Instalación de Subversion Uno de los sistemas de control de versiones mas utilizados en la actualidad es Subversion. Es un sistema fácil de usar y esta disponible para todas las plataformas populares, como Linux, Mac OSX y Windows.

La instalación depende de la plataforma, pero generalmente hay paquetes disponibles para instalar con el administrador de paquetes propio de cada sistema.

Por ejemplo, en el caso de Ubuntu o Debian Linux, basta ejecutar el siguiente comando:

```
apt-get install subversion
```

En el caso de Red Hat o CentOS, el comando es muy similar:

```
yum install subversion
```

Otras distribuciones de Linux y otros sistemas operativos tienen sus propios mecanismos de instalación, pero no difieren mucho de lo anterior.

Arranque del servidor Una vez instalado Subversion, puede iniciarse el servicio utilizando el comando svnserve, de la siguiente manera:

```
svnserve -d
```

El servicio espera peticiones en el puerto 3690, por lo que si se tiene un firewall debe abrirse dicho puerto para entrada y salida.

Creación de un repositorio Por supuesto, el servicio recién iniciado no tiene ninguna utilidad hasta que es creado un repositorio para servir. Esto se hace con el comando svnadmin, el cual recibe como parámetro el path dentro del sistema de archivos donde se desea crear el repositorio:

```
svnadmin create /path/del/repositorio
```

El path puede ser cualquiera en el sistema, pero hay que asegurarse que los usuarios tengan acceso de lectura por lo menos en todo el path elegido. Al crear el repositorio, Subversion genera una serie de directorios dentro del path elegido. Por ejemplo, si creamos el repositorio en /opt/svn:

```
svnadmin create /opt/svn
cd /opt/svn
ls
conf  db  format  hooks  locks  README.txt
```

El contenido del repositorio se guarda dentro del directorio db, pero por supuesto nunca hay que modificar nada dentro del mismo.

El repositorio que hemos creado tiene un URL que se forma utilizando el protocolo svn con el host donde se encuentra el servicio de svnserve y el path completo al repositorio. En este caso: svn:*localhost/opt/svn.* ***Configuración de un repositorio*** *Por el momento, el único directorio que debe interesarnos dentro del repositorio es conf, pues ahí se guardan los archivos de configuración. El archivo principal de configuración de Subversion es svnserve.conf y contiene las siguientes declaraciones, omitiendo los comentarios: [general] anon-access = read auth-access = write password-db = passwd authz-db = authz realm = My First Repository [sasl] use-sasl = false min-*

