

Práctica1: Creación y Gestión de Repositorios con Git y GitHub

Objetivo

Aprender a crear, configurar y gestionar un repositorio utilizando **Git** desde la terminal y conectarlo con **GitHub** para compartir código y colaborar con otros.

1. Preparación del entorno

1.1 Verificar si Git está instalado

```
git --version
```

Si no aparece un número de versión, instala Git:

- **Windows:** <https://git-scm.com/download/win>
- macOS: <https://git-scm.com/download/mac>
- Linux (Debian/Ubuntu):
- `sudo apt install git`

```
Microsoft Windows [Versión 10.0.26100.6584]  
(c) Microsoft Corporation. Todos los derechos reservados.  
  
C:\Users\LUIS>git --version  
git version 2.51.0.windows.1
```

1.2 Configuración inicial

Reemplaza los datos con tu nombre y correo (usa el mismo correo de GitHub):

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

```
git config --global user.email "tucorreo@ejemplo.com"
```

Puedes verificar la configuración con:

```
git config --list
```

```
C:\Users\LUIS>git config --list  
diff.astextplain.textconv=astextplain  
filter.lfs.clean=git-lfs clean -- %f  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
filter.lfs.required=true  
http.sslbackend=schannel  
core.autocrlf=true  
core.fscache=true  
core.symlinks=false  
pull.rebase=false  
credential.helper=manager  
credential.https://dev.azure.com.usehttppath=true  
init.defaultbranch=master  
core.editor="C:\Users\LUIS\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait  
user.name=LUIS  
user.email=luisrodriguezregueiro@gmail.com  
alias.s=status --short  
alias.lg=log --graph --abbrev-commit --decorate --format=format:'%C(bold blue)%h%C(reset) - %C(bold green)(%ar)%C(reset)  
%C(white)%s%C(reset) %C(dim white)- %an%C(reset)%C(bold yellow)%d%C(reset)' --all
```

2. Crear un repositorio local

2.1 Crear una carpeta de trabajo

mkdir mi-primer-repo

cd mi-primer-repo

```
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical>mkdir mi-primer-repo
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical>cd
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical>cd mi-primer-repo
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical\mi-primer-repo>
```

mi-primer-repo	12/10/2025 12:53	Carpeta de archivos	
Entregable	12/10/2025 12:50	Texto OpenDocum...	11 KB
Práctica1	07/10/2025 11:19	Documento Adob...	3.877 KB
rubrica_Practica1	07/10/2025 11:28	Documento Adob...	53 KB

2.2 Inicializar Git

git init

Git creará una carpeta oculta .git que contiene la base de datos del repositorio.

```
Símbolo del sistema
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical\mi-primer-repo>git init
Initialized empty Git repository in C:/Users/LUIS/Desktop/Módulo/SOTRONDIO/Entornos de desarrollo/Practica GIT-GITHUB/Práctical/Práctical/mi-primer-repo/.git/
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical\mi-primer-repo>
```

Nombre	Fecha de modificación	Tipo	Tamaño
.git	12/10/2025 12:55	Carpeta de archivos	

3. Realizar el primer commit

3.1 Crear un archivo

echo "# Mi primer repositorio" > README.md

```
Símbolo del sistema
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical\mi-primer-repo>git init
Initialized empty Git repository in C:/Users/LUIS/Desktop/Módulo/SOTRONDIO/Entornos de desarrollo/Practica GIT-GITHUB/Práctical/Práctical/mi-primer-repo/.git/

C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical\mi-primer-repo>
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical\mi-primer-repo>echo "Mi p
rimer repositorio" > README.md

C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical\mi-primer-repo>
```

