



Subversion en Debian Sarge 3.1

REVISIONS

Versión	Data	Autor	Comentarios
0.1	29/10/2006	Daniel Díaz De La Iglesia	Versión inicial
0.2	30/10/2006	Daniel Díaz De La Iglesia	Corrección erratas referente a clientes de subversion.

Paquetes necesarios

- **Apache 2**
Mediante DAV será o encargado de servir os repositorios svn.
- **Modulo subversion para Apache 2**
É o módulo que permite a Apache 2 funcionar con subversion.
- **subversion**
É o sistema de control de versións que imos empregar.
- **subversion-tools**
Diversas ferramentas e utilidades para subversion

Instalacion de paquetes necesarios

```
apt-get install apache2 libapache2-svn subversion subversion-tools
```

Preparando o sistema

Os diferentes proxectos de subversion almacenaríanse nunha partición aparte, para desta maneira, ter cercado este servicio e evitar que os usuarios de subversion poidan provocar malfuncionamento no resto dos servicios.

Para iso, procederemos a montar unha partición dedicada en `/var/lib/subversion`. Crearemos unha particion con fdisk e logo fomatearemos esa partición, no meu caso, fixeno usando reiserfs e dándolle 1 GB de capacidade. O tamaño dependerá dos proxectos que vai xestionar subversion.

```
mkfs.reiserfs /dev/sda29
```

Editaremos fstab pra que ó arrancar monte a partición dos repositorios:

```
jed /etc/fstab
```

Engadimos a seguinte liña, tendo en conta que `/dev/sda29` é a partición que usamos pra subversión e os seus repositorios.

```
/dev/sda29 /var/lib/subversion reiserfs defaults 0 2
```

Creamos o directorio do novo punto de montaxe, e montamos a partición:

```
mkdir /var/lib/subversion  
mount /var/lib/subversion
```

Creando repositorios

O noso primeiro repositorio chamaráse *repositorio1*.

- Crearemos a carpeta do novo repositorio
- Con `svnadmin` crearemos o estrutura do novo repositorio subversion.
- Cambiaremos o propietario a `www-data`, que é o usuario que emprega `apache2`.

```
mkdir /var/lib/subversion/repositorio1
svnadmin create /var/lib/subversion/repositorio1
chown www-data:www-data /var/lib/subversion/repositorio1 -R
```

Con iso o repositorio xa está configurado.

Configurando apache2

Elexir porto

No meu caso, estou usando apache para servir páxinas, polo que o porto 80 o teño ocupado para a versión 1.x de apache.

É por iso, que cambiei o porto que usa apache2. No meu caso, elixín o 8383

```
jed /etc/apache2/ports.conf
```

Arranque automático

En sarge, para que apache2 arranque automaticamente, hai que editar o ficheiro `/etc/default/apache2`. Seguramente isto débese a que por defecto en Sarge apache1 está escoitando polo porto 80, o igual que a configuración por defecto de apache2.

Cambiamos a opción de 1 a 0.

```
jed /etc/default/apache2
```

```
# 0 = start on boot; 1 = don't start on boot  
NO_START=0
```

Reiniciamos apache2

```
/etc/init.d/apache2 start
```

Neste intre apache2 debería estar escoitando no porto que configuramos, para iso, abriremos un navegador e probaremos a conectamos. Si estamos nunha consola na propia máquina faremos:

```
lynx http://localhost:8383
```

E debería aparecernos a páxina por defecto de Apache 2. Si non é así, deberemos mirar o ficheiro de logs de Apache2 e revisar cal é o erro.

