

# MANUAL GIT Y GITHUB

## 1. Instalar Git

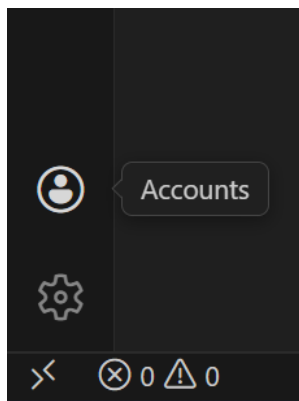
1. Entra en la web de Git: <https://git-scm.com/>  
Busca en Google: **"Git download"** y abre la página oficial de Git.
  2. Descarga e instala: el instalador para tu sistema:
    - Acepta que añada Git al **PATH** (suele venir por defecto).
  3. Comprueba que funciona:
    - Abre una terminal (o la propia terminal de VS Code).
    - Escribe: **git --version**
- 

## 2. Crear cuenta en GitHub

1. Ve a la página de **GitHub**: <https://github.com/>
  2. Pulsa en **Sign up / Sign up for GitHub**.
  3. Rellena el registro.
- 

## 3. Iniciar sesión en GitHub desde VS Code

1. Abre **Visual Studio Code**.
2. En la barra izquierda, busca el icono de Accounts (la siguiente imagen puede variar):



3. Pulsa en alguna opción de Sign in o login y VS Code abrirá el navegador:
    - Autorizas a VS Code a usar tu cuenta GitHub. Aceptas/Confirmas.
  4. Vuelve a VS Code: ya deberías estar logueado con tu usuario de GitHub.
-

## 4. Configurar Git (solo la primera vez)

En la **terminal integrada de VS Code** (View > Terminal, o Ctrl + ñ):

Pon tu nombre y correo (los que quieras que aparezcan en los commits (actualizaciones de código); suelen ser los de GitHub):

```
git config --global user.name "Tu Nombre"
```

```
git config --global user.email "tu_correo@loquesea.com"
```

Puedes comprobar la configuración con:

```
git config --global --list
```

---

## 5. Crear un proyecto nuevo y subirlo a GitHub (desde la terminal de VS Code)

### 5.1 Crear carpeta del proyecto

1. En tu sistema, crea una carpeta, por ejemplo:  
C:\Users\TuUsuario\Proyectos\mi\_proyecto\_git
2. Abre esa carpeta con VS Code:  
**File > Open Folder...** → selecciona mi\_proyecto\_git.

### 5.2 Crear algún archivo

En VS Code, crea un archivo sencillo, por ejemplo main.py:

```
print("Hola Git y GitHub desde VS Code")
```

Guárdalo.

### 5.3 Inicializar el repositorio Git

En la **terminal integrada** (ya estando en la carpeta del proyecto, lo verás en la ruta):

```
git init
```

Esto crea un repositorio Git local (aparecerá una carpeta oculta .git).

### 5.4 Ver el estado

```
git status
```

Deberías ver main.py como “untracked file”.

### 5.5 Añadir archivos al control de versiones

```
git add .
```

(El punto significa “todos los archivos cambiados/creados”).

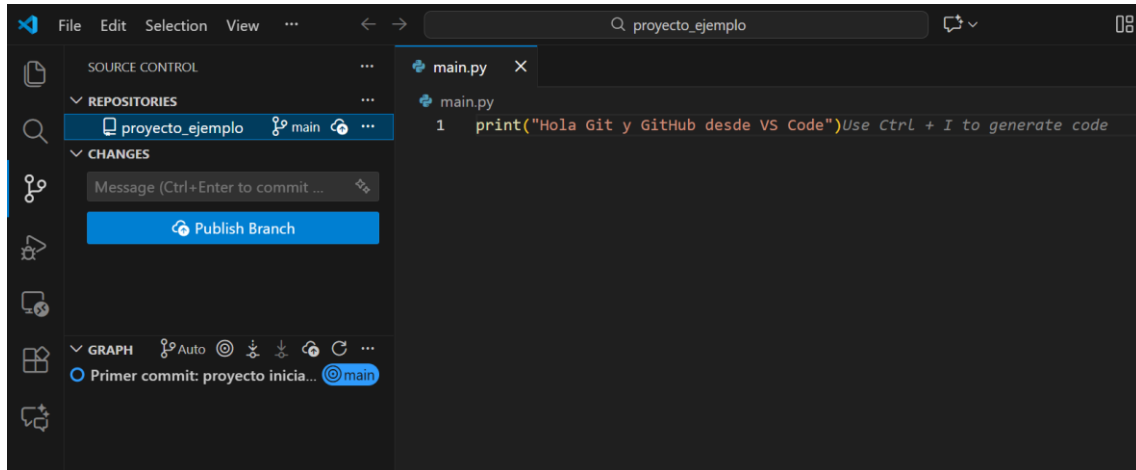
### 5.6 Hacer el primer commit

```
git commit -m "Primer commit: proyecto inicial"
```

