**INFORME N°1: GIT**



**1. Introducción**

Git es un sistema de control de versiones distribuido, diseñado para manejar todo tipo de proyectos, desde pequeños hasta muy grandes, con velocidad y eficiencia. Fue creado por Linus Torvalds en 2005 para el desarrollo del kernel de Linux.

Git permite a múltiples desarrolladores trabajar en el mismo proyecto de manera simultánea sin interferir entre sí, y mantiene un historial completo de los cambios realizados.

**2. ¿Qué es un sistema de control de versiones?**

Un sistema de control de versiones (VCS, por sus siglas en inglés) permite gestionar los cambios en archivos y proyectos a lo largo del tiempo. Algunas de sus funciones principales son:

* Guardar diferentes versiones del mismo archivo.
* Comparar cambios entre versiones.
* Volver a una versión anterior si es necesario.
* Colaborar con otros desarrolladores sin sobrescribir el trabajo ajeno.

**3. Características principales de Git**

* Distribuido: Cada desarrollador tiene una copia completa del repositorio, con todo el historial.
* Velocidad: Git es muy rápido al comparar, fusionar y registrar cambios.
* Ramas (branches): Facilita la creación de ramas para trabajar en nuevas funcionalidades sin afectar el código principal.
* Integridad: Utiliza SHA-1 para asegurar la integridad de los datos.
* Colaborativo: Permite que varias personas trabajen en el mismo proyecto de forma paralela y eficiente.

**4. Flujo de trabajo básico en Git**

1. Inicializar un repositorio:  
   git init – Crea un nuevo repositorio Git.
2. Agregar archivos al área de preparación:  
   git add archivo.txt – Añade archivos al área de staging.
3. Guardar cambios (commit):  
   git commit -m "Mensaje descriptivo" – Guarda un snapshot del proyecto.
4. Ver historial de commits:  
   git log – Muestra los commits realizados.
5. Crear una rama:  
   git branch nueva-rama
6. Cambiar de rama:  
   git checkout nueva-rama
7. Fusionar ramas:  
   git merge rama-a-fusionar
8. Clonar un repositorio remoto:  
   git clone https://url-del-repo.git
9. Enviar cambios al servidor remoto:  
   git push
10. Obtener cambios del servidor remoto:  
    git pull

**5. GitHub y otros servicios**

Aunque Git es una herramienta local, existen plataformas como GitHub, GitLab, y Bitbucket que permiten alojar repositorios en la nube, facilitar el trabajo en equipo, la revisión de código (pull requests), y la integración continua.