Modulo subversion

O ficheiro de configuración do módulo de subversion sobre apache2, é */etc/apache2/mods-enabled/dav_svn.conf*

Editaremoslo, comentando a configuración para o repositorio que ten por defecto e crearemos a nosa configuración específica para cada novo repositorio, poñendo esto:

```
<Location /repositorio1>
DAV svn
AuthType Basic
AuthName "Servidor subversion"
AuthUserFile /etc/apache2/dav_repositorio1.passwd

SVNPATH /var/lib/subversion/repositorio1
#Pra que non entre calquera
Require valid-user

</Location>
```

Neste ficheiro de configuración, indicamos:

- **AuthUserFile**
Indicamos o ficheiro que almacenará os usuarios/passwords do repositorio.
- **SVNPATH**
Ruta do repositorio
- **Require valid-user**
Desta maneira, sempre preguntará para acceder ó repositorio, incluso pra ler os contidos.

Xestión Usuarios:

Teremos que dar de alta usuarios, a primeira vez non existirá o ficheiro de passwords, polo que empregaremos a opción *-c* pra crealo:

```
htpasswd2 -c /etc/apache2/dav_repositorio1.passwd usuario1
htpasswd2 /etc/apache2/dav_repositorio1.passwd usuario2
```

Probamos a conectamos a través dun navegador.

```
lynx http://localhost:8383/repositorio1
```

Si todo sale correctamente, tras introducir usuario/contrasinal, deberíamos ver unha paxina mostrando o contido do repositorio subversion en cuestión.

Si da un erro, o máis aconsellable e ver los logs de apache2, que nos dirán cal é o erro concreto.

Cientes para subversion

Consola

Desde consola, podemos usar os comandos que proporciona o paquete subversion. É moi recomendable aprender a utilizar subversion por consola, pois así aprenderemos o seu funcionamento. Ademáis, nada nos garantiza ter un cliente gráfico de subversion en calquera máquina.

Para iso non temos máis que empregar svn. Para calquera dúbida: `svn help`.

```
ideafix$ svn help
```

```
usage: svn <subcommand> [options] [args]
Subversion command-line client, version 1.4.0.
Type 'svn help <subcommand>' for help on a specific subcommand.
Type 'svn --version' to see the program version and RA modules
  or 'svn --version --quiet' to see just the version number.
Most subcommands take file and/or directory arguments, recursing
on the directories. If no arguments are supplied to such a
command, it recurses on the current directory (inclusive) by default.
Available subcommands:
```

```
add
blame (praise, annotate, ann)
cat
checkout (co)
cleanup
commit (ci)
copy (cp)
delete (del, remove, rm)
diff (di)
export
help (?, h)
import
info
list (ls)
lock
log
merge
mkdir
move (mv, rename, ren)
propdel (pdel, pd)
propedit (pedit, pe)
propget (pget, pg)
proplist (plist, pl)
propset (pset, ps)
resolved
revert
status (stat, st)
switch (sw)
unlock
update (up)
```

