

# Índice

|   |          |
|---|----------|
| <b>Resumen</b>                                  | <b>1</b> |
| Comandos Básicos de Git . . . . .               | 1        |
| Sincronización con Repositorio Remoto . . . . . | 1        |
| Ramas (Branches) . . . . .                      | 1        |
| Manejo de Cambios . . . . .                     | 2        |
| Otros útiles . . . . .                          | 2        |

## Resumen

### Comandos Básicos de Git

1. `git init`  
**Utilidad:** Inicializa un nuevo repositorio Git en el directorio actual.
2. `git clone <url>`  
**Utilidad:** Clona un repositorio remoto a tu máquina local.
3. `git status`  
**Utilidad:** Muestra el estado del repositorio (archivos modificados, en staging, sin seguimiento, etc.).
4. `git add <archivo>` / `git add .`  
**Utilidad:** Agrega archivos al área de staging (preparación para commit).
5. `git commit -m "mensaje"`  
**Utilidad:** Guarda los cambios del staging en el historial del repositorio.
6. `git log`  
**Utilidad:** Muestra el historial de commits.

### Sincronización con Repositorio Remoto

7. `git remote add origin <url>`  
**Utilidad:** Vincula el repositorio local con uno remoto.
8. `git push origin <rama>`  
**Utilidad:** Sube los cambios locales al repositorio remoto.
9. `git pull origin <rama>`  
**Utilidad:** Trae cambios desde el repositorio remoto y los fusiona con tu rama local.
10. `git fetch`  
**Utilidad:** Descarga cambios del remoto sin fusionarlos automáticamente.

### Ramas (Branches)

11. `git branch`  
**Utilidad:** Lista las ramas existentes.
12. `git branch <nombre>`  
**Utilidad:** Crea una nueva rama.
13. `git checkout <rama>`  
**Utilidad:** Cambia a la rama especificada.
14. `git checkout -b <rama>`  
**Utilidad:** Crea y cambia a una nueva rama en un solo paso.
15. `git merge <rama>`  
**Utilidad:** Fusiona la rama especificada con la actual.
16. `git branch -d <rama>`  
**Utilidad:** Elimina una rama local.

## Manejo de Cambios

17. `git diff`  
**Utilidad:** Muestra diferencias entre archivos modificados y el último commit.
18. `git reset <archivo>`  
**Utilidad:** Quita archivos del staging area.
19. `git reset --hard <commit>`  
**Utilidad:** Revierte el repositorio a un commit específico (elimina cambios posteriores).
20. `git stash`  
**Utilidad:** Guarda cambios temporales sin hacer commit.
21. `git stash pop`  
**Utilidad:** Recupera los cambios guardados con stash.

## Otros útiles

22. `git tag <nombre>`  
**Utilidad:** Marca un punto específico (como una versión) en la historia del proyecto.
23. `git rebase <rama>`  
**Utilidad:** Reaplica commits de una rama sobre otra base (reescribe el historial).
24. `git cherry-pick <commit>`  
**Utilidad:** Aplica un commit específico de otra rama en la rama actual.