Sistemas de Control de Versiones

René Mayorga
rmayorga@debian.org

Comunidad de usuarios Debian de El Salvador

30 de enero de 2009

Esquema

¿Qué es un sistema de control de versiones?

¿Ideas?

¿Qué es un sistema de control de versiones?

¿Ideas?

¿Quien usa sistemas de control de versiones aqui?

¿Qué es un sistema de control de versiones?

¿Ideas?

¿Quien usa sistemas de control de versiones aqui?

¿Cuales?

¿Qué es un sistema de control de versiones?

¿Ideas?

¿Quien usa sistemas de control de versiones aqui?

¿Cuales?

Un par de conceptos

Un par de conceptos

- CVS, o Control Versions System es el concepto detras de todo el tema. Un Control de Versiones es una manera ordenada de mantener un rastro de modificaciones y almacenar versiones y revisiones especificas.

Un par de conceptos

- CVS, o Control Versions System es el concepto detras de todo el tema. Un Control de Versiones es una manera ordenada de mantener un rastro de modificaciones y almacenar versiones y revisiones especificas.
- Repositorio, Es el lugar donde se almacena el código fuente por lo general este es accesible a todos los usuarios que colaboran o que siguen el desarrollo.

Un par de conceptos

- CVS, o Control Versions System es el concepto detras de todo el tema. Un Control de Versiones es una manera ordenada de mantener un rastro de modificaciones y almacenar versiones y revisiones especificas.
- Repositorio, Es el lugar donde se almacena el código fuente por lo general este es accesible a todos los usuarios que colaboran o que siguen el desarrollo.
- CheckOut, es la accin de desbloquear una copia de trabajo por lo general, esta copia se almacena en el equipo local.

Un par de conceptos

- Tag, es una copia exacta de una versión especifica, es decir, esta versión esta liberada y no sufrira más cambios.

Un par de conceptos

- Tag, es una copia exacta de una versión especifica, es decir, esta versión esta liberada y no sufrira más cambios.
- trunk/master, esta es la rama principal de desarrollo por lo general todo el desarrollo de un proyecto se lleva a cabo sobre esta rama.

Un par de conceptos

- Tag, es una copia exacta de una versión especifica, es decir, esta versión esta liberada y no sufrira más cambios.
- trunk/master, esta es la rama principal de desarrollo por lo general todo el desarrollo de un proyecto se lleva a cabo sobre esta rama.
- branch, o ramas, son bifurcaciones del proyecto, estan puden llevar un nuevo feature, o caracterisitcas totalmente inestables destinadas a pruebas de concepto, o un desarrollo especifico con fines especificos.

Un par de conceptos

- Tag, es una copia exacta de una versión especifica, es decir, esta versión esta liberada y no sufrira más cambios.
- trunk/master, esta es la rama principal de desarrollo por lo general todo el desarrollo de un proyecto se lleva a cabo sobre esta rama.
- branch, o ramas, son bifurcaciones del proyecto, estan puden llevar un nuevo feature, o caracterisitcas totalmente inestables destinadas a pruebas de concepto, o un desarrollo especifico con fines especificos.
- commit, o push, es la acción de enviar cambios locales al servidor.

Distribuidos y centralizados

- Centralizados, son aquellos sistemas de control de versiones donde la información es administrada y mantenida en un repositorio central haciendo imperatia la necesidad de tener siempre un enlace disponible para tareas como la creación de tags o brances.

Distribuidos y centralizados

- Centralizados, son aquellos sistemas de control de versiones donde la información es administrada y mantenida en un repositorio central haciendo imperatia la necesidad de tener siempre un enlace disponible para tareas como la creación de tags o brances.
- Distribuidos, con este esquema el servidor básicamente solo mantiene una copia del repositorio, pero cada equipo se convierte en un repositorio, permitiendo fácilmente la bifurcación en ramas y el manejo del repositorio de una manera más flexible.

vamos al grano, SVN

SVN es conocido así por ser el nombre del cliente, el software en si es llamado subversion.

vamos al grano, SVN

SVN es conocido así por ser el nombre del cliente, el software en si es llamado subversion.

