

Introducción a GIT



¿Qué es el control de versiones?

El control de versiones es la gestión del estado de los archivos mediante el registro de las actualizaciones del contenido de un determinado archivo o carpeta.

El control de versiones se realiza generalmente mediante una herramienta denominada "sistema de control de versiones", que se describe a continuación.

Si intenta realizar el control de versiones sin utilizar una herramienta, ocurrirá lo siguiente

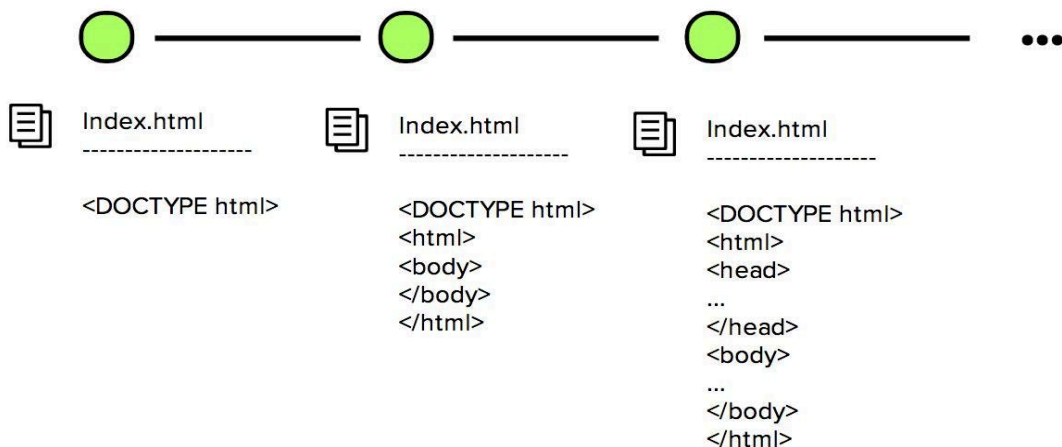
- No puedo decir cuál es el último archivo.
- Nombres de archivo incoherentes y difíciles de gestionar
- No está inmediatamente claro qué cambios se han hecho en el archivo desde el anterior.



Cambio 1

Cambio 2

Cambio 3



¿Por qué utilizar un sistema de control de versiones?

Los sistemas de control de versiones solucionan estos problemas y ofrecen varias ventajas:

Ventajas:

1. Es fácil registrar los cambios en los archivos.
2. Fácil restauración a una versión grabada específica.

Desventajas:

1. La configuración inicial y la construcción para la operación pueden llevar tiempo.
2. En proyectos de gran escala, las operaciones como la bifurcación y la integración pueden volverse complejas.

Tipos de sistemas de control de versiones

Existen dos tipos principales de sistemas de control de versiones: centralizados y descentralizados.

1. Control de versiones centralizado

La gestión en un sistema centralizado es sencilla en su estructura e implica dos tareas principales

1. Reflejar las actualizaciones en el servidor
2. Obtener actualizaciones del servidor.



Ventajas

1. Debido a que el repositorio central está gestionado de manera centralizada, la gestión es fácil."
2. Dado que el historial se guarda únicamente en el servidor, el consumo de recursos del lado del cliente es bajo."

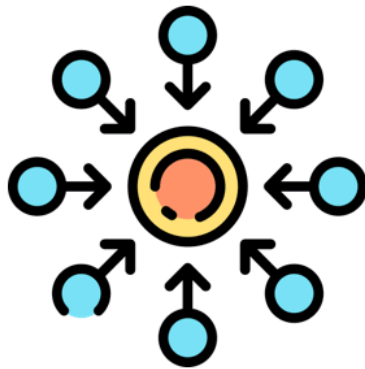
Desventajas

1. Reflejar un fichero en estado incompleto afecta inmediatamente al fichero de cada trabajador que recibe el cambio.
2. El servidor debe estar constantemente conectado durante el trabajo.

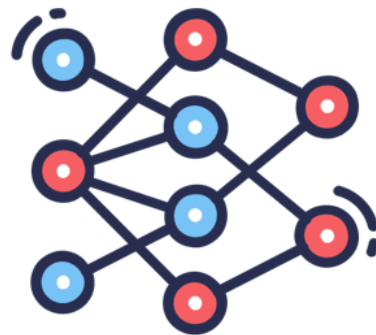
Un sistema de control de versiones centralizado típico es el servicio [Subversion](#).

2. Control de versiones descentralizado

En este modelo, cada desarrollador tiene una copia completa del historial de versiones en su computadora. No es necesario estar siempre conectado al servidor central, ya que los cambios locales se sincronizan solo cuando es necesario.



