

# Sistema de control de versiones

Iván Guadalupe Bustamante Cortés  
lupitobtte\_17@hotmail.com  
Universidad de la Sierra Sur

2022/05/09

## 1. Introducción

Un sistema de control de versiones es de las herramientas más útiles en la vida de un programador, pues esta herramienta tiene la función de poder gestionar todos los cambios realizados en el código fuente del programa que se está realizando, pues en caso de que este llegue a fallar en algún punto, el programador puede regresar a una versión anterior la cual resultaba estable. A medida que los entornos de desarrollo se aceleran, los sistemas de control de versiones ayudan a los equipos de software a trabajar de forma más rápida e inteligente.

Este tipo de sistemas también es de gran utilidad para equipos de desarrollo, pues con este puede controlar las versiones de distintos módulos y llegando a un punto en el cual los distintos trabajadores consideren estables y finalizados sus módulos puedan unirlos. El control de versiones ayuda a los equipos a resolver este tipo de problemas al realizar un seguimiento de todos los cambios individuales de cada colaborador y al contribuir a evitar que el trabajo concurrente entre en conflicto.

La ventajas que tiene usar un control de versiones son:

- Tienes un historial de cambios.
- Creación de ramas y fusiones.
- Trazabilidad.
- Detección de errores mas fácilmente.

## 2. Desarrollo

### 2.1. ¿Que es un sistema de control de versiones?

“El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo de tal manera que sea posible recuperar versiones específicas más adelante.” (Medium, 2017) Esta tarea se realiza con el fin de poder observar

y tener los distintos cambios realizados sobre el archivo, pues en caso de cometerse algún error, se puede regresar a alguna versión anterior del archivo en el cual aún resultaba estable.

## **2.2. Sistemas de Control de Versiones Locales**

Un método de control de versiones, usado por muchas personas, es copiar los archivos a otro directorio. Este método es muy común porque es muy sencillo, pero también es tremendamente propenso a errores. Es fácil olvidar en qué directorio te encuentras y guardar accidentalmente en el archivo equivocado o sobrescribir archivos que no querías. “Los sistemas de control de versiones locales en vez de mantener las versiones como archivos independientes, los almacenaban en una base de datos. Cuando era necesario revisar una versión anterior del proyecto se usaba el sistema de control de versiones en vez de acceder directamente al archivo, de esta manera en cualquier momento solo se tenía una copia del proyecto, eliminando la posibilidad de confundir o eliminar versiones.” (Medium, 2017)

## **2.3. Sistemas de Control de Versiones Centralizados**

“Un sistema de control de versiones centralizado (SCVC) utiliza un servidor central para almacenar todos los archivos y permite el trabajo colaborativo de un equipo. Trabaja sobre un repositorio único al que los usuarios pueden acceder desde un servidor central.

El repositorio indica un servidor central que está directamente conectado al puesto de trabajo de cada programador. Todos los programadores pueden actualizar (update) sus puestos de trabajo con los datos presentes en el repositorio o pueden hacer cambios (commit) en los mismos. Cada operación se realiza directamente en el repositorio.” (Opentix, 2019)

Esta configuración ofrece muchas ventajas, especialmente frente a VCS locales. Por ejemplo, todas las personas saben hasta cierto punto en qué están trabajando los otros colaboradores del proyecto. Los administradores tienen control detallado sobre qué puede hacer cada usuario, y es mucho más fácil administrar un CVCS que tener que lidiar con bases de datos locales en cada cliente.

## **2.4. Sistemas de Control de Versiones Distribuidos**

“Cada usuario tiene su propio repositorio. Los distintos repositorios pueden intercambiar y mezclar revisiones entre ellos. Es frecuente el uso de un repositorio, que está normalmente disponible, que sirve de punto de sincronización de los distintos repositorios locales. Ejemplos: Git y Mercurial.” (Aulas software libre, 2021). Estos ofrecen soluciones para los problemas que han sido mencionados. En un sistema de Control de Versiones Distribuido, los clientes no solo descargan la última copia instantánea de los archivos, sino que se replica completamente el repositorio. De esta manera, si un servidor deja de funcionar y estos sistemas estaban colaborando a través de él, cualquiera de los repositorios disponibles en los clientes puede ser copiado al servidor con el fin de restaurarlo.