3.2 Ver el estado del repositorio

git status

3.3 Añadir el archivo al área de preparación

git add README.md

3.4 Guardar los cambios en un commit

git commit -m "Primer commit: añadir README"

```
Símbolo del sistema
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical\mi-primer-repo>git init
Initialized empty Git repository in C:/Users/LUIS/Desktop/Módulo/SOTRONDIO/Entornos de desarrollo/Practica GIT-GITHUB/Práctical/Práctical/mi-primer-repo/.git/

C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical\mi-primer-repo>
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical\mi-primer-repo>echo "Mi primer repositorio" > README.md

C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical\mi-primer-repo>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical\mi-primer-repo>git add README.md

C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical\mi-primer-repo>git commit -m "Primer commit:añadir REA
DME"
[master (root-commit) 90d84d3] Primer commit:añadir README
1 file changed, 1 insertion(+)
create mode 100644 README.md

C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical\mi-primer-repo>
```

SOTRONDIO > Entornos de desarrollo > Practica GIT-GITHUB > Práctica1 > Práctica1 > mi-primer-repo				
Ordenar Ver ...				
Nombre	Fecha de modificación	Tipo	Tamaño	
.git	12/10/2025 12:57	Carpeta de archivos		
README	12/10/2025 12:56	Archivo de origen ...	1 KB	

4. Conectar con GitHub

4.1 Crear una cuenta en GitHub

Ir a <https://github.com> y registrarse si aún no tienes cuenta.

4.2 Crear un nuevo repositorio en GitHub

En GitHub:

- Haz clic en **New repository**.
- Ponle el mismo nombre que tu carpeta local (mi-primer-repo).
- No marques ninguna opción de inicialización (sin README, sin .gitignore).
- Copia la URL del repositorio (HTTPS o SSH).

4.3 Conectar el repositorio local con GitHub

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

4.4 Verificar la conexión

```
git remote -v
```

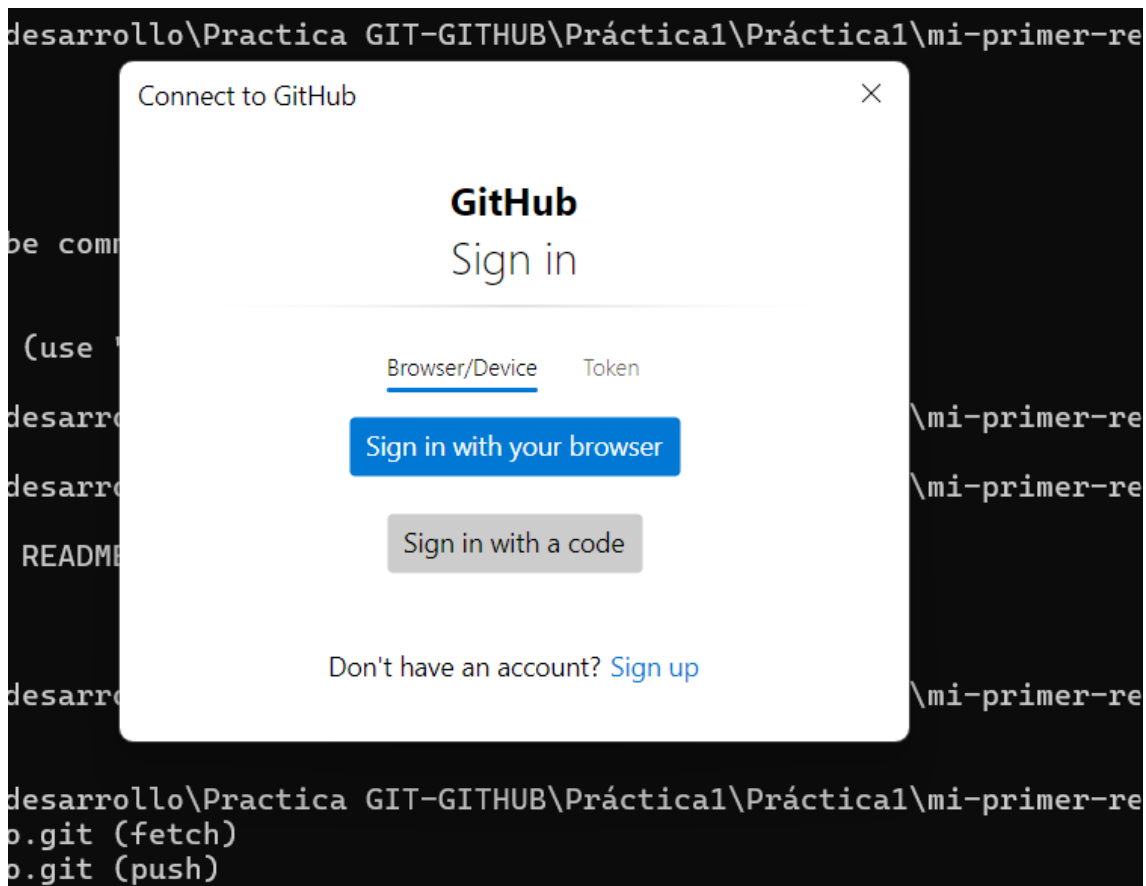
```
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical\mi-primer-repo>git remote add origin https://github.com/tu-usuario/mi-primer-repo.git
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical\mi-primer-repo>git remote -v
origin  https://github.com/tu-usuario/mi-primer-repo.git (fetch)
origin  https://github.com/tu-usuario/mi-primer-repo.git (push)
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctical\Práctical\mi-primer-repo>
```

