

Subir archivos a Github

1. Inicializa el repositorio localmente (si aún no lo has hecho)

git init

2. Añade todos los archivos a la zona de preparación (staging area)

git add .

3. Haz un commit de los archivos

git commit -m "Mensaje del commit"

4. Vincula tu repositorio local con el repositorio remoto de GitHub

Este comando solo lo necesitas si aún no has vinculado tu repositorio local al remoto:

git remote add origin https://github.com/tu-usuario/tu-repo.git

5. Sube los cambios al repositorio remoto

❑ **Si quieres subir los cambios a la rama main (asumiendo que esa es la rama principal):**

```
git push -u origin main
```

❑ **Si tu rama principal se llama master:**

```
git push -u origin master
```

Nota: Si ya has subido previamente commits a este repositorio y solo estás agregando nuevos cambios, puedes utilizar simplemente:

```
git push
```

Verifica el estado del repositorio:

❑ Ejecuta el siguiente comando en la terminal para ver qué archivos han sido modificados o agregados:

```
git status
```

❑ **Añade todos los archivos modificados y agregados a la zona de preparación:**

❑ Para añadir todos los archivos que han cambiado, utiliza:

```
git add .
```

❑ Esto añadirá todos los archivos modificados, nuevos o eliminados al área de preparación.

❑ **Realiza un commit de los cambios:**

□ Crea un commit con un mensaje que describa los cambios realizados:

```
git commit -m "Descripción de los cambios realizados"
```

Realizar un pull para obtener los cambios remotos

Antes de hacer push, debes realizar un git pull para obtener los cambios que ya existen en el repositorio remoto y combinarlos con tus cambios locales:

```
git pull origin main --rebase
```

Empuja los cambios al repositorio remoto:

□ Para enviar los cambios a la rama principal `main` en GitHub, usa:

```
git push origin master
```

Cambiar de master a main en tu local

```
git branch -m master main
```

Esto descargará los cambios remotos de la rama main y aplicará tus cambios locales encima de esos cambios. La opción `--rebase` es útil para mantener un historial de commits más limpio, aplicando tus commits locales después de los remotos.

1. Cambiar de rama

El uso más común de git checkout es para moverse entre ramas en tu repositorio. Cuando quieras cambiar a una rama específica, puedes usar:

```
git checkout <nombre-de-la-rama>
```