Entre las ventajas que tiene este sistema se encuentran:

- No es necesario estar conectado para guardar cambios.

- Posibilidad de continuar trabajando si el repositorio remoto no está accesible.
- El repositorio central está más libre de ramas de pruebas.
- Se necesitan menos recursos para el repositorio remoto.
- Más flexibles al permitir gestionar cada repositorio personal como se quiera.

## 2.5. Repositorios de código fuente

“Es el lugar en el que se almacena y se puede realizar la distribución del código de una aplicación o un programa. Este debe ser un servidor seguro que utiliza sistemas de control de versiones. Debe contener las diferentes versiones de la aplicación o programa, disponiendo de un historial con los cambios realizados sobre el original y sobre cada nueva versión. Además, debe permitir poder revertir esos cambios. Y permitir que la aplicación o programa pueda ser utilizado en paralelo por diferentes usuarios al mismo tiempo, en la misma o en sus diferentes versiones.” (Lab, 2019)

Este tipo de sistemas facilitan la interacción entre distintos miembros de un equipo de desarrollo de código, pues al permitir la elaboración de código entre distintas personas al mismo tiempo, el tiempo de codificación dura menos.

## 2.6. GUI de Sistemas de versiones

Los entornos gráficos de sistemas de control de versiones nos ayudan a utilizar de una manera más fácil estos sistemas, pues no todas las personas se sienten cómodas al trabajar con comandos, es por eso que se crearon entornos gráficos para poder trabajar esto, ejemplo de ello es GitKraken o GitHub.

“Se puede acceder y administrar Git utilizando herramientas de línea de comandos. Pero si eres nuevo en Git, entonces quizás quieras comenzar con algo más manejable.

Los clientes de la interfaz gráfica de usuario (GUI, por sus siglas en inglés) son herramientas que proporcionan una visualización alternativa para Git.” (A., 2022) Ejemplos de GUI's:

- QGit
- Gitg
- Git Force
- Sourcetree
- GitHub
- Tortoise Git
- GitUp
- GitBox

- GitX-dev
- GitKraken
- SmartGit

### 3. Conclusiones

En conclusión Los sistemas de control de versiones facilitan mucho el desarrollo de software, pues este te permite tener guardadas distintas versiones de tu proyecto, por lo cual es fácil regresar a una versión anterior en caso de encontrar algún error, al igual que por si alguna circunstancia pierdes tu proyecto, sistemas como GitHub te permiten guardarlo en internet y poder contar con tu proyecto desde cualquier parte del mundo solo con tener acceso a internet. Al igual para que sea más fácil para los recién llegados, los desarrolladores han creado clientes de interfaz gráfica de usuario Git para varias plataformas.

### 4. Bibliografía

- A., D. (08 de Febrero de 2022). *Hostinger*. Obtenido de <https://www.hostinger.mx/tutoriales/mejores-clientes-git-gui#:~:text=Git%20es%20un%20sistema%20de,contenido%20y%20c%C3%B3digo%20en%20repositorios>.
- Aulas software libre*. (14 de Octubre de 2021). Obtenido de <https://aulassoftwarelibre.github.io/taller-de-git/cvs/>
- Lab, T. B. (16 de Febrero de 2019). Obtenido de <https://theblackboxlab.com/2021/02/22/que-son-los-repositorios-de-codigo-y-cuales-son-sus-beneficios/>
- Medium*. (28 de Enero de 2017). Obtenido de <https://medium.com/@jointdeveloper/sistemas-de-control-de-versiones-qu%C3%A9-son-y-por-qu%C3%A9-amarlos-24b6957e716e>
- Opentix*. (28 de Julio de 2019). Obtenido de <https://www.opentix.es/sistema-de-control-de-versiones/>