Informe:

Comandos Básicos de GIT

# 1. Introducción a Git

# Git es un sistema de control de versiones distribuido, diseñado para manejar todo tipo de proyectos con eficiencia y rapidez. Fue creado por Linus Torvalds en 2005, el mismo creador del núcleo de Linux, con el objetivo de gestionar el desarrollo del propio kernel de Linux de manera colaborativa. A diferencia de otros sistemas de control de versiones centralizados, Git permite que cada desarrollador tenga una copia completa del historial del proyecto en su propia máquina, lo cual brinda mayor flexibilidad, seguridad y autonomía en el desarrollo. Su arquitectura distribuida también facilita el trabajo en equipo, ya que múltiples desarrolladores pueden trabajar simultáneamente en diferentes partes del código sin interferir entre sí. En este informe se detallarán los comandos esenciales de Git para un manejo eficiente del control de versiones.

# 2. Comandos Iniciales de Git

## git init

Crea un nuevo repositorio Git en el directorio actual.

git init

## git clone

Clona un repositorio remoto a tu máquina local.

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

## git config

Permite establecer configuraciones como el nombre de usuario o el correo electrónico.

git config --global user.name "Tu Nombre"  
git config --global user.email "tuemail@ejemplo.com"

# 3. Gestión de Cambios

## git status

Muestra el estado actual del repositorio: qué archivos han sido modificados, añadidos o eliminados.

git status

## git add

Agrega archivos al área de preparación (staging area).

git add archivo.txt  
git add .

## git commit

Guarda los cambios en el repositorio con un mensaje.

git commit -m "Mensaje del commit"

# 4. Trabajo con Ramas (Branches)

## git branch

Muestra todas las ramas existentes. También puede crear una nueva rama.

git branch nueva-rama

## git checkout

Permite cambiar de rama o restaurar archivos.

git checkout main  
git checkout -b nueva-rama

## git merge

Fusiona los cambios de una rama a otra.

git merge main

# 5. Repositorios Remotos

## git remote add

Asocia un repositorio remoto al local.

git remote add origin main

## git push

Envía los cambios al repositorio remoto.

git push origin main

## git pull

Descarga y fusiona cambios del repositorio remoto.

git pull origin main

# 6. Historial y Control de Versiones

## git log

Muestra el historial de commits.

git log

## git diff

Muestra las diferencias entre versiones.

git diff

## git reset

Permite deshacer cambios.

git reset --hard HEAD

# 7. Conclusiones

Git es una herramienta fundamental para trabajar de forma organizada, segura y colaborativa en el desarrollo de software. A través de comandos como git init, add, commit, push y pull, puedes llevar un control detallado de los cambios en tu proyecto, colaborar con otros sin pisarte el trabajo, y mantener un historial claro de todo lo que ha pasado.

Usar Git correctamente te permite:

* Guardar versiones de tu proyecto como puntos seguros de recuperación.
* Colaborar en equipo sin sobrescribir el trabajo de otros.
* Subir y sincronizar tu código con plataformas como GitHub.
* Traer actualizaciones y fusionar cambios con facilidad.
* Resolver conflictos y manejar ramas para probar nuevas ideas sin dañar el proyecto principal.

Dominar estos comandos permite a los equipos tener mayor control, trazabilidad y calidad en sus proyectos.