**Informe sobre los Comandos de Git**

**Introducción**

Git es un sistema de control de versiones distribuido que permite a los desarrolladores rastrear cambios en el código fuente durante el desarrollo de software. Es una herramienta esencial para la colaboración en proyectos de programación, ya que facilita la gestión de versiones y la integración de cambios realizados por diferentes miembros del equipo.

**Comandos Básicos de Git**

1. **git init**
   * **Descripción**: Inicializa un nuevo repositorio de Git.
   * **Uso**: git init
   * **Ejemplo**: git init my-repo
2. **git clone**
   * **Descripción**: Clona un repositorio existente.
   * **Uso**: git clone <url-del-repositorio>
   * **Ejemplo**: git clone https://github.com/user/repo.git
3. **git status**
   * **Descripción**: Muestra el estado del repositorio, incluyendo archivos modificados y cambios no confirmados.
   * **Uso**: git status
4. **git add**
   * **Descripción**: Añade archivos al área de preparación (staging area).
   * **Uso**: git add <archivo>
   * **Ejemplo**: git add index.html
5. **git commit**
   * **Descripción**: Confirma los cambios en el repositorio.
   * **Uso**: git commit -m "Mensaje del commit"
   * **Ejemplo**: git commit -m "Añadido archivo index.html"
6. **git push**
   * **Descripción**: Envía los cambios confirmados al repositorio remoto.
   * **Uso**: git push origin <rama>
   * **Ejemplo**: git push origin main
7. **git pull**
   * **Descripción**: Descarga y fusiona cambios del repositorio remoto.
   * **Uso**: git pull origin <rama>
   * **Ejemplo**: git pull origin main
8. **git branch**
   * **Descripción**: Muestra las ramas existentes o crea una nueva rama.
   * **Uso**: git branch (para listar ramas) o git branch <nombre-de-la-rama> (para crear una nueva rama)
   * **Ejemplo**: git branch feature-x
9. **git checkout**
   * **Descripción**: Cambia a otra rama o restaura archivos.
   * **Uso**: git checkout <rama> (para cambiar de rama) o git checkout -- <archivo> (para restaurar un archivo)
   * **Ejemplo**: git checkout main
10. **git merge**
    * **Descripción**: Fusiona cambios de una rama en la rama actual.
    * **Uso**: git merge <rama>
    * **Ejemplo**: git merge feature-x

**Comandos Avanzados de Git**

1. **git rebase**
   * **Descripción**: Reaplica commits sobre una base diferente.
   * **Uso**: git rebase <rama>
   * **Ejemplo**: git rebase main
2. **git stash**
   * **Descripción**: Guarda cambios temporales sin confirmarlos.
   * **Uso**: git stash
   * **Ejemplo**: git stash save "Cambios temporales"
3. **git tag**
   * **Descripción**: Crea una etiqueta en un commit específico.
   * **Uso**: git tag <nombre-de-la-etiqueta>
   * **Ejemplo**: git tag v1.0.0
4. **git log**
   * **Descripción**: Muestra el historial de commits.
   * **Uso**: git log
   * **Ejemplo**: git log --oneline
5. **git remote**
   * **Descripción**: Gestiona repositorios remotos.
   * **Uso**: git remote add <nombre> <url>
   * **Ejemplo**: git remote add origin https://github.com/user/repo.git

**Conclusión**

Git es una herramienta poderosa y flexible que facilita la colaboración y el control de versiones en proyectos de desarrollo de software. Conocer y dominar los comandos básicos y avanzados de Git es esencial para cualquier desarrollador que trabaje en equipo o en proyectos de código abierto.