



¿QUÉ ES GIT?

Git es un sistema de control de versiones distribuido, gratuito y de código abierto, diseñado para manejar todo, desde proyectos pequeños hasta proyectos muy grandes con rapidez y eficiencia.

Git es fácil de aprender y tiene una huella pequeña con un rendimiento increíblemente rápido. Supera las herramientas de SCM como Subversion, CVS, Perforce y ClearCase con características como bifurcaciones locales baratas, áreas de preparación convenientes y múltiples flujos de trabajo.

Características

- **Pequeño y rápido:** Git es rápido. Con Git, casi todas las operaciones se realizan localmente, lo que le otorga una gran ventaja de velocidad en los sistemas centralizados que constantemente tienen que comunicarse con un servidor en algún lugar.

Git fue creado para funcionar en el kernel de Linux, lo que significa que ha tenido que manejar grandes repositorios desde el primer día. Git está escrito en C, lo que reduce la sobrecarga de los tiempos de ejecución asociados con lenguajes de nivel superior. La velocidad y el rendimiento han sido un objetivo primordial de diseño del Git desde el principio.

- **Repartido**

Una de las mejores características de cualquier SCM distribuido, incluido Git, es que está distribuido. Esto significa que en lugar de hacer un "checkout" de la punta actual del código fuente, se hace un "clon" de todo el repositorio.

Copias de seguridad múltiples: Esto significa que incluso si está utilizando un flujo de trabajo centralizado, cada usuario tiene esencialmente una copia de seguridad completa del servidor principal. Cada una de estas copias podría ser empujada hacia arriba para reemplazar el servidor principal en caso de una falla o corrupción. En efecto, no hay un punto único de falla con Git a menos que haya solo una copia del repositorio.

Cualquier flujo de trabajo: Debido a la naturaleza distribuida de Git y al excelente sistema de ramificación, se puede implementar una cantidad casi infinita de flujos de trabajo con relativa facilidad.

Flujo de trabajo estilo subversión: Un flujo de trabajo centralizado es muy común, especialmente cuando las personas hacen la transición desde un sistema centralizado. Git no le permitirá empujar si alguien lo ha hecho desde la última vez que lo buscó, por lo que un modelo centralizado en el que todos los desarrolladores empujan al mismo servidor funciona bien.

Administrador de integración de flujo de trabajo: Otro flujo de trabajo Git común involucra a un administrador de integración, una sola persona que se compromete con el repositorio 'bendito'. Varios desarrolladores luego clonan desde ese repositorio, lo empujan



a sus propios repositorios independientes, y le piden al integrador que introduzca sus cambios. Este es el tipo de modelo de desarrollo que se ve a menudo con los repositorios de código abierto o GitHub.

Flujo de trabajo de dictadores y tenientes: Para proyectos más masivos, un flujo de trabajo de desarrollo como el del kernel de Linux a menudo es efectivo. En este modelo, algunas personas ('tenientes') están a cargo de un subsistema específico del proyecto y se fusionan en todos los cambios relacionados con ese subsistema. Otro integrador (el 'dictador') puede obtener cambios solo de sus lugartenientes y luego enviarlos al repositorio 'bendito' del que todos los clones vuelven a clonar.

- **Aseguramiento de datos**

El modelo de datos que utiliza Git garantiza la integridad criptográfica de cada bit de su proyecto. Cada archivo y confirmación se suma a su suma de comprobación y se recupera cuando se retira. Es imposible sacar algo de Git que no sean los bits exactos que pones.

También es imposible cambiar un archivo, una fecha, un mensaje de confirmación o cualquier otro dato en un repositorio Git sin cambiar las ID de todo lo que aparece después. Esto significa que si tiene un ID de compromiso, puede estar seguro de que no solo su proyecto es exactamente el mismo que cuando se cometió, sino que no se cambió nada en su historial.

La mayoría de los sistemas de control de versiones centralizados no proporcionan dicha integridad de forma predeterminada.

- **Área de ensayo**

A diferencia de los otros sistemas, Git tiene algo llamado "área de preparación" o "índice". Esta es un área intermedia donde las confirmaciones se pueden formatear y revisar antes de completar la confirmación.

Una cosa que diferencia a Git de otras herramientas es que es posible configurar rápidamente algunos de sus archivos y confirmarlos sin confirmar todos los demás archivos modificados en su directorio de trabajo o tener que enumerarlos en la línea de comandos durante la confirmación.

Esto le permite escalar solo porciones de un archivo modificado. Se acabaron los días de hacer dos modificaciones lógicamente no relacionadas a un archivo antes de que se diera cuenta de que olvidó cometer una de ellas. Ahora puede configurar el cambio que necesita para la confirmación actual y configurar el otro cambio para la próxima confirmación. Esta función escala hasta tantos cambios diferentes en su archivo como sea necesario.

Por supuesto, Git también hace que sea más fácil ignorar esta función si no desea ese tipo de control; simplemente agregue una '-a' a su comando de confirmación para agregar todos los cambios a todos los archivos en el área de almacenamiento.



- **Libre y de código abierto**

Git se lanza bajo la licencia pública general GNU versión 2.0, que es una licencia de código abierto. El proyecto Git eligió usar GPLv2 para garantizar su libertad de compartir y cambiar el software libre, para asegurarse de que el software sea gratuito para todos sus usuarios.

Sin embargo, restringen el uso del término "Git" y los logotipos para evitar confusiones. Por favor, consulte la política de marcas para más detalles.

Fuente: <https://git-scm.com/>