Taller de Git y GitHub

Guillermo Valdés Lozano

26 de septiembre de 2015

Documento protegido por GFDL

A cerca de Git

- Git es un software de control de revisiones: Es capaz de recordar los estados previos en que se hayan guardado los muchos archivos de un sistema/documentación/sitio.

A cerca de Git

- Git es un software de control de revisiones: Es capaz de recordar los estados previos en que se hayan guardado los muchos archivos de un sistema/documentación/sitio.
- Se puede comunicar con computadoras y servidores remotos, para lograr así la descarga, sincronización y actualización.

A cerca de Git

- Git es un software de control de revisiones: Es capaz de recordar los estados previos en que se hayan guardado los muchos archivos de un sistema/documentación/sitio.
- Se puede comunicar con computadoras y servidores remotos, para lograr así la descarga, sincronización y actualización.
- Usado por los desarrolladores del kernel **Linux** y diseñado por **Linus Torvalds**.

A cerca de Git

- Git es un software de control de revisiones: Es capaz de recordar los estados previos en que se hayan guardado los muchos archivos de un sistema/documentación/sitio.
- Se puede comunicar con computadoras y servidores remotos, para lograr así la descarga, sincronización y actualización.
- Usado por los desarrolladores del kernel **Linux** y diseñado por **Linus Torvalds**.
- Es Software Libre con licencia GPL versión 2.

A cerca de GitHub

- GitHub es el repositorio más grande y popular de software que usa Git.

A cerca de GitHub

- GitHub es el repositorio más grande y popular de software que usa Git.
- Ofrece **alojamiento ilimitado** en la nube gratuito **para todo lo que sea abierto**.

A cerca de GitHub

- GitHub es el repositorio más grande y popular de software que usa Git.
- Ofrece **alojamiento ilimitado** en la nube gratuito **para todo lo que sea abierto**.
- Las **condiciones de uso** especifican que debe ser un humano de más de 13 años.

A cerca de GitHub

- GitHub es el repositorio más grande y popular de software que usa Git.
- Ofrece **alojamiento ilimitado** en la nube gratuito **para todo lo que sea abierto**.
- Las **condiciones de uso** especifican que debe ser un humano de más de 13 años.
- Que uno es **responsable** de la cuenta y de todo el contenido que subas.

A cerca de GitHub

- GitHub es el repositorio más grande y popular de software que usa Git.
- Ofrece **alojamiento ilimitado** en la nube gratuito **para todo lo que sea abierto**.
- Las **condiciones de uso** especifican que debe ser un humano de más de 13 años.
- Que uno es **responsable** de la cuenta y de todo el contenido que subas.
- Que **NO** es para contenido ilegales o no autorizados.

Versión instalada

Verifique que tenga instalado Git.

Para averiguar la versión instalada

```
$ git version
git version 2.4.6
```

Versión instalada

Verifique que tenga instalado Git.

Para averiguar la versión instalada

```
$ git version
git version 2.4.6
```

Para instalar en la famila de Debian

```
$ sudo apt-get install git-cvs
```

Versión instalada

Verifique que tenga instalado Git.

Para averiguar la versión instalada

```
$ git version
git version 2.4.6
```

Para instalar en la famila de Debian

```
$ sudo apt-get install git-cvs
```

Ejecute Git sin parámetros para mostrar una ayuda simple

```
$ git
```

Configuraciones globales

Es obligado que configure su nombre e e-mail.

Definir su nombre y correo electrónico

```
$ git config –global user.name "Tu Nombre Completo"
$ git config –global user.email tunombre@servidor.com
```

Configuraciones globales

Es obligado que configure su nombre e e-mail.

Definir su nombre y correo electrónico

```
$ git config –global user.name "Tu Nombre Completo"
$ git config –global user.email tunombre@servidor.com
```

Usar colores en la terminal

```
$ git config –global color.ui auto
```

Configuraciones globales

Es obligado que configure su nombre e e-mail.

Definir su nombre y correo electrónico

```
$ git config –global user.name "Tu Nombre Completo"
$ git config –global user.email tunombre@servidor.com
```

Usar colores en la terminal

```
$ git config –global color.ui auto
```

Generar un par de llaves OpenSSH

```
$ ssh-keygen
```

Iniciar un repositorio

Para iniciar un repositorio cámbiese al directorio base del mismo y ejecute git init.

Crear su Primer Repositorio

```
$ cd ~/Documentos/GitHub/PrimerRepositorio
$ git init
```

Todos los comandos Git de este repositorio debe ejecutarlos en este directorio. Se creará un directorio oculto con nombre .git.

Iniciar un repositorio

Para iniciar un repositorio cámbiese al directorio base del mismo y ejecute git init.

Crear su Primer Repositorio

```
$ cd ~/Documentos/GitHub/PrimerRepositorio
$ git init
```

Todos los comandos Git de este repositorio debe ejecutarlos en este directorio. Se creará un directorio oculto con nombre .git.

Configure lo que NO se compartirá

```
$ nano .git/info/exclude
```

Agregar novedades al repositorio

Haga cambios en los archivos o cree nuevos.

Revisar el status

```
$ cd ~/Documentos/Prueba
$ git status
```

Agregar novedades al repositorio

Haga cambios en los archivos o cree nuevos.

Revisar el status

```
$ cd ~/Documentos/Prueba
$ git status
```

Agregar archivos al repositorio local

```
$ git add .
```

Agregar novedades al repositorio

Haga cambios en los archivos o cree nuevos.

