Manual paso a paso para clonar, publicar y administrar un repositorio en GitHub

Este manual te guía desde cero en el uso de Git y GitHub, explicando cómo instalar, configurar, clonar, publicar y administrar un repositorio, con buenas prácticas y recursos útiles.



🔧 1. Tener Git instalado

Antes de comenzar, asegúrate de tener:

- Una cuenta en GitHub.
- **Git** instalado en tu computadora:
 - Windows: Descárgalo desde git-scm.com.
 - Linux:

```
sudo apt update && sudo apt install git -y
```

MacOS:

```
brew install git
```

• Un editor de código como Visual Studio Code.

Verifica la instalación de git con:

```
git --version
```

2. Configurar usuario en Git

Configura tu nombre y correo (el mismo que usas en GitHub):

```
git config --global user.name "Tu Nombre"
git config --global user.email "tu_correo@example.com"
```

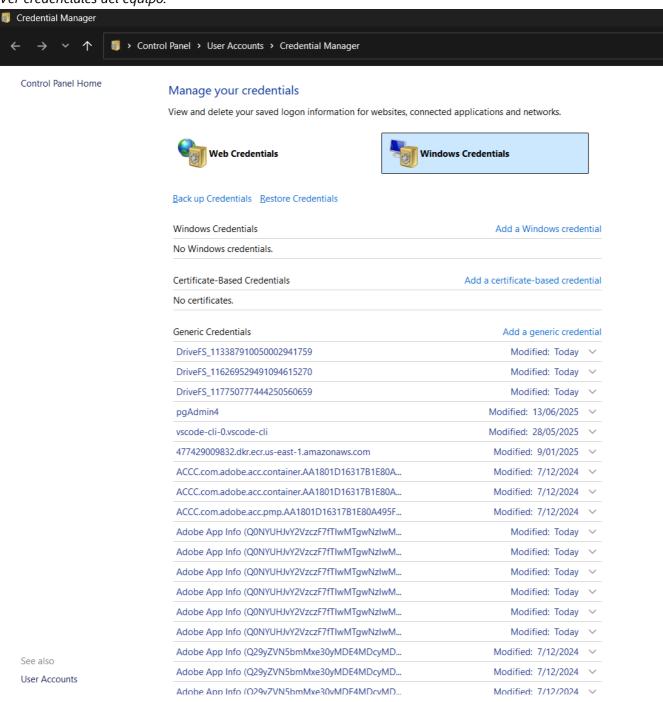
Verifica la configuración:

```
git config --list
```

2.1. Credenciales y autenticación

- En Windows, puedes usar credenciales de Windows o autenticación de GitHub.
- Si usas un equipo público, cierra sesión y elimina credenciales.
- Para mayor seguridad, puedes usar llaves SSH (Guía oficial).

Ver credenciales del equipo:



Internet or network address: git:https://github.com

User name: ariel5253
Password: ••••••

Persistence: Local computer

Edit Remove

☐ 3. Clonar un repositorio

Para copiar un repositorio de GitHub a tu computadora:

- 1. Copia la URL del repositorio (HTTPS o SSH).
- 2. Ejecuta el comando en la ruta donde quieres guardar el proyecto (ejemplo: C:\www\corhuila):

```
git clone https://github.com/usuario/mi-proyecto.git
```

3. Entra al directorio:

cd mi-proyecto

4. Obtener cambios del repositorio remoto

Para actualizar tu copia local con los cambios de GitHub:

git pull

Si trabajas con ramas:

git pull origin nombre-rama

💪 5. Publicar cambios en GitHub

Para subir tus cambios al repositorio remoto:

1. Descarga primero los cambios remotos:

```
git pull
```

2. Verifica el estado de los archivos:

```
git status
```

3. Agrega los archivos modificados:

```
git add .
```

4. Realiza un commit con un mensaje descriptivo (preferiblemente en inglés):

```
git commit -m "Describe your change"
```

5. Sube los cambios:

```
git push
```

% Buenas prácticas

- Haz commits pequeños y frecuentes.
- Usa mensajes claros y descriptivos.
- Sincroniza antes de trabajar (git pull).
- Usa ramas para nuevas funciones o correcciones.
- No subas información sensible (contraseñas, API keys).

Resumen rápido de comandos

```
# Clonar un repositorio
git clone URL

# Ver estado de cambios
git status

# Agregar archivos
git add archivo.txt

# Guardar cambios
git commit -m "mensaje"
```

```
# Subir cambios
git push
# Descargar cambios
git pull
```