

Juan José Delgado Quesada
Chair Rama Estudiantil IEEE-UCR
Estudiante de Bachillerato en Ingeniería Eléctrica
Universidad de Costa Rica

Linux - Git

Taller introductorio

Table of Contents

Git

Linux

script

Introducción a los comandos básicos

¿Qué es Git?

Sistema de Control de versiones

Registra los cambios que se realizan sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que se pueda recuperar versiones específicas más adelante.

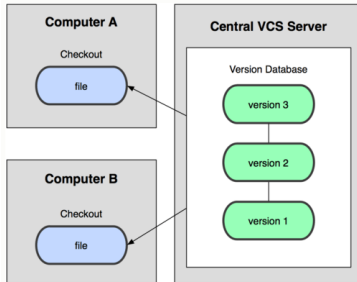
Permite:

- Revertir archivos a un estado anterior.
- Revertir el proyecto entero a un estado anterior.
- Comparar cambios a lo largo del tiempo.
- Ver quién modificó por última vez algo que puede estar causando un problema.
- Examinar quién introdujo un error y cuándo.

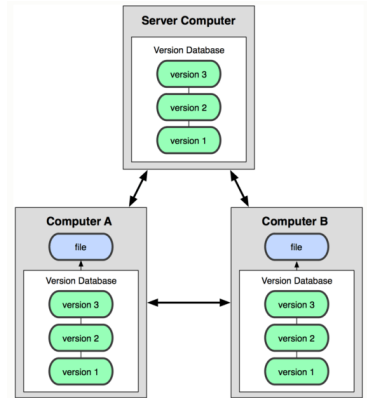
Error común



Centralizado Vs. Distribuido



(a) Centralizado

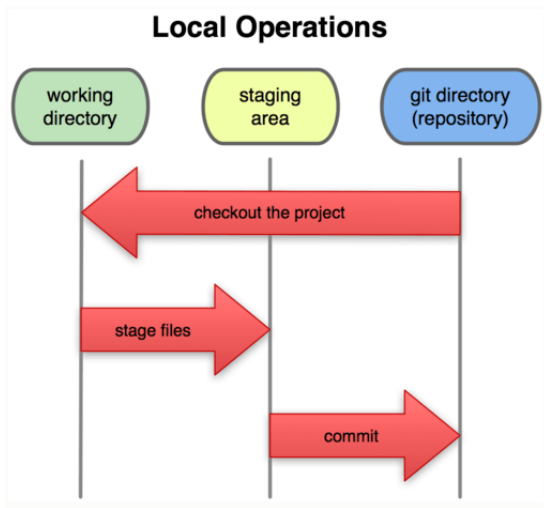


(b) Distribuido

Características

- Casi cualquier operación es local.
- Integridad (Verificación por medio de checksum).
- Generalmente solo se añade información.
- Tres secciones principales de un proyecto de Git: el directorio de Git (Git directory), el directorio de trabajo (working directory), y el área de preparación (staging area).

Secciones principales



Instalando Git desde código fuente

Instalar dependencias

- `$ yum install curl-devel expat-devel gettext-devel \openssl-devel zlib-devel`
- `$ apt-get install libcurl4-gnutls-dev libexpat1-dev gettext \libz-dev libssl-dev`

Descarga

- <http://git-scm.com/download>

Instalando Git desde código fuente

Compilación e instalación

- `$ tar -zxf git-1.6.0.5.tar.gz`
- `$ cd git-1.6.0.5`
- `$ make prefix=/usr/local all`
- `$ sudo make prefix=/usr/local install`

Instalando Git por paquete

Linux

- `$ yum install git-core` ó `$ apt-get install git`

Mac

- <http://sourceforge.net/projects/git-osx-installer/>

Windows

- <http://msysgit.github.com/>

Configuración inicial

Comandos

- `$ git config --list`
- `$ git config --global user.name "NAME"`
- `$ git config --global user.email "EMAIL"`
- `$ git config --global core.editor "EDITOR"`
- `$ git config --global merge.tool "TOOL"`

Nota: Git acepta kdiff3, tkdiff, meld, xxdiff, emerge, vimdiff, gvimdiff, ecmerge, y opendiff como herramientas válidas.