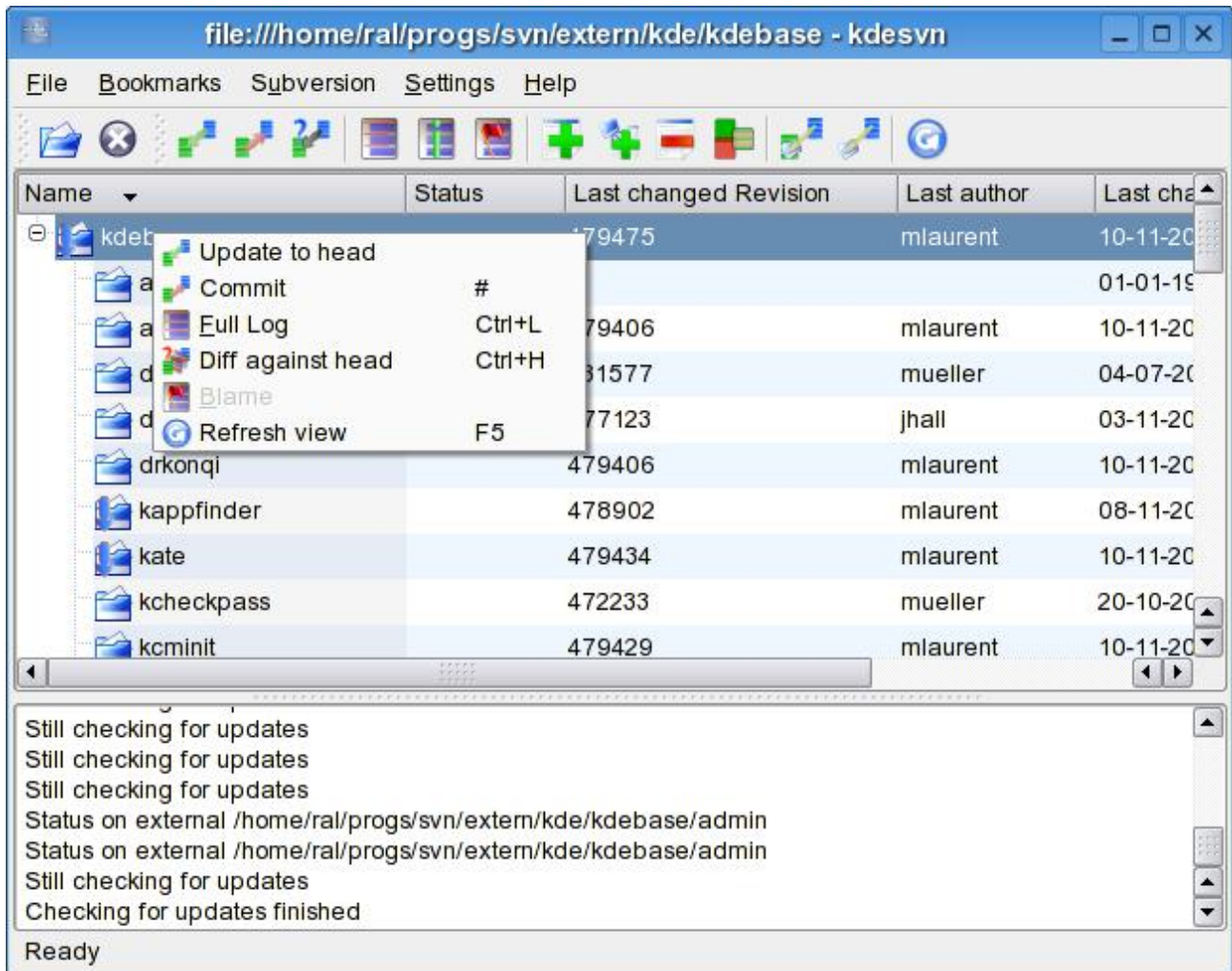
```
Subversion is a tool for version control.
For additional information, see http://subversion.tigris.org/
```

Aplicación gráfica

kdesvn

En etch esta disponible *kdesvn*, que aunda que está nunha versión moi nova, realmente é funcional e promete.

```
apt-get install kdesvn kompare
```



