

Sistema de control de versiones

Heber Esaú Hernández Ramírez
heberesauhernan@gmail.com
Universidad de la Sierra Sur

Mayo 2022

1 Introducción

En el siguiente proyecto hablaremos sobre los sistemas de control de versiones que podemos encontrar en todas nuestras plataformas y como implementarlos en nuestro proyecto, uno de los mas populares que existen en estos tiempos son gitHub entre otros. Sistema de control de versiones. Software que administra el acceso a un conjunto de ficheros, y mantiene un historial de cambios realizados. El control de versiones es útil para guardar cualquier documento que cambie con frecuencia, como código fuente, documentación o ficheros de configuración. Un sistema de control de versiones debe proporcionar: Mecanismo de almacenamiento de los elementos que deba gestionar. Posibilidad de realizar cambios sobre los elementos almacenados. Registro histórico de las acciones realizadas con cada elemento o conjunto de elementos.

2 Desarrollo

2.1 Tipos de sistemas de control de versiones

Los sistemas de control de versiones se pueden clasificar en 2 grandes grupos

Centralizados: En un sistema de control de versiones centralizado todos nuestros fuentes y sus versiones están almacenados en un único directorio (llamado repositorio de fuentes) de un ordenador (un servidor). Todos los desarrolladores que quieran trabajar con esos fuentes, deben pedirle al sistema de control de versiones una copia local para trabajar. En ella realizan todos sus cambios y cuando están listos y funcionando, le dicen al sistema de control de versiones que guarde los fuentes modificados como una nueva versión. Algunos ejemplos son CVS y subversión.

Distribuidos: En un sistema de control de versiones distribuido no hay un repositorio central. Todos los desarrolladores tienen su propia copia del repositorio, con todas las versiones y toda la historia. Por supuesto, según van desarrollando y haciendo cambios, sus fuentes y versiones van siendo distintas unas de otras. Sin embargo, los sistemas de control de versiones distribuidos permiten que en cualquier momento dos desarrolladores cualesquiera puedan "sincronizar" sus repositorios. Si uno de los desarrolladores ha tocado determinados fuentes y el otro no, los modificados se convierten en la versión más moderna. Ejemplos: Git y Mercurial.

Funcionamiento: Todos los sistemas de control de versiones

se basan en disponer de un repositorio, que es el conjunto de información gestionada por el sistema. Este repositorio contiene el historial de versiones de todos los elementos gestionados. Cada uno de los usuarios puede crearse una copia local duplicando el contenido del repositorio para permitir su uso. Es posible duplicar la última versión o cualquier versión almacenada en el historial. Este proceso se suele conocer como checkout o desproteger. Para modificar la copia local existen dos semánticas básicas: Exclusivos Para poder realizar un cambio es necesario marcar en el repositorio el elemento que se desea modificar y el sistema se encargará de impedir que otro usuario pueda modificar dicho elemento.

Colaborativos En el que cada usuario se descarga la copia, la modifica, y el sistema automáticamente combina las diversas modificaciones. El principal problema es la posible aparición de conflictos que deban ser solucionados manualmente o las posibles inconsistencias que surjan al modificar el mismo fichero por varias personas no coordinadas. Además, esta semántica no es apropiada para ficheros binarios.

2.2 Repositorios de código fuente

El repositorio de código fuente está diseñado para guardar de forma segura grandes cantidades de código fuente de soluciones de software o páginas web de los programadores, desarrolladores de software y diseñadores. Dependiendo de la naturaleza del código, los ingenieros de software pueden compartir estos archivos con los equipos de trabajo o pueden compartirlos públicamente con todos, o pueden restringir el acceso a los archivos solo para ellos. Hay muchos repositorios de código fuente eficientes disponibles para realizar estas tareas. En este artículo, analizamos los repositorios de código fuente mejor valorados por usuarios y revisores de todo el mundo.

2.3 GitHub

Fundado en 2008, GitHub arrasó en el mercado muy poco tiempo después de su lanzamiento. GitHub, el repositorio de código favorito de casi todos los ingenieros de software del mundo, ofrece alojamiento en Internet y control de versiones mediante Git, un software que permite realizar un seguimiento de los cambios y una colaboración fluida.