Ayuda

Comandos

- `$ git help <comando>`
- `$ git <comando> -help`
- `$ man git-<comando>`

Obtener un repositorio Git

Dos formas:

- `$ git init`
- `$ git clone git://github.com/schacon/grit.git`

Guardando cambios en el repositorio

Ejemplos

- Comprobando el estado de tus archivos.
- Ignorando archivos.
- Viendo cambios preparados y no preparados.
- Saltarse el área de preparación.
- Mover y eliminar archivos.

Otras posibilidades

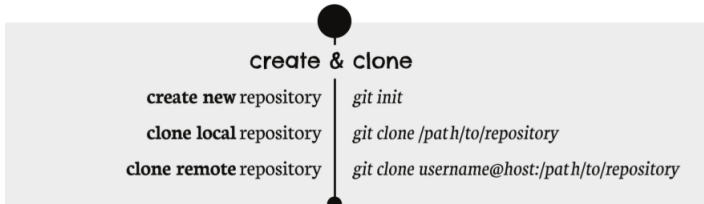
Ejemplos

- Modificar la última confirmación.
- Deshacer la preparación de un archivo.
- Deshacer modificaciones de un archivo.

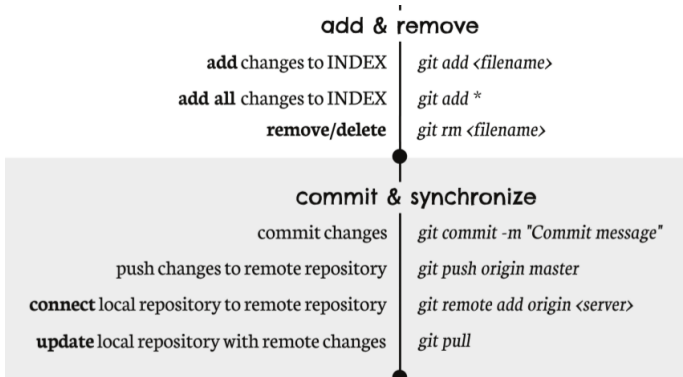
Comandos básicos

git cheat sheet

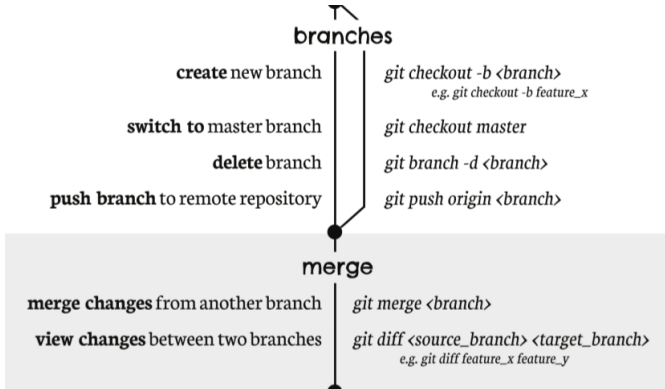
learn more about git the simple way at rogerdudler.github.com/git-guide/
cheat sheet created by Nina Jaeschke of ninagrafik.com



