

0. Introducción a Git

¿Qué es Git y por qué usarlo?

Git es un sistema de control de versiones distribuido que te permite gestionar y realizar un seguimiento de los cambios en archivos de código fuente durante el desarrollo de software. Fue creado por Linus Torvalds en 2005, inicialmente para el desarrollo del kernel de Linux.

Beneficios de usar Git:

- **Control de versiones:** Te permite llevar un registro detallado de todos los cambios realizados en tu proyecto, lo que facilita revertir cambios no deseados o analizar el historial de modificaciones.
- **Colaboración eficiente:** Git facilita el trabajo colaborativo entre múltiples desarrolladores, permitiendo que todos trabajen simultáneamente en diferentes partes de un proyecto sin interferir con el trabajo de los demás.
- **Copia de seguridad distribuida:** Como sistema distribuido, cada desarrollador tiene una copia completa del repositorio con todo su historial, lo cual actúa como una copia de seguridad.
- **Fusión y ramificación poderosas:** Git proporciona herramientas eficientes para manejar ramas y fusionar cambios, lo que facilita el desarrollo de nuevas características o la corrección de errores en paralelo.

Historia y Evolución de Git

- **2005:** Linus Torvalds crea Git para gestionar el desarrollo del kernel de Linux debido a la necesidad de un sistema de control de versiones más eficiente y rápido que los existentes.
- **2006:** Git se establece como el sistema de control de versiones oficial para el kernel de Linux.
- **2008:** Se lanza GitHub, una plataforma que utiliza Git como base, popularizando el uso de Git entre los desarrolladores de software.
- **2010 en adelante:** Git se convierte en el estándar de facto para el control de versiones en la industria del software, adoptado por la mayoría de las empresas de tecnología y proyectos de código abierto.

Comparación entre Git y otros sistemas de control de versiones

Sistemas centralizados (como Subversion, CVS):

- Mantienen un solo repositorio central al que los desarrolladores deben conectarse.
- Ventajas: Simplicidad y un único punto de control.
- Desventajas: Dependencia del servidor central, menos flexibilidad para el trabajo fuera de línea, y menor eficiencia en la fusión de cambios.

Sistemas distribuidos (como Git, Mercurial):

- Cada desarrollador tiene una copia completa del repositorio, incluyendo todo su historial.
- Ventajas: Mayor flexibilidad, soporte para trabajo sin conexión, fusión más eficiente, y no depende de un único punto de falla.
- Desventajas: Curva de aprendizaje inicial más pronunciada debido a la mayor complejidad de las operaciones.