encryption = 0 max-encryption = 256 La sección general define primero el tipo de acceso permitido al repositorio. Por defecto, los usuarios anónimos tiene permiso para leer y solo los autenticados pueden escribir. En la mayoría de los ambientes corporativos, es deseable que los usuarios anónimos no tengan ningún acceso, por lo que la declaración correspondiente debe cambiarse por la siguiente: anon-access = write Generalmente este es el único cambio que tiene sentido en los valores de acceso. Las siguientes dos opciones, password-db y authz-db se refieren a nombres de archivos en el mismo directorio que contienen la base de datos de contraseñas y las definiciones de grupos y permisos por path, respectivamente. La opción realm se utiliza para nombrar al repositorio y puede tener cualquier valor deseado. Finalmente, la sección sasl es para definir si se desea utilizar cifrado para las contraseñas. En caso de que el valor de use-sasl sea true, el archivo definido arriba en password-db no se utiliza. El uso de SASL requiere tener instalado el soporte para SASL en el sistema. ***Base de datos de contraseñas*** *El archivo passwd contiene la definición de usuarios y contraseñas del repositorio. Es simplemente un archivo de texto donde se define un usuario con su contraseña en cada linea: [users] juan = secreto En el ejemplo, se define un usuario llamado juan, con la contraseña secreto. Para agregar usuarios simplemente hay que poner una nueva linea con el nombre del usuario y la contraseña, separados por el signo =.* ***Definiciones de grupos y permisos por path*** *El archivo authz contiene las definiciones de grupos y permisos por path. Por defecto contiene solamente ejemplos: [aliases] # joe = /C=XZ/ST=Dessert/L=Snake City/O=Snake, Ltd./OU=Institute/CN=Joe Average [groups] # harry_and_sally = harry,sally # harry_sally_and_joe = harry,sally,&joe # [/foo/bar] # harry = rw # &joe = r # * = # [repository:/baz/fuz] # @harry_and_sally = rw # * = r Dado que el repositorio que recién hemos creado no tiene contenido aun, no podemos configurar otro path mas que la raíz. Para permitir al usuario juan que definimos arriba acceso de lectura y escritura al repositorio, basta agregar las siguientes lineas: [/] juan = rw También es posible agregar grupos de usuarios en la sección groups. Basta poner el nombre de grupo y a continuación una lista de nombres de los que fueron definidos en el archivo passwd, separados por comas.* ***Estructura del repositorio*** *Una vez creado y configurado el repositorio es importante definir su estructura antes de comenzar a importar contenido. Independientemente de la organización de carpetas que se decida adoptar, en un repositorio de Subversion se recomienda tener una carpeta principal por cada proyecto. A su vez, dentro de cada proyecto es usual utilizar una carpeta con la versión oficial del código, llamada trunk, así como carpetas para etiquetar versiones y para realizar pruebas, llamadas tags y branches respectivamente.* ***import - Como importar un proyecto*** *La estructura inicial de directorios puede crearse paso a paso, pero en muchas ocasiones es mejor importarla en un solo paso utilizando ya sea un proyecto ya existente o simplemente una estructura de carpetas. El comando de Subversion para hacer esto se llama svn import. Para importar una estructura de directorios al repositorio que creamos con anterioridad, podemos usar una secuencia de comandos como la que sigue: mkdir proyecto_ejemplo cd proyecto_ejemplo mdkir trunk tags branches cd .. svn import proyecto_ejemplo svn:*localhost/opt/svn/proyecto_ejemplo

```
Adding         proyecto_ejemplo/trunk
Adding         proyecto_ejemplo/branches
Adding         proyecto_ejemplo/tags
```

Committed revision 1 La primera vez que nos conectamos al repositorio, Subversion nos pedirá la contraseña para entrar, asumiendo que nuestro nombre de usuario es el mismo con el que estamos conectados en nuestro sistema. Si esto no es así, basta presionar la tecla enter sin escribir nada y Subersion nos preguntara el nombre de usuario primero.

Antes de hacer el import, Subversion abrirá una ventana del editor defecto del sistema, para que escribamos un mensaje que explique el cambio. Esto debe hacerse en todas las operaciones de escritura al repositorio y es útil ser concisos pero al mismo tiempo informativos al poner el comentario.

Nótese que al final de la operación, Subversion nos informa el numero de versión que se aplica a estos cambios. Cada operación donde se cambia el repositorio aumenta el numero de versión por uno, independientemente de la cantidad de documentos modificados en ella.

ls - Como listar los contenidos del repositorio Para verificar que el import funciono correctamente, podemos pedir un listado del contenido de la nueva carpeta en el repositorio, utilizando el comando svn ls:

```
svn ls svn://localhost/opt/svn/proyecto_ejemplo
branches/
tags/
trunk/
mkdir - Como crear directorios en el repositorio
```

Otra manera de crear la estructura del repositorio es creando las carpetas directamente, utilizando el comando svn mkdir:

```
svn mkdir svn://localhost/opt/svn/otro_ejemplo -m 'nuevo proyecto'
```

Committed revision 2 En este caso, en lugar de esperar a que se nos muestre una ventana del editor, enviamos el mensaje junto con el comando utilizando la opción -m. Esto podemos hacerlo con todos los comandos que escriben en el repositorio en lugar de utilizar el editor.

Comandos básicos de Subversion Una vez que se tiene un proyecto o estructura en el repositorio, la manera de trabajar con Suversion es extraer una copia del proyecto para realizar cambios y subirlos al terminar. Esta copia del proyecto se conoce como copia de trabajo y Subversion puede determinar exactamente que documentos se han agregado o han sido modificados mientras trabajos en ella.

checkout - Como crear una copia de trabajo El proceso de obtener del repositorio una copia del proyecto se conoce como svn checkout. El parámetro que se pasa al comando además del path en el repositorio que queremos copiar es el nombre de la carpeta donde colocaremos la copia:

```
svn co svn://localhost/opt/svn/proyecto_ejemplo proyecto_ejemplo
A    proyecto_ejemplo/trunk
A    proyecto_ejemplo/tags
A    proyecto_ejemplo/branches
```

