

Taller: Crear un proyecto en Git y GitHub

Objetivo:

Aprender a crear un proyecto local en Git, gestionarlo con ramas y subirlo a GitHub.

Enunciado del Taller

1. Crear un proyecto llamado miPrimerGit usando Git.
2. Subirlo a GitHub.
3. Realizar las siguientes actividades:
 - a) Crear con Visual Studio Code (VSC) una página web index.html con un saludo e incluir la imagen imagen1.jpg.
 - b) Crear una rama llamada agregar-imagen2, donde se incluya imagen2.jpg.
 - c) Hacer un merge de la rama con main.
 - d) Subir todo a GitHub.
 - e) Realizar el ejercicio del numeral 12.

Desarrollo Paso a Paso

1. Instalar Git en Windows

- a) Descargar desde: <https://git-scm.com/download/win>

verificar que se instalo correctamente: en powershell o en la consola creada por el instalador denominada MINGW64 teclear el siguiente comando:

```
git --version
```

2. Crear una cuenta en GitHub

Ingresar a: <https://github.com>

Registrar: Usuario, correo electrónico y contraseña.

3. Abrir la consola de Git

Buscar Git Bash (se instala con Git). Permite usar comandos de Git similares a Linux.

4. Configurar la identidad

```
git config --global user.name "usuario"  
git config --global user.email "correo_github@example.com"
```

El correo debe ser el mismo usado en GitHub.

5. Crear el proyecto

```
mkdir miPrimerGit  
cd miPrimerGit
```

Crear index.html y agregar imagen1.jpg en la carpeta.

6. Iniciar el repositorio Git

```
git init  
git add index.html imagen1.jpg  
git commit -m "Primer commit: index.html e imagen1.jpg"
```

7. Crear el repositorio en GitHub

Entrar a GitHub, crear nuevo repositorio: miPrimerGit (vacío, sin README).

8. Vincular proyecto local con GitHub

```
git remote add origin https://github.com/usuario/miPrimerGit.git  
git remote -v  
git branch -M main  
git push -u origin main
```

9. Crear una rama para agregar otra imagen

```
git branch  
git branch agregar-imagen2  
git checkout agregar-imagen2
```

Agregar imagen2.jpg y actualizar index.html

```
git add .  
git commit -m "Agrego segunda imagen"
```

10. Fusionar la rama con main

```
git checkout main  
git merge agregar-imagen2
```

11. Subir cambios a GitHub

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

12. Ejercicio final

Agregar taller.pdf y actualizar repositorio:

```
git add taller.pdf  
git commit -m "Agrego taller.pdf"  
git push  
git pull
```

Nota 1

`git push -u origin main`

Nota: cuando se creó el proyecto Git debe tener cuidado de haber cambiado 'master' por 'main'; de lo contrario use 'master' en lugar de 'main'.

Mucho cuidado: cuando salga la ventana de autorizaciones, inicialmente usar por default el "explorador" y con mucho cuidado y paciencia teclear las credenciales de la cuenta GitHub y esperar hasta que se realice el push.

Importante: en este paso no lo vaya a interrumpir.

Si por algún motivo no le funciona, intente hacerlo usando device code o desinstale Git y vuelva a realizar todo el proceso reseteando todo el historial del explorador.

Nota 2: Sincronización entre Git local y GitHub

Existen dos formas de mantener sincronizado el proyecto entre su computadora local y GitHub:

1. **Flujo normal (local → GitHub):**

- Realizar cambios en local.
- Ejecutar ``git add``, ``git commit``.
- Subir con ``git push``.

2. **Flujo inverso (GitHub → local):**

- Subir o modificar archivos directamente en GitHub (por ejemplo, crear o subir un PDF desde el navegador).
- Actualizar la copia local con ``git pull``.

Importante:

- Antes de realizar un ``git push`` desde el local, conviene ejecutar ``git pull`` para traer posibles cambios desde GitHub.
- Si un mismo archivo se modifica en ambos lados sin sincronizar, aparecerá un **merge conflict**, que se debe resolver manualmente.