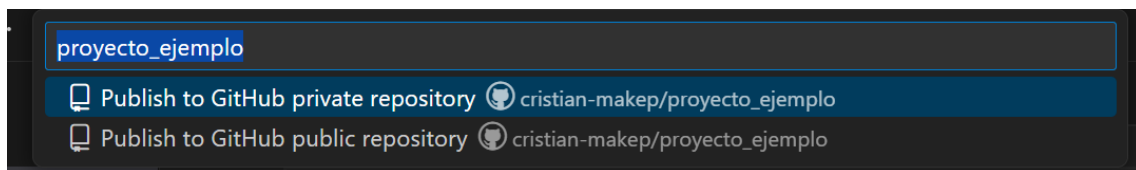
Ya tienes la **historia local** de tu proyecto creada.

## 5.7 Crear un repositorio en Github

Ir al icono de la barra lateral izquierda que son unos círculos con líneas que representa Github y comprobar que nuestro fichero y cambios aparecen aquí. Luego pulsar **“Publish Branch”** o cualquier referencia a publicar o crear repositorio que aparezca



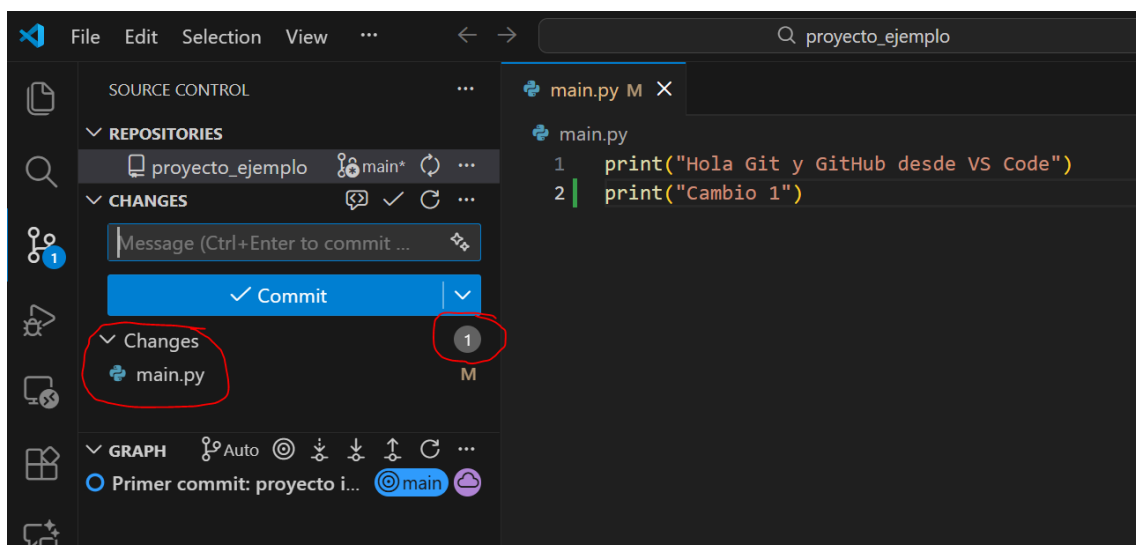
Tras pulsar es posible que nos pida loguearnos y tendremos que seleccionar si queremos crear un repositorio público o privado



Vamos a nuestro Github y comprobamos que nuestro repositorio y código están ahí.

