**CONTROL DE VERCIONES**

**El control de versión es un sistema que registra los cambios realizados en el archivo o conjunto de archivos a lo largo del tiempo para poder recuperar versiones anteriores o versiones específicas más adelante es un concepto clave para cualquier programador**

[**Git**](https://git-scm.com/)**:** es una de las mejores herramientas de control de versiones disponible en el mercado actual. Es un modelo de repositorio distribuido compatible con sistemas y protocolos existentes como HTTP, FTP, SSH y es capaz de manejar eficientemente proyectos pequeños a grandes.

DESVENTAS;

Por qué es tan difícil de aprender Git? - Quora. Todo nuevo conocimiento tiene su curva de aprendizaje y como todo, vas a requerir de consultar los tutoriales y practicar y practicar y practicar. Eventualmente se te hara muy sencillo y no vas a concebir como poder hacer un proyecto sin un repositorio de código

[**CVS**](http://www.nongnu.org/cvs/)**:** es otro sistema de control de versiones muy popular. Es un modelo de repositorio cliente-servidor donde varios desarrolladores pueden trabajar en el mismo proyecto en paralelo. El cliente CVS mantendrá actualizada la copia de trabajo del archivo y requiere intervención manual sólo cuando ocurre un conflicto de edición.

DESVENTAJA;

 No soporta revisiones firmadas, como SVN. Poco soporte para el control de código fuente distribuio

[**Apache Subversion**](https://subversion.apache.org/)**(SVN):** abreviado como SVN, apunta a ser el sucesor más adecuado. Es un modelo de repositorio cliente-servidor donde los directorios están versionados junto con las operaciones de copia, eliminación, movimiento y cambio de nombre.

DESVENTAJA;

* La principal desventaja de Subversion es que es más lento que CVS y que una verificación local de Subversion requiere mayor espacio en disco.
* 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.
* El manejo de archivos binarios los trata internamente como si fuera de texto no como de Subversion.

[**Mercurial**](https://www.mercurial-scm.org/)**:** es una herramienta distribuida de control de versiones que está escrita en Python y destinada a desarrolladores de software. Los sistemas operativos que admite son similares a Unix, Windows y macOS. Tiene un alto rendimiento y escalabilidad con capacidades avanzadas de ramificación y fusión y un desarrollo colaborativo totalmente distribuido. Además, posee una interfaz web integrada.

DESVENTAJA;

* Pocas caracterizas añadidas por default
* Comunidad de desarrollo muy pequeña

[**Monotone**](https://www.monotone.ca/)**:** está escrito en C ++ y es una herramienta para el control de versiones distribuido. El sistema operativo que admite incluye Unix, Linux, BSD, Mac OS X y Windows. Brinda un buen apoyo para la internacionalización y localización. Además, utiliza un protocolo personalizado m,,

DESVENTAJA;

 Problemas de rendimiento en algunas operaciones. No permite hacer checkout ni commit detrás del proxy.