Checked out revision 2 Los archivos del proyecto quedan guardados en la carpeta proyecto_ejemplo y Subversion nos informa que la versión que ha obtenido es la 2. Una vez que se ha realizado el checkout podemos cambiarnos al directorio del proyecto y comenzar a trabajar.

info - Como obtener información básica del repositorio Al cambiarnos dentro del directorio de la copia de trabajo, Subversion puede reconocer que estamos utilizando un repositorio. En cualquier momento podemos obtener los datos del repositorio donde estamos conectados utilizando el comando svn info:

```
cd proyecto_ejemplo
svn info
Path: .
URL: svn://localhost/opt/svn/proyecto_ejemplo
```

```
Repository Root: svn://localhost/opt/svn
Repository UUID: 073e038a-3ebf-4a60-b88a-b0abaccd7367
Revision: 2
Node Kind: directory
Schedule: normal
Last Changed Author: juan
Last Changed Rev: 2
Last Changed Date: 2010-04-09 00:30:57 -0500 (Fri, 09 Apr 2010)
```

El comando svn info nos devuelve entre otras cosas el URL de donde se extrajo el directorio donde estamos trabajando (URL), el URL de la raíz del repositorio (Repository Root), la revision o versión al momento del checkout (Revision), el autor del ultimo cambio (Last Changed Author) y la fecha de ese cambio (Last Changed Date).

status - Como conocer el estado de nuestras modificaciones Una vez que comenzamos a hacer modificaciones dentro del directorio del proyecto, Subversion lleva la cuenta de los cambios que hemos realizado y en cualquier momento podemos consultarlos:

```
cd trunk
echo "La capital de Francia es Tokio" > info.txt
svn status
?      info.txt
```

En el ejemplo anterior, creamos un archivo de texto con una sola linea, llamado info.txt. Una vez creado el archivo, utilizamos el comando svn status para mostrar como Subversion ha detectado que existe un nuevo archivo en el directorio. El signo de interrogación que aparece antes del nombre, significa que el archivo en cuestión no esta bajo control de versiones y Subversion lo desconoce.

add - Como agregar documentos al proyecto Para agregar ese archivo al proyecto, utilizamos el comando svn add:

```
svn add info.txt
A      info.txt
```

Subversion agrega el archivo info.txt a los que se encuentran bajo control de versiones, por lo que el status muestra ahora la letra A junto al nombre. Es importante hacer notar que este comando únicamente tiene efecto en nuestra copia de trabajo y no sube de inmediato el archivo al repositorio.

El comando svn add no esta limitado a agregar un solo archivo, por supuesto. Es posible incluir como parámetro cualquier cantidad de archivos. Si se agrega un directorio, todos los archivos contenidos en el serán agregados recursivamente al proyecto.

commit - Como guardar nuestros cambios en el repositorio Podemos hacer todos los cambios que necesitemos en nuestra copia de trabajo, si bien se recomienda subir la información al menos al final de cada sesión de trabajo y de preferencia cada vez que terminemos una tarea especifica de edición. La razón es que mientras mas tiempo pasemos sin subir los cambios, mas difícil puede resultar integrarlos con otros cambios al repositorio, especialmente si muchas personas tienen acceso al mismo.

A la operación de subir los cambios al repositorio se le llama commit. Una vez que hemos terminado nuestra sesión de trabajo, utilizamos ese comando para guardarlos en el repositorio:

```
svn commit -m 'se agrego archivo info'
Adding         trunk/info.txt
Transmitting file data .
Committed revision 3.
```

El comando svn commit guarda todos los cambios realizados desde que inicio la sesión. En caso de no querer guardar todo, es posible especificar los archivos que deben subirse.

Ciclo de trabajo con Subversion Para utilizar Subversion eficientemente, la rutina de trabajo que utilizamos debe cambiar un poco para incluir los momentos en que actualizamos o subimos archivos. Ademas, a lo largo del tiempo, el repositorio ira evolucionando y encontraremos necesidad de revisar cambios anteriores y, si trabajamos con otras personas, de resolver conflictos.