Comandos básicos



Comandos básicos

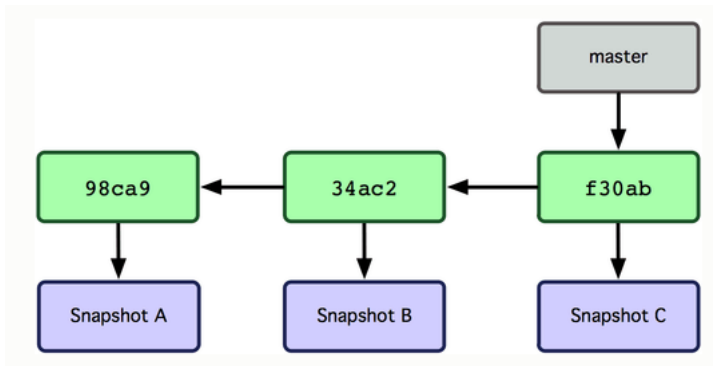


Comandos básicos

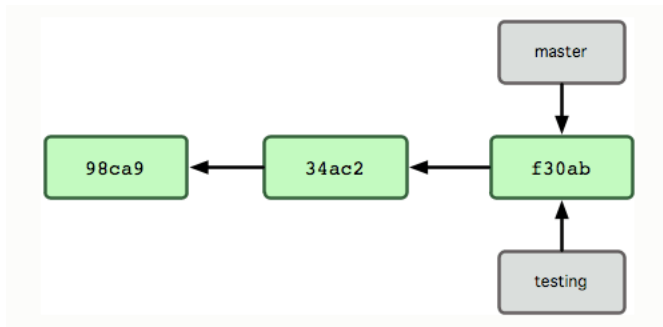


Concepto de Branch

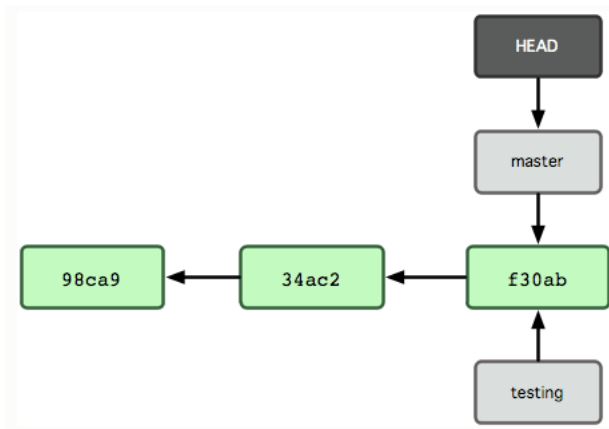
Origen master



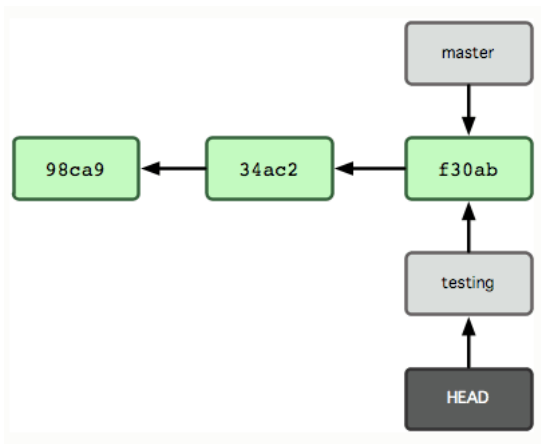
New Branch: git branch



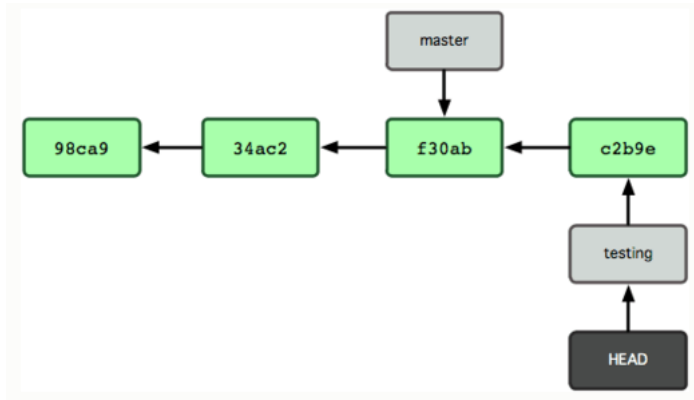
Concepto del puntero HEAD



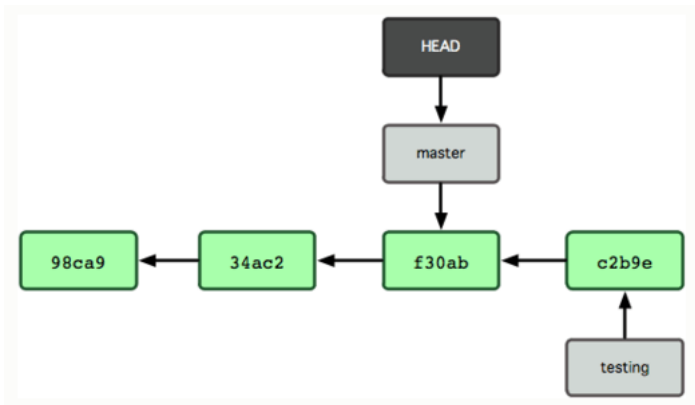
Puntero HEAD al saltar de rama



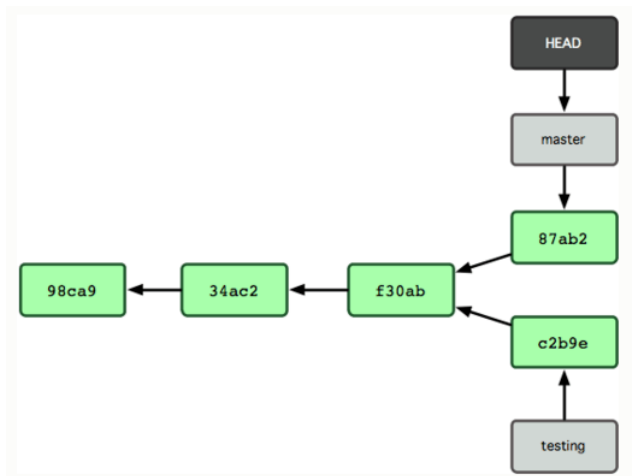
Avance de la rama apuntada por HEAD



Git Checkout a la rama master



Divergencia en los registros de las ramas



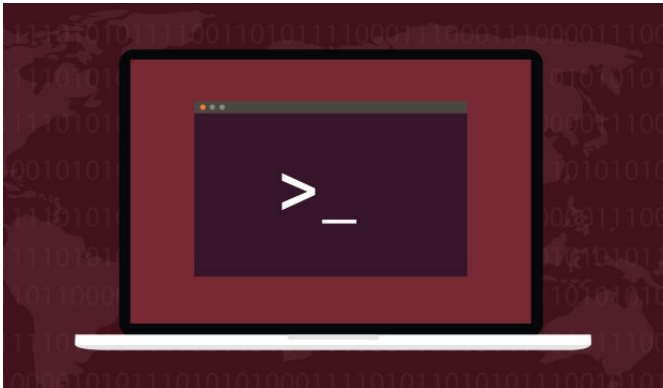
Resumen de comandos:

- git branch
- git checkout
- git checkout -b
- git merge
- git branch -d
- git branch --merged
- git branch --no-merged
- git push --set-upstream origin
- git push origin --delete

Introducción a Linux

Shell de Linux

Intérprete de línea de comandos.



JUAN @ LEON : ~\$
usuario - servidor - normal / superusuario

Sudo: Superusuario Do

Estructura del directorio de Linux

- ¿ Dónde se encuentran los programas ?
- ¿ Dónde están los archivos de configuración ?
- ¿ Dónde podría encontrar los archivos de registro para esta aplicación ?

Directorios

Dir

/	El directorio llamado "root". Es el punto de partida para la jerarquía del sistema de archivos. Tenga en cuenta que esto no está relacionado con la raíz, o superusuario.
