

Tutorial Básico de Git y GitHub

Curso: Temas Selectos de Python para Ciencias de la Tierra

Semestre 2026-I

1. Introducción

Git es un sistema de control de versiones que nos permite trabajar con código y documentos de forma organizada. GitHub es una plataforma en línea donde podemos alojar repositorios y colaborar. En este tutorial aprenderás a:

- Crear una cuenta en GitHub
- Configurar Git en tu entorno
- Clonar un repositorio
- Descargar actualizaciones (`git pull`, `git fetch`)
- Subir cambios (`git commit`, `git push`)
- Evitar y resolver conflictos comunes

2. Creación de cuenta en GitHub

1. Ve a <https://github.com> y selecciona **Sign up**.
2. Ingresa tu correo, usuario y contraseña.
3. Verifica tu cuenta (posible verificación 2FA si la activas).

3. Configuración inicial de Git

En la terminal (o consola de JupyterHub), configura tu nombre y correo:

```
git config --global user.name "Tu Nombre"
git config --global user.email "tuemail@encit.unam.mx"
```

Para verificar:

```
git config --list
```

4. Clonar un repositorio

Para obtener una copia local de un repositorio remoto:

```
git clone URL_DEL_REPOSITORIO
```

Ejemplo:

```
git clone https://github.com/JLGarciaFranco/TemasSelectos.git
```

5. Actualizar desde el repositorio remoto

5.1. `git pull`

Descarga e integra cambios de la rama remota:

```
git pull
```

Este comando trae cambios y los fusiona automáticamente.

5.2. `git fetch`

Descarga cambios pero no los fusiona todavía:

```
git fetch
```

Para ver los cambios antes de aplicarlos:

```
git log HEAD..origin/main
```

6. Trabajar con cambios propios

6.1. Ver el estado

```
git status
```

6.2. Añadir cambios al área de preparación

```
git add archivo1 archivo2
```

O para todos los cambios:

```
git add .
```

6.3. Guardar cambios localmente

```
git commit -m "Mensaje descriptivo de los cambios"
```

6.4. Enviar cambios al repositorio remoto

```
git push
```

7. Buenas prácticas para el curso

- Mantén dos carpetas:
 1. **Carpeta de descarga:** aquí clonamos el repositorio del curso y ejecutamos `git pull` para recibir las actualizaciones.
 2. **Carpeta de trabajo:** copia aquí los notebooks que quieras modificar. Así no afectas la versión original del repositorio.
- Cambia el nombre de los notebooks que copies a tu carpeta de trabajo para evitar confusiones, por ejemplo: `notebook_original.ipynb` → `notebook_mialumno.ipynb`.
- Haz siempre `git pull` antes de empezar a trabajar.
- No modifiques directamente los archivos en la carpeta de descarga del curso.

8. Resolución de problemas comunes

8.1. Error: `git pull` falla por cambios locales

Este error aparece cuando has modificado un archivo que también ha cambiado en el repositorio remoto.

Soluciones:

1. Si quieres conservar tus cambios pero aplicarlos después:

```
git stash
git pull
git stash pop
```

2. Si no quieres conservar los cambios locales:

```
git reset --hard HEAD
git pull
```

3. Mejor aún: evita modificar los notebooks en la carpeta de descarga (ver buenas prácticas).

8.2. Error: credenciales solicitadas en cada git pull

Si tu entorno no guarda las credenciales, puedes activar el guardado:

```
git config --global credential.helper store
```

Ten en cuenta que en entornos temporales (como contenedores no persistentes) esto se pierde al reiniciar.