

## Configuración inicial

1. `git config --global user.name "Tu Nombre"`  
Configura el nombre de usuario.
2. `git config --global user.email "tuemail@dominio.com"`  
Configura el correo electrónico.
3. `git config --global core.editor "editor"`  
Configura el editor de texto para Git (por ejemplo, nano, vim, o code).

## Comandos básicos

1. `git init`  
Inicializa un nuevo repositorio en el directorio actual.
2. `git clone <URL>`  
Clona un repositorio remoto en el sistema local.
3. `git status`  
Muestra el estado actual del repositorio (archivos modificados, no rastreados, etc.).
4. `git add <archivo>`  
Agrega un archivo específico al área de preparación.
5. `git add .`  
Agrega todos los archivos nuevos y modificados al área de preparación.
6. `git commit -m "Mensaje del commit"`  
Guarda los cambios preparados en el historial del repositorio.
7. `git commit -am "Mensaje del commit"`  
Prepara y guarda los cambios en archivos ya rastreados en un solo paso.

## Ramas (branches)

1. `git branch`  
Lista todas las ramas locales.
2. `git branch <nombre-rama>`  
Crea una nueva rama.
3. `git checkout <nombre-rama>`  
Cambia a otra rama.
4. `git checkout -b <nombre-rama>`  
Crea y cambia a una nueva rama en un solo paso.
5. `git merge <nombre-rama>`  
Fusiona la rama especificada con la rama actual.

6. `git branch -d <nombre-rama>`  
Elimina una rama local.
- 

### **Trabajando con remoto**

1. `git remote add origin <URL>`  
Conecta el repositorio local a un remoto.
2. `git remote -v`  
Muestra las URL de los remotos configurados.
3. `git fetch`  
Descarga las actualizaciones del remoto sin fusionarlas.
4. `git pull`  
Descarga y fusiona las actualizaciones del remoto en la rama actual.
5. `git push origin <nombre-rama>`  
Sube los cambios de la rama actual al remoto.
6. `git push -u origin <nombre-rama>`  
Establece el seguimiento de la rama actual con el remoto.

### **Deshacer cambios**

1. `git reset <archivo>`  
Saca un archivo del área de preparación.
2. `git reset --soft HEAD~1`  
Revierte el último commit, manteniendo los cambios en el área de preparación.
3. `git reset --hard HEAD~1`  
Revierte el último commit y elimina los cambios del área de trabajo.
4. `git restore <archivo>`  
Descarta cambios no guardados en un archivo.
5. `git restore --staged <archivo>`  
Saca un archivo del área de preparación.

### **Historial**

1. `git log`  
Muestra el historial de commits.
2. `git log --oneline`  
Muestra el historial en una sola línea por commit.
3. `git log --graph --oneline --all`  
Muestra el historial en formato gráfico simplificado.

### **Etiquetas (tags)**

1. `git tag`  
Lista todas las etiquetas.
2. `git tag <nombre-etiqueta>`  
Crea una nueva etiqueta ligera.
3. `git tag -a <nombre-etiqueta> -m "Mensaje"`  
Crea una etiqueta anotada.
4. `git push origin <nombre-etiqueta>`  
Sube una etiqueta al remoto.

### **Otros comandos útiles**

1. `git stash`  
Guarda temporalmente los cambios no confirmados.
2. `git stash apply`  
Restaura los cambios guardados con `git stash`.
3. `git diff`  
Muestra las diferencias entre el área de trabajo y el área de preparación.
4. `git show <hash>`  
Muestra detalles de un commit específico.
5. `git rebase <rama>`  
Reaplica los commits de la rama actual sobre otra.
6. `git clean -fd`  
Elimina archivos no rastreados y directorios.