5. Subir los cambios a GitHub

5.1 Subir la rama principal

```
git branch -M main
```

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



5.2 Autenticación en GitHub

Si GitHub te pide autenticación, genera un **Personal Access Token (PAT)** y úsalo como contraseña.

Cómo generar un PAT (actualizado 2025)

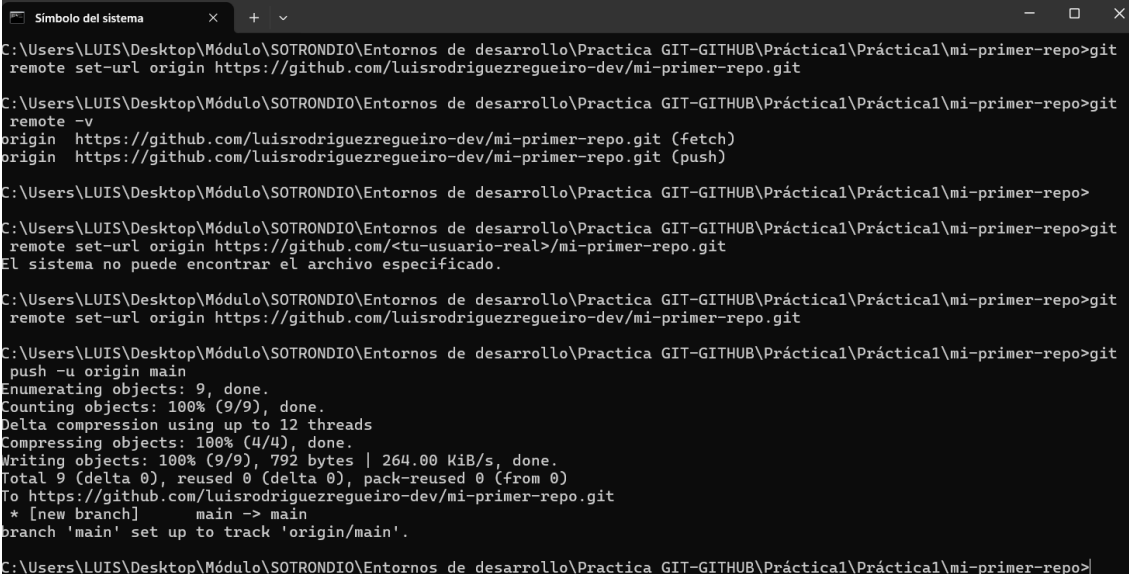
1. Inicia sesión en GitHub.
2. Haz clic en tu foto (arriba a la derecha) → **Settings**.
3. En el menú lateral, selecciona **Developer settings**.
4. Ve a Personal access tokens → Fine-grained tokens.
5. Haz clic en Generate new token.
6. Ponle un nombre (por ejemplo: *Git desde terminal*).
7. Establece una **fecha de expiración** (por ejemplo, 90 días).
8. En **Resource owner**, selecciona tu cuenta.
9. En **Repository access**, elige *All repositories* o uno específico.
10. En Permissions, concede Read and write para “Contents”.