/bin	Binarios y otros programas ejecutables
/etc	Archivos de configuración de sistema
/home	Directorios raíz
/opt	Software opcional o de terceros
/tmp	Espacio temporal, que normalmente se borra al reiniciar
/usr	Programas relacionados con el usuario
/var	Datos variables, mayormente ficheros log

Directorios

/boot	Archivos necesarios para iniciar el sistema
/cdrom	Grabación para CD-ROMs.
/cgroup	Jerarquía de Grupos de Control
/dev	Los archivos de dispositivo, normalmente controlados por el sistema operativo y los administradores del sistema.
/etc	Archivos de configuración del sistema Archivos del sistema compartidos.
/export	Normalmente se encuentran en sistemas Solaris
/home	Directorios raíz
/lib	Librerías del Sistema

Comandos básicos de linux

ls

Lista el contenido del directorio. Lo usará para mostrar información acerca de archivos y directorios.

cd

Cambia el directorio actual a dir. Si ejecuta cd sin especificar un directorio, cd cambia el directorio actual al directorio de inicio. Esta es la forma de navegar por el sistema.

Comandos básicos de linux

`pwd`

Muestra el nombre del directorio de trabajo actual.

`cat`

Concatena y muestra archivos.

`echo`

Muestra los argumentos de la pantalla.

Comandos básicos de linux

man

Muestra el manual para el comando solicitado.

exit, log out, Ctrl + d

Sale del Shell o la sesión actual.

clear

Borra la pantalla.

