



Control de versiones con Subversion

Martín Gaitán y Pablo Martínez
FCEFyN, Universidad Nacional de Córdoba
Junio de 2007

- ◇ Qué es Control de Versiones
- ◇ Qué es Subversion
- ◇ Diferencias con CVS, características y arquitectura
- ◇ Breve historia
- ◇ Vocabulario básico
- ◇ Infografía del proceso ante una 'modificación'
- ◇ Demostraciones de diferentes clientes
- ◇ Recursos
- ◇ Preguntas y comentarios finales



¿Qué es Control de Versiones? 1/2

- ◇ Gestión del desarrollo de cada elemento de un proyecto a lo largo del tiempo
- ◇ Proporciona:
 - Mecanismo de almacenaje de cada elemento que deba gestionarse (archivos de código, imágenes, documentación...)
 - Posibilidad de añadir, modificar, mover, borrar...
 - Historial de las acciones realizadas con cada elemento pudiendo volver a un estado anterior
 - Otros: generación de informes de cambios, informes de estado, marcado con nombre identificativo, etc.
- ◇ Se utiliza un repositorio, donde se almacena la información de todo el desarrollo



¿Qué es Control de Versiones? 2/2

- ◇ Útil para trabajar individualmente o en grupo
- ◇ Servidor local o remoto
- ◇ Permite desarrollos colaborativos, incluso concurrentemente!
- ◇ Todo equipo profesional de desarrollo de software lo utiliza



¿Qué es Subversion?

- ◇ Un software para control de versiones
- ◇ Fácil de usar
- ◇ Diseño simple, potente, seguro
- ◇ Software Libre y gratuito
 - Licencia APACHE/BSD
- ◇ Tanto clientes y servidor, funcionan en muchos S.O.
 - Incluidos GNU/Linux, Windows, Mac OS, etc.

◇ Envíos atómicos

- Todos los cambios, o ninguno.
- Aporta seguridad en la integridad de los datos

◇ Versionado de directorios

- Se lleva un control de cambios a través del tiempo del directorio.
- Sistema de archivos “virtual ” que sigue los cambios sobre árboles de directorios completos a través del tiempo

◇ Verdadero historial de versiones

- Se puede añadir, borrar, copiar, y renombrar archivos y directorios.
- Cada fichero nuevo añadido comienza con un historial nuevo, limpio y completamente propio.

◇ Elección de las capas de red

- Posee abstracción del acceso al repositorio
- Como módulo de Apache:
 - Estabilidad e interoperabilidad
 - Autenticación, autorización, compresión de la conexión, etc.
 - Puertos estándar (administradores de red agradecidos)
- Como servidor independiente (svnserve):
 - Muy ligero, habla un protocolo propio.
 - Puede ser encaminado fácilmente a través de un túnel SSH.

◇ Manipulación consistente de datos

- Encuentra diferencias usando un algoritmo en modo binario
- Funciona idénticamente con texto o archivos binarios.
- Todos los tipos de archivos se guardan comprimidos en el repo.