GitHub viene equipado con muchas características admirables y eficientes. Para enumerar algunos, 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.

2.4 BitBucket

Desarrollado por los gigantes tecnológicos, Atlassian en 2008, BitBucket es otro repositorio de código muy aclamado. Al igual que GitHub, Bitbucket también incorpora Git que habilita y permite el acceso, el almacenamiento, el uso compartido y la revisión de archivos de código sin problemas. Bitbucket Server, que también se conocía inicialmente como Stash, combina el servidor Git y una solución de interfaz web personalizada y de diseño propio desarrollada

con Java y Apache Experto. BitBucket ofrece a sus usuarios muchas funciones diversas, como la capacidad de extraer solicitudes con revisión de código y comentarios, Bitbucket Pipelines, verificación en dos pasos y verificación requerida en dos pasos, inclusión en la lista blanca de IP

2.5 FuenteForge

Lanzado en 1999, SourceForge es un servicio diverso de gestión de proyectos de software de código abierto. Al ser la plataforma más larga del mercado, la plataforma es la primera opción de muchos desarrolladores de software de todas las plataformas de sistemas operativos, es decir, usuarios de Linux, Windows y macOS, todos prefieren SourceForge para alojar sus aplicaciones de software.

Junto con un repositorio de código fuente de alto funcionamiento, 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. Al ser una plataforma de código abierto, los usuarios pueden tener en sus manos el paradigma de forma gratuita.

2.6 Entornos graficos

El control de versiones, también conocido como "control de código fuente", 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. 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. Son especialmente útiles para los equipos de DevOps, ya que les ayudan a reducir el tiempo de desarrollo y a aumentar las implementaciones exitosas. Un buen software de control de versiones soporta el flujo de trabajo preferido de un desarrollador sin imponer una forma determinada de trabajar. Idealmente, también funciona en cualquier plataforma, en vez de ordenar qué sistema operativo o cadena de herramientas deben utilizar los desarrolladores. Los sistemas de control de versiones excepcionales facilitan un flujo sencillo y continuo de cambios en el código en vez del mecanismo frustrante y burdo del bloqueo de archivos, que da luz verde a un desarrollador a expensas de bloquear el progreso de los demás. El software de control de versiones es una parte esencial del día a día de las prácticas profesionales del equipo de software moderno. Los desarrolladores de software individuales que están acostumbrados a trabajar con un sistema de control de versiones potente en sus equipos suelen reconocer el increíble valor que el control de versiones también les da incluso en los proyectos pequeños en los que trabajan solos. Una vez acostumbrados a las potentes ventajas de los sistemas de control de versiones, muchos desarrolladores no se plantearían trabajar sin ellos incluso para los proyectos que no son de software. Utilizar un software de control de versiones es una práctica recomendada para los equipos de software y de DevOps de alto rendimiento. El control de versiones también ayuda a los desarrolladores

a moverse más rápido y permite que los equipos de software mantengan la eficiencia y la agilidad a medida que el equipo se escala para incluir más desarrolladores.

3 Conclusiones

El tema es muy interesante y con esto aprendemos muchas cosas nuevas de las herramientas las cuales se pueden utilizar para la creacion y la gestión de diversos proyectos informaticos. Aprendimos varias formas de hacer un proyecto o mejor dicho aprendimos de más herramientas las cuales antes de este trabajo no sabiamos cuales existian, el proyecto nos sirvio mucho para nuestro aprendizaje y para nuestro crecimiento en la parte de la teoria. Pero tambien en la practica ya que ahora sabemos como podemos crear nuestros proyectos

4 Referencias

Los cinco principales repositorios de codigo fuente. Recuperado el 09 de mayo de 2022, de <https://codepre.com/los-5-principales-repositorios-de-codigo-fuente.html?msclkid=da931af7cf5e11ec8c7>

Sistema de control de versiones. (2022, 09 mayo). Control de versiones. Recuperado 09 de mayo de 2022, de https://es.wikibooks.org/wiki/Manual_b

Qué es Gradle y sus características?. (2022, 19 febrero). *osmosislatina*. Recuperado el 18 de abril de 2022, de <https://openwebinars.net/blog/que-es-gradle/>