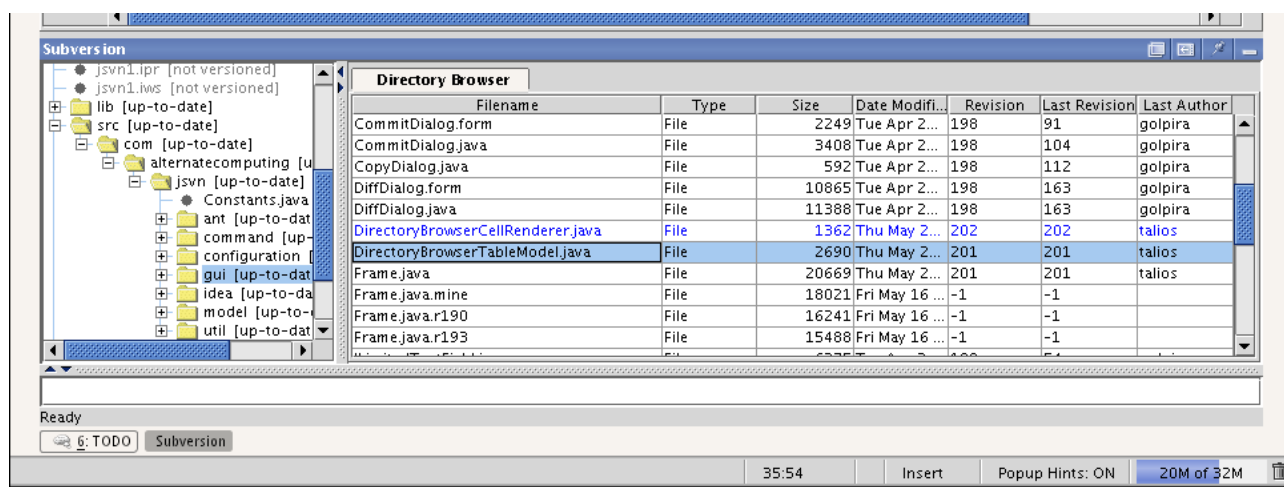
jsvn

Existe unha aplicación java chamada *jsvn* que podemos atopar en <http://jsvn.alternatecomputing.com/>

Ten boa pinta, pero non garda os datos de configuración e hai que indicar repositorios, usuarios, etc cada vez que a arrancamos.

Descargaremosla de alí e para executala:

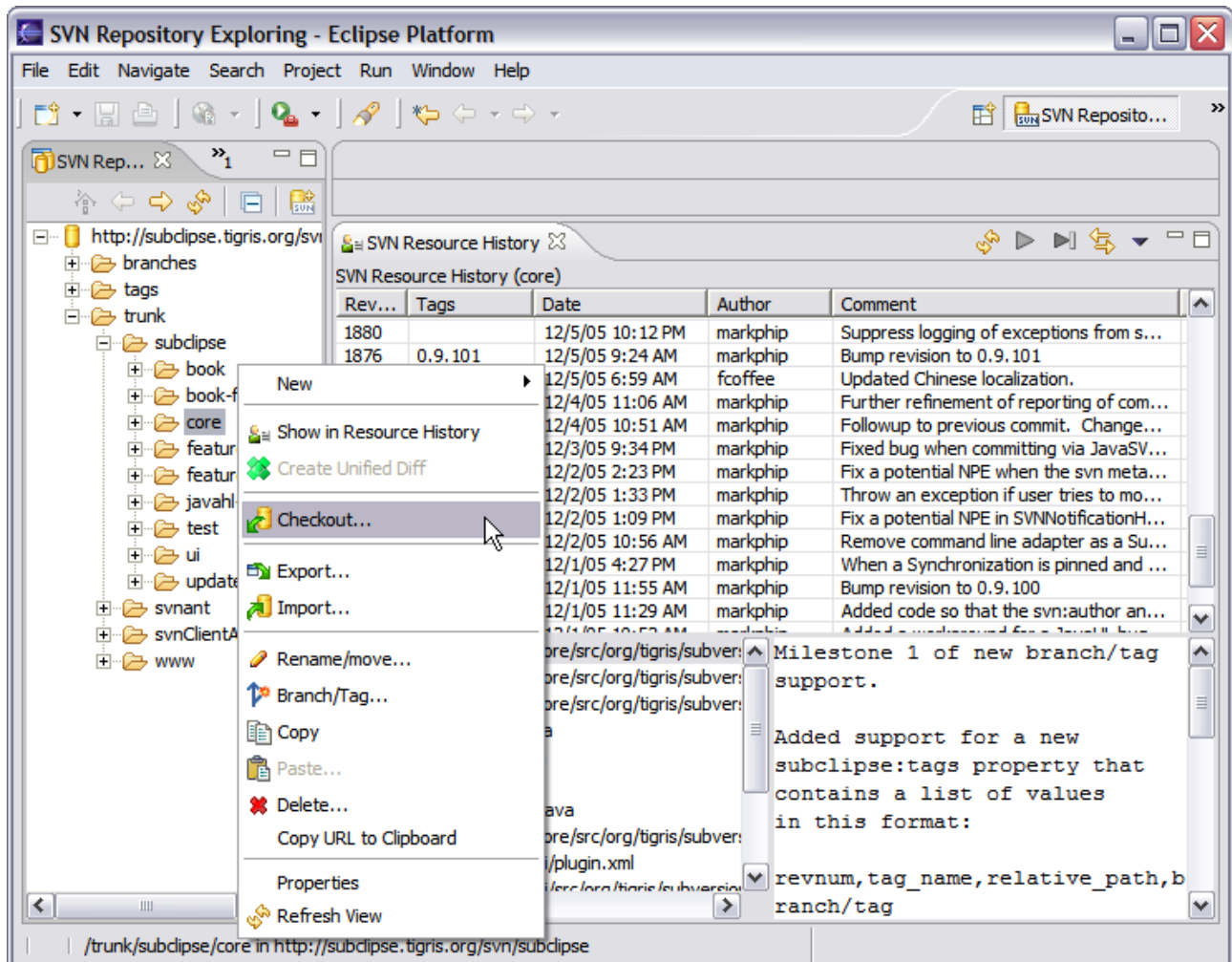
```
java -jar jsvn_version.jar
```



IDEs de desenrolo

Eclipse

Eclipse foi dos primeiros IDEs en ter soporte para subversion. Para a súa instalación, na URL <http://subclipse.tigris.org/install.html> explica todo o proceso de instalar e configurar.



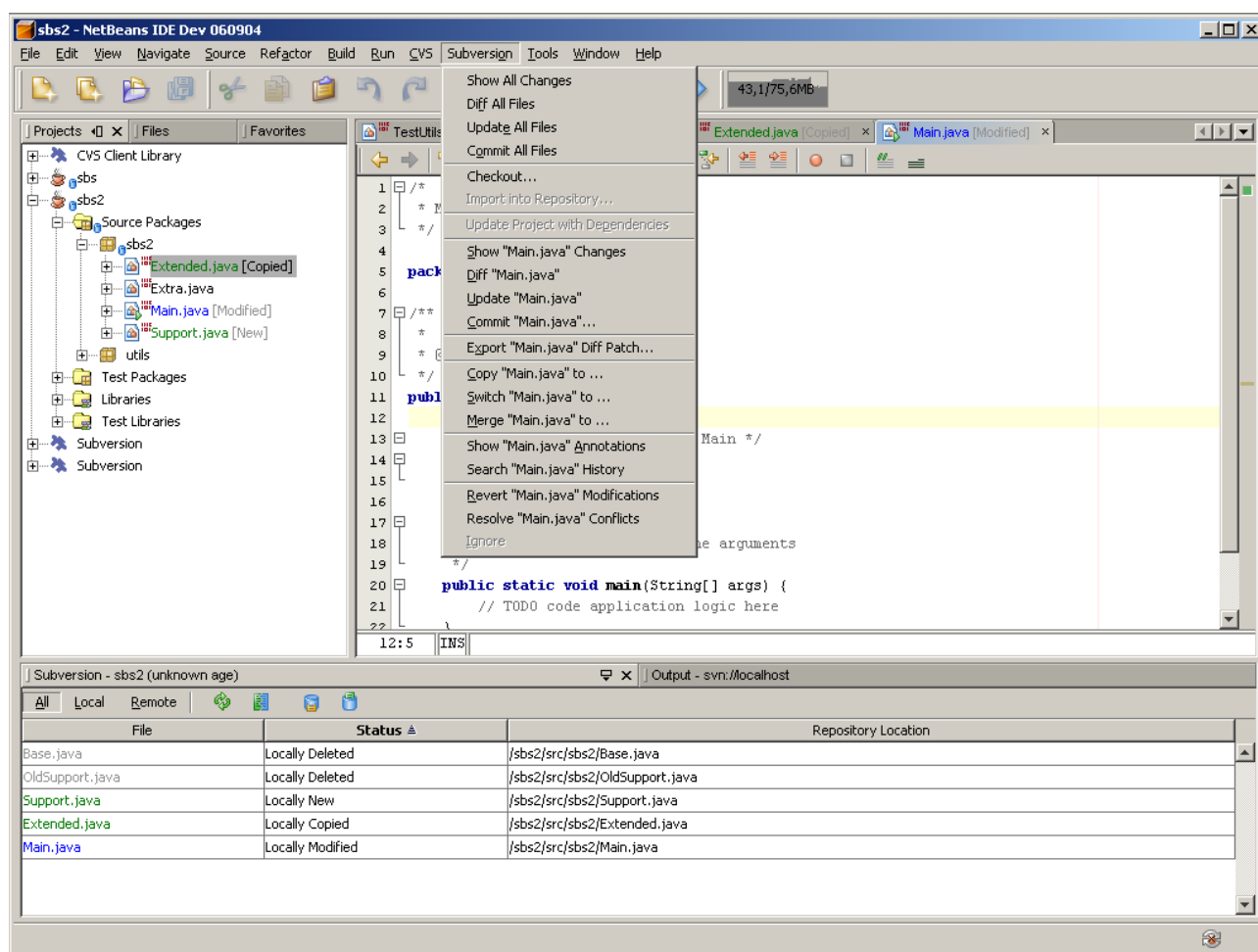
Máis información en <http://subclipse.tigris.org/>

