

- 1 **Configuración inicial:** Antes de comenzar, configura tu nombre de usuario y dirección de correo electrónico en Git. Esto es importante para que tus contribuciones queden registradas correctamente.

```
git config --global user.name "Tu Nombre"
git config --global user.email "tu@email.com"
```

- 2 **Iniciar un repositorio:** Puedes iniciar un nuevo repositorio Git en un directorio existente utilizando el comando `git init`. Por ejemplo:

```
mkdir mi_proyecto
cd mi_proyecto
git init
```

- 3 **Agregar archivos al área de preparación (Staging Area):** Antes de hacer una confirmación (commit), debes agregar los archivos que deseas rastrear al área de preparación. Utiliza `git add` para hacerlo. Por ejemplo, para agregar todos los archivos al área de preparación:

```
git add .
```

- 4 **Realizar una confirmación (Commit):** Después de agregar archivos al área de preparación, realiza una confirmación para guardar los cambios en el repositorio. Usa `git commit`:

```
git commit -m "Mensaje descriptivo del cambio"
```

- 5 **Ver el historial de confirmaciones:** Puedes ver el historial de confirmaciones en tu repositorio usando `git log`:

```
git log
```

- 6 **Crear una rama (Branch):** Las ramas te permiten trabajar en diferentes características o correcciones al mismo tiempo. Crea una nueva rama con `git branch`:

```
git branch nombre_de_la_rama
```

- 7 **Cambiar de rama (Checkout):** Cambia entre ramas usando `git checkout`:

```
git checkout nombre_de_la_rama
```

- 8 **Combinar ramas (Merge):** Para combinar una rama con otra, utiliza `git merge`. Por ejemplo, para combinar una rama llamada `nueva-funcionalidad` con la rama principal:

```
git checkout rama_principal
git merge nueva-funcionalidad
```


- 9 **Actualizar y obtener cambios (Pull y Fetch):** Para obtener cambios de un repositorio remoto, utiliza `git pull` o `git fetch`. `git pull` obtiene y fusiona automáticamente los cambios, mientras que `git fetch` solo obtiene los cambios.

```
git pull origin nombre_de_la_rama
```

- 10 **Subir cambios a un repositorio remoto (Push):** Para enviar tus cambios locales a un repositorio remoto, utiliza `git push`:

```
git push origin nombre_de_la_rama
```

Estos son los conceptos básicos de Git que te ayudarán a empezar. Puedes combinarlos y adaptarlos a tu flujo de trabajo. Git es una herramienta versátil y poderosa que te permitirá trabajar en proyectos de programación de manera colaborativa y controlada. ¡Buena suerte con tu aprendizaje!

From ChatGPT, designed with  FancyGPT