11. Haz clic en **Generate token**.
12. Copia el token (solo se muestra una vez).

Luego, cuando Git te pida credenciales:

- **Usuario:** tu nombre de usuario de GitHub.
- **Contraseña:** pega el token generado.

Si usas Windows o macOS, el token se guardará automáticamente en el gestor de credenciales.



```
Símbolo del sistema
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctica1\Práctica1\mi-primer-repo>git remote set-url origin https://github.com/luisrodriguezregueiro-dev/mi-primer-repo.git
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctica1\Práctica1\mi-primer-repo>git remote -v
origin https://github.com/luisrodriguezregueiro-dev/mi-primer-repo.git (fetch)
origin https://github.com/luisrodriguezregueiro-dev/mi-primer-repo.git (push)
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctica1\Práctica1\mi-primer-repo>
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctica1\Práctica1\mi-primer-repo>git remote set-url origin https://github.com/<tu-usuario-real>/mi-primer-repo.git
El sistema no puede encontrar el archivo especificado.
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctica1\Práctica1\mi-primer-repo>git remote set-url origin https://github.com/luisrodriguezregueiro-dev/mi-primer-repo.git
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctica1\Práctica1\mi-primer-repo>git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (9/9), 792 bytes | 264.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/luisrodriguezregueiro-dev/mi-primer-repo.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo\Practica GIT-GITHUB\Práctica1\Práctica1\mi-primer-repo>
```

6. Clonar un repositorio existente

Para copiar un proyecto de GitHub a tu computadora:

```
git clone https://github.com/usuario/repositorio.git
```

8. Buenas prácticas

- Haz commits pequeños y descriptivos.
- Usa ramas para nuevas características o correcciones.
- Sincroniza con git pull antes de empezar a trabajar.
- Incluye un README.md que describa el proyecto.
- Utiliza .gitignore para excluir archivos innecesarios.