Centralized



Distributed

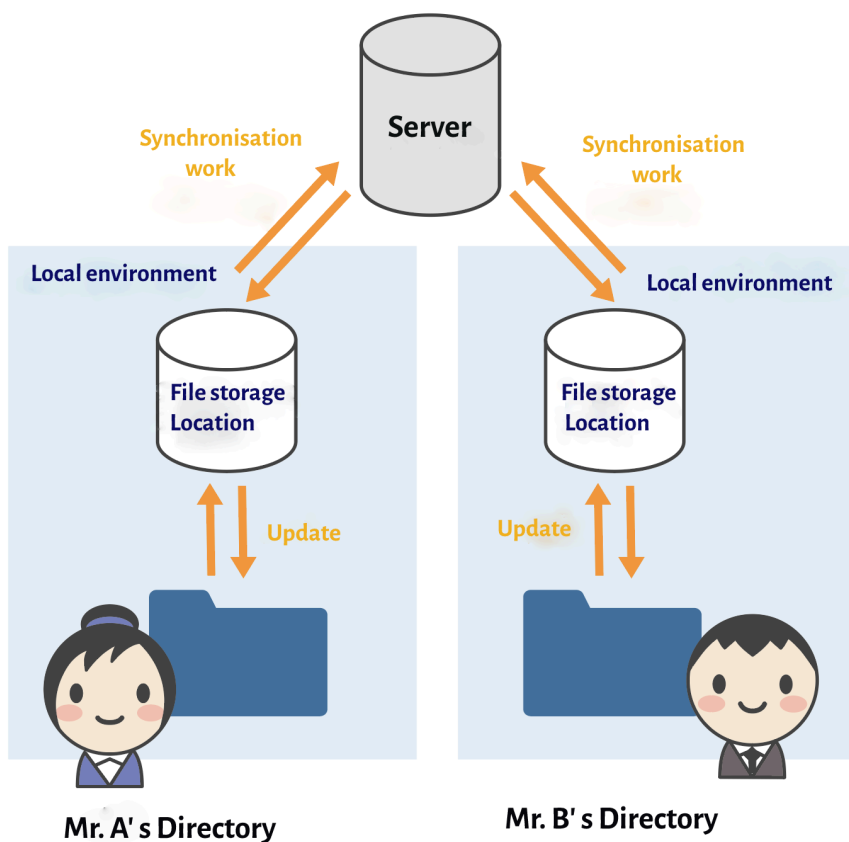
Arquitectura cliente-servidor

Se trata de un método para lograr la coherencia global duplicando el almacenamiento de archivos en el entorno de cada trabajador y sincronizando los cambios.

A diferencia del método centralizado, los cambios no afectan a otros trabajadores hasta que finaliza el proceso de sincronización.

Esto significa que el archivo sólo puede modificarse dentro del entorno del trabajador, y que puede crearse un registro de esa versión o destruir los cambios.

Si hay diferencias en los cambios realizados con otros trabajadores en el momento de la sincronización, pueden corregirse en ese momento.



Claro, aquí tienes la traducción al español:

"Existen servicios de sistemas de control de versiones distribuidos como [Git](#) y [Mercurial](#)".

Las ventajas y desventajas de los sistemas distribuidos son las siguientes:

Ventaja

1. El repositorio local tiene una copia completa para que puedas trabajar sin conexión a Internet
2. Las operaciones flexibles de ramificación y fusión permiten que cada desarrollador trabaje de forma independiente
3. Incluso si el servidor central deja de funcionar, puedes seguir trabajando en otros repositorios

Desventaja

1. Las configuraciones y operaciones iniciales tienden a ser complicadas y los costos de aprendizaje son altos.
2. Alto consumo de espacio en disco ya que cada cliente mantiene un repositorio completo

Los servicios centralizados requieren comunicación con el servidor para cada operación de actualización. Por otro lado, los servicios descentralizados no necesitan comunicarse con el servidor, excepto para las tareas de sincronización. Por lo tanto, los servicios descentralizados son, en general, más rápidos.

En términos de funcionalidad, Subversion tiene unos cinco años más que otros servicios y, por lo tanto, es considerado inferior.

Además, los servicios centralizados no requieren mucho tiempo de aprendizaje debido a la sencillez de sus mecanismos. Los servicios distribuidos, en cambio, requieren entender la diferencia entre trabajar dentro del propio entorno y el proceso de sincronización con otros entornos. Sin embargo, los servicios distribuidos pueden aprovechar sus características para adaptarse de manera flexible a diferentes situaciones de control de versiones.



Es importante destacar que Git es, con diferencia, el sistema de control de versiones más popular.

Ahora aprenderemos sobre Git, el estándar de control de versiones en el desarrollo de software moderno.

Resumen

1. Se puede utilizar un sistema de control de versiones para gestionar fácilmente las versiones modificadas de los archivos mediante el registro y la restauración de los cambios realizados en los archivos
2. Los sistemas de control de versiones incluyen servicios centralizados y descentralizados, pero Git descentralizado acapara actualmente la mayor parte de los usuarios.

Documentos e informaciones

1. [Subversión](#)
2. [Git](#).
3. [.Mercurial](#).