Subversion tiene varios comandos para apoyarnos en ese ciclo básico de trabajo. En esta sección conoceremos algunos de los mas importantes.

update - Como trabajar con la versión mas reciente Lo primero que debemos hacer diariamente al iniciar una sesión de trabajo, es actualizar nuestra copia de trabajo del repositorio, para asegurarnos de trabajar con la versión mas reciente de nuestros documentos. El comando para hacer esto se llama svn update:

```
svn update
At revision 3.
```

El comando actualiza los archivos que han cambiado, integrando al mismo tiempo nuestros cambios y nos muestra el status de lo que ha sido modificado, junto con la versión a la que nos hemos actualizado. En el ejemplo anterior no hubo cambios que integrar.

Ahora supongamos que alguien ha agregado un titulo al archivo info.txt y ha subido sus cambios. Si hacemos un update ahora, veremos la diferencia:

```
svn update
U    trunk/info.txt
Updated to revision 4.
```

En este caso, Subversion nos muestra el status U, que significa que un documento existente fue modificado.

log - Como revisar la historia de un documento Como el archivo info.txt ha sido modificado, quizá deseamos saber quien realizo la modificación y cuando. Subversion ofrece el comando svn log para poder conocer la historia de commits de un archivo:

```
svn log info.txt
------------------------------------------------------------------------
r4 | predro | 2010-04-09 23:02:29 -0500 (Fri, 09 Apr 2010) | 1 line
```

```
se agrego titulo
------------------------------------------------------------------------
r3 | juan | 2010-04-09 22:41:55 -0500 (Fri, 09 Apr 2010) | 1 line
```

```
se agrego archivo info
```

```
------------------------------------------------------------------------
```

El comando nos muestra revisión, autor, fecha y comentario por cada cambio que se ha hecho al archivo. En este caso, podemos ver que el usuario pedro agrego un titulo a nuestro archivo.

diff - Como revisar los cambios que hemos realizado en una sesión Si además de conocer al autor del cambio y su comentario queremos saber exactamente que texto ha cambiado en nuestro archivo, podemos utilizar el comando svn diff de Subversion para hacerlo:

```
svn diff -r3:4 info.txt
Index: info.txt
===================================================================
--- info.txt        (revision 3)
+++ info.txt        (revision 4)
@@ -1 +1,3 @@
+Sabia usted que...
+
La capital de Francia es Tokio
```

El comando svn diff acepta el parámetro -r para especificar los números de versiones entre los que queremos conocer la diferencia. En este caso necesitamos conocer los cambios entre las revisiones 3 y 4, por lo que pasamos esos números. Se puede omitir el parámetro -r y entonces Subversion nos dará las diferencias entre el estado actual del archivo y el estado que tenia la ultima vez que actualizamos el repositorio.

Lo que nos muestra el comando son las lineas que difieren entre una versión y otra. Las lineas que tienen el símbolo + al lado izquierdo son las lineas que fueron agregadas entre la primera y la segunda versión especificadas. En caso de que se hayan eliminado algunas lineas, estas tendrán el símbolo - a su lado izquierdo.

blame - Como saber quien modifico una parte especifica de un documento Todavía podemos averiguar mas información sobre la historia de cambios del archivo. El comando svn blame nos muestra la ultima revisión en que ha cambiado cada linea del archivo, junto con el nombre del autor del cambio:

```
svn blame info.txt
4      pedro Sabia usted que...
4      pedro
3      juan  La capital de Francia es Tokio
```

cat - Como ver el contenido de versiones anteriores de un documento Subversion nos permite también conocer el contenido completo de algún archivo en el momento en que determinada revisión fue subida al repositorio. Por ejemplo, para ver el contenido del archivo info.txt en la revisión 3:

```
svn cat -r3 info.txt
La capital de Francia es Tokio
```

revert - Como regresar un documento a su estado inicial en una sesión En ocasiones, después de haber realizado algunos cambios en un archivo, nos damos cuenta de que no queremos conservarlos, sino que deseamos volver a la versión original del mismo. El comando svn revert anula

cualquier cambio realizado a un archivo en la sesión actual, volviendo al estado que tenia al momento de actualizar el repositorio por ultima vez:

```
svn revert info.txt
Reverted info.txt
```

From:
https://wiki.merkatu.info/ - **Wiki de merkatu**

Permanent link:
https://wiki.merkatu.info/manual_subversion?rev=1462437511

Last update: **2017/03/27 17:43**