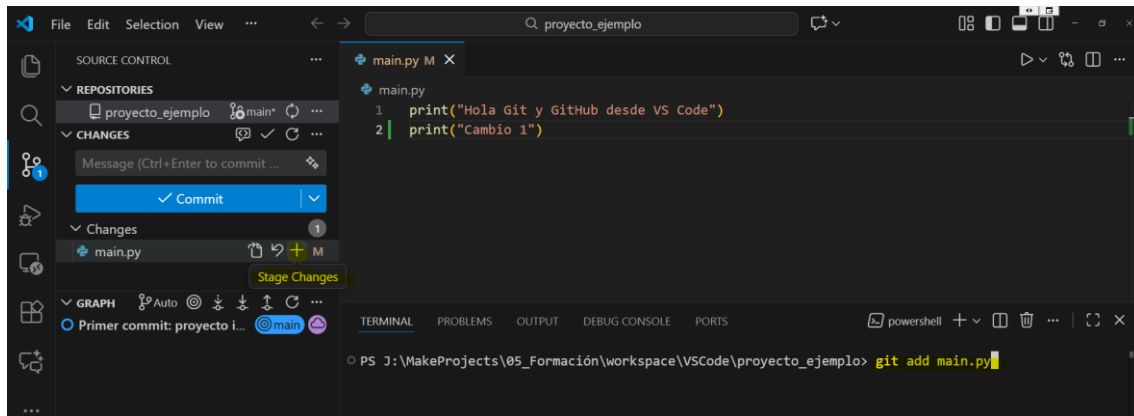
## 5.8 Actualizar un repositorio en Github

1. Añade una nueva línea de código en el fichero y comprobamos que GIT detecta el cambio:

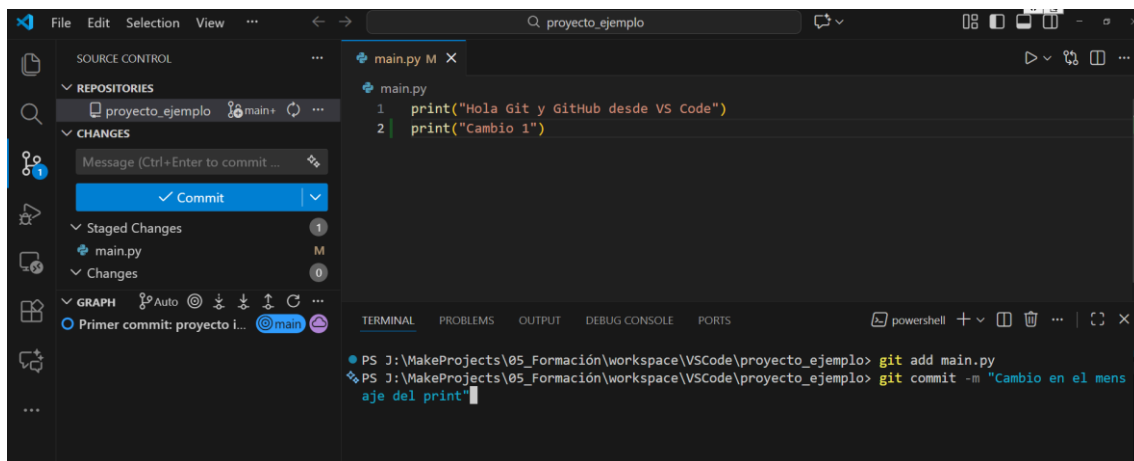


2. Ahora registraremos el cambio, tenemos dos opciones:

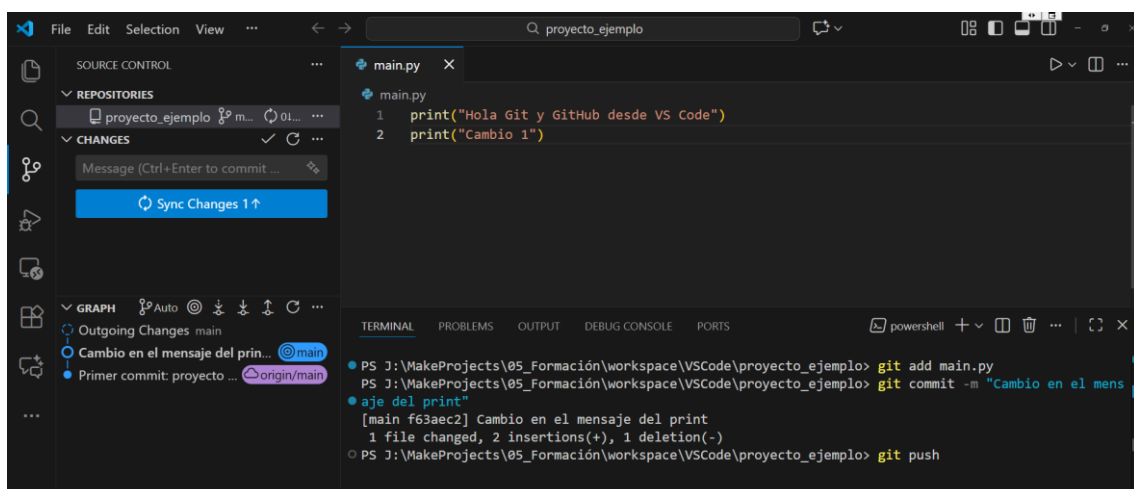
- En “Source Control”, pasa el ratón sobre el archivo y dale al +.
- Por terminal: **git add main.py** o **git add .** (para múltiples cambios)



3. A continuación, haremos el commit, la actualización, para ello tenemos dos opciones:
  - En “Source Control”, escribir un mensaje y pulsar Commit
  - En terminal: **git commit -m "Cambio en el mensaje del print"**



4. Por último subimos el cambio al repositorio de Github:
  - En “Source Control”, pulsamos “Syn Changes”
  - Por terminal: **git push**



Si todo va bien, al refrescar la página de GitHub **verás tus archivos**.

---

## 6. Descargar ese mismo proyecto en otra carpeta (clonar)

Ahora quieres **bajarte ese proyecto** a otra ubicación y con otro nombre de carpeta.

### 6.1 Clonar el repositorio

1. En tu ordenador, crea una nueva carpeta general, por ejemplo:  
C:\Users\TuUsuario\Proyectos\copias\_proyectos
2. Abre ahí una terminal (o usa VS Code y navega con cd):

```
cd C:\Users\TuUsuario\Proyectos\copias_proyectos
```

3. Clona el repo de GitHub:

```
git clone https://github.com/tu_usuario/mi_proyecto_git.git mi_proyecto_git
```

- El último parámetro (mi\_proyecto\_git) es el **nombre de la carpeta donde se clonará**.
  - Si lo omites, usará el nombre original (mi\_proyecto\_git).
4. Abre esta nueva carpeta en VS Code:
    - **File > Open Folder...** → mi\_proyecto\_git\_copia.

Ahora tienes **otra copia del mismo proyecto** pero en otra ruta.

---

## 7. Hacer un cambio en la copia y subir ese cambio a GitHub

### 7.1 Modificar un archivo

En mi\_proyecto\_git, abre main.py y cambia algo, por ejemplo:

```
print("Hola Git y GitHub desde la COPIA del proyecto")
```

Guarda el archivo.

### 7.2 Ver el estado en Git

En la terminal de VS Code (dentro de mi\_proyecto\_git\_copia):

```
git status
```

Verás main.py como “modified”.

### 7.3 Añadir el cambio y hacer commit

```
git add main.py
```

```
git commit -m "Cambio mensaje en la copia del proyecto"
```

### 7.4 Subir el cambio a GitHub

Como este repositorio ya se clonó desde GitHub, el remoto origin ya está configurado. Solo necesitas hacer:

```
git push
```

Listo: si vas a GitHub y recargas la página del repositorio, verás el nuevo mensaje en el archivo.

---