Directorios

.

Este directorio.

..

El directorio padre.

\

Separador de directorio. Directorios terminan con una barra inclinada y esto usualmente se asume.

Directorios y comandos

-

Directorio pasado.

mkdir y rmdir

Crea un directorio y elimina un directorio vacío respectivamente.

rm -rf

Elimina recursivamente el directorio y todos los archivos y directorios dentro de él. Utilizar con precaución.

Listar archivos y entender las salidas

ls -l

Salida en formato largo.

Artículo	Valor
Permisos	-rw-rw-r--
Número de enlaces	1
Nombre del propietario	bob
Nombre del grupo	users
Número de bytes en el archivo	10400
Última fecha de modificación	Sep 27 08:52
Nombre fichero	sales.data

Listar archivos y entender las salidas

ls -a

Los archivos y directorios que empiezan con un punto se consideran ocultos y no se muestran por defecto.

cambiar el orden

- `ls -l -a`
- `ls -la`
- `ls -al`
- `ls -a -l`

Listar archivos y entender las salidas

ls -F

Para revelar de que tipo es cada archivo.

Símbolo	Significado
/	Directorio.
@	Link. El archivo que sigue al símbolo -> es el destino del enlace.
*	Programa ejecutable

Directorios y comandos

`ls -t / ls -tr`

Archivos por tiempo y en orden inverso.

`ls -R`

Muestra archivos y directorios de forma recursiva.

`tree / -d -C`

Lista contenido de directorios en formato de árbol.

Directorios y comandos

chmod

Modificar los permisos de los archivos. Se puede realizar de forma explícita o en binario.

find

Buscar archivos dentro de todo un árbol de directorios. Existen diferentes atributos por nombre, fecha, etc.

diff

Muestra la diferencia que existen entre varios archivos.

Otros comandos:

- `grep`
- `rm "file"`
- `rm -r "directorio"`
- `rm -f "file"`
- `cp "Source_file" "Destino"`
- `cp -r "source_directory" "destino"`
- `mv "Source_file" "destino"`
- `history`

Desarrollo de scripts

¿Qué es un script?

- 1 Es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.
- 2 El uso habitual de los scripts (script, archivo de órdenes, archivo de procesamiento por lotes o guion) es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario.

Pasos para hacer un script

- 1 Crear un archivo de texto ASCII con cualquier editor de texto.
- 2 Nombrar el archivo. La terminación “.sh” es opcional.
- 3 Encabezado: La primera línea sirve para indicar que shell utilizamos (bash) y donde puede ser encontrado. Esta línea debe ser la primera de todos los scripts que realicemos.

#!/bin/bash

Pasos para hacer un script

- 1 Escribir las instrucciones, comandos o código.
- 2 Modificar los permisos del archivo.
- 3 Probarlo.

source "script"

- 4 Ejecutarlo.

./"script"

ejemplo1.sh

Aclaración sobre comando echo

- echo esto es un asterisco * sin comillas
- echo esto es un dólar y tres letras \$ABC sin comillas
- echo "esto es un asterisco * entre comillas dobles"
- echo "esto es un dólar y tres letras \$ABC entre comillas dobles"
- echo 'esto es un asterisco * entres comillas simples'
- echo 'esto es un dólar y tres letras \$ABC entre comillas simples'

ejemplo2.sh

Ejecución de varias líneas

```
;
```

Si tenemos que ejecutar varias líneas y queremos escribirlas en una sola.

```
\
```

Escribir una sola línea en varias.

ejemplo3.sh

Uso de variables

No tienen tipo definido ni necesitan ser declaradas antes de poder ser usadas.

```
#!/bin/bash  
DECIR="Hola Mundo"  
echo $DECIR
```

Expansiones

```
echo pwd  
echo $(pwd)
```

ejemplo4.sh

Operación Aritmética

`$(())` o `$([])` o `let`

```
#!/bin/bash
PRECIO_COPAS=60
BEBEDORES=10
PAGO=$((PRECIO_COPAS / BEBEDORES))
echo Se tiene que pagar $PAGO dólares por persona.
```

ejemplo5.sh

Funciones

Estructura

```
function nombre_función {  
  líneas de la función  
}
```

Global Vs. Local

ejemplo6.sh

¡Muchas Gracias!