

# Sistema de control de versiones

Floriceli Gonzalez Ruiz

gonzalezruizfloriceli183@gmail.com

Universidad de la Sierra Sur

08/05/2022

## 1 Introducción

A lo largo de los años el desarrollo de nuevas tecnologías, así como los sistemas de control de versiones nos facilitaron tener un mejor control de las versiones que vamos realizando día a día de algún proyecto que estemos realizando ya que estas herramientas de control de versiones nos permitieron tener muchos beneficios para el desarrollo de algún proyecto ya que este nos permite que guardemos una copia de nuestro proyecto por si se hace un cambio que no queríamos o que nos ocasiono error.

## 2 Desarrollo

### 2.1 Sistema de control de versiones

Los sistemas de control de versiones son una categoría de herramientas de software que ayudan a registrar los cambios realizados en los archivos al realizar un seguimiento de las modificaciones realizadas en el código.

#### **Características**

Mejora la velocidad de desarrollo del proyecto al proporcionar una colaboración eficiente, Aprovecha la productividad, acelera la entrega de productos y las habilidades de los empleados a través de una mejor comunicación y asistencia, Reduzca las posibilidades de errores y conflictos mientras desarrolla el proyecto a través de la trazabilidad de cada pequeño cambio.

Los empleados o colaboradores del proyecto pueden contribuir desde cualquier lugar independientemente de las diferentes ubicaciones geográficas a través de este VCS. Para cada colaborador diferente del proyecto, se mantiene una copia de trabajo diferente y no se fusiona con el archivo principal a menos que se valide la copia de trabajo. El ejemplo más popular es Git, Helix core, Microsoft TFS, Ayuda en la recuperación en caso de cualquier desastre o situación contingente, Nos informa sobre quién, qué, cuándo y por qué se han realizado los cambios.

#### **Arquitectura**

Un sistema de control de versiones generalmente tiene tres requisitos funcionales básicos, a saber:

Almacenamiento de contenido, Seguimiento de cambios en el contenido (historial, incluidos los metadatos combinados) y Distribuir el contenido y la historia con los colaboradores

### **Tipos de sistemas de control de versiones**

**Sistemas Locales de Control de Versiones:** Es una de las formas más sencillas y cuenta con una base de datos que mantiene todos los cambios en los archivos bajo control de revisión. RCS es una de las herramientas VCS más comunes. Mantiene conjuntos de parches (diferencias entre archivos) en un formato especial en el disco. Al sumar todos los parches, puede volver a crear el aspecto de cualquier archivo en cualquier momento.

**Sistemas de control de versiones centralizados:** los sistemas de control de versiones centralizados contienen solo un repositorio globalmente y cada usuario debe comprometerse a reflejar los cambios propios en el repositorio. Es posible que otros vean sus cambios al actualizar.

**Sistemas de control de versiones distribuidos:** Los sistemas de control de versiones distribuidos contienen múltiples repositorios. Cada usuario tiene su propio repositorio y copia de trabajo. Solo confirmar sus cambios no le dará a otros acceso a sus cambios. Esto se debe a que la confirmación reflejará esos cambios en su repositorio local y debe enviarlos para que sean visibles en el repositorio central. De manera similar, cuando actualiza, no obtiene los cambios de otros a menos que primero haya extraído esos cambios en su repositorio.

## **2.2 Repositorios de código fuente**

Un repositorio de código fuente es un archivo con el código, así como la instalación de alojamiento para estos archivos de software, donde también puede tener la documentación técnica del proyecto, páginas web, fragmentos, parches, etc. a los que se puede acceder públicamente (código abierto). ) o en privado.

### **Tipos**

Hay dos tipos principales de plataformas de repositorio de código: sistemas de control de versiones centralizados (VCS) y sistemas de control de versiones distribuidos (DVCS). El primero es un tipo más antiguo, donde solo hay un repositorio de código central y usted, como desarrollador, solo puede contribuir a ese repositorio. Si no tiene acceso a Internet y el código está ubicado en un servidor externo, mala suerte: no puede confirmar los cambios. Un ejemplo de dicho sistema es Subversion (también conocido como SVN) y CVS.

Afortunadamente, hay VCS distribuidos, que admiten varios repositorios para el proyecto. De esta forma, cada desarrollador puede tener su propio repositorio local (uno o varios), y trabajar de forma autónoma con cada uno. Dicho DVCS le permite trabajar incluso en condiciones en las que no tiene conexión a Internet (por ejemplo, en algún lugar de una playa en Tailandia).

### **Ventajas**

. Menos repositorios para instalar, configurar y realizar un seguimiento de la dirección (incluso si puede factorizar esas operaciones, ya que probablemente sean las mismas). . Una y única dirección de repositorio para usar y

buscar proyectos. . Es más fácil configurar los derechos de los usuarios sobre cada proyecto que crear un perfil para cada usuario en cada repositorio y luego configurar el acceso para cada usuario en cada repositorio. . El uso de un repositorio para cada proyecto cuando no necesita separar especialmente esos proyectos en su mayoría agrega confusión en su trabajo.