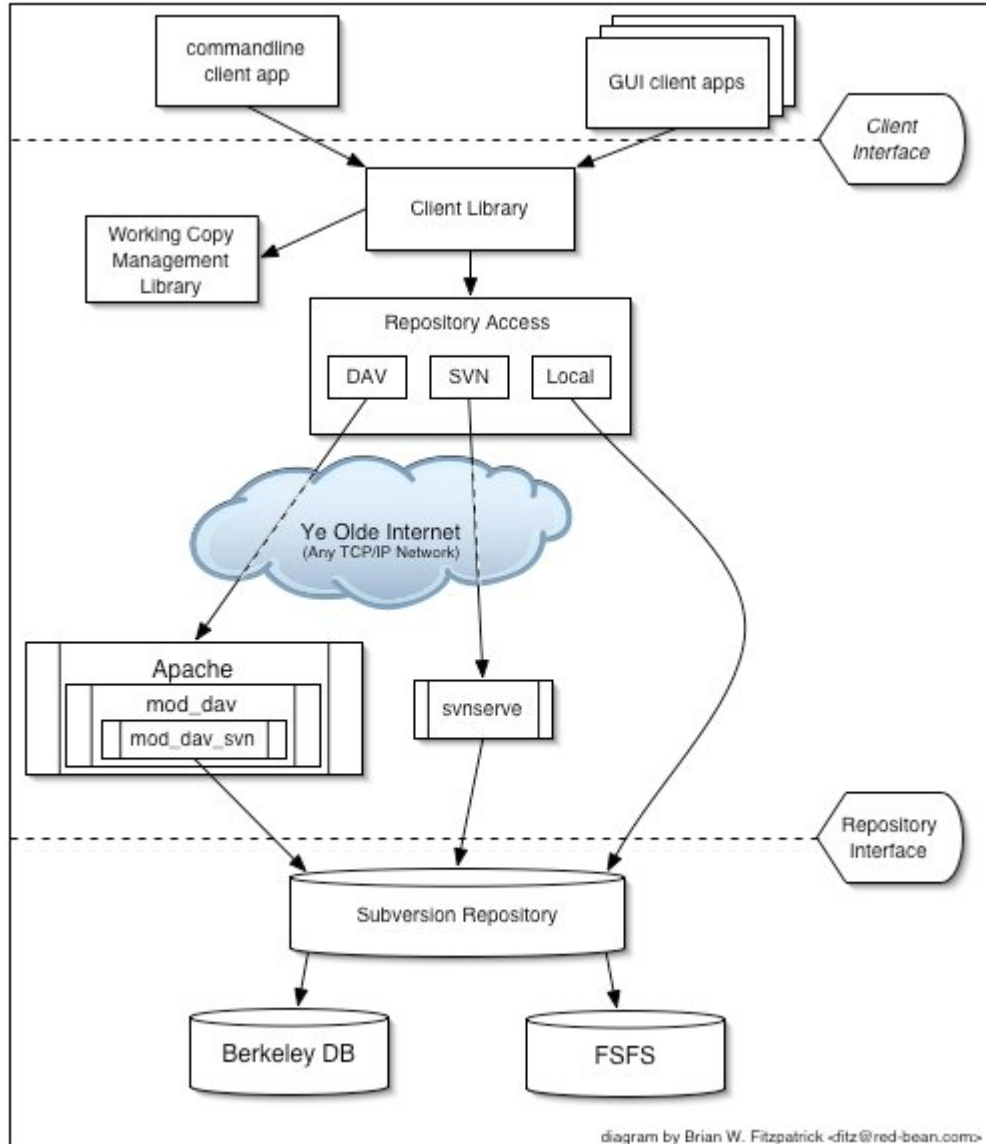
◇ Ramificación y etiquetado eficientes

- El costo de crear una nuevas ramas (branch) o etiquetas (tag) no es proporcional al tamaño del proyecto
- Se usa un mecanismo similar al 'enlace duro' (“acceso directo”)

◇ Totalmente reutilizable

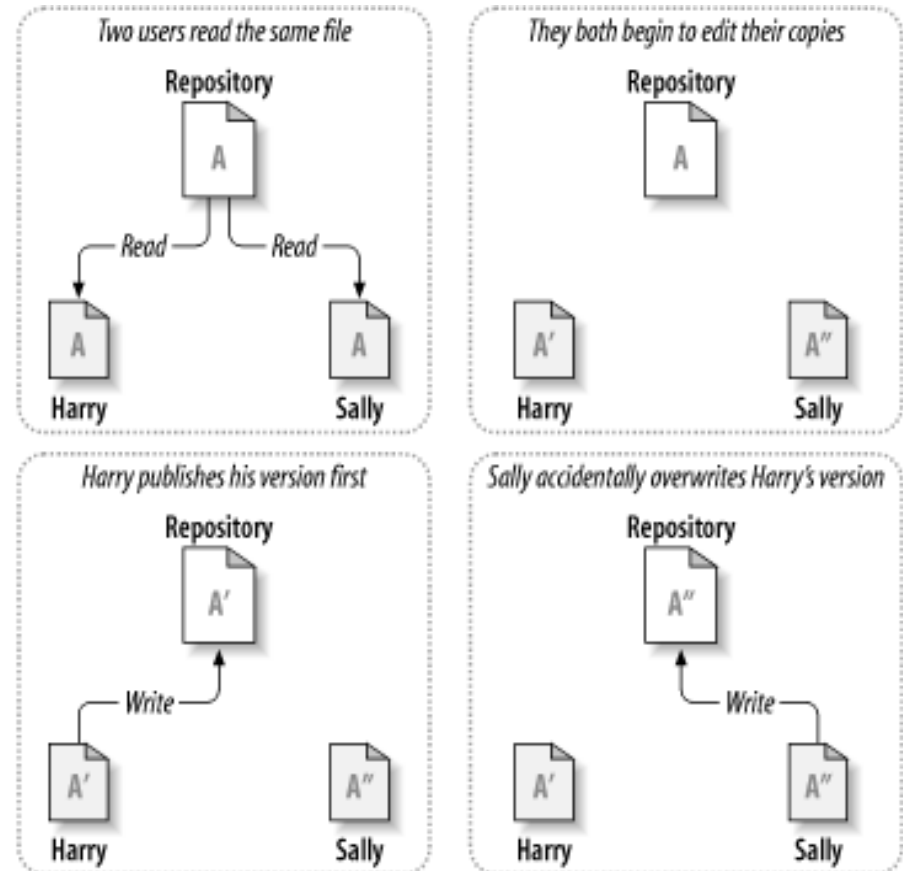
- SVN está compuesto por bibliotecas compartidas en C con APIs claras.
- Hay implementaciones en distintos lenguajes (java, python, etc)

