Git

Danna Valentina Rodriguez Clavijo

Johan Fernando Ochoa Lozano

Servicio Nacional de Aprendizaje (SENA)

Centro de Electricidad, Electrónica y Telecomunicaciones

Ficha 2996150

Isaura Novoa

24 de febrero de 2025

**Git**

El Git fue creado por Linus Torvalds en 2005, a causa de la necesidad de una herramienta adecuada para gestionar la carga de trabajo, ya que el desarrollo del kernel de linux en ese año, había alcanzado un tamaño considerable de líneas de código, lo que hacía que los diversos cambios se gestionaban a través de correos electrónicos de patch, los cuales eran archivos que contenían la diferencia entre la versión original y la versión modificada .

Ante la dificultad de gestionar la creciente cantidad de cambios, Linus Torvalds y otros desarrolladores decidieron crear una herramienta propia que cumpliera con sus requerimientos, como un flujo de trabajo distribuido, seguridad contra la corrupción de datos y alto rendimiento. En solo unas semanas, Git fue desarrollado y se convirtió en la herramienta oficial para la gestión de versiones del kernel de Linux, lanzando su versión 1.0 en diciembre de ese mismo año.

¿Qué es git?

Git es un sistema de control de versiones gratuito y de código abierto. A diferencia de otros sistemas, Git opera de forma distribuida, lo que facilita el trabajo colaborativo en el desarrollo de software. Aunque se considera difícil de dominar debido a sus funcionalidades avanzadas, más del 90% de los desarrolladores lo utilizan como su herramienta principal para gestionar el historial de cambios en su código.

Características principales

* Desarrollo distribuido: Permite que cada desarrollador trabaje de manera independiente sin depender de un servidor central.
* Alto rendimiento en proyectos grandes: Facilita el trabajo rápido en proyectos extensos o con muchos archivos.
* Historial de cambios: Mantiene un registro detallado de todas las modificaciones realizadas, permitiendo incluir nuevos cambios sin perder la secuencia original.

**Comandos de Git**

**git add:** Agrega archivos al área de preparación.

git add .

**git init:** Crea un nuevo repositorio GIT en el directorio actual o en un nuevo proyecto

git init “nombre de proyecto”

**git clone:** Copia un repositorio existente, ya sea remoto o local.

git clone <https://github.com/usuario/repo.git>

**git commit:** Guarda los cambios preparados en un repositorio con un mensaje descriptivo.

git commit -m “descripción del cambio”

**git status:** Muestra el estado del repositorio, indicando que archivos han sido modificados o añadidos.

git status

**git push:** Envía tus commits locales al repositorio remoto.

git push origin main

**git pull:**Descarga y fusiona los cambios del repositorio remoto al local.

git pull

**git branch:** Muestra las ramas actuales, crea una nueva rama o elimina una existente.

git branch nombre-rama

git branch –d nombre-rama

**git checkout:** Cambia entre ramas o crea una nueva rama.

git checkout nombre -rama

**git merge:** Fusiona una rama con otra. Esto suele utilizarse después de terminar el desarrollo en una rama.

git merge nombre-rama

**git log:** Muestra el historial de commits del repositorio.

git log

**git reset:** Deshace cambios locales y reinicia el área de preparación al último commit.

git reset - -hard HEAD

**git rm:** Elimina archivos en el área de preparación y del directorio de trabajo.

git rm archivo.txt

**git stash:** Guarda temporalmente los cambios no confirmados y limpia el area de trabajo para poder cambiar las ramas o realizar otras tareas sin perder tu trabajo.

git stash

**git fetch:** Descarga los cambios del repositorio remoto sin fusionarlos inmediatamente.

git fetch origin

**Bibliografía**

Aulab. (n.d.). *Nacimiento de Git*. Aulab. <https://aulab.es/articulos-guias-avanzadas/55/nacimiento-de-git>

SWH Hosting. (n.d.). *Comandos básicos de Git: Guía para principiantes*. SWH Hosting.<https://www.swhosting.com/es/blog/comandos-basicos-de-git-guia-para-principiantes>