



# Control de Versiones con Subversion y TortoiseSVN

Juan Luis Serradilla ([juanlu@um.es](mailto:juanlu@um.es))  
Sección de Metodología, Normalización y Calidad del Software  
ATICA  
Universidad de Murcia



# ¿Qué es el “control de versiones”?

- Capacidad de recordar todos los cambios que se hacen tanto en la estructura de directorios como en el contenido de los ficheros.
- Cuando más de una persona trabaja con los mismos archivos – y aún cuando es una sola persona – resulta imprescindible mantener cierto control sobre los cambios que se realizan: quién, cuándo, qué,.... .
- Si los cambios realizados por dos personas son incompatibles y es necesario tomar una decisión sobre la forma definitiva del archivo.
- Nos damos cuenta que los últimos cambios realizados no siguen el camino apropiado y debemos volver atrás.



# ¿Qué es “Subversion”?

- Herramienta de código abierto, multiplataforma (Win32, Linux, Mac, etc), para el control de versiones de ficheros electrónicos, como son el software o la documentación.
- Se basa en un repositorio central que actúa como un servidor de ficheros, con la capacidad de recordar todos los cambios que se hacen tanto en sus directorios como en sus ficheros.
- El repositorio incrementa un número global de revisión con cada conjunto de cambios enviados (commit) al mismo. Es posible copiar y renombrar ficheros; crear una rama del proyecto es tan fácil como copiar un directorio. También se puede pedir una salida con las diferencias entre dos revisiones arbitrarias, o que recupere algún sub-árbol de la revisión N.
- <http://subversion.tigris.org>



## ¿Qué es “TortoiseSVN”?

- TortoiseSVN es un cliente gratuito de código abierto para el sistema de control de versiones Subversion.
- Integrado en la shell de Windows, por ejemplo el “explorador”. Y ni siquiera está obligado a usar el Explorador de Windows. Los menús contextuales de TortoiseSVN también funcionan en otros administradores de archivos, y en la ventana Fichero/Abrir que es común a la mayoría de aplicaciones estándar de Windows.
- Todos los comandos de Subversion están disponibles desde el menú contextual del explorador. TortoiseSVN añade su propio submenú allí.
- <http://tortoisesvn.tigris.org>



# ¿Qué necesitamos para empezar?

- Un cliente, como puede ser TortoiseSVN, o el propio que incorpora Subversion.
  - La versión más reciente de TortoiseSVN es la 1.2, con capacidad para bloquear.  
<http://tortoisesvn.tigris.org/download.html>
  - La versión 1.2 de Subversion:  
[http://subversion.tigris.org/project\\_packages.htn](http://subversion.tigris.org/project_packages.htn)
- Un repositorio, que podemos crear en nuestro servidor SVN, si disponemos de uno; o bien, en local con el propio cliente TortoiseSVN.
  - TortoiseSVN -> Crear repositorio



## ¿Necesito un servidor SVN?

- Sólo en el caso de que me interese mantener una gestión de usuarios con sus controles de accesos, y permitir el acceso mediante HTTP (webdav).
- Si se trata de un repositorio privado, puedo crearlo en mi propio PC, sin necesidad de un servidor SVN. En este caso, puedo hacerlo directamente con TortoiseSVN, incluso en una unidad de red.
- Si finalmente opto por la opción del servidor SVN, en la máquina destinada a tal fin, tendré que instalar un servidor web Apache (en el caso de que quiera habilitar el acceso mediante HTTP).
- ¿Puedo montar un servidor SVN sin Apache? Sí, con el servidor nativo “svnserve”, sin acceso HTTP.



# Crear el Repositorio con TortoiseSVN

- Seleccionar directorio donde crear el repositorio.
- ¿Cómo accedo al cliente TortoiseSVN? A través del “explorador de archivos”, desde el menú contextual que aparece pulsando el botón derecho.
- Opción “Crear repositorio aquí”. Inicialmente vacío.
- ¿Cómo cargo datos en el repositorio? Me sitúo en el directorio que quiero cargar, y con la opción “Importar” de TortoiseSVN, indico en qué repositorio lo quiero importar.
- ¿Cómo puedo ver lo que hay en el repositorio? Usaré la opción “Navegador”. El directorio donde he creado el repositorio no debo tocarlo.



