## **Informe sobre los Comandos GIT**

### **Introducción**

GIT es un sistema de control de versiones distribuido, utilizado principalmente para el desarrollo de software. Fue creado por Linus Torvalds en 2005, y desde entonces se ha convertido en una herramienta esencial para programadores y equipos de desarrollo. GIT permite llevar un registro de los cambios realizados en un proyecto, colaborar con otros desarrolladores y mantener versiones estables del código.

### **Comandos GIT más usados**

A continuación, se presentan algunos de los comandos GIT más utilizados, con una breve explicación de cada uno:

#### **git init**

Este comando se usa para **inicializar un nuevo repositorio GIT** en un directorio. Crea una carpeta oculta .git que contiene todos los archivos necesarios para el control de versiones.

bash

CopiarEditar

git init

#### **git clone**

Permite **clonar un repositorio remoto** en tu máquina local. Es útil cuando se desea empezar a trabajar en un proyecto existente.

bash

CopiarEditar

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

#### **git add**

Agrega archivos al área de preparación (*staging area*) para ser incluidos en el próximo commit. Se puede agregar un solo archivo o todos los cambios.

bash

CopiarEditar

git add archivo.txt  
git add .

#### **git commit**

Guarda los cambios del área de preparación en el historial del proyecto. Es recomendable agregar un mensaje descriptivo.

bash

CopiarEditar

git commit -m "Mensaje del commit"

#### **git status**

Muestra el estado actual del repositorio: qué archivos han sido modificados, cuáles están en el área de preparación y cuáles no.

bash

CopiarEditar

git status

#### **git log**

Muestra el historial de commits realizados en el repositorio. Incluye información como el autor, fecha y mensaje del commit.

bash

CopiarEditar

git log

#### **git pull**

Descarga los cambios del repositorio remoto y los fusiona con el repositorio local.

bash

CopiarEditar

git pull

#### **git push**

Envía los commits locales al repositorio remoto. Es usado para compartir los cambios con otros desarrolladores.

bash

CopiarEditar

git push origin main

#### **git branch**

Permite crear, listar o eliminar ramas (*branches*) en un proyecto. Las ramas se usan para trabajar en nuevas características sin afectar el código principal.

bash

CopiarEditar

git branch  
git branch nueva-rama

#### **git checkout**

Sirve para cambiar de rama o restaurar archivos. También se usa para moverse a un commit anterior.

bash

CopiarEditar

git checkout main  
git checkout nombre-rama

#### **git merge**

Fusiona los cambios de una rama con otra. Generalmente se usa para combinar una rama de desarrollo con la rama principal.

bash

CopiarEditar

git merge nombre-rama

### **Conclusión**

El conocimiento y dominio de los comandos GIT es fundamental para cualquier desarrollador, ya que permite trabajar de manera ordenada, segura y colaborativa. Usar GIT correctamente reduce errores, mejora la organización del código y facilita el trabajo en equipo.