Revisar el status

```
$ cd ~/Documentos/Prueba
$ git status
```

Agregar archivos al repositorio local

```
$ git add .
```

Hacer un corte: es su respaldo y lo deja listo para subir

```
$ git commit -m "He hecho unas mejoras para aprender."
```

Agregar novedades al repositorio

Haga cambios en los archivos o cree nuevos.

Revisar el status

```
$ cd ~/Documentos/Prueba
$ git status
```

Agregar archivos al repositorio local

```
$ git add .
```

Hacer un corte: es su respaldo y lo deja listo para subir

```
$ git commit -m "He hecho unas mejoras para aprender."
```

Revisar la bitácora

```
$ git log
```

Descargue un repositorio desde GitHub

Obtener **software público** sin tener una cuenta en GitHub es sencillo.

Descargar Twitter Bootstrap

```
$ mkdir -p ~/Descargas/Git
$ cd ~/Descargas/Git
$ git clone https://github.com/twbs/bootstrap.git
$ cd bootstrap
```

Actualize sus copias locales

Los repositorios pueden recibir actualizaciones frecuentes.

Sincronizar el estado de su copia

```
$ git fetch && git status
```

Actualize sus copias locales

Los repositorios pueden recibir actualizaciones frecuentes.

Sincronizar el estado de su copia

```
$ git fetch && git status
```

Descargar y actualizar su copia

```
$ git pull
```

Actualize sus copias locales

Los repositorios pueden recibir actualizaciones frecuentes.

Sincronizar el estado de su copia

```
$ git fetch && git status
```

Descargar y actualizar su copia

```
$ git pull
```

Revisar la bitácora

```
$ git log
```

Ramas

Un *branch* es una **rama** que permite establecer una ruta distinta; que más adelante podría integrarse a la **rama principal**.

Mostrar la rama en uso

```
$ git branch
```

Ramas

Un *branch* es una **rama** que permite establecer una ruta distinta; que más adelante podría integrarse a la **rama principal**.

Mostrar la rama en uso

```
$ git branch
```

Agregar una nueva rama y cambiarse a ésta

```
$ git branch guillermo
$ git checkout guillermo
```

Ramas

Un *branch* es una **rama** que permite establecer una ruta distinta; que más adelante podría integrarse a la **rama principal**.

Mostrar la rama en uso

```
$ git branch
```

Agregar una nueva rama y cambiarse a ésta

```
$ git branch guillermo
$ git checkout guillermo
```

Revise las ramas y sus últimos comentarios

```
$ git branch -v
```

Cómo funcionan las ramas

Si elimina o renombra archivos o directorios deberá usar el parámetro –**all** al agregar.

Agregue novedades y haga un corte

```
$ git status
$ git add . –all
$ git status
$ git commit -m "He hecho un par de mejoras."
```

Cómo funcionan las ramas

Si elimina o renombra archivos o directorios deberá usar el parámetro –**all** al agregar.

Agregue novedades y haga un corte

```
$ git status
$ git add . –all
$ git status
$ git commit -m "He hecho un par de mejoras."
```

Cámbiese a la rama master, queda como antes de sus cambios

```
$ git checkout master
```

Cómo funcionan las ramas

Si elimina o renombra archivos o directorios deberá usar el parámetro –**all** al agregar.

Agregue novedades y haga un corte

```
$ git status
$ git add . –all
$ git status
$ git commit -m "He hecho un par de mejoras."
```

Cámbiese a la rama master, queda como antes de sus cambios

```
$ git checkout master
```

Regrese a su rama, regresarán los cambios

```
$ git checkout guillermo
```

Fusionar una rama

Lo más sano es ir integrando novedades a las ramas. Cuando queden listas, se integran a la rama **master**. Lo que es lo mismo, mantenga **master** *atrás* como lo estable y a las ramas *adelante* con las novedades.

Fusionar la rama guillermo en master

```
$ git checkout master
$ git merge guillermo
```

Fusionar una rama

Lo más sano es ir integrando novedades a las ramas. Cuando queden listas, se integran a la rama **master**. Lo que es lo mismo, mantenga **master** *atrás* como lo estable y a las ramas *adelante* con las novedades.

Fusionar la rama guillermo en master

```
$ git checkout master
$ git merge guillermo
```

Verifique que master y la rama guillermo son iguales

```
$ git branch -v
```

Fusionar una rama

Lo más sano es ir integrando novedades a las ramas. Cuando queden listas, se integran a la rama **master**. Lo que es lo mismo, mantenga **master** *atrás* como lo estable y a las ramas *adelante* con las novedades.

Fusionar la rama guillermo en master

```
$ git checkout master
$ git merge guillermo
```

Verifique que master y la rama guillermo son iguales

```
$ git branch -v
```

Cuando ya no la necesite; puede eliminar una rama

```
$ git branch -d guillermo
```

Subir una rama a GitHub

Cuando tenga avances terminados o *commiteados* que compartir con sus colegas en GitHub.

Subir una rama a GitHub

```
$ git push origin nombredelarama
```

Subir una rama a GitHub

Cuando tenga avances terminados o *commiteados* que compartir con sus colegas en GitHub.

Subir una rama a GitHub

```
$ git push origin nombredelarama
```

Sus amigos pueden bajar su rama

```
$ git fetch origin nombredelarama
$ git checkout nombredelarama
$ git pull origin nombredelarama
```

Más órdenes útiles

No hay mejor forma de aprender que usándolo.

Sincronice su copia local y lea los cambios de las ramas

```
$ git fetch
$ git branch -v
```

Más órdenes útiles

No hay mejor forma de aprender que usándolo.

Sincronice su copia local y lea los cambios de las ramas

```
$ git fetch
$ git branch -v
```

Para regresar al pasado, destruyendo lo nuevo

```
$ git reset –hard 1234567
```