### **Desventajas**

Una desventaja menor es que envía spam a su lista de repositorios con repositorios inútiles. Por lo general, comienzo mis repositorios como privados y tan pronto como tienen algo que vale la pena mirar, los hago públicos. De esa manera, el páramo de nuevas ideas que nunca pasó de unas pocas horas retocando no distrae la atención del código interesante. Esto solo importa si espera que las personas vean su lista de repositorios para encontrar algo interesante. (*Repositorios de código fuente*,(2022))

## **2.3 Entornos graficos para el sistema de control de versiones**

Las principales ventajas de utilizar un sistema de control de versiones incluyen la optimización del proceso de desarrollo, la gestión del código para múltiples proyectos y el mantenimiento de un historial de todos los cambios dentro de un código.

Un software de control de versiones guarda todos los cambios en un repositorio. Por lo tanto, si los desarrolladores cometen un error, pueden deshacerlo. Al mismo tiempo, pueden comparar el nuevo código con versiones anteriores para resolver su queja. Esto puede reducir en gran medida los errores humanos y las consecuencias no deseadas. Una gran opción para cualquier empresa de desarrollo web en todo el mundo.

### **Tipos**

1. GitHub GitHub ayuda a los equipos de software a colaborar y mantener el historial completo de cambios de código. Puede realizar un seguimiento de los cambios en el código, retroceder el reloj para deshacer errores y compartir sus esfuerzos con otros miembros del equipo.

Es un repositorio para albergar proyectos Git. Para aquellos que se preguntan ¿Qué es Git? Es un sistema de control de versiones de código abierto que presenta ramificaciones locales, múltiples flujos de trabajo y áreas de preparación convenientes. El control de versiones de Git es una opción fácil de aprender y ofrece una velocidad de operación más rápida.

2. habichuelas mágicas Beanstalk es una opción ideal para aquellos que necesitan trabajar desde lugares remotos. Este software se basa en el navegador y la nube, lo que permite a los usuarios codificar, confirmar, revisar e implementar mediante un navegador. Se puede integrar con plataformas de mensajería y correo electrónico para colaboraciones eficientes relacionadas con códigos y actualizaciones. Es compatible con Git y SVN y viene con funciones de análisis integradas.

3. Por fuerza Perforce ofrece capacidades de control de versiones a través de su HelixCore. El HelixCore viene con una plataforma única para una colaboración en equipo fluida y soporte para flujos de trabajo de desarrollo centralizados y distribuidos.

**Ventajas** Las principales ventajas de utilizar un sistema de control de versiones incluyen la optimización del proceso de desarrollo, la gestión del código para múltiples proyectos y el mantenimiento de un historial de todos los cambios dentro de un código.

Un software de control de versiones guarda todos los cambios en un repositorio. Por lo tanto, si los desarrolladores cometen un error, pueden deshacerlo. Al mismo tiempo, pueden comparar el nuevo código con versiones anteriores para resolver su queja. Esto puede reducir en gran medida los errores humanos y las consecuencias no deseadas. Una gran opción para cualquier empresa de desarrollo web en todo el mundo.

### **Desventajas**

Es más complejo que los sistemas centralizados tradicionales porque entran en juego más repositorios, más operaciones y más posibilidades para trabajar en equipo, que hay que decidir. La curva de aprendizaje es empinada. Lo básico lo aprendes enseguida, pero la realidad te demuestra que no es suficiente "tocar de oído" con él. La documentación es tan compleja que muchas veces no resulta de ayuda. Los comandos y algunos conceptos que usa pueden llegar a ser confusos, al igual que algunos mensajes que muestra. Por defecto, se lleva mal con archivos binarios muy grandes, como vídeos o documentos gráficos muy pesados. Por suerte existen soluciones para ello (Git LFS).

*(Entornos graficos para el sistema de control de versiones,(2022))*

## **3 Conclusiones**

Como conclusion, las herramientas que se mencionan en este documento nos permite guardar la diferente versiones de nuestro proyecto a los largo de su creacion ya que estos nos hace tener un respaldo y version anterior de nuestro proyecto sin cambio por si ocurre un problema y una falla en el que no archivo anterior no tenia, por eso es muy importante ir guardando todas las versiones de nuestro proyecto.

## **4 Referencias**

Version Control Systems - GeeksforGeeks. GeeksforGeeks.

(2022). Retrieved 9 May 2022, from <https://www.geeksforgeeks.org/version-control-systems/>.

How to Choose Code Repository for Your Project [Prices Updated] - HUSPI. HUSPI.

(2022). Retrieved 9 May 2022, from <https://huspi.com/blog-open/software-code-repositories/#:~:text=hosting%20service%20platforms>

.-,A%20source-code%20repository%20is%20an%20archive%20with%20the%20code,open-source)%20or%20privately.

Top 10 Version Control Systems | HackerNoon. Hackernoon-com.translate.goog.

(2022). Retrieved 9 May 2022, from [https://hackernoon-com.translate.goog/top-10-version-control-systems-4d314cf7adea?\\_x\\_tr\\_sl](https://hackernoon-com.translate.goog/top-10-version-control-systems-4d314cf7adea?_x_tr_sl)

=en&\_x\_tr\_tl=es&\_x\_tr\_hl=es&\_x\_tr\_pto=wapp.