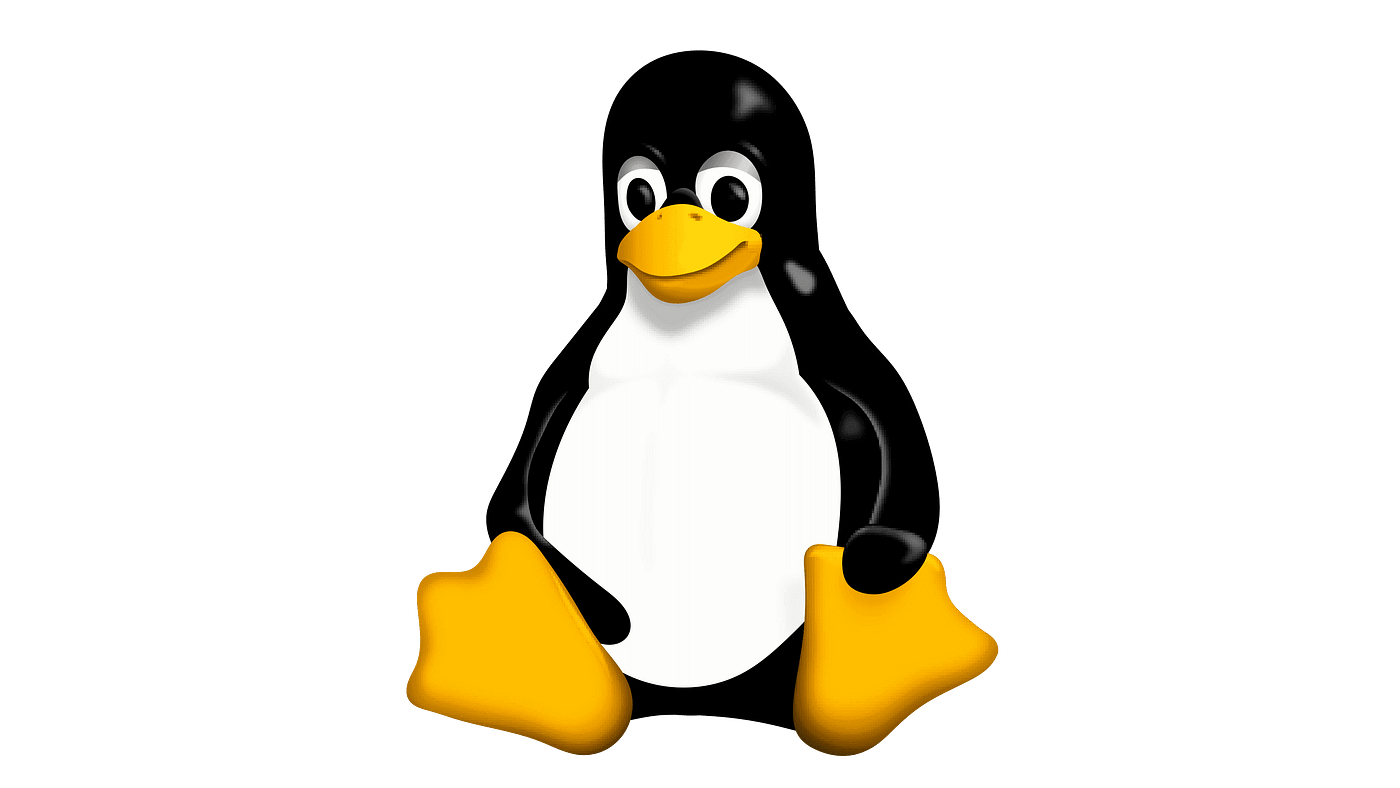
**6. Ventajas de Git**

* Permite trabajar sin conexión a internet.
* Facilita la colaboración entre varios desarrolladores.
* Tiene una gran comunidad y documentación.
* Es compatible con servicios de integración continua (CI/CD).

**7. Conclusión**

Git es una herramienta esencial para el desarrollo de software moderno. Su uso permite mantener un registro organizado y seguro de los cambios, favoreciendo el trabajo en equipo y la calidad del código. Aprender a usar Git es una habilidad fundamental para cualquier desarrollador o persona que trabaje en proyectos tecnológicos.

**INFORME N°2: COMANDOS DE LINUX**



**Introducción**

Linux es un sistema operativo libre y de código abierto, ampliamente utilizado en servidores, sistemas embebidos y equipos personales. Uno de los aspectos más poderosos de Linux es su terminal o línea de comandos, que permite a los usuarios controlar el sistema de manera eficiente y detallada.

El uso de comandos en Linux es esencial para tareas de administración, configuración del sistema, automatización y desarrollo.

**2. ¿Qué es la terminal de Linux?**

La terminal es una interfaz de línea de comandos (CLI) que permite interactuar directamente con el sistema operativo usando texto. Aunque existen entornos gráficos (GUI), la terminal ofrece mayor control y flexibilidad.

**3. Comandos básicos de Linux**

|  |  |
| --- | --- |
| pwd | Muestra la ruta del directorio actual. |
| ls | Lista los archivos y carpetas del directorio. |
| cd | Cambia de directorio. Ej: cd /home/usuario |
| mkdir | Crea un nuevo directorio. |
| rm | Elimina archivos o directorios. Ej: rm archivo.txt |
| cp | Copia archivos o carpetas. Ej: cp archivo.txt copia.txt |
| mv | Mueve o renombra archivos. |
| touch | Crea un archivo vacío. |
| echo | Muestra un mensaje en la terminal o escribe texto en un archivo. |
| cat | Muestra el contenido de un archivo. |
| clear | Limpia la terminal. |
| man | Muestra el manual de un comando. Ej: man ls |
| exit | Cierra la terminal. |

**4. Gestión de archivos y directorios**

* ls -l: lista con detalles (permisos, fecha, tamaño).
* ls -a: incluye archivos ocultos.
* rm -r: elimina carpetas y su contenido.
* cp -r: copia directorios de forma recursiva.

**5. Permisos y usuarios**

Linux usa un sistema de permisos para proteger archivos:

|  |  |
| --- | --- |
| chmod | Cambia permisos. Ej: chmod 755 archivo.sh |
| chown | Cambia el propietario de un archivo. |
| sudo | Ejecuta comandos como administrador. |

Permisos básicos:

* r (lectura)
* w (escritura)
* x (ejecución)

**6. Comandos para procesos y sistema**

|  |  |
| --- | --- |
| ps | Muestra procesos activos. |
| top | Muestra uso de recursos en tiempo real. |
| kill | Finaliza un proceso. Ej: kill 1234 |
| df -h | Muestra espacio en disco. |
| free -m | Muestra uso de memoria RAM. |
| uname -a | Información del sistema. |
| reboot | Reinicia el sistema. |
| shutdown now | Apaga el sistema. |

**7. Red y conexiones**

|  |  |
| --- | --- |
| ping | Verifica conexión a una IP o dominio. |
| ifconfig o ip a | Muestra configuración de red. |
| netstat -tuln | Puertos abiertos y servicios en uso. |
| curl | Realiza peticiones web. |
| wget | Descarga archivos desde internet. |

**8. Comandos útiles adicionales**

|  |  |
| --- | --- |
| history | Muestra el historial de comandos. |
| alias | Crea atajos de comandos. |
| grep | Busca texto dentro de archivos. |
| find | Busca archivos en el sistema. |
| tar | Comprime y descomprime archivos .tar. |
| zip/unzip | Comprime y descomprime archivos .zip. |

**9. Conclusión**

Conocer y dominar los comandos de Linux permite a los usuarios realizar tareas de forma rápida, precisa y automatizada. Es una herramienta fundamental para administradores de sistemas, programadores y usuarios avanzados. Aprender a usar la terminal es una inversión clave para trabajar eficazmente en entornos basados en Linux.