



Manual uso de subversion

Programación Paralela

José María Cecilia Canales

Grado en Informática

1.- Introducción.

Realizar una aplicación software suele ser tarea de un grupo de programadores, que con un conjunto de requisitos software bien definido, pueden programar una solución modular que permita trabajar de forma independiente a cada uno de ellos, teniendo al finalizar, la aplicación perfectamente cohesionada y en producción.

Podemos definir *subversion* como un sistema de control de versiones, donde a través de un repositorio es posible compartir información. Un repositorio es un sitio en la red donde se guarda información, tanto en forma de bases de datos, como en archivos independientes, pudiendo acceder los clientes simultáneamente, siguiendo una arquitectura cliente-servidor.

El trabajo colaborativo es perfectamente asumible con este sistema, cada usuario crea una copia local de los archivos, reflejo del estado actual de los mismos, o en alguna de las confirmaciones anteriores que hayan sufrido. Con esta copia, todos los usuarios pueden modificar de forma independiente estos archivos, y al finalizar, cada uno puede confirmar sus modificaciones.

Este sistema permite realizar un seguimiento sobre directorios y archivos, permitiendo acceder a una versión anterior de los mismos. Esta característica permite poder volver a versiones estables de nuestros códigos, y realizar un seguimiento sobre los archivos modificados.

En condiciones de concurrencia, el sistema informa de posibles conflictos a la hora de confirmar las actualizaciones de los archivos, informándonos que vamos a realizar una actualización de ciertos archivos sin tener sus últimas modificaciones realizadas por otros usuarios, pero al final todo, la confirmación de modificaciones depende de cada usuario.

2.- Instalación.

Primero debemos descargar los paquetes para que funcione correctamente desde la página de descarga que figura a continuación, seleccionando correctamente el sistema operativo con el que estemos trabajando.

<http://subversion.apache.org/packages.html>

También podemos optar directamente porque el sistema operativo descargue subversion y la instale, desde sus propios repositorios. Por ejemplo en Ubuntu debemos usar:

\$ sudo apt-get install subversion libapache2-svn (con permisos de administrador)

3.- Componentes.

Tenemos un conjunto de órdenes en la línea de comandos para configurar subversion. Brevemente a continuación se describen los seis más utilizados.

svnadmin. Herramienta con la que podemos crear, configurar o reparar un repositorio.

svnversion. Con esta orden obtenemos la versión actual de los archivos con los que estamos trabajando en la copia local.

svnlook. Herramienta que permite inspeccionar el estado del repositorio donde se encuentra alojado nuestro repositorio.

svnservice. Con esta herramienta nos permite usar el repositorio a través de la red.

svnsync. Permite hacer espejos de nuestro repositorio (*mirros*) a través de la red.

svn. Es el programa cliente de línea de comandos.

Cada uno de estos comandos, tiene un conjunto de opciones asociadas, y que de forma particular y dependiendo de la configuración que queramos conocer o aplicar a nuestro repositorio, debemos consultar en su ayuda con **--help**.

4.- Manejo de subversion.

Como toda herramienta de trabajo colaborativo, tenemos que tener presente que más de un usuario puede tener acceso al mismo.

4.1 Descarga de “Working Copy” o Copia para trabajar.

Cuando descarguemos la revisión de los archivos con los que queremos trabajar, generaremos una copia local. La orden para realizar esta descarga con la última versión de los archivos es la siguiente:

svn checkout svn+ssh://usuario@servidor/alumno@servidor/directorio/nombre_repositorio directorio_destino

Si no especifica el último parámetro, el directorio con los ficheros del proyecto se creará con el nombre **svnroot** en el directorio actual. También puede especificar «**co**» en lugar de **checkout**.

4.2 Trabajar con “Working Copy”.

Una vez tengamos la copia local, se pueden realizar cuantas modificaciones se desee, lo que incluye crear/borrar ficheros/directorios, cambiar el contenido de los mismos, etc. Sin embargo, se debe tener en cuenta algunos detalles si quiere que subversion controle todos los cambios que vaya a realizar. Las operaciones típicas de añadir/borrar ficheros, modificar ficheros, etc, en subversion se realizan de la siguiente manera:

Modificar un fichero existente. Simplemente, se modifica, no necesita hacer nada especial.

Añadir un fichero/directorio. Crear o copiar un fichero y decirle a subversion que controle los cambios sobre el mismo. Para ello, hay que ejecutar:

`svn add fichero/directorio`

Crear y añadir a la vez un directorio con la orden:

`svn mkdir directorio`

Borrar un fichero o directorio, hay que utilizar la orden:

`svn delete fichero/directorio`

Esta orden, además, borrará físicamente el fichero o directorio especificado. En el caso de un directorio, se borrará todo lo que haya en él.

Renombrar un fichero o directorio, la siguiente orden:

`svn rename fichero/directorio nuevofichero/nuevodirectorio`

Tras ejecutar la orden, se puede comprobar que el fichero o directorio efectivamente ha cambiado de nombre.

En las operaciones que acabamos de indicar hay que darse cuenta de que **NO** necesitamos especificar dónde está nuestro repositorio. Esto es así siempre que las operaciones anteriores se realicen estando en **ALGUNO** de los directorios que hemos copiado desde el repositorio. Esta es *la manera más cómoda de usar* subversion.

4.3 Confirmación de cambios.

Los cambios realizados son **LOCALES**, o dicho de otro modo, no son visibles al resto de usuarios del repositorio. Para que nuestros cambios sean definitivos y, por tanto, **PÚBLICOS**, debemos ejecutar la orden:

`svn commit -m "comentario"`

o

`svn commit`

En ambas órdenes podemos sustituir **commit** por **ci**. Cuando los cambios se hacen permanentes se puede incluir un pequeño comentario que explique qué es lo que hemos cambiado. En el primer caso el comentario se indica con la opción **-m**, mientras que en el segundo se abre un editor de textos (el que tengamos configurado por defecto en las variables de entorno **VISUAL** o **EDITOR**) en el que podremos detallar los cambios realizados.

El repositorio se puede configurar para que se envíe un correo electrónico cada vez que se modifique. Hay que tener en cuenta que, cuando el repositorio se modifica, la copia del proyecto que se encuentra en el ordenador personal puede quedar **OBSOLETA**. Para actualizarla y, por tanto, recibir todos los cambios que se han producido desde la última actualización, se debe ejecutar la siguiente orden:

svn update

Por último, si antes de enviar los cambios al repositorio deseas comprobar qué cosas ha cambiado, puede ejecutar la orden:

svn status

Ten en cuenta que las órdenes **commit**, **update** y **status** actúan recursivamente a partir del directorio **ACTUAL**. Si se quiere que actúen en todo el proyecto, nos debemos colocar previamente en el directorio **RAÍZ** del mismo.

5.- Más información.

Podemos encontrar un libro sobre subversion en:

<http://svnbook.red-bean.com/>

Con respecto al entorno WINDOWS, hay clientes para Windows pero no servidores. En cualquier caso, lo que se necesita es un cliente. Puede encontrar un cliente para Windows con interfaz de línea de órdenes en:

http://subversion.tigris.org/project_packages.html

Con interfaz gráfica para Windows lo podemos encontrar en:

<http://tortoisesvn.tigris.org/>

En internet podemos encontrar muchas más páginas, blogs y artículos de investigación que realicen estudios y aportaciones sobre subversion, aunque en este manual se trata de aprender las bases para comenzar a manejar el entorno y su modo de trabajo.