

# Git: Un sistema de control de versiones

Este trabajo explora el concepto de Git, un sistema de control de versiones ampliamente utilizado en el desarrollo de software y otros proyectos colaborativos. veremos sobre sus funciones principales, comandos esenciales, y la importancia de las ramas en Git.

Jeremy Tomas Diaz Gomera (2022-0471)

[Jdiaz1423/Itla1 \(github.com\)](#).

# ¿Qué es Git?

Git es un sistema de control de versiones distribuido que facilita el seguimiento de cambios en archivos, código fuente y otros tipos de datos a lo largo del tiempo. Actúa como una herramienta fundamental para el desarrollo de software, ya que permite a los equipos colaborar de forma eficiente, rastrear cambios, recuperar versiones anteriores, y gestionar código de manera organizada. En esencia, Git se encarga de registrar el historial de cada archivo, lo que permite volver a versiones previas si es necesario.





# El comando git init

El comando `git init` es el primer paso para crear un repositorio Git dentro de un directorio específico. Al ejecutar este comando, se crea un directorio oculto llamado `.git` dentro del directorio actual. Este directorio contiene toda la información sobre el repositorio, incluyendo el historial de versiones, las ramas, y otros datos importantes. En esencia, `git init` convierte un directorio en un repositorio Git, preparándolo para el control de versiones.





# Ramas en Git

Una rama en Git es una línea de desarrollo independiente que se crea a partir del estado actual del repositorio. Cada rama representa una línea de trabajo distinta. Las ramas permiten a los desarrolladores trabajar en nuevas funcionalidades, solucionar errores, o experimentar con cambios sin afectar el código principal. Al crear una rama, se crea una copia del repositorio, lo que permite realizar cambios sin modificar la línea principal del código. Una vez que los cambios se completan, la rama se puede fusionar con la línea principal, incorporando las nuevas funcionalidades o soluciones.

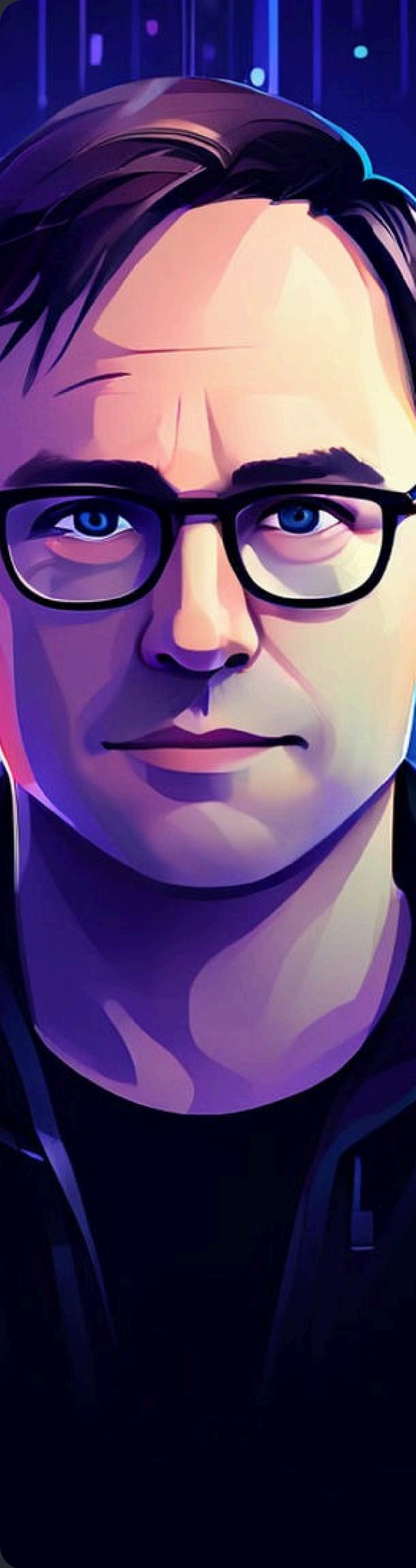


# Determinar la rama actual

Para saber en qué rama te encuentras en un repositorio Git, puedes utilizar el comando `git branch`. Este comando muestra todas las ramas disponibles en el repositorio, con un asterisco (\*) junto al nombre de la rama actual. Por ejemplo, si ves `* master`, significa que estás en la rama master.

```
git branch
```

```
git branch
```



# El creador de Git

Git fue creado por Linus Torvalds, el mismo desarrollador responsable del kernel de Linux. El desarrollo de Git comenzó en 2005, con el objetivo de gestionar el código fuente del kernel de Linux de una manera más eficiente. La necesidad de un sistema de control de versiones robusto y flexible llevó a Torvalds a desarrollar Git, el cual ha ganado popularidad a lo largo de los años y se ha convertido en una herramienta esencial para el desarrollo de software en todo el mundo.

# Comandos esenciales de Git

Comando	Descripción
git add	Agrega archivos al índice de Git para ser rastreados.
git commit	Graba los cambios en el repositorio, creando un nuevo "commit".
git status	Muestra el estado actual del repositorio, incluyendo archivos modificados o sin rastrear.
git push	Envía cambios del repositorio local a un repositorio remoto.
git pull	Descarga cambios desde un repositorio remoto y los fusiona con el repositorio local. Cambia la rama actual en el repositorio.
git checkout	Muestra el historial de commits del repositorio.
git log	Muestra las diferencias entre el estado actual del repositorio y una versión anterior.
git diff	

# Repositorios de Git

Algunos de los repositorios Git más reconocidos y utilizados son:

- [GitHub](#)
- [GitLab](#)
- [Bitbucket](#)

Estos sitios web ofrecen plataformas de alojamiento de código fuente que permiten a los desarrolladores alojar sus proyectos de código abierto o privados, colaborar con otros, y acceder a herramientas para el desarrollo de software.