INFORME SOBRE GIT

Alumno: Thaily Abigail Puente Esparta

Codigo: SV72461269

Docente: Roberto Pineda

2025

Contenido

Introducción

GIT es una herramienta de control de versiones que se ha vuelto esencial para cualquier persona que trabaje en proyectos de programación. La descubrí hace poco, y la verdad es que me ha facilitado mucho el trabajo, sobre todo cuando trabajo en equipo. GIT fue creado por Linus Torvalds en 2005, con el objetivo de gestionar de forma eficiente los cambios en el código fuente. Si alguna vez has trabajado en un proyecto con varias personas, sabes lo difícil que puede ser asegurarte de que todos tienen la versión más actualizada del código. GIT resuelve ese problema, y mucho más.

¿Qué es GIT?

GIT es, básicamente, un sistema de control de versiones que me permite registrar los cambios que realizo en los archivos, de manera que siempre puedo saber qué cambios se hicieron, cuándo y por quién. Esto es fundamental cuando estás trabajando en proyectos grandes, porque si no usáramos una herramienta como GIT, sería un caos intentar seguirle el rastro a todo lo que se ha modificado.

Lo que me encanta de GIT es que no solo me permite ver el historial de cambios, sino que también facilita el trabajo en equipo. Imagina que estoy desarrollando una aplicación con otros programadores, y cada uno trabaja en diferentes partes del código. Con GIT, todos podemos trabajar en paralelo sin pisarnos el trabajo, y luego fusionamos nuestros cambios de manera ordenada.

GIT fue creado por Linus Torvalds, y nació de su necesidad de tener una herramienta eficiente para gestionar el código de Linux. Al principio, GIT fue pensado como un sistema de control de versiones de bajo nivel, pero con el tiempo ha evolucionado y ahora es una herramienta completa que puedo usar en cualquier tipo de proyecto. Se basó en otras herramientas anteriores como BitKeeper y Monotone, pero ha superado a estas por su velocidad y eficiencia.

Comentado [TP1]: Al comienzo no sabia que git pudiera hacer tanto, ya que solo tenia un breve conociemto sobre el github. Mes sorprendiendo mucho sobre sus comandos ya que me recuerda un poco a los comados de Kali Linux

Características

- **Distribuido**: Cada copia del repositorio contiene todo el historial del proyecto.
- **Velocidad**: GIT es rápido en operaciones como confirmaciones (commits), cambios de rama (branching) y fusiones (merging).
- **Integridad**: Usa funciones hash SHA-1 para asegurar la integridad de los datos.

- **Ramas y fusiones**: Las ramas en GIT son fáciles de crear y administrar, lo que permite probar nuevas funciones sin afectar la versión principal.

¿Para qué nos sirve GIT?

GIT tiene muchas ventajas, especialmente cuando trabajas en equipo:

1. **Control de versiones:** GIT guarda cada cambio que haces en el código, así que puedes ver cómo ha evolucionado el proyecto con el tiempo. Si algo sale mal, siempre puedes volver a una versión anterior del código sin perder nada.
2. **Trabajo colaborativo:** Si estás trabajando en colaborativo, GIT te permite hacer cambios sin pisar el trabajo de los demás. Cada miembro del equipo puede trabajar en su propia "rama" y luego fusionar su trabajo con el de los demás cuando lo necesite.
3. **Seguridad:** Si tenemos algo que va mal, GIT te permite deshacer cambios fácilmente. Es como tener una máquina del tiempo para tu código, lo cual es súper útil cuando estás haciendo algo importante.
4. **Desarrollo distribuido:** La diferencia con otras herramientas, GIT no depende de un servidor central. Cada persona tiene una copia completa del proyecto en su computadora, por lo que puedes seguir trabajando incluso sin conexión a internet.

Aplicaciones que usan GIT

Hay muchas aplicaciones y plataformas que hacen uso de GIT porque es una herramienta muy usada. Algunas de las más conocidas son:

- **GitHub:** Es la más popular. Aquí puedes almacenar tu código, colaborar con otros desarrolladores y compartir tu trabajo con el mundo. Es muy usado para proyectos de código abierto.
- **GitLab:** Es muy similar a GitHub, pero con algunas características extras como la integración con DevOps (automatización de pruebas y despliegues).
- **Bitbucket:** Es usado por muchos equipos, especialmente aquellos que trabajan con herramientas de Atlassian (como Jira y Trello).
- **Visual Studio Code:** Es un editor de código que tiene una integración increíble con GIT. Desde ahí puedes hacer commits, ver el historial y trabajar con ramas sin tener que salir del editor.
- **Sourcetree:** Es una aplicación con una interfaz gráfica para GIT, que es ideal si no te gustan mucho los comandos y prefieres hacer todo con clics.

Comandos de Git

Configuración Inicial

- git config --global user.email "tuemail@mail.com"
- git config --global user.name "TuNombre"
- git config --global color.ui true

Crear o Clonar Repositorio

- Crear repositorio: git init
- Clonar repositorio: git clone <url>

Añadir Cambios

- Todos los archivos: git add .
- Uno específico: git add archivo.txt

Guardar Cambios

- git commit -m "mensaje"
- Modificar último commit: git commit --amend

Subir Cambios

- git push origin master
- Subir tags: git push --tags

Ver Cambios

- Ver historial: git log
- Ver diferencias: git diff

Deshacer Cambios

- git reset --soft HEAD^
- git reset --hard HEAD^

Remotos

- Agregar remoto: git remote add origin <url>
- Ver remotos: git remote -v

Ramas

- Crear rama: git branch nueva
- Cambiar: git checkout rama

Comentado [TP2]: Configuré mi correo y nombre

Comentado [TP3]: Este comando lo descubrí intentando resolver un problema, se que sirve para activar grupal

Comentado [TP4]: Lo uso para crear una repo y poder clonar

Comentado [TP5R4]:

Comentado [TP6]: Lo use para añadir los cambios y añadir un archivo especifico

Comentado [TP7]: Lo uso para guardar y modificar

Comentado [TP8]: Ver el historial y verificar las versiones

Comentado [TP9]: Para añadir un remoto y verificar la url

Etiquetas

- Ver: git tag
- Crear: git tag -a v1.0 -m "Versión 1.0"

Rebase

- git rebase rama
- git rebase --continue / --skip / --abort

Otros

- Estado actual: git status
- Eliminar archivo: git rm archivo.txt
- Fusionar ramas: git merge rama

Conclusión

Lo útil que es GIT para mantener todo organizado. Si trabajo solo o en equipo, siempre puedo tener control sobre las versiones de mi proyecto y colaborar con otros de manera eficiente. GIT es una herramienta esencial para cualquier desarrollador y me ha ayudado mucho en mis proyectos, tanto personales como en equipo. Es como tener un "asistente" que siempre me permite retroceder y ver el historial de cambios, lo cual es perfecto para evitar errores.

Comentado [TP10]: Lo que me sorprendió fue el hecho puedo subir mi repo desde mi mismo visual studio y no tener que hacerlo desde el el github como suelo hacerlo yo.

Bibliografía

- Git. (s. f.). *Documentación oficial de Git*. Recuperado el 19 de abril de 2025, de https://git-scm.com/docs/git
- GitBits. (s. f.). *Git info*. Recuperado el 19 de abril de 2025, de https://github.com/gitbits/git-info
- GitHub. (s. f.). *Acerca de GitHub y Git*. Recuperado el 19 de abril de 2025, de https://docs.github.com/es/get-started/start-your-journey/about-github-and-git