# Modificar los datos del Repositorio

- El repositorio SVN no se puede “tocar” directamente, hay que hacerlo usando un cliente, como TortoiseSVN.
- Si quiero modificar el contenido de un directorio del repositorio, primero debo crear un “directorio de trabajo” en mi PC, donde descargaré los datos con los que voy a trabajar.
- ¿Cómo se crea un directorio de trabajo? Creando un directorio más desde el “explorador de archivos”, y usando el comando “Obtener” de TortoiseSVN. Indicaré el directorio local donde está el repositorio, o la URL, en el caso de tener un repositorio accesible por HTTP.

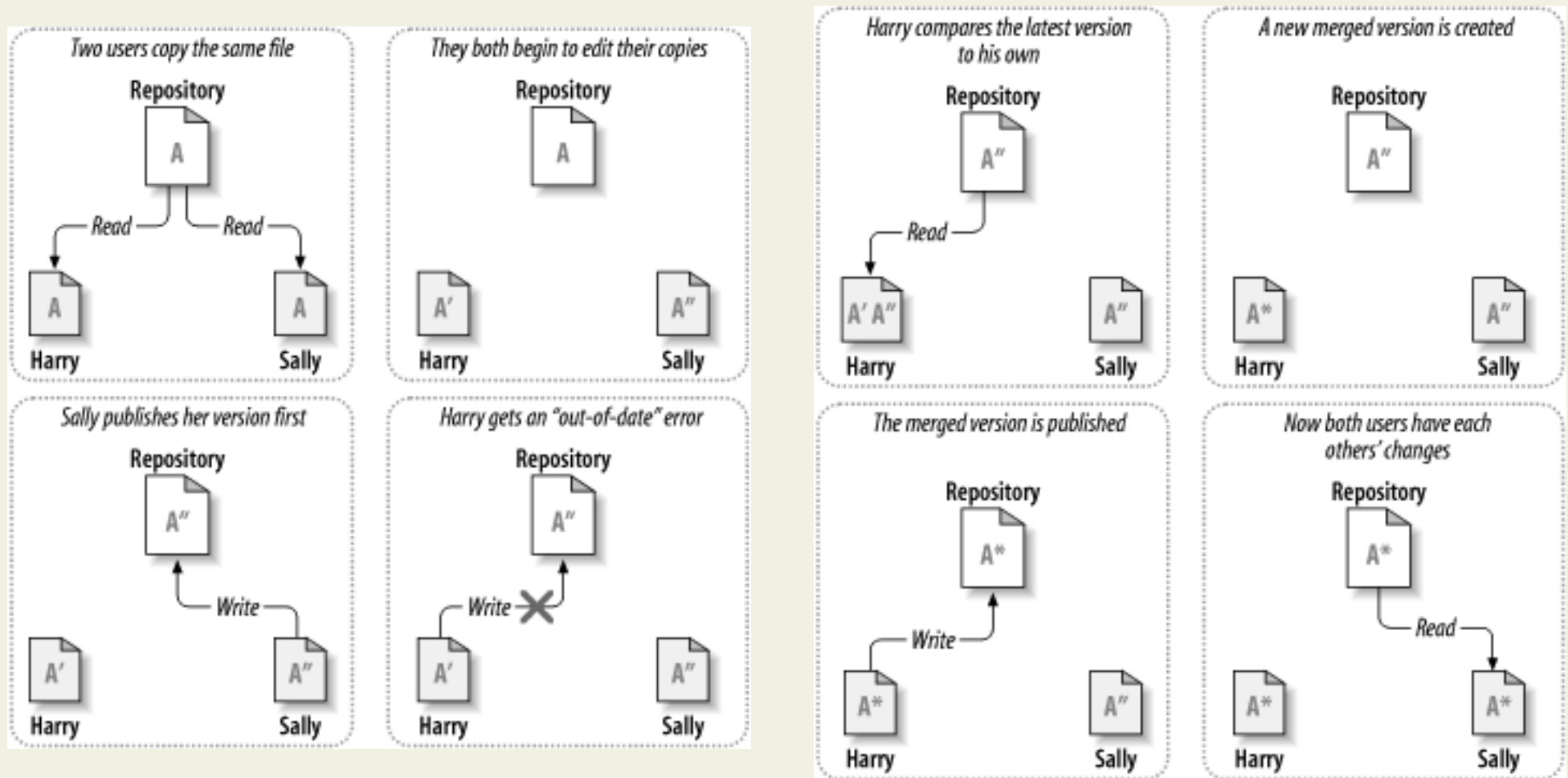




# Modificar los datos sin bloquearlos

- Modelo copiar-modificar-fundir. No bloquea. Mejora la productividad al tener los ficheros siempre accesibles.
- Si dos usuarios acceden al mismo fichero y lo modifican, el primero que lo actualice no tendrá problemas para dejarlo en el repositorio; pero el segundo ya no podrá hacerlo, y SVN le avisará de que hay un conflicto y el fichero ha sido modificado, dándole la opción de revisar las tres versiones (la original, la suya y la del otro) y resolver el conflicto.
- Si los ficheros en conflicto son de texto, SVN puede ayudarnos a mezclar los cambios.
- El proceso a seguir será:
  - “Actualizar” directorio de trabajo
  - Modificar datos
  - “Validar” datos enviando cambios al repositorio

