Trabajo de programación

¿Qué es un subversionador o administrador de versiones?

Es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante. Es el mandato de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Ayuda virtualmente en todos los aspectos al dirigir un proyecto

¿Para qué funciona?

Para facilitar la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas.

¿Cómo funcionan?

Este puede funcionar almacenando, restaurando y realizando el seguimiento de elementos de una lista y de archivos de una biblioteca siempre que éstos cambien.

¿Qué es un Git?

Git es un software de [control de versiones](http://es.wikipedia.org/wiki/Control_de_versiones) diseñado pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de [código fuente](http://es.wikipedia.org/wiki/C%C3%B3digo_fuente).

¿Cómo funciona un Git?

¿Qué hace?

Llevar cuenta de los cambios que hacemos en nuestros archivos.

Proteger nuestro código

Protege de errores de hardware como por ejemplo si un disco duro se quema

Contra corrupción de archivos

Contra errores de programación (por ejemplo metemos una función que hace que nuestro programa no corra nos podemos devolvernos a una versión anterior estable y lo podemos corregir para que corra correctamente).

Entre las características más relevantes se encuentran:

* Fuerte apoyo al desarrollo no lineal, por ende rapidez en la gestión de ramas y mezclado de diferentes versiones.
* Gestión distribuida
* Gestión eficiente de proyectos grandes, dada la rapidez de gestión de diferencias entre archivos, entre otras mejoras de optimización de velocidad de ejecución.

[Programado en](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n): [C](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_C), [Bourne Shell](http://es.wikipedia.org/wiki/Bourne_Shell" \o "Bourne Shell), [Perl](http://es.wikipedia.org/wiki/Perl)

Git hub: Es una [forja](http://es.wikipedia.org/wiki/Forja_(software)) para alojar proyectos utilizando el sistema de [control de versiones](http://es.wikipedia.org/wiki/Control_de_versiones) [Git](http://es.wikipedia.org/wiki/Git). Utiliza el [framework](http://es.wikipedia.org/wiki/Framework" \o "Framework) [Ruby on Rails](http://es.wikipedia.org/wiki/Ruby_on_Rails) por GitHub, Inc.

Características de GitHub útiles para el trabajo en equipo

* Un wiki que funciona con [gollum](https://github.com/github/gollum), el cual opera con Git para el mantenimiento de las distintas versiones de las páginas.
* Un [sistema de seguimiento de problemas](https://github.com/rails/rails/issues), que al estilo del clásico sistema de tickets, permiten a los miembros de tu equipo, 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](https://github.com/rails/rails/commit/0495736339df48a625fd58f4304b7b497b02d859#comments), donde se pueden añadir anotaciones en cualquier punto de un fichero
* Un [visor de ramas](https://github.com/rails/rails/branches) donde se pueden comparar los progresos realizados en las distintas ramas de nuestro repositorio.

¿Qué es Mercurial?

Es un [sistema de control de versiones](http://es.wikipedia.org/wiki/Sistema_de_control_de_versiones) [multiplataforma](http://es.wikipedia.org/wiki/Multiplataforma), para [desarrolladores de software](http://es.wikipedia.org/wiki/Desarrollador_de_software).

¿Para qué sirve?

Está implementado principalmente haciendo uso del [lenguaje de programación](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [Python](http://es.wikipedia.org/wiki/Python), pero incluye una implementación binaria de [diff](http://es.wikipedia.org/wiki/Diff" \o "Diff) escrita en [C](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_C).

