

0.1 INTRODUCCIÓN

Los sistemas de control de versiones son una herramienta que permiten gestionar las modificaciones realizados en archivos de texto. Facilitando así la identificación de todos los cambios existentes entre dos versiones de un mismo archivo. A la hora de trabajar con código esta información es clave para poder identificar y corregir los problemas.

0.2 DESARROLLO

0.3 Sistema de Control de Versiones

Es la práctica de rastrear y gestionar los cambios en el código de software. Los sistemas de control de versiones son herramientas de software que ayudan a los equipos de software a gestionar los cambios en el código fuente a lo largo del tiempo. [1]

0.4 Características

Resolución de conflictos: Es muy probable que los miembros del equipo tengan la necesidad de realizar cambios en el mismo archivo de código fuente al mismo tiempo. Un VCS monitoriza y ayuda a poder resolver los conflictos entre varios desarrolladores.

Revertir y deshacer los cambios en el código fuente: Al empezar a monitorizar un sistema de archivos de códigos fuente, existe la posibilidad de revertir y deshacer rápidamente a una versión estable conocida.

Copia de seguridad externa del código fuente: Se debe crear una instancia remota del VCS que se puede alojar de forma externa con un tercero de confianza y con ello, se conservará una copia del código fuente.

0.5 Tipos de Sistemas de Control de Versiones

Git: 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.

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.

Apache Subversion (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.

Mercurial: 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.

Monotone: 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 muy eficiente y robusto llamado Netsync.

0.6 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. [?]

0.7 Tipos

GitHub: La plataforma brinda capacidades para escribir mejor código, conversaciones fluidas y revisiones de código, colaboraciones en equipo, integración continua y canalizaciones de implementación continua para probar,

lanzar e implementar software sin el uso de sitios web o plataformas de terceros, gestión de proyectos junto con código en problemas y proyectos, seguridad de código, acceso de usuario controlado, diversas capacidades de alojamiento de código para permitir el alojamiento donde sea necesario, etc.

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

FuenteForge: SourceForge proporciona muchas funciones, como seguimiento de errores integrado, un wiki para documentación, duplicación de descargas para equilibrio de carga, código host para Git, Mercurial y Subversion, listas de correo de desarrolladores y usuarios, funciones foros de discusión encadenados, reseñas escritas por usuarios y calificaciones, foros de soporte al usuario, un boletín de noticias, la capacidad de ver el historial de confirmaciones como un gráfico, un microblog para publicar actualizaciones de proyectos, etc.

0.8 Beneficios

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 de ciberseguridad y la creación constante de copias de seguridad.

Permite disponer y acceder a un historial de cambios. Cada programador tiene la obligación de señalar qué cambios ha realizado, quién los ha llevado a cabo y también cuando se hicieron. Esto permite un mayor control sobre cada versión, permite corregir cambios y facilita que cada cambio realizado se vea como un nuevo estado.

Facilita el entendimiento del proyecto, sus avances y el estado actual de la última versión. Algo posible gracias a que cada pequeño cambio realizado por un programador debe ir acompañado por un mensaje explicativo de la tarea realizada. Así evita a los demás programadores perder tiempo cuestionándose el por qué de los cambios realizados. Y facilita la corrección de errores si los hubiera y son detectados por otro programador.

0.9 Entorno Gráfico para el Sistemas de Control de Versiones

GitHub Desktop: Este programa es una extensión del flujo de trabajo de GitHub a través de una interfaz gráfica de usuario que nos permita administrar su código sin necesidad de usar la línea de comandos.

GitKraken: Se trata de una aplicación para Git compatible con sitios como GitHub, Bitbucket y Gitlab. Cuenta con versiones tanto gratuitas, indicada para equipos pequeños como versiones premium y empresariales para equipos más grandes.

0.10 CONCLUSIONES

Los sistemas de control de versiones (VCS por sus siglas en inglés) se utilizan ampliamente en diversos proyectos de desarrollo de software en la actualidad, debido a su flexibilidad, escalabilidad, potencia y facilidad de uso.

Bibliography

- [1] Atlassian. (z.d.). Qué es el control de versiones — Atlassian Git Tutorial. Geraadpleegd op 9 mei 2022, van <https://www.atlassian.com/es/git/tutorials/what-is-version-control>
- [2] The Black Box Lab. (2021, 22 februari). Qué son los repositorios de código y cuáles son sus beneficios. Geraadpleegd op 9 mei 2022, van <https://theblackboxlab.com/2021/02/22/que-son-los-repositorios-de-codigo-y-cuales-son-sus-beneficios/>: