



Uso básico de GitHub

Vamos a subir nuestro proyecto a GitHub.

1. Creamos un repositorio en GitHub

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).



Owner * / Repository name *

 RocioIES /

✓ Mi_Proyecto is available.

Great repository names are short and memorable. Need inspiration? How about [scaling-telegram](#) ?

Description (optional)

- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

- ☐ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template:

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License:

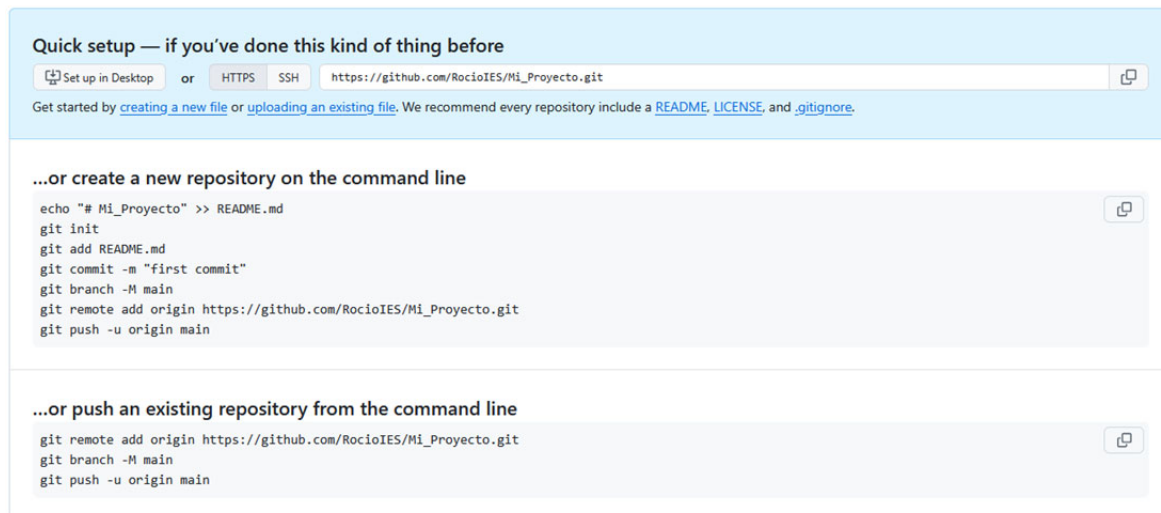
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

 You are creating a public repository in your personal account.

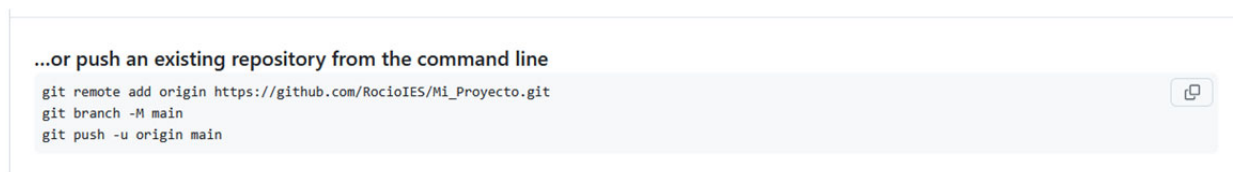
Create repository



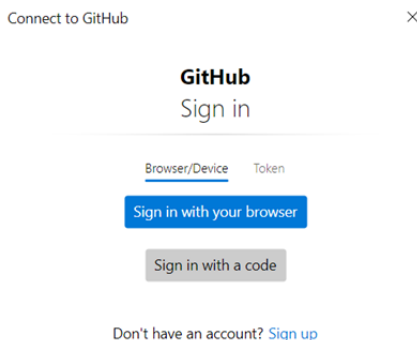
2. Ventana que se muestra a continuación:



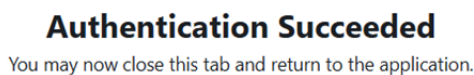
3. Como nosotros ya tenemos un repositorio en local, tenemos que optar por lo que se muestra:



4. Copiamos las tres líneas en nuestro repositorio en local y se nos muestra el mensaje:



5. Pulsamos en iniciar sesión con el navegador ponemos nuestras credenciales y nos mostrará:





6. En local (VS Code) se pueden ver los mensajes de subida de los archivos de forma correcta y en el repositorio creado en GitHub ya se han subido los archivos, además de los commits, ramas, etc.
7. A partir de ahora puedo seguir trabajando en local y cuando quiera volver a subir las modificaciones sólo tengo que dar la orden: `git push`
8. En GitHub puedo crear el archivo `Readme.md`, donde describo el contenido de mi repositorio. Al guardar los cambios (`commit changes`) se creará el fichero junto al resto de ficheros que ya teníamos.
9. Si queremos sincronizar lo que hemos creado en GitHub con nuestro repositorio local utilizamos la orden: `git pull` (en VS Code).
10. Si creamos un tag:

```
git tag -a v1.0 -m "uso básico de GitHub"
```

No se actualizan en el repositorio remoto con la orden `git push`, sino que hay que teclear la orden:

```
git push origin --tags
```

Conceptos

Remote

Remote en Git es una **referencia a un repositorio alojado en otro lugar** como GitHub.

Origin → Cuando se clona un repositorio desde GitHub, Git **automáticamente crea un remote llamado origin**, que es simplemente un *alias* para la URL del repositorio (hace referencia a la URL).

Si tecleamos la orden:

```
git remote -v
```

Se mostrará por ejemplo:

```
origin https://github.com/miprofesor/proyecto.git (fetch)
origin https://github.com/miprofesor/proyecto.git (push)
```

Esto significa:

- Cuando hacemos `git fetch`, Git traerá datos desde origin.
- Cuando hacemos `git push`, Git enviará tus cambios a origin.

Si queremos colaborar con otro repositorio además del original, se puede añadir así:

```
git remote add upstream https://github.com/otrafuente/proyecto.git
```

Para eliminarlo:

```
git remote remove upstream
```

Si cambia el nombre del repositorio en GitHub debemos ejecutar:

```
git remote set-url origin git@github.com:usuario/proyecto.git
```

Push

Fetch

Pull

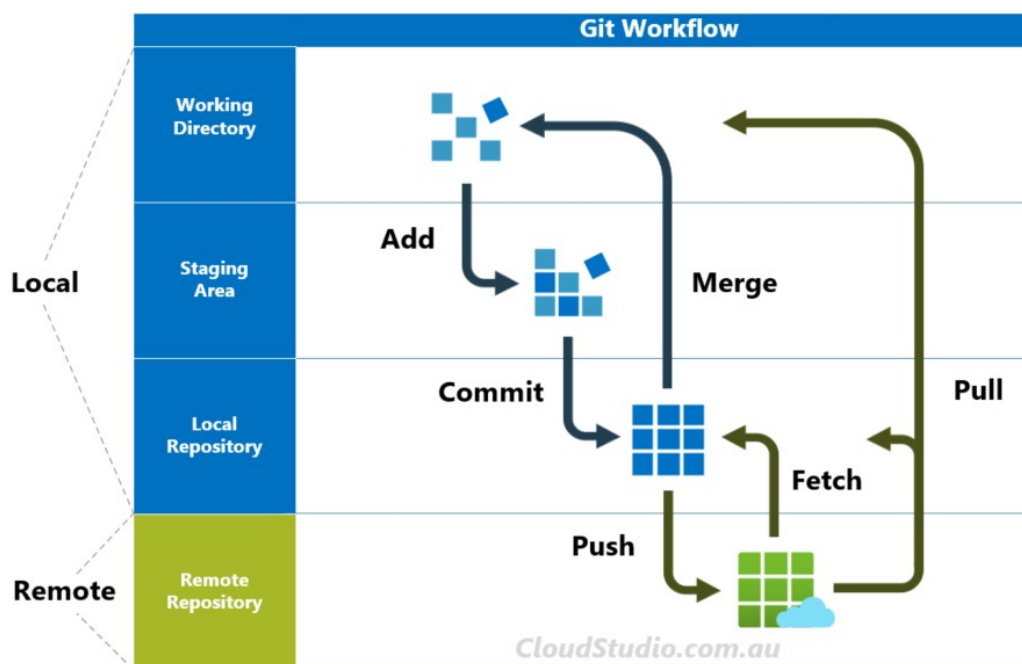


Imagen de <https://cloudstudio.com.au/2021/06/26/git-command/>