Netbeans:

A versión 5.5 ten no seu *Update Center* o plugin para *subversión*, pero necesita que esté instalado o comando *svn* no sistema (superiora 1.3).

A próxima versión de Netbeans a 6, xa terá incorporado por defecto o soporte para *subversión*.

En Sarge non hai a versión 1.3 de *subversión*, pero poderemos baixalo de fontes como *sarge-backports*



Máis información en <http://subversion.netbeans.org/>

Copias de seguridade

Os nosos repositorios están configurados e en uso. Neste momento, temos que asegurar salvagardar a información que está ahi, para evitar que calquera anomalía ou corrupción do repositorio faga que se perda todo o traballo e o histórico das revisións.

Fixen un par de utilidades sinxelas, que empaquetei para Sarge nun paquete chamado *subversion-backup_0.0.7_i386.deb* que poderemos descargar de <http://www.diazdelaiglesia.com/drupal/?q=node/70>

Baixaremos o paquete *subversion-backup_0.0.7_i386.deb* e como root o instalaremos:

```
dpkg -i subversion-backup_0.0.7_i386.deb
```

As utilidades deste paquete, ubicadas en */opt/utilidades/svn/backup/* son:

- `svn_backup`
Realiza un backup de tódolos repositorios.
- `svn_restore`
A partir dun backup concreto, e indicando o repositorio, fai unha recuperación do repositorio.

svn_backup

O script de *svn_backup*, feito en bash, é moi sinxelo, poderemolo ver na seguinte páxina. Deberemos editar este script e cambiar os seguintes parámetros se é necesario:

- **DIRECTORIO_REPOSITORIOS**
Por defecto ten o valor `/var/local/subversion`
Esta é a ruta onde están os repositorios. Unha ruta habitual e recomendable, pode ser `/var/lib/subversion`.
- **DIRECTORIO_BACKUP**
Por defecto `/var/copias`
Esta é a ruta onde almacenará o backup de cada repositorio.
Debemos ter en conta que esta carpeta vai crecer de forma periódica, no caso de que metamos este script no cron. É recomendable facer unha partición aparte, como no caso da partición de repositorios que fixemos antes.
Ademais, recomendable de forma periódica (diaria, semanal, etc), copiar os backups a outro medio, por máis seguridade.

