Puedes tener varios, cada uno de los cuales puede ser de sólo lectura, o de lectura/escritura, según los permisos que tengas.

### Mostrar repositorios remotos:

Para ver qué repositorios remotos tienes configurados, puedes ejecutar el comando git remote. Mostrará una lista con los nombres de los remotos que hayas especificado. Si has clonado tu repositorio, deberías 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
```

### **Añadir repositorios remotos:**

Para añadir un nuevo repositorio Git remoto, asignándole un nombre con el que referenciarlo fácilmente, ejecuta `git remote add [nombre] [url]`:

```
$ 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
```

Ahora puedes usar la cadena "pb" en la línea de comandos, en lugar de toda la URL. Por ejemplo, si quieres recuperar toda la información de Paul que todavía no tienes en tu repositorio, puedes ejecutar `git fetch pb`:

### **Recibir de los repositorios remotos:**

```
$ 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. (Veremos qué son las ramas y cómo utilizarlas en más detalle en el Capítulo 3.)

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. Tendrás que unir ambos manualmente a posteriori

### **Enviando a tus repositorios remotos:**

Cuando tu proyecto se encuentra en un estado que quieres compartir, tienes que enviarlo a un repositorio remoto. El comando que te permite hacer esto es sencillo: `git push`

[nombre-remoto][nombre-rama]. Si quieres enviar tu rama maestra (master) a tu servidor origen (origin), ejecutarías esto para enviar tu trabajo al servidor:

```
$ git push origin master
```

### Revisar un repositorio remoto:

Si quieres ver más información acerca de un repositorio remoto en particular, puedes usar el comando `git remote show [nombre]`. Si ejecutas este comando pasándole el nombre de un repositorio, como `origin`, obtienes algo así:

```
$ git remote show origin
* remote origin
URL: git://github.com/schacon/ticgit.git
Remote branch merged with 'git pull' while on branch master
master
Tracked remote branches
master
ticgit
```

### Eliminar y renombrar repositorios remotos

Si quieres renombrar una referencia a un repositorio remoto, en versiones recientes de Git puedes ejecutar `git remote rename`. Por ejemplo, si quieres renombrar `pb` a `paul`, puedes hacerlo de la siguiente manera:

```
$ git remote rename pb paul
$ git remote
origin
paul
```

Conviene mencionar que esto también cambia el nombre de tus ramas remotas. Lo que antes era referenciado en `pb/master` ahora está en `paul/master`.

Si por algún motivo quieres eliminar una referencia —has movido el servidor o ya no estás usando un determinado mirror, o quizás un contribuidor ha dejado de contribuir— puedes usar el comando `git remote rm`:

```
$ git remote rm paul
$ git remote
origin
```