Aquí tienes un resumen de los comandos más comunes que deberías usar en la terminal cuando ya estás conectado a Git y trabajando en un proyecto:

### 1. \*\*Inicializar Git (si no lo has hecho aún)\*\*

```bash

git init

```

Este comando crea un nuevo repositorio Git local si no existe aún.

### 2. \*\*Verificar el estado de tu repositorio\*\*

```bash

git status

```

Este comando muestra los archivos modificados, no rastreados (untracked) o preparados para ser confirmados (staged).

### 3. \*\*Agregar archivos al área de preparación (staging area)\*\*

```bash

git add <nombre\_del\_archivo>

```

o para agregar todos los archivos:

```bash

git add .

```

Este comando mueve los archivos al área de preparación para ser confirmados.

### 4. \*\*Confirmar los cambios (commit)\*\*

```bash

git commit -m "Mensaje explicativo del commit"

```

Este comando guarda una versión de tus cambios en el historial del repositorio local.

### 5. \*\*Conectar tu repositorio local con el remoto\*\*

(Solo lo haces una vez, cuando conectas tu repositorio local a uno remoto)

```bash

git remote add origin https://github.com/usuario/repositorio.git

```

### 6. \*\*Obtener los últimos cambios del repositorio remoto\*\*

Si el repositorio remoto tiene cambios que no tienes en local, primero debes integrarlos:

```bash

git pull origin main --rebase

```

Este comando baja los cambios del remoto y los integra en tu rama actual.

### 7. \*\*Enviar tus cambios al repositorio remoto\*\*

Después de hacer un `commit`, puedes enviar tus cambios al repositorio remoto:

```bash

git push origin main

```

### 8. \*\*Verificar la historia de commits\*\*

Para ver el historial de commits de tu proyecto:

```bash

git log

```

### 9. \*\*Resolver conflictos\*\* (si aparecen durante el pull o el rebase)

- \*\*Editar los archivos en conflicto\*\*, luego:

```bash

git add <archivo\_resuelto>

git rebase --continue

```

### Flujo típico de trabajo en Git:

1. \*\*Ver el estado del repositorio\*\*:

```bash

git status

```

2. \*\*Agregar archivos modificados\*\*:

```bash

git add .

```

3. \*\*Confirmar cambios\*\*:

```bash

git commit -m "Descripción de los cambios"

```

4. \*\*Obtener cambios del remoto (si es necesario)\*\*:

```bash

git pull origin main --rebase

```

5. \*\*Enviar cambios al repositorio remoto\*\*:

```bash

git push origin main

```

Con estos comandos, tendrás el flujo básico cubierto para trabajar con Git desde la terminal.