# Modificar los datos sin bloquearlos

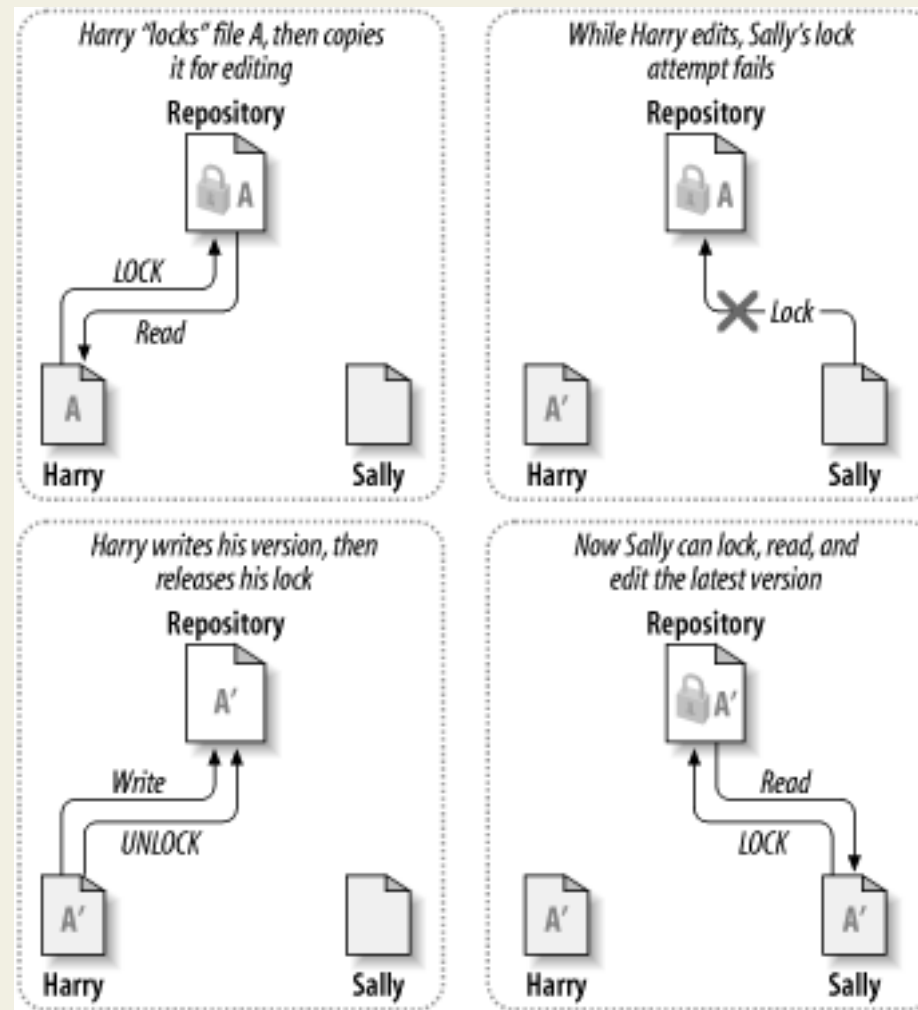




# Modificar los datos bloqueándolos

- Modelo bloquear-modificar-desbloquear. Bloquea. Asegura que nadie más puede modificar los datos.
- Si dos usuarios acceden al mismo fichero para modificarlo, el primero que lo ha bloqueado no tendrá problemas para hacerlo; pero el segundo ya no podrá hacer el bloqueo, y SVN le avisará de que el fichero en cuestión ya está bloqueado; pudiendo verlo con la opción “Navegador”.
- El proceso a seguir será:
  - Actualizar directorio de trabajo
  - Bloquear datos
  - Modificar datos
  - Validar datos enviando cambios al repositorio, y quitando el bloqueo

# Modificar los datos bloqueándolos





# Interacción con el repositorio

- Crear directorio de trabajo y cargarlo con “Obtener”.
- Crear fichero y/o directorios, añadirlos al “control de versiones” con “Añadir”, y subirlos con “Validar”.
- Antes de modificar un fichero, pensemos si necesitamos bloquearlo. Una vez hechos los cambios, lo podemos subir con “Validar”.
- Si queremos borrar ficheros/directorios, hacerlo con “Eliminar”, y subiremos los cambios con “Validar”.
- Antes de “Validar” los cambios, podemos hacer todas las operaciones de añadir, borrar y modificar que queramos. Al “Validar”, se subirán todos los cambios al repositorio en una sola transacción.



# Crear “directorio de trabajo”

- Crear un directorio “normal”.
- Usar la operación “Obtener” para vincularlo con un repositorio SVN, o con una rama del mismo. Se bajarán los datos del repositorio (si los hay).
  - Sólo se hace una vez. Esta operación lo convierte en “directorio de trabajo”.
  - Crea un directorio oculto “.svn” en cada subdirectorio.
- Podemos crear tantos directorios de trabajo como queramos.
- Lo podemos borrar cuando queramos, pues los datos están en el repositorio y podemos crear uno nuevo (iii ojo con los datos que no hayamos subido !!!).



# Actualizar el directorio de trabajo

- Para que los cambios que hagan otros usuarios al repositorio, se reflejen en nuestro directorio de trabajo, usaremos la opción “Actualizar”.