Arquitectura de Subversion



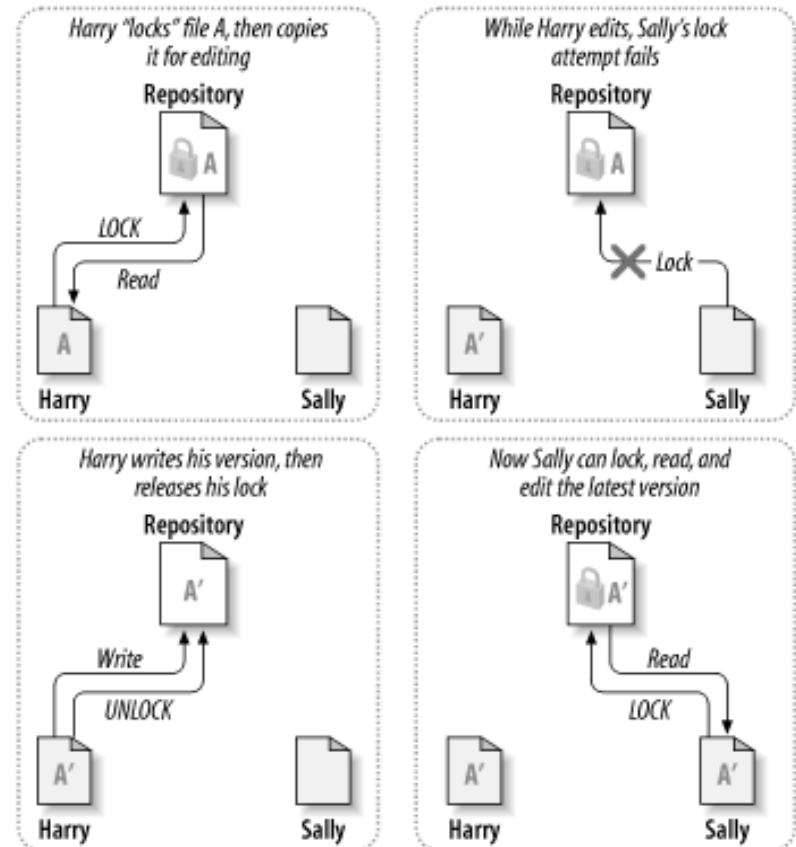
- ◇ Comenzado a principios de 2000 (CollabNet)
- ◇ Objetivo: Mejorar CVS
- ◇ Versión 1.0 liberada en 2004
 - Contemplaba todas de las funcionalidades de CVS y muchas mejoras
- ◇ La versión actual es la 1.4.3
- ◇ Grandes proyectos y empresas lo adoptaron
 - KDE, GCC, Python, Samba, Mono, PuTTY, Zope, Plone, CUPS...
 - Google y Sourceforge lo ofrecen como servicio
 - ¿Y vos?...

¿Cómo permitirá el sistema a los usuarios compartir información, pero al mismo tiempo impidiendo que se sobrescriban de forma accidental?



