**Distintos sistemas de control de versión aplicado al desarrollo de software**

Es la gestión de los diversos cambios que se realizan sobre algún producto o una configuración del mismo. Una versión es el estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificación.

Estos sistemas facilitan la administración de las distintas versiones de cada producto desarrollado así como las posibles especializaciones realizadas. El control de versiones se realiza principalmente en la industria informática para controlar las distintas versiones del código fuente dando lugar a los sistemas de control de código fuente o SCM. Sin embargo los mismos conceptos son también aplicables a documentos imágenes sitios web etc.

Existen varios sistemas de control de versión los cuales nombraremos y detallaremos a continuación:

* Plastic SCM
* Git
* Mercurial
* Perforce
* Fossil SCM
* Team Foundation Server.

**CVS.-** Concurrent Versions System o simplemente CVS, también conocido como Concurrent Versioning System, es una aplicación informática que implementa un sistema de control de versiones: mantiene el registro de todo el trabajo y los cambios en los ficheros (código fuente principalmente) que forman un proyecto (de programa) y permite que distintos desarrolladores (potencialmente situados a gran distancia) colaboren. CVS se ha hecho popular en el mundo del software libre. Sus desarrolladores difunden el sistema bajo la licencia GPL.Características:

CVS utiliza una arquitectura cliente-servidor: un servidor guarda la(s) versión(es) actual(es) del proyecto y su historial. Los clientes se conectan al servidor para sacar una copia completa del proyecto. Esto se hace para que eventualmente puedan trabajar con esa copia y más tarde ingresar sus cambios con comandos GNU.

Típicamente, cliente y servidor se conectan utilizando Internet, pero con el sistema CVS el cliente y servidor pueden estar en la misma máquina. El sistema CVS tiene la tarea de mantener el registro de la historia de las versiones del programa de un proyecto solamente con desarrolladores locales. Originalmente, el servidor utilizaba un sistema operativo similar a Unix, aunque en la actualidad existen versiones de CVS en otros sistemas operativos, incluido Windows. Los clientes CVS pueden funcionar en cualquiera de los sistemas operativos más difundidos.

Varios clientes pueden sacar copias YOU del proyecto al mismo tiempo. Posteriormente, cuando actualizan sus modificaciones, el servidor trata de acoplar las diferentes versiones. Si esto falla, por ejemplo debido a que dos clientes tratan de cambiar la misma línea en un archivo en particular, entonces el servidor deniega la segunda actualización e informa al cliente sobre el conflicto, que el usuario deberá resolver manualmente. Si la operación de ingreso tiene éxito, entonces los números de versión de todos los archivos implicados se incrementan automáticamente, y el servidor CVS almacena información sobre la actualización, que incluye una descripción suministrada por el usuario, la fecha y el nombre del autor y sus archivos de registro (log).

**Subversion (software).-** Apache Subversión (abreviado frecuentemente como SVN, por el comando svn) es una herramienta de control de versiones open source basada en un repositorio cuyo funcionamiento se asemeja enormemente al de un sistema de ficheros. Es software libre bajo una licencia de tipo Apache/BSD.

Utiliza el concepto de revisión para guardar los cambios producidos en el repositorio. Entre dos revisiones sólo guarda el conjunto de modificaciones (delta), optimizando así al máximo el uso de espacio en disco. SVN permite al usuario crear, copiar y borrar carpetas con la misma flexibilidad con la que lo haría si estuviese en su disco duro local. Dada su flexibilidad, es necesaria la aplicación de buenas prácticas para llevar a cabo una correcta gestión de las versiones del software generado.

Subversión puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintas computadoras. A cierto nivel, la posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones. Y puesto que el trabajo se encuentra bajo el control de versiones, no hay razón para temer por que la calidad del mismo vaya a verse afectada —si se ha hecho un cambio incorrecto a los datos, simplemente deshaga ese cambio. Características:

* Se sigue la historia de los archivos y directorios a través de copias y renombrados.
* Las modificaciones (incluyendo cambios a varios archivos) son atómicas.
* La creación de ramas y etiquetas es una operación más eficiente. Tiene coste de complejidad constante (O(1)) y no lineal (O(n)) como en CVS.
* Se envían sólo las diferencias en ambas direcciones (en CVS siempre se envían al servidor archivos completos).
* Puede ser servido mediante Apache, sobre WebDAV/DeltaV. Esto permite que clientes WebDAV utilicen Subversion de forma transparente.
* Maneja eficientemente archivos binarios (a diferencia de CVS que los trata internamente como si fueran de texto).
* Permite selectivamente el bloqueo de archivos. Se usa en archivos binarios que, al no poder fusionarse fácilmente, conviene que no sean editados por más de una persona a la vez.
* Cuando se usa integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autentificar archivos (SQL, LDAP, PAM, etc.)

**Microsoft Visual SourceSafe.-** Es una herramienta de Control de versiones que forma parte de Microsoft Visual Studio aunque está siendo sustituido por Visual Studio Team Foundation Server.

