**1. Configuración inicial**

* **git config --global user.name "TuNombre"**  
  Establece el nombre de usuario que se asociará con tus commits.
* **git config --global user.email "tuemail@example.com"**  
  Establece el correo electrónico asociado con tus commits.
* **git config --list**  
  Muestra todas las configuraciones actuales de Git.

**2. Comandos básicos**

* **git init**  
  Inicializa un repositorio Git en el directorio actual.
* **git clone <URL>**  
  Clona un repositorio remoto en tu máquina local.
* **git add <archivo>**  
  Agrega archivos específicos al área de preparación (staging). Puedes usar git add . para agregar todos los cambios.
* **git commit -m "Mensaje del commit"**  
  Crea un commit con un mensaje que describe los cambios.
* **git status**  
  Muestra el estado actual de los archivos y los cambios en el repositorio.

**3. Ramas (Branches)**

* **git branch**  
  Lista todas las ramas en el repositorio.
* **git branch <nombre\_rama>**  
  Crea una nueva rama con el nombre especificado.
* **git checkout <nombre\_rama>**  
  Cambia a la rama especificada.
* **git checkout -b <nombre\_rama>**  
  Crea y cambia a una nueva rama al mismo tiempo.
* **git merge <nombre\_rama>**  
  Fusiona la rama especificada con la rama actual.
* **git branch -d <nombre\_rama>**  
  Elimina una rama local.

**4. Sincronización con el repositorio remoto**

* **git remote add origin <URL>**  
  Vincula un repositorio remoto (habitualmente se usa origin como nombre).
* **git push origin <nombre\_rama>**  
  Envía los commits de la rama local especificada al repositorio remoto.
* **git push -u origin <nombre\_rama>**  
  Envía la rama y configura el repositorio remoto para hacer el seguimiento de la rama automáticamente.
* **git pull**  
  Trae los cambios del repositorio remoto a la rama local y los fusiona.
* **git fetch**  
  Descarga el historial y los datos de las ramas sin hacer merge automático.

**5. Historial y revertir cambios**

* **git log**  
  Muestra el historial de commits.
* **git log --oneline**  
  Muestra un resumen del historial de commits en una sola línea.
* **git diff**  
  Muestra las diferencias entre los cambios no guardados y la última versión.
* **git reset <archivo>**  
  Quita un archivo del área de preparación.
* **git reset --hard <commit>**  
  Restaura el estado del repositorio al commit especificado (¡ten cuidado con este comando!).

**6. Comandos avanzados**

* **git stash**  
  Guarda temporalmente los cambios no confirmados para liberar el área de trabajo.
* **git stash pop**  
  Restaura los cambios guardados previamente con git stash.
* **git rebase <nombre\_rama>**  
  Aplica los cambios de una rama sobre otra base, útil para limpiar el historial de commits.

**7. Comandos de GitHub específicos**

* **gh repo clone <usuario>/<repositorio>**  
  Clona un repositorio desde GitHub (requiere la CLI de GitHub instalada).
* **gh repo create**  
  Crea un nuevo repositorio en GitHub desde la terminal.
* **gh pr create**  
  Crea un Pull Request (PR) en GitHub desde la terminal.
* **gh pr checkout <ID\_PR>**  
  Cambia a una rama basada en un Pull Request específico.