◇ La solución bloqueo-modificación-desbloqueo

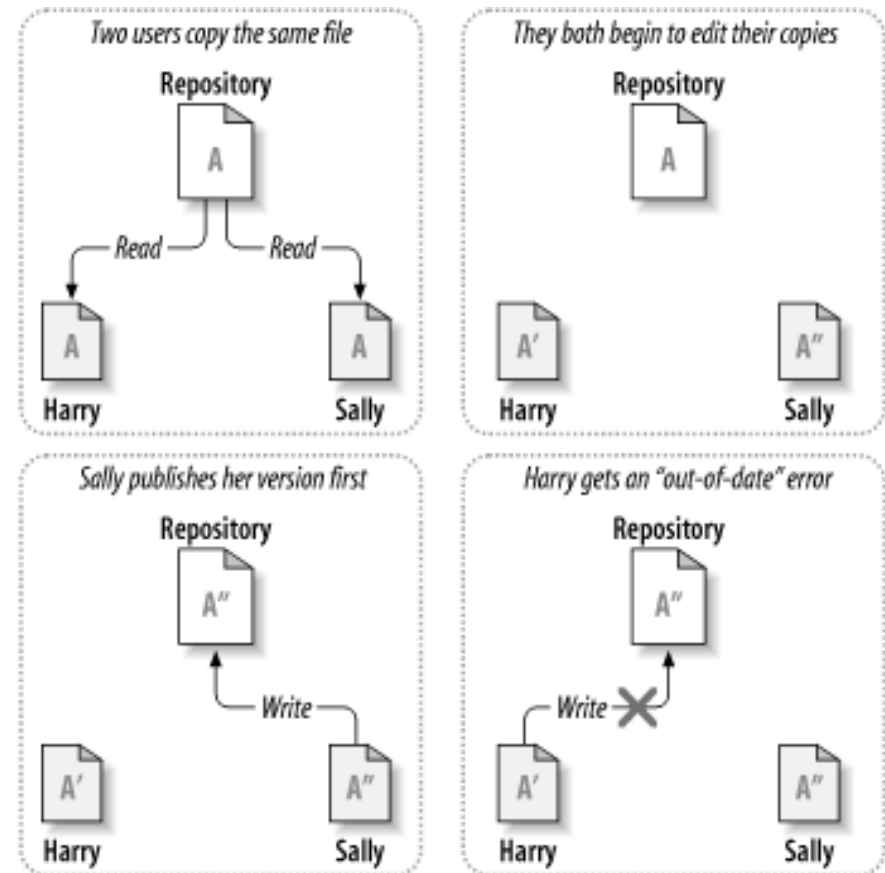
- El repositorio sólo permite a una persona modificar un archivo al mismo tiempo.
- Si 'Harry' ha bloqueado el archivo, entonces 'Sally' no puede hacerle cambios.
- Sólo puede leer el archivo y esperar a que lo desbloquee



- ◇ Bloquear puede causar problemas de gestión
 - Demoras innecesarias y pérdida de tiempo.
- ◇ *Bloquear puede causar una serialización innecesaria*
 - *Si los cambios no se solapan los usuarios podrían editar el archivo simultáneamente*
- ◇ *Bloquear puede causar una falsa sensación de seguridad*
 - *Cuando hay una dependencia entre varios archivos, no se soluciona nada bloqueando sólo uno, ya que una dependencia puede cambiar y el proyecto pierde integridad.*