Historia:

Mercurial fue escrito originalmente para funcionar sobre Linux. Ha sido adaptado para [Windows](http://es.wikipedia.org/wiki/Microsoft_Windows), [Mac OS X](http://es.wikipedia.org/wiki/Mac_OS_X) y la mayoría de otros sistemas [tipo Unix](http://es.wikipedia.org/wiki/Unix). Mercurial es, sobre todo, un programa para la [línea de comandos](http://es.wikipedia.org/wiki/L%C3%ADnea_de_comandos). Todas las operaciones de Mercurial se invocan como opciones dadas a su programa motor, hg (cuyo nombre hace referencia al [símbolo químico](http://es.wikipedia.org/wiki/S%C3%ADmbolo_qu%C3%ADmico) del [mercurio](http://es.wikipedia.org/wiki/Mercurio_(elemento))). 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](http://es.wikipedia.org/wiki/Archivo_de_texto) como [binarios](http://es.wikipedia.org/wiki/Archivo_binario); y capacidades avanzadas de ramificación e integración, todo ello manteniendo sencillez conceptual. Incluye una interfaz web integrada.

El creador y desarrollador principal de Mercurial es [Matt Mackall](http://es.wikipedia.org/w/index.php?title=Matt_Mackall&action=edit&redlink=1" \o "Matt Mackall (aún no redactado)). El [código fuente](http://es.wikipedia.org/wiki/C%C3%B3digo_fuente) se encuentra disponible bajo los términos de la licencia [GNU GPL](http://es.wikipedia.org/wiki/GNU_General_Public_License) versión 2, lo que clasifica a Mercurial como [software libre](http://es.wikipedia.org/wiki/Software_libre).

Bitbucket: es un servicio de alojamiento basado en web, para los proyectos que utilizan el sistema de control de revisiones[Mercurial](http://es.wikipedia.org/wiki/Mercurial) y [Git](http://es.wikipedia.org/wiki/Git). Bitbucket ofrece planes comerciales y gratuitos. Se ofrece cuentas gratuitas con un número ilimitado de repositorios privados. El servicio está escrito en [Python](http://es.wikipedia.org/wiki/Python).

¿Qué es Subversion?

Subversion es un sistema de control de versiones libre y de código fuente abierto. Es decir, Subversion maneja ficheros y directorios a través del tiempo. Hay un árbol de ficheros en unrepositorio central. El repositorio es como un servidor de ficheros ordinario, excepto porque recuerda todos los cambios hechos a sus ficheros y directorios. Ésto le permite recuperar versiones antiguas de sus datos, o examinar el historial de cambios de los mismos. En este aspecto, mucha gente piensa en los sistemas de versiones como en una especie de“máquina del tiempo ”.

Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos ordenadores. A cierto nivel, la capacidad para que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar mas 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 por la pérdida de ese conducto único—si se ha hecho un cambio incorrecto a los datos, simplemente deshaga ese cambio.

¿Subversion solo sirve en Windows?

Subversion está construido sobre una capa de portabilidad llamada APR (la biblioteca Apache Portable Runtime), lo cual significa que Subversion debería funcionar en cualquier sistema operativo donde lo haga el servidor httpd Apache: Windows, Linux, todos los sabores de BSD, Mac OS X, Netware y otros

Git =Administrador de versiones desentralizado

Git y mercurial= son multiplataforma

Subversion= Sentralizado, solo se le puede enviar a un unico servidor

Control de versiones= Nos permite trabajar en equipo

Programming work

What is a manager or subversionador versions?

It is a system that records changes made to a file or set of files over time so that you can later retrieve specific versions.It is the mandate of the various changes that are performed on the elements of a product or configuration. Help in virtually all aspects of running a project.

What works?

To facilitate the administration of the different versions of each product developed and made ​​possible specializations.

How do they work?

This can work storing, restoring and by tracking items in a list and library files whenever they change.

What is Git?

Git is a version control software designed with efficiency and reliability of maintaining versions of applications when they have a large number of source files.

How does a Git works?

What do you do?

Keeping track of the changes we make in our archives.   
Protect our code.   
Protects from hardware errors such as if a hard disk is burned.   
Against file corruption.   
Against programming errors (eg we put a function that does not run our program we can restore to a previous stable version and we can fix it to run properly).