

Sistema de control de versiones

Maario Hecxai Valencia Reyes

2022/05/09

1 Introducción

Los sistemas de control de versiones son una tecnología que funciona como herramienta, con la cual se puede administrar y gestionar los cambios que se realizan en archivos de tipo texto. Los archivos que se utilizan en esta herramienta se almacenan en un repositorio.

2 Desarrollo

2.1 Sistema de control de versiones

2.1.1 Características

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]

[?]



Figure 1: Sistema de control de versiones

2.1.2 Arquitectura

clasificación de los sistemas de control de versiones:

- Distribuidos: cada usuario tiene su propio repositorio. Los distintos repositorios pueden intercambiar y mezclar revisiones entre ellos.
- Centralizados: existe un repositorio centralizado de todo el código, del cual es responsable un único usuario. [1]

2.1.3 Tipos

Sistemas de Control de Versiones Locales

Los sistemas de control de versiones locales en vez de mantener las versiones como archivos independientes, los almacenaban en una base de datos.

Sistemas de Control de Versiones Centralizados

Para facilitar la colaboración de múltiples desarrolladores en un solo proyecto los sistemas de control de versiones evolucionaron: estos se almacenaban en un servidor.

Sistemas de Control de Versiones Distribuidos

Un solo repositorio centralizado y optó por darle a cada desarrollador una copia local de todo el proyecto, de esta manera se construyó una red distribuida de repositorios.

2.2 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.

2.2.1 Tipos

Los dos repositorios de código más utilizados son Git y SVN.

2.2.2 Ventajas de Git

1. Es gratis de utilizar.
2. Es muy rápido. Por ejemplo los comandos `git pull/ git push`.
3. Puede manejar grandes proyectos.

2.2.3 Desventajas de Git

1. La documentacion es complicada.

2.2.4 Ventajas de SVN

1. Se sigue la historia de los archivos y directorios a través de copias y renombrados.
2. Modificaciones atómicas.
3. La creacionde branch, y tags son operaciones eficientes.

2.2.5 Desventajas de SVN

1. La autorización de acceso depende de la ruta de acceso.

2.3 Entornos gráficos para el sistema de control de versiones

2.3.1 Tipos

Es otro sistema de control de versiones distribuido como mercurial y Git, tiene una interfaz amigable, se hace llamar a si mismo el control de versiones para

humanos.

2.3.2 Ventajas Bazaar(bzr)

1. Se puede utilizar para adaptarse a cualquier tipo de proyecto.
2. Es fácil de modificar muy intuitivo.
3. Integrable se puede añadir proyectos existentes.

2.3.3 Desventajas de Bazaar(bzr)

1. Se trabaja siempre con acceso al servidor remoto.
2. No es muy popular.

3 Conclusiones

Una herramienta de software que te permita gestionar las versiones. Esta forma práctica que los usuarios toman respecto a los sistemas de control de versiones, ayudará a los desarrolladores tener eficiencia, agilidad y tiempo para poder tener trabajos colaborativos de mayor.

References

- [1] Atlassian. (s. f.). *Qué es el control de versiones* Atlassian Git Tutorial. (27 agosto 2013).
- [2] Atlassian. (s. f.). *Software de control de versiones* descripción general. Bitbucket. (12 de enero 2019).