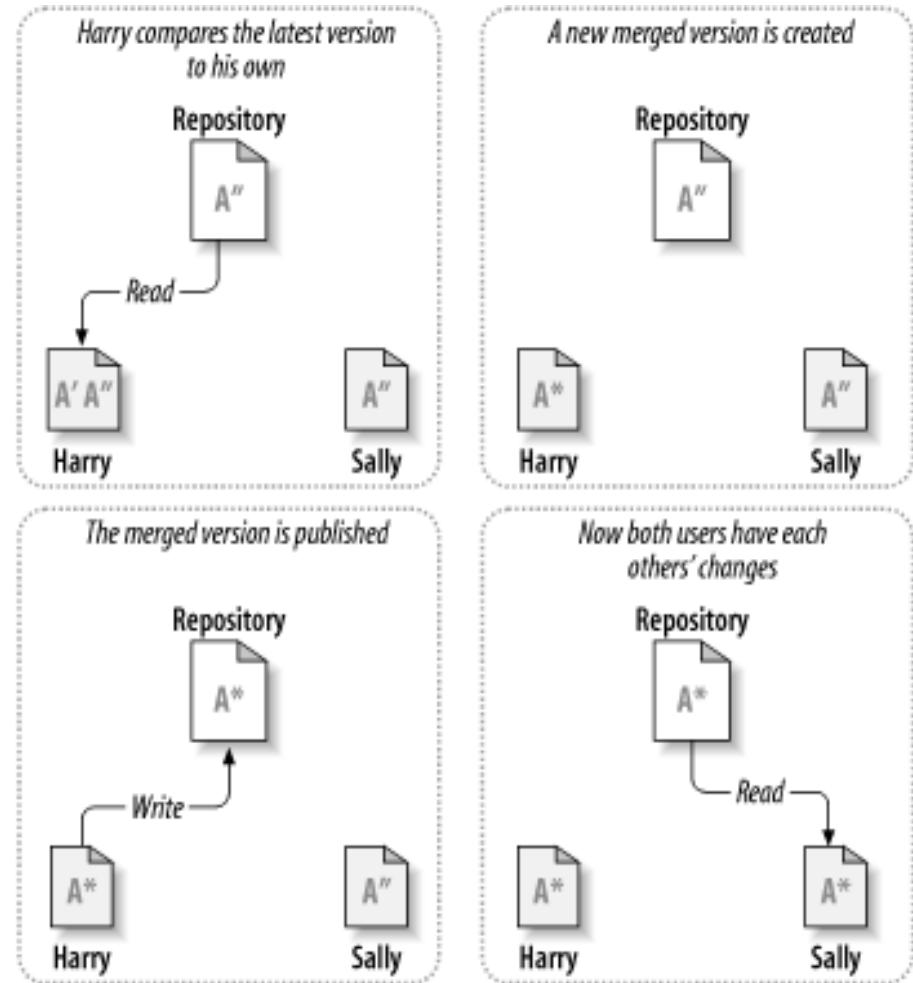
◇ La solución copiar-modificar-mezclar

- Se crea una réplica local del repositorio
- Los usuarios pueden trabajar en paralelo
- Finalmente, todas las copias privadas se combinan en una nueva versión final.



◇ ¿Pero qué ocurre si los cambios de los usuarios se solapan?

- Se produce un conflicto
- Se puede resolver de tres formas:
 - Editando manualmente las líneas conflictivas
 - Eligiendo una de las versiones
 - O deshaciendo todos los cambios locales



◇ Conclusiones

- *El tiempo en resolver los conflictos es mucho menor que el tiempo perdido por un sistema de bloqueos.*
- *Los usuarios pueden trabajar en paralelo, sin tener que esperarse el uno al otro.*
- *La mayoría de los cambios concurrentes no se solapan en absoluto; los conflictos son poco frecuentes*
- *Un factor crítico para la productividad: la comunicación entre los usuarios. ¡Hay que conversar con los compañeros!*

- ◇ 0) Crear copia local (checkout)
 - Se puede especificar una revision o fecha particular
- ◇ 1) Actualizar la copia de trabajo (update)
 - Permite recuperar las ultimas modificaciones del repositorio
- ◇ 2) Hacer cambios
 - Add, delete, copy, move
- ◇ 3) Examinar cambios
 - status, diff, revert
- ◇ 4) Fusionar cambios
 - merge, resolved
- ◇ 5) Enviar cambios (commit)
 - Requiere un mensaje 'log' que detalle las modificaciones hechas

◇ Copia de trabajo

- Es la copia local de los archivos de un repositorio, en un momento del tiempo o revisión específicos.

◇ Check-out (co)

- Crea una copia de trabajo local desde el repositorio. Se puede especificar una revisión específica.

◇ Commit (ci)

- Cuando una copia de los cambios hechos a una copia local es integrada sobre repositorio.

◇ Import

- Una importación es la acción de copiar un árbol de directorios local (que no es en ese momento una copia de trabajo) en el repositorio por primera vez.

◇ Actualizar (update)

- Una actualización integra los cambios que han sido hechos en el repositorio (por ejemplo por otras personas) en la copia de trabajo local.

◇ Conflicto

- Ocurre cuando se realizan dos cambios al mismo documento, y el sistema es incapaz de reconciliar los mismos.

◇ Resolver

- La intervención del usuario para atender un conflicto entre diferentes cambios al mismo documento.

◇ Clientes

- TortoiseSVN (MS Windows)
- Subclipse (plugin para Eclipse, multiplataforma)
- RapidSVN (multiplataforma)
- Qsvn (Linux/KDE)
- muchos más...

◇ Navegadores de repositorios

- ViewVC
- WebSVN

◇ Sistemas integrados

- Trac: SVN+wiki+gestión de bugs

◇ Sitio oficial de Subversion

– <http://subversion.tigris.org/>

◇ *Libro libre 'Control de versiones con Subversion'*

– <http://svnbook.red-bean.com/>

◇ *Hosting SVN gratuito*

– <http://code.google.com/hosting>

– <http://sourceforge.net>

– ¿proximamente la facu?

◇ **Dónde encontrar esta presentación**

– En la página de la carrera

– <http://www.efn.unc.edu.ar/escuelas/computacion/>