Este es un sistema de control de versiones centralizado, fue diseñado como reemplazo de CVS,

vamos al grano, SVN

SVN es conocido así por ser el nombre del cliente, el software en si es llamado subversion.

Este es un sistema de control de versiones centralizado, fue diseñado como reemplazo de CVS,

¿Alguien uso CVS?

vamos al grano, SVN

SVN es conocido así por ser el nombre del cliente, el software en si es llamado subversion.

Este es un sistema de control de versiones centralizado, fue diseñado como reemplazo de CVS,

¿Alguien uso CVS?

¿Alguien uso SVN?

SVN

Veamos un par de comandos

creando el repositorio

Necesitamos instalar un par de paquetes

creando el repositorio

Necesitamos instalar un par de paquetes

```
apt-get install subversion
```

esto instalara el cliente de subversion y las herramientas necesarias. Luego podemos crear un repositorio local.

creando el repositorio

Necesitamos instalar un par de paquetes

```
apt-get install subversion
```

esto instalara el cliente de subversion y las herramientas necesarias. Luego podemos crear un repositorio local.

```
@tepitzin:/tmp/svn $ svnadmin create repositorio
```

creando el repositorio

Necesitamos instalar un par de paquetes

```
apt-get install subversion
```

esto instalara el cliente de subversion y las herramientas necesarias. Luego podemos crear un repositorio local.

```
@tepitzin:/tmp/svn $ svnadmin create repositorio
```

Si somos curioso y examinamos el contenido del directorio podremos notar que son ciertos archivos y directorios que contendran el repositorio.

SVN

Ahora necesitamos hacer un checkout de nuestro repositorio

SVN

Ahora necesitamos hacer un checkout de nuestro repositorio

```
@tepitzin:~/temp $ svn co file:///tmp/svn/repositorio repo
Revisin obtenida: 0
```

SVN

Ahora necesitamos hacer un checkout de nuestro repositorio

```
@tepitzin:~/temp $ svn co file:///tmp/svn/repositorio repo
Revisin obtenida: 0
```

Con esto tenemos nuestro rep sincronizado en el directorio repo

SVN

Veamos que paso aquí

SVN

Veamos que paso aquí

- revisión, si observamos, nuestro cliente svn nos devolvio un mensaje, comentandonos que se obtuvo la revisión 0. SVN usa estos números de revisiones para llevar el control de los cambios estos números son enteros secuenciales, y una revisión aplica a todos los ficheros y directorios de nuestro repositorio.

SVN

Veamos que paso aquí

- revisión, si observamos, nuestro cliente svn nos devolvio un mensaje, comentandonos que se obtuvo la revisión 0. SVN usa estos números de revisiones para llevar el control de los cambios estos números son enteros secuenciales, y una revisión aplica a todos los ficheros y directorios de nuestro repositorio.
- El directorio esta vacio!, y es que no existe una infraestructura de desarrollo todavía, nosotros necesitamos crearla, básicamente crear los directorios.

SVN y la estructura de directorios

Bueno, vamos con los directorios. por lo general, necesitarmos un trunk, un directorio para nuestros tags y posiblemente un directorio para los branches

SVN

Los directorios una vez hagamos el checkout simplemente necesitamos crearlos, con un simple vulgar y silvestre mkdir. es fácil

SVN

Los directorios una vez hagamos el checkout simplemente necesitamos crearlos, con un simple vulgar y silvestre mkdir. es fácil

```
mkdir trunk tags branches
```

SVN

Los directorios una vez hagamos el checkout simplemente necesitamos crearlos, con un simple vulgar y silvestre mkdir. es fácil

```
mkdir trunk tags branches
```

Ya explicamos que era eso de branches, trunk y tags

SVN

Los directorios una vez hagamos el checkout simplemente necesitamos crearlos, con un simple vulgar y silvestre mkdir. es fácil

```
mkdir trunk tags branches
```

Ya explicamos que era eso de branches, trunk y tags

¿Alguien se acuerda?