- Debemos “Actualizar” nuestro directorio de trabajo siempre antes de hacer cualquier cambio, para asegurarnos que estamos modificando la última versión del repositorio.
- Al “Actualizar” nuestro directorio de trabajo, se harán todos los cambios necesarios (añadir/borrar/modificar ficheros/directorios) para que los datos sean idénticos a los del repositorio (en ese momento).



# Subir cambios al repositorio

- Una vez que hayamos hecho los cambios oportunos en nuestro directorio de trabajo, los podemos subir al repositorio con la opción “Validar”.
- No olvidemos rellenar el campo “comentario”, ya que cuando consultemos el “log” con los cambios veremos estos “comentarios”, tal cual.
- Con cada operación de “validación” (subir cambios al repositorio), se incrementará en uno el contador de revisiones (que inicialmente está a cero).
- Los cambios se envían en forma de “transacciones”, de forma que o se validan todos o ninguno.
- Si al “validar” se producen “conflictos” (alguien ha subido cambios de un mismo fichero antes que nosotros), nos avisará (abortando la transacción).





# Resolver conflictos

- Si al “Validar” nos encontramos con “conflictos”, procederemos a “Actualizar” el directorio de trabajo, de forma que aparecerán, al menos, dos ficheros nuevos: uno con los cambios que han subido antes que nosotros, y otro con el estado del fichero antes de iniciar nuestros cambios.
- Con la opción “Editar conflictos” podemos incorporar los cambios “del otro” en nuestro fichero.
- Finalmente con la operación “Revolver conflicto”, validaremos nuestro fichero como “bueno”, de forma que podremos “Validar” los cambios. Ojo, porque podemos decirle que hemos resuelto el conflicto, sin haberlo hecho realmente.
- Para evitar conflictos tenemos la opción de “Bloquear” un fichero, antes de iniciar el proceso de cambios.



# Bloquear para evitar conflictos

- La mejor forma de evitar conflictos es tomar la costumbre de “Bloquear” siempre un fichero antes de modificarlo.
- Si el fichero ya está bloqueado no nos dejará “Bloquear” y nos dirá quién lo tiene.
- Con la opción “Navegador” también podremos ver quién ha puesto un bloqueo.