O contido de svn_backup:

```
#!/bin/bash
#####
#
# Realizacion de backup de subversion
# Daniel Diaz De La Iglesia
# 22/10/2006
#####

DIRECTORIO_REPOSITORIOS="/var/local/subversion"
DIRECTORIO_BACKUP="/var/copias"
DATA=`date "+%Y%m%d%H%M%S"`

echo "SCRIPT de realizaci♦ de backups de repositorios SVN"
echo ""
echo "Considerando como ruta repositorios: ${DIRECTORIO_REPOSITORIOS}"
echo ""
echo ""
for i in $( ls ${DIRECTORIO_REPOSITORIOS} ); do

    echo "Procedendo con repositorio $i"
    echo ""
    REPOSITORIO=${DIRECTORIO_REPOSITORIOS}/${i}
    FICHEIRO_BACKUP=${DIRECTORIO_BACKUP}/${DATA}_backup_svn_${i}.gz

    /usr/bin/svnadmin dump ${REPOSITORIO} | gzip -9 > ${FICHEIRO_BACKUP}

    echo ""
    echo "Realizouse correctamente."
    echo ""
    echo "Se desexa restaurar:"
    echo " Paso1 -> gunzip -c ${FICHEIRO_BACKUP} | svnadmin load ${REPOSITORIO}"
    echo " Paso2 -> chown www-data:www-data ${REPOSITORIO}/* -R"
    echo " "

    echo ""
```

done

svn_restore

Permite recuperar un repositorio concreto.

A este script temos que pasarlle dous parámetros:

- O primeiro parámetro, é a copia de seguridade, por exemplo `/var/copias/20061001_repositorio1.gz`
- O segundo parámetro, é a ruta do repositorio a recuperar, por exemplo `/var/local/subversion/repositorio1`

Datos a modificar no script, si non son os correctos:

- **USUARIO_SVN**
Por defecto é `www-data`
Esté e o usuario que emprega `apache2` pra conectarse co repositorio.
- **GRUPO_SVN**
Por defecto é `www-data`
O grupo que emprega `apache2`.

O contido `restore_svn`:

```
#!/bin/bash
echo " recuperacion repositorio svn"
echo " "
echo " Empleo: restore_svn copia_seguridad.gz ruta_repositorio"
echo ""

USUARIO_SVN=www-data
GRUPO_SVN=www-data
COPIA_SEGURIDAD=$2
RUTA_REPOSITORIO=$1
erro=""

if [ -f ${COPIA_SEGURIDAD} ]
```

```

then
echo "atopado ${COPIA_SEGURIDAD}"

else

erro="Non existe ${COPIA_SEGURIDAD}."

fi

if [ -f ${RUTA_REPOSITORIO} ]

then
echo "Existe ruta repositorio"

else

erro="${erro} Non existe ${RUTA_REPOSITORIO}"
echo "Procedendo a creala"

echo "svnadmin create ${RUTA_REPOSITORIO}"
fi

if [ erro == "" ]
then

echo "gunzip -c ${COPIA_SEGURIDAD} | svnadmin load ${RUTA_REPOSITORIO}"
echo "chown ${USUARIO_SVN}:${GRUPO_SVN} ${RUTA_REPOSITORIO} -R"
else

echo "Errores atopados:"
echo "${erro}"
echo ""

```

```
fi
```

Configurando backups periódicos

Executamos crontab en modo edición para planificar a copia.

```
crontab -e
```

Cada liña é unha tarefa a realizar, e por exemplo, faremos a seguinte planificación:

```
0 6 * * * /opt/utilidades/backup_svn
```

Esta planificación fará que as 6:00, tódolos días, se executará o backup de subversión.