SVN

Hora de hacer nuestro primer commit

SVN

Hora de hacer nuestro primer commit

Para comenzar a llevar tracking de los archivos, necesitamos agregarlos para usamos svn add

```
@tepitzin:~/temp/repo $ svn add branches tags trunk
A         branches
A         tags
A         trunk
@tepitzin:~/temp/repo $
```

SVN

Ahora hacemos el primer commit al repositorio

```
@tepitzin:~/temp/repo $ svn ci -m "creando la estructura"
Aadiendo       branches
Aadiendo       tags
Aadiendo       trunk

Commit de la revisin 1.
```

SVN

Ahora hacemos el primer commit al repositorio

```
@tepitzin:~/temp/repo $ svn ci -m "creando la estructura"
Aadiendo       branches
Aadiendo       tags
Aadiendo       trunk

Commit de la revisin 1.
```

Si notamos la revisión cambio, ahora es 1

SVN

Ok, olvidemos un segundo de los slides y vamos a hacerlo practico

SVN

Ok, olvidemos un segundo de los slides y vamos a hacerlo practico

recordemos, es un taller :)

SVN - terminando

- svn co

SVN - terminando

- svn co
- svn up

SVN - terminando

- svn co
- svn up
- svn add/rm/mv/mkdir

SVN - terminando

- svn co
- svn up
- svn add/rm/mv/mkdir
- svn ci

SVN - terminando

- svn co
- svn up
- svn add/rm/mv/mkdir
- svn ci
- svn status

SVN - terminando

- svn co
- svn up
- svn add/rm/mv/mkdir
- svn ci
- svn status
- svn diff

SVN - terminando

- svn co
- svn up
- svn add/rm/mv/mkdir
- svn ci
- svn status
- svn diff
- svn log

SVN - terminando

- svn co
- svn up
- svn add/rm/mv/mkdir
- svn ci
- svn status
- svn diff
- svn log
- svn blame

Git

git es un sistema de control de versiones distribuido escrito originalmente por Linus Torvalds, actualmente es uno de los sistemas de control de versiones más populares que existen

git

Ok, creamos un repo, pero primero, le decimos a git quien soy

git

Ok, creamos un repo, pero primero, le decimos a git quien soy

```
[~]$ git config --global user.name "Rene Mayorga"
[~]$ git config --global user.email rmayorga@debian.org
```

Ahora si, a crear el repo

Ahora si, a crear el repo

```
@tepitzin:/tmp/repo $ git init
Initialized empty Git repository in /tmp/repo/.git/
```

Ahora si, a crear el repo

```
@tepitzin:/tmp/repo $ git init
Initialized empty Git repository in /tmp/repo/.git/
```

Con esto estamos listos, podemos agregar archivos con git add, y luego a hacer commits con git commit

git

dejemonos de slides y pasemos a jugar con git

recapitulando git

- git init

recapitulando git

- git init
- git branch

recapitulando git

- git init
- git branch
- git log

recapitulando git

- git init
- git branch
- git log
- git merge

recapitulando git

- git init
- git branch
- git log
- git merge
- git status

recapitulando git

- git init
- git branch
- git log
- git merge
- git status
- git commit

recapitulando git

- git init
- git branch
- git log
- git merge
- git status
- git commit
- git pull

recapitulando git

- git init
- git branch
- git log
- git merge
- git status
- git commit
- git pull
- git push

recapitulando git

- git init
- git branch
- git log
- git merge
- git status
- git commit
- git pull
- git push
- git checkout

recapitulando git

- git init
- git branch
- git log
- git merge
- git status
- git commit
- git pull
- git push
- git checkout
- git add/rm/mv