# Exportar datos

- La opción “Exportar” me permite volcar el contenido del repositorio (o un subdirectorio del mismo) en un directorio cualquiera de mi máquina (sin generar los “.svn”).
- Util para generar ficheros “entregables”: ejecutables de una aplicación, documentación en PDF, etc. Previamente he diseñado la estructura del repositorio para que los “entregables” estén en un subdirectorio independiente.



## Crear etiquetas (tags)

- Una etiqueta para SVN es una copia de un directorio, que se hace con el objetivo de obtener una "foto" del mismo, y que no tenemos intención de modificar. Opción “Rama/Etiqueta”.
- Es útil para crear versiones. Por ejemplo, si tengo un repositorio con dos directorios en la raíz, principal y versiones, y bajo versiones quiero ir creando directorios con las diferentes versiones que se vayan publicando, puedo crear una etiqueta /versiones/v1.0 que será una copia de /principal en un determinado instante. De esta forma podría tener un directorio /versiones/v1.0/bin con los ejecutables de la versión 1.0; que ser una copia (en un determinado instante) de /principal/bin.



## Crear ramas (branches)

- Una rama de desarrollo SVN, también es una copia de un directorio, pero en este caso nuestra intención es modificarla, para conseguir un producto final diferente y alternativo al original.
- Muy útiles para personalizar una aplicación para diferentes clientes; por ejemplo, puedo tener un directorio ramas en la raíz de mi repositorio, y crear una rama en /ramas/cliente1 que sea una copia de /principal; con la intención de personalizar mi aplicación para “cliente1”. Así tendré un directorio /ramas/cliente1/bin, con los ejecutables de la aplicación para “cliente1”, que será una copia (modificable) de /principal/bin.



# Estructura del Repositorio

- Separar:
  - Línea principal
    - Fuentes/Originales
    - Ejecutables/Entregables
  - Versiones
  - Ramas
- Objetivos:
  - Generar fácilmente directorio con ejecutables/entregables
  - Poder hacerlo con cualquier versión o rama.

```
/principal  
  ./fuentes  
  ./bin  
/versiones  
  ./v1.0  
    ./fuentes  
    ./bin  
  ./v1.1  
    ./fuentes  
    ./bin  
/ramas  
  ./r1  
    ./fuentes  
    ./bin  
  ./r2  
    ./fuentes  
    ./bin
```



# Errores comunes

- El error más habitual se produce al copiar en nuestro directorio de trabajo un directorio que pertenece a otro directorio de trabajo (o al mismo).
  - ¿Por qué no debemos hacer ésto? Porque cada subdirectorio de un directorio de trabajo incluye un directorio oculto (".svn") que contiene la información de sincronización con el repositorio que usa TortoiseSVN. Al copiar el directorio, copiamos también el ".svn" y, al hacerlo en una ubicación diferente a la original, los datos de sincronización ya no se corresponden con los originales.
  - ¿Cómo hacerlo bien? Si necesitamos hacer ésto, después de copiar el directorio, nos aseguraremos de borrar los ".svn". Posteriormente, si lo queremos incluir en el control de versiones, usaremos la opción "Añadir".
  - iii Ojo !!! no debemos tocar los directorios ".svn".



# Enlaces

- ATICA (Presentación y Normativa relacionada)  
<http://www.um.es/atica/documentos/PREsubversion.pdf>  
<http://www.um.es/atica/documentos/NORPuestaEnLineaAplicacionesWeb.pdf>
- Subversion:  
<http://subversion.tigris.org>
- Manual de Subversion en castellano:  
<http://svnbook.red-bean.com/nightly/es/index.html>
- TortoiseSVN:  
<http://tortoisesvn.tigris.org>
- Manual de TortoiseSVN en castellano:  
[http://tortoisesvn.sourceforge.net/docs/nightly/TortoiseSVN\\_es/index.html](http://tortoisesvn.sourceforge.net/docs/nightly/TortoiseSVN_es/index.html)
- Subclipse (plugin de SVN para el IDE Eclipse):  
<http://subclipse.tigris.org/#subclipse>