

# Sistemas de control de versiones

lagunasgonzalezstephanie123@gmail.com

Universidad de la Sierra Sur

2022/05/09

## 1 Introducción

En la actualidad estamos acostumbrados a realizar muchas versiones de código, incluso muchas veces algunas versiones las solemos perder, obstruir la versión que se llevaba, incluimos nuevas cosas, pero también errores y preferimos regresar a la anterior donde si funcionaba y así nos la pasamos todo el tiempo avanzando y regresando, por ello, en este ensayo se les explicara un poco acerca de los sistemas de control de versiones, sus características, su arquitectura que tiene implementada, y los tipos, también se abordará un poco sobre los repositorios de código fuente, los tipos, ventajas y desventajas de estos, por último se explicará sobre el tema de entornos gráficos para el sistema de control de versiones, sus tipos, las ventajas y desventajas que se han encontrado hasta el momento, con esta información tendrán un panorama más claro acerca de estos conceptos y podrán entender muchas cosas que hasta el momento no entendían.

## 2 Desarrollo

### 2.1 Sistemas de control de versiones

#### 2.1.1 Características

- Guarda el registro histórico de las modificaciones realizadas a cada archivo y/o carpeta.
- Añade trazabilidad al desarrollo de software. Por lo tanto, permite identificar cambios entre una versión y otra.
- Colaboración de varios desarrolladores en el mismo proyecto.

#### 2.1.2 Arquitectura

**Locales:** Consiste en crear copias locales de tus archivos en otra carpeta de tu computadora. Este sistema se caracteriza por ser fácil de implementar. Es decir, copias, pegas y renombas directorios.

**Centralizados:** Se pasan las copias en un servidor dedicado. Es decir, un sistema centralizado es un sistema cliente – servidor.

**Distribuidos:** Tiene una copia central, pero cada programador puede replicar una copia local del código fuente. Sobre todo, al ser un sistema distribuido, no requieres estar conectado al servidor central para poder trabajar.

#### 2.1.3 Tipos

- Git: 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.
- CVS: Es un modelo de repositorio cliente-servidor donde varios desarrolladores pueden trabajar en el mismo proyecto en paralelo.
- Apache Subversion (SVN): Es un modelo de repositorio cliente-servidor donde los directorios están versionados junto con las operaciones de copia, eliminación, movimiento y cambio de nombre.

## 2.2 Repositorios de códigos fuente

### 2.2.1 Tipos

**GitHub:** software que permite realizar un seguimiento de los cambios y una colaboración fluida.

**BitBucket:** incorpora Git que habilita y permite el acceso, el almacenamiento, el uso compartido y la revisión de archivos de código sin problemas.

**FuenteForge:** proporciona muchas funciones, como seguimiento de errores integrado, duplicación de descargas para equilibrio de carga, etc.

### 2.2.2 Ventajas

- Permite el trabajo en paralelo de dos o más usuarios de una aplicación o programa sin que se den por válidas versiones con elementos que entren en conflicto entre sí.
- La seguridad de un repositorio de código en un servidor es máxima ya que este garantiza la misma mediante diferentes métodos avanzados.

## 2.3 Entornos gráficos para el sistema de control de versiones

### 2.3.1 Ventajas

- Todos los cambios que se realizan en el repositorio quedan registrados por medio de un historial.
- Con la creación de ramas y fusiones, todos los integrantes del equipo pueden trabajar al mismo tiempo.

### 2.3.2 Desventajas

- Muchos comandos y el significado de los comandos.
- Complejidad, recomendado para proyectos grandes.

### 3 Conclusiones

Con toda la información obtenida, se pueden llegar a variedad de conclusiones, por ejemplo que los sistemas de control de versiones nos pueden dar una gran ayuda en nuestro trabajo, debido a que estos nos guardan los registros históricos de las modificaciones y estos identifican los cambios que existen entre una versión y otra, además que estos contienen una arquitectura, lo cual fue nuevo para mí, ya que yo aún no sabía de qué existían tres tipos, los locales, centralizados y distribuidos. También que los repositorios de código fuente existen una gran variedad, pero cada uno funciona de una manera diferente, claro que no cambian tanto, algunas cosas son muy similares, en tanto a sus ventajas y desventajas son muy parecidas, ya que estos tienen un fin en común, ahora ya se conoce un poco sobre el control de versiones y algunos tipos que nos pueden facilitar el trabajo.

### 4 Referencias

- [1] Galvan, O. (2022). Lo que muchos no saben sobre los sistemas de control de versiones.. OsvaldoGR. Retrieved 9 May 2022, from <https://osvaldogalvan.com/blog/lo-que-muchos-no-saben-sobre-los-sistemas-de-control-de-versiones>.
- [2] 5 softwares de control de versiones. Drauta. (2022). Retrieved 9 May 2022, from <https://www.drauta.com/5-softwares-de-control-de-versiones#:~:text=1%20Git%3A%20es%20una%20de%20las%20mejores%20herramientas>.
- [3] (2022). Retrieved 9 May 2022, from <https://codepre.com/los-5-principales-repositorios-de-codigo-fuente.html?msclkid=973d08d6cf3b11ecaf4ab8a68dc080ff>.
- [4] Git - Acerca del Control de Versiones. Git-scm.com. (2022). Retrieved 9 May 2022, from <https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Acerca-del-Control-de-Versiones>.