UNIVESIDAD TÉCNICA NACIONAL

INGENIERÍA EN SOFTWARE

PROGRAMACION 1

PROFESOR:

EFREN JIMENEZ DELGADO

ALUMNO:

NORMAN MUÑOZ GOMEZ

SEGUNDO CUATRIMESTRE 2014

**Subversion**

Subversión (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.

**Ventajas**

* 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).

**Uso y reconocimiento**

Subversión es muy conocido en la comunidad de software libre y se utiliza en muchos proyectos, como:

* Apache Software Foundation
* Django (migrado a Git)
* Free Pascal
* FreeBSD
* GCC
* KDE (en proceso de migración a Git)
* Ruby
* Tigris.org

Servicios que proporcionan almacenamiento usando Subversión gratuito para proyectos de software libre:

* SourceForge
* Google Code
* Project Kenai
* CodePlex
* Forja de Conocimiento Libre de la Comunidad RedIRIS de RedIRIS

**Administración de versiones**

**Git**

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o front end como Cogito o StGIT. 2 Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. 3 Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux.

Características

* El diseño de Git se basó en BitKeeper y en Monotone.
* Fuerte apoyo al desarrollo no lineal, por ende rapidez en la gestión de ramas y mezclado de diferentes versiones. Git incluye herramientas específicas para navegar y visualizar un historial de desarrollo no lineal. Una presunción fundamental en Git es que un cambio será fusionado mucho más frecuentemente de lo que se escribe originalmente, conforme se pasa entre varios programadores que lo revisan.
* Gestión distribuida. Al igual que Darcs, BitKeeper, Mercurial, SVK, Bazaar y Monotone, Git le da a cada programador una copia local del historial del desarrollo entero, y los cambios se propagan entre los repositorios locales. Los cambios se importan como ramas adicionales y pueden ser fusionados en la misma manera que se hace con la rama local.
* Los almacenes de información pueden publicarse por HTTP, FTP, rsync o mediante un protocolo nativo, ya sea a través de una conexión TCP/IP simple o a través de cifrado SSH. Git también puede emular servidores CVS, lo que habilita el uso de clientes CVS pre-existentes y módulos IDE para CVS pre-existentes en el acceso de repositorios Git.
* Los repositorios Subversion y svk se pueden usar directamente con git-svn.
* Gestión eficiente de proyectos grandes, dada la rapidez de gestión de diferencias entre archivos, entre otras mejoras de optimización de velocidad de ejecución.
* Todas las versiones previas a un cambio determinado, implican la notificación de un cambio posterior en cualquiera de ellas a ese cambio (denominado autenticación criptográfica de historial). Esto existía en Monotone.

**Mercurial**

Mercurial es un sistema de control de versiones multiplataforma, para desarrolladores de software. Está implementado principalmente haciendo uso del lenguaje de programación Python, pero incluye una implementación binaria de diff escrita en C. Mercurial fue escrito originalmente para funcionar sobre Linux. Ha sido adaptado para Windows, Mac OS X y la mayoría de otros sistemas tipo Unix. Mercurial es, sobre todo, un programa para la línea de comandos. Todas las operaciones de Mercurial se invocan como opciones dadas a su programa motor, hg (cuyo nombre hace referencia al símbolo químico del mercurio).

Las principales metas de desarrollo de Mercurial incluyen un gran rendimiento y escalabilidad; desarrollo completamente distribuido, sin necesidad de un servidor; gestión robusta de archivos tanto de texto como binarios; y capacidades avanzadas de ramificación e integración, todo ello manteniendo sencillez conceptual.1 Incluye una interfaz web integrada.

El creador y desarrollador principal de Mercurial es Matt Mackall. El código fuente se encuentra disponible bajo los términos de la licencia GNU GPL versión 2, lo que clasifica a Mercurial como software libre.

**GitHub**

Git es uno de los sistemas de control de versiones más populares entre los desarrolladores. Y parte culpa de su popularidad la tiene GitHub, un excelente servicio de alojamiento de repositorios de software con este sistema, que lejos de quedarse en esta funcionalidad, ofrece hoy en día un conjunto de características muy útiles para el trabajo en equipo.

No en vano, es el servicio elegido por proyectos de software libre como jQuery, reddit, Sparkle, curl, Ruby on Rails, node.js, ClickToFlash, Erlang/OTP, CakePHP, Redis, y otros muchos. Además, algunas de las grandes empresas de Internet, como Facebook, alojan ahí sus desarrollos públicos, tales como el SDK, librerías, ejemplos, etc.

Nosotros no queríamos ser menos, y hemos creado una cuenta de GitHub para GenbetaDev donde publicar ocasionalmente el código que utilicemos para ciertos artículos. Aunque no busquéis mucho en ella por ahora que de momento está vacía.

Y ahora paso a hablaros un poco de qué es GitHub y cuáles son las ventajas evidentes de utilizar un servicio como éste.

Pero hoy en día GitHub es mucho más que un servicio de alojamiento de código. Además de éste, se ofrecen varias pequeñas herramientas en línea muy útiles para el trabajo en equipo. Entre ellas, cabe destacar:

* Un wiki que funciona con gollum, el cual opera con Git para el mantenimiento de las distintas versiones de las páginas.
* Un sistema de seguimiento de problemas, que al estilo del clásico sistema de tickets, permiten a los miembros de tu equipo (o a cualquier usuario de GitHub si tu repositorio es público) abrir un ticket detallando un problema que tenga con tu software o una sugerencia que desee hacer al mismo.
* Una herramienta de revisión de código, donde se pueden añadir anotaciones en cualquier punto de un fichero (ej: “Esto es mejor que lo extraigamos a una nueva clase”), y debatir sobre determinados cambios realizados en un commit específico.
* Un visor de ramas donde se pueden comparar los progresos realizados en las distintas ramas de nuestro repositorio.

**Bitbucket**

Bitbucket es un servicio de alojamiento basado en web, para los proyectos que utilizan el sistema de control de revisiones Mercurial y Git. Bitbucket ofrece planes comerciales y gratuitos. Se ofrece cuentas gratuitas con un número ilimitado de repositorios privados (que puede tener hasta cinco usuarios en el caso de cuentas gratuitas) desde septiembre de 2010 , los repositorios privados no se muestran en las páginas de perfil - si un usuario sólo tiene depósitos privados, el sitio web dará el mensaje "Este usuario no tiene repositorios". El servicio está escrito en Python.2

Es similar a GitHub, que utiliza Git. En una entrada de blog del 2008,3 Bruce Eckel hace una comparación favorablemente de bitbucket frente a Launchpad, que utiliza Bazaar.