9. Actividad final

1. Crea tu propio repositorio local y súbelo a GitHub. En el repositorio debes incluir TODAS las operaciones y ejemplos vistos en clase hasta el momento.

```
error: remote origin already exists.

C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo>git push -u origin main
remote: Repository not found.
fatal: repository 'https://github.com/TU_USUARIO/NOMBRE_REPO.git/' not found

C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo>git remote -v
origin https://github.com/TU_USUARIO/NOMBRE_REPO.git (fetch)
origin https://github.com/TU_USUARIO/NOMBRE_REPO.git (push)

C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo>git remote set-url origin https://github.com/luisrodriguezregueiro-dev/repositorio-temario.git

C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo>git remote -v
origin https://github.com/luisrodriguezregueiro-dev/repositorio-temario.git (fetch)
origin https://github.com/luisrodriguezregueiro-dev/repositorio-temario.git (push)

C:\Users\LUIS\Desktop\Módulo\SOTRONDIO\Entornos de desarrollo>git push -u origin main
Enumerating objects: 102, done.
Counting objects: 100% (102/102), done.
Delta compression using up to 12 threads
Compressing objects: 100% (88/88), done.
Writing objects: 100% (102/102), 62.81 MiB | 1.68 MiB/s, done.
Total 102 (delta 14), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (14/14), done.
To https://github.com/luisrodriguezregueiro-dev/repositorio-temario.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

luisrodriguezregueiro-dev Añadir practica GIT-GITHUB		4dfb1a7 · 6 minutes ago	🕒 7 Commits
BBDD	Añadir asignaturas para convalidar	4 days ago	
Digitalización	Añadir asignaturas para convalidar	4 days ago	
Entornos de desarrollo	Añadir practica GIT-GITHUB	6 minutes ago	
IPEI1	Añadir asignaturas para convalidar	4 days ago	
Inglés	Añadir asignaturas para convalidar	4 days ago	
Lenguaje de marcas	Actualización 02-bases	last week	
Programación	Añadir acc directo a proyectos Programación en Eclipse	2 days ago	
Sistemas informáticos	Añadir asignaturas para convalidar	4 days ago	
Sostenibilidad	Añadir asignaturas para convalidar	4 days ago	

1. Prepara un glosario con los términos o instrucciones que utilizas en esta actividad.

Glosario de términos y explicación de los mismos:

Repositorio: Espacio donde se almacena y organiza el código de un proyecto. Puede ser local o remoto (en GitHub).

Git: Sistema de control de versiones que permite registrar los cambios realizados en los archivos de un proyecto y volver a versiones anteriores si es necesario.

GitHub: Plataforma en línea que permite alojar repositorios Git, colaborar con otras personas y compartir proyectos.

Terminal o línea de comandos: Herramienta que permite ejecutar instrucciones directamente mediante texto, sin necesidad de una interfaz gráfica.

git --version: Comando que muestra la versión de Git instalada, útil para verificar que está correctamente configurado.

git config: Sirve para establecer o consultar la configuración de Git, como el nombre de usuario o el correo electrónico asociados a los commits.

git init: Inicializa un nuevo repositorio Git en la carpeta actual, creando la estructura necesaria para comenzar a trabajar con control de versiones.

mkdir: Crea una nueva carpeta o directorio en el sistema.

cd: Permite moverse entre carpetas en la terminal.

echo: Crea o escribe contenido dentro de un archivo desde la línea de comandos.

README.md: Archivo de texto (en formato Markdown) que describe el propósito, uso e información básica del proyecto.

git status: Muestra el estado actual del repositorio: qué archivos han cambiado, cuáles están preparados para un commit y cuáles no.

git add: Agrega archivos al área de preparación (staging area) antes de realizar un commit.

git commit -m: Guarda los cambios en el historial del repositorio con un mensaje descriptivo que explica lo que se hizo.

git branch -M main: Renombra la rama principal del proyecto como 'main', que es el nombre estándar actual en GitHub.

git remote add origin: Conecta el repositorio local con un repositorio remoto en GitHub mediante una URL.

git remote -v: Muestra las direcciones (URLs) de los repositorios remotos configurados.

git push -u origin main: Envía los commits locales al repositorio remoto en GitHub y establece una relación de seguimiento entre ambas ramas.

git clone: Copia un repositorio remoto (de GitHub u otra fuente) a tu equipo local, incluyendo todo su historial de cambios.

Personal Access Token (PAT): Clave de acceso generada en GitHub que sustituye a la contraseña al autenticarte desde la terminal.

git pull: Descarga los cambios más recientes del repositorio remoto y los combina con tu versión local.

.gitignore: Archivo que indica qué elementos deben excluirse del control de versiones (por ejemplo, archivos temporales o de configuración local).

commit: Registro individual de cambios realizados en el código. Cada commit guarda una versión concreta del proyecto.

push: Acción de subir los commits locales a un repositorio remoto en GitHub.

pull: Acción de descargar y fusionar los cambios realizados por otros colaboradores en el repositorio remoto.

rama (branch): Línea de desarrollo independiente dentro del proyecto. Permite trabajar en nuevas funciones sin afectar la versión principal.

10. Entregable

Capturas de los pasos realizados

Evidencias del repositorio local y de GitHub

Enlace al repositorio GitHub

Sube el entregable también a GitHub