**UNIVERSIDAD AUTONOMA “GABRIEL RENE MORENO”**

**FACULTAD INTEGRAL DEL CHACO**

**CARRERA ING. INFORMATICA**

**SISTEMAS DE CONTROL DE VERSIONES**

**(VCS)**

**NOMBRE: Maria Paola Vargas Fernandez**

**MATERIA: Programación de Aplicación en Tiempo Real**

**SIGLA: ELC – 104**

**DOCENTE: ING. Luis Ariel Vega**

**Sistemas de control de versiones (VCS)**

También conocido como sistema de control de revisiones o de fuentes, es una herramienta de software que monitoriza y gestiona cambios en un sistema de archivos

**Listas de 5 sistemas de control de versiones (VCS)**

### CVS (Concurrent versions system)

### Subversión (SVN).

### Mercurial (hg).

### Bazaar (bzr).

### Visual SourceSafe (Microsoft).

**CVS (Concurrent versions system)**

Concurrent versions system desarrollado por Dick Grune a mediados de los años ochenta basado en RSC creada a modo de conjunto de scripts, trabajando simultáneamente de forma independiente, pasando cada trabajo sobre una copia del proyecto, es un sistema de control de versiones que mantiene el registro del trabajo y los cambios en los ficheros permitiendo a distintos des arrolladores que colaboren.

**Ventajas.**

* Utiliza arquitectura cliente- servidor.
* El servidor guarda las versiones y el historial.
* Los cliente se conectan y hacen una copia completa del Proyecto.
* Funciona en cualquier sistema operativo.
* Varios personas pueden sacar una copia al mismo tiempo.
* Puede mantener distintas ramas de un Proyecto.

**Desventajas.**

* No soporta refactorización de sistemas de forma automática o versionadas.
* Limitada para UTF-8 Unicode o archivos con contenido diferecnte ACII.
* El protocolo no soporta eliminación de directorios o re nombrarlos.
* Visual Studio carece de soporte nativo a CVS,

### Subversión(SVN).

Es un sistema de control de versiones que remplaza a CVS, con opciones más amplia, la mayoría de proyectos de código abierto utilizan SVN como SourceForge, Apache, Python, Ruby, se puede acceder al repositorio a través de las redes, lo que permite ser usado por muchas personas en distintos ordenadores fomentando la colaboración y así progresando rápidamente, ya que el proyecto se encuentra en un control de versiones no hay temor de que el proyecto acabe afectado por algún error.

**Ventajas.**

* Se sigue la historia de los archivos y directorios a través de copias y renombrados.
* Modificaciones atómicas.
* La creación de branch, y tags son operaciones eficientes.
* Se envían sólo las diferencias en ambas direcciones.
* Maneja eficazmente los archivos binarios.
* Permite el bloqueo de archivos.

**Desventejas.**

* El manejo de cambio de nombres de archivos no es completo. Lo maneja como la suma de una operación de copia y una de borrado.
* No resuelve el problema de aplicar repetidamente parches entre ramas, no facilita llevar la cuenta de qué cambios se han realizado. esto se resuelve siendo cuidadoso con los mensajes de commit.

### Mercurial (hg).

Es un sistema de control de versiones distribuido de código abierto como Git, fue diseñado para proyectos grandes, los equipos de desarrolladores pequeños no pueden utilizarlo, es muy rápido y los que crearon este sistema le pusieron de nombre de mercurial que lo atribuyen al dios Mercurio.

**Ventajas.**

* Rápido escalable.
* Mas simple que Git.
* No hay tantas funciones para aprender, son similares a los otros sistemas.
* Viene equipado con una interfaz Web independiente y una amplia documentación en la comprensión de mercurial.
* Es liviano.

**Desventajas.**

* Solo se usa en proyectos grandes.

### Bazaar(bzr).

Es otro sistema de control de versiones distribuido como mercurial y Git, tiene una interfaz amigable, se hace llamar a sí mismo el control de versiones para humanos, la principal característica de bazar es el control bean que tiene una configuración, como se muestra los flujos de trabajo adaptándose al entorno y configuración del usuario.

**Ventajas.**

* Se puede utilizar para adaptarse a cualquier tipo de proyecto.
* Es fácil de modificar muy intuitivo.
* Integrable  se puede añadir proyectos existentes.
* Tiene una comunidad fuerte que mantiene las cosas, como plug-ins y bastantes herramientas de terceros como GUI para añadir interfaces gráficas para el sistema.

**Desventajas.**

* Se trabaja siempre con acceso al servidor remoto.
* No es muy popular.
* Consume ancho de banda.
* Estadísticamente es lento.
* El Launchpad es nuevo.

### Visual SourceSafe (Microsoft).

Es una herramienta de control de versiones que forma parte de Microsoft visual studio, es un sistema basado en un equipo anfitrión, a diferencia de los otros sistemas de control de versiones que se basan en cliente servidor, para implementar el acceso compartido VSS sw emplea el protocolo de archivos compartidos SMB lo cual crea algunos inconvenientes.

**Ventajas.**

* Permite un manejo relativo simple de versiones sobre un ordenador individual y trabajo relativamente pequeños.

**Desventajas.**

* El método de acceso que constituye su repositorio SMB impide que sea manipulado de manera externa.
* Equipos de trabajo grande de acceso concurrente provoca lentitud en el sistema.
* Solo un programador puede modificar el código.
* Las herramientas de gestión que reunifica el código no son realmente buenas comparadas con otros gestores de código fuente.
* Inestable cuando se suben archivos binarios de gran tamaño.