SourceSafe fue creado por la compañía americana One Tree Software a principios de los años noventa. Su primera versión fue la 3.1 coincidiendo con el mismo número de versión del sistema operativo Windows. Microsoft adquirió los derechos sobre SourceSafe que era un programa de 16 bits y lo liberó en 1995 como un programa de 32 bits, con la versión 4.0

SourceSafe es un sistema basado en un equipo anfitrión a diferencia de la mayoría de los programas de control de versiones que son basados en Cliente-Servidor donde el repositorio de control de cambios reside en el equipo servidor y los clientes toman de allí la última versión para modificarla y posteriormente ingresarla con las modificaciones realizadas. Como en todos los programas de control de versiones, se basa en obtener una copia de trabajo ("check-out" o "desproteger"), realizar cambios sobre la copia y reingresarla al repositorio. Para lograr el acceso compartido al repositorio, VSS emplea el protocolo de archivos compartidos SMB lo cual crea algunos inconvenientes (ver Desventajas, más adelante).

* **Ventajas:**

Para las personas que desarrollan programas en el sistema operativo Windows, resulta una herramienta útil ya que se integra fuertemente con el entorno de desarrollo integrado o IDE de Visual Studio permitiendo un manejo relativamente simple de versiones sobre una computadora individual y en equipos de trabajo relativamente pequeños.

**Rational ClearCase.-** Es una familia de herramientas de software de computadoras que supports software gestión de la configuración (SMC) de código fuente y otros activos de desarrollo de software. Incluye control de revisiones. ClearCase constituye la base para la gestión de configuración en las grandes y medianas empresas y tiene capacidad para proyectos con cientos o miles de desarrolladores. Es desarrollado por IBM. ClearCase admite dos modelos de gestión de configuración: UCM (Gestión del Cambio Unificado) y ClearCase base. UCM ofrece un modelo de salida de la caja mientras ClearCase base proporciona una infraestructura básica (UCM está construido sobre ClearCase base). Ambos pueden ser configurados para soportar una amplia variedad de necesidades. ClearCase puede acomodar grandes archivos binarios, un gran número de archivos, y grandes tamaños de repositorio. Es compatible con ramificaciones, etiquetado y control de versiones de los directorios.

**Darcs.-** Es un sistema de control de versiones distribuido creado por David Roundy. Las características clave son: la habilidad de escoger que cambios aceptar de otros repositorios, interacción con otros repositorios locales (en disco) o repositorios remotos vía SSH, HTTP o correo electrónico, y una inusual interfaz interactiva. Los desarrolladores también hacen hincapié en el uso de herramientas de software avanzadas para la verificación de la corrección: el sistema de tipo expresivo del lenguaje de programación funcional Haskell hace cumplir algunas propiedades, y las pruebas al azar a través de QuickCheck verifica muchos otros.2 El nombre es un acrónimo recursivo para el sistema de control de versiones avanzada Darcs Advanced Revision Control System.3

**Modelo.-** Darcs trata los parches como ciudadanos de primera clase. Para el usuario, un repositorio puede ser visto como un conjunto de parches, donde cada parche no es necesariamente requerido con relación a otros parches es decir, el conjunto de parches es sólo un conjunto parcialmente ordenado. En muchos casos, los parches pueden ser transmitidos de forma independiente entre varios repositorios.

Muchas operaciones de ramificación, fusión y cherry-picking que requieren comandos adicionales con sistemas basadas en snapshots como Git o Mercurial, se pueden hacer directamente con Darcs con los comandos comunes "pull" y "push". En cuanto a la interfaz de usuario, esto significa que Darcs tiene menos comandos. Además estos comandos son más interactivos ya que se puede elegir con más precisión que parches se quiere intercambiar con repositorios remotos.

En el sistema de archivos, los parches de un repositorio están ordenados linealmente. Darcs calcula automáticamente si los parches se pueden reordenar (una operación llamada conmutación), y la forma de hacerlo. Estos cálculos implementar la llamada "teoría de parches".

Un parche Darcs puede contener cambios de los siguientes tipos:

* Cambios en las líneas
* Creación y eliminación de archivos y directorios
* Movimiento de archivos y directorios

Sustitución de palabras (utilizado normalmente en la refactorización de código, por ejemplo, cambiar el nombre de todas las apariciones de "foo" en un archivo determinado)

La noción de dependencia entre los parches se define sintácticamente. Intuitivamente, un parche B depende de otro parche A si A proporciona los contenidos que B modifica. Esto significa que los parches que modifican diferentes partes del código se consideran, de forma predeterminada, independientes. Para hacer frente a los casos en que esto no es deseable, Darcs permite al usuario especificar dependencias explícitas entre parches.

**Bazaar.-** Es un sistema de control de versiones distribuido patrocinado por Canonical Ltd., diseñado para facilitar la contribución en proyectos de software libre y opensource.

Bazaar puede ser usado por un usuario único trabajando en múltiples ramas de un contenido local, o por un equipo colaborando a través de la red.

Bazaar está escrito en lenguaje de programación Python y tiene versiones empaquetadas para la mayoría de distribuciones GNU/Linux, así como Mac OS X y MS Windows. Bazaar es software libre y parte del proyecto GNU.

Alojamiento de código fuente

Los siguientes sitios web proporcionan alojamiento gratis en repositorios utilizando Bazaar:

* Launchpad
* GNU Savannah
* Sourceforge
* Proyectos

Proyectos destacados que utilizan Bazaar como sistema de control de versiones:

* Ubuntu2
* GNU Mailman3 4
* GNU Emacs5
* Inkscape6
* MySQL7
* Gnash8 9
* Squid10