Identifica herramientas de versionamiento. GA7-220501096-01-AA1-EV03.

PRESENTADO POR:

LUIS EDUARDO CHALA SERRADA

**Ficha programa de formación: 2455277**



INSTRUCTOR:

[HENRY TORRES PERTUZ](https://sena.territorio.la/perfil.php?persona=1770103&grupo=1423502)

TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SOFTWARE

SERVICIO NACIONAL DE APRENDIZAJE – SENA

CENTRO DE LOGÍSTICA Y PROMOCIÓN ECOTURÍSTICA DEL MAGDALENA​

2023

Introducción:

Git es un popular sistema de control de versiones muy utilizado por desarrolladores de software y otros profesionales para gestionar los cambios en el código y otros activos digitales. Git puede utilizarse en entornos locales y remotos, y cada opción ofrece características y ventajas únicas. En este informe, compararemos Git local y Git remoto, incluyendo sus objetivos, diferencias, características y comandos clave.

Objetivo:

El objetivo de este informe es proporcionar una visión general de las diferencias entre Git local y Git remoto, incluyendo sus características, ventajas y limitaciones. Al comprender las diferencias entre estas dos opciones, los desarrolladores de software y otros profesionales pueden tomar decisiones informadas sobre qué enfoque se adapta mejor a sus necesidades específicas.

Tabla de diferencias y características entre Git local y Git remoto:

| **Características** | **Git Local** | **Git Remoto** |
| --- | --- | --- |
| Almacenamiento | Repositorio local en tu máquina. | Repositorios remotos alojados en servidores o servicios como GitHub, GitLab o Bitbucket. |
| Acceso a otros repositorios | No se puede acceder directamente a repositorios de otras máquinas. | Permite acceder a repositorios de otros desarrolladores o equipos a través de URL remotas. |
| Trabajo colaborativo | No es óptimo para el trabajo colaborativo ya que no permite compartir cambios directamente con otros desarrolladores. | Facilita el trabajo en equipo, permitiendo compartir y colaborar en el mismo proyecto con otros desarrolladores. |
| Ramas | Puedes crear y trabajar con ramas localmente. | Permite crear, gestionar y colaborar en ramas tanto localmente como en el repositorio remoto. |
| Sincronización | No hay una sincronización automática con otros repositorios. Para compartir cambios, es necesario utilizar otras herramientas o métodos manuales. | La sincronización es más sencilla, ya que puedes hacer "push" para enviar cambios y "pull" para obtener cambios del repositorio remoto. |
| Historial de cambios | Puedes revisar el historial de cambios localmente. | El historial de cambios de todos los colaboradores está disponible en el repositorio remoto. |
| Copias de seguridad | Al estar solo en el repositorio local, los datos están más expuestos a pérdidas en caso de fallo de hardware o problemas en la máquina. | Los repositorios remotos actúan como copias de seguridad y protegen tus datos en caso de problemas locales. |
| Requiere conexión a internet | No es necesario tener conexión a internet para trabajar en el repositorio local. | Es necesario tener una conexión a internet para interactuar con el repositorio remoto. |

Tabla de comandos comunes de Git local y Git remoto:

| **Comando** | **Descripción** |
| --- | --- |
| **Git Local** |  |
| **git init** | Inicializa un nuevo repositorio local. |
| **git add <archivo>** | Agrega un archivo específico al área de preparación (staging). |
| **git commit -m "mensaje"** | Crea un nuevo commit con los cambios preparados en el área de preparación. |
| **git status** | Muestra el estado actual del repositorio local, incluyendo los cambios sin preparar y los commits pendientes. |
| **git log** | Muestra el historial de commits del repositorio local. |
| **Git Remoto** |  |
| **git clone <URL>** | Clona un repositorio remoto en tu máquina local. |
| **git push** | Envía los cambios locales al repositorio remoto. |
| **git pull** | Obtiene y fusiona los cambios del repositorio remoto en tu repositorio local. |
| **git remote -v** | Muestra las URL y nombres de los repositorios remotos configurados. |
| **git fetch** | Descarga los cambios del repositorio remoto, pero no los aplica en tu rama local. |

Conclusión:

Git es un potente sistema de control de versiones que puede utilizarse en entornos locales y remotos. Cada enfoque tiene sus propias características y ventajas, y los desarrolladores de software y otros profesionales deben considerar cuidadosamente sus necesidades antes de elegir una opción sobre la otra. Al comprender las diferencias entre Git local y Git remoto, los desarrolladores pueden tomar decisiones informadas sobre qué enfoque se adapta mejor a sus necesidades específicas.

Referencias bibliográficas:

**Git verificar si repositorio remoto tiene cambios (Git pull,fetch y log)**

[Git verificar si repositorio remoto tiene cambios (Git pull,fetch y log) (unidadvirtual.com)](https://www.unidadvirtual.com/git-verificar-si-repositorio-remoto-tiene-cambios-git-pull-fetch-y-log)

**Acerca de los repositorios remotos**

[Acerca de los repositorios remotos - GitHub Enterprise Server 3.6 Docs](https://docs.github.com/es/enterprise-server@3.6/get-started/getting-started-with-git/about-remote-repositories)

2.5 Fundamentos de Git - Trabajar con Remotos

[Git - Trabajar con Remotos (git-scm.com)](https://git-scm.com/book/es/v2/Fundamentos-de-Git-Trabajar-con-Remotos)

Hoja de referencias para principiantes

[Hoja de referencia de 32 comandos Git para principiantes - Geekflare](https://geekflare.com/es/git-commands-cheat-sheet/)

**Git vs GitHub – ¿Qué es el Control de Versiones y Cómo Funciona?**

[Git vs GitHub – ¿Qué es el Control de Versiones y Cómo Funciona? (freecodecamp.org)](https://www.freecodecamp.org/espanol/news/git-vs-github-what-is-version-control-and-how-does-it-work/)