

# Guía de aprendizaje 3: Enlace y Sincronización con Repositorios Remotos (GitHub)

Colaboración y Repositorios Remotos: Creación, Clonación y el Flujo Push/Pull.

Tema que se va a ver en la guía de estudio

Creación de cuenta en GitHub, creación de repositorios remotos, conexión de repositorios locales y remotos (git remote add), clonación (git clone), y comandos de sincronización (git push, git pull).

## Objetivo general

- Lograr la integración del trabajo local con el servidor en la nube (GitHub) para habilitar la colaboración y la persistencia de cambios.

## Objetivos específicos

- Crear una cuenta y un repositorio en GitHub.
- Enlazar un repositorio local existente (con su historial) a un repositorio remoto vacío.
- Diferenciar y ejecutar correctamente git clone vs. git remote add.
- Utilizar git push para enviar cambios locales al servidor remoto.
- Utilizar git pull para traer y fusionar cambios del servidor remoto.



Servicio Nacional de Aprendizaje – SENA

Centro de Tecnología de la Manufactura Avanzada – Regional Antioquia

## Introducción

Para que el trabajo en equipo sea posible, se necesitan repositorios remotos que actúen como puntos de sincronización centralizados, a los que todos los desarrolladores puedan acceder. GitHub es el servicio más popular para alojar estos repositorios. Esta guía detalla cómo se crea la base remota y cómo se transfieren los commits entre la máquina local y el servidor, mediante los comandos push (empujar) y pull (jalar).

## Desarrollo del tema:

### 1. Creación de una Cuenta y Repositorio en GitHub:

- **Cuenta:** Visite <https://github.com>, seleccione un nombre de usuario, proporcione un correo y una contraseña, y verifique su dirección de correo (**el correo debe ser el mismo usado en la guía anterior**).
- **Repositorio Remoto:** Un repositorio se utiliza para organizar un solo proyecto.

#### Paso a Paso de Creación de Repositorio Remoto (ej. estructura-datos):

- En GitHub, haga clic en (+) en la esquina superior derecha y seleccione Nuevo repositorio.
- Asigne el nombre deseado, por ejemplo, estructura-datos.
- Escriba una breve descripción.
- Seleccione la visibilidad (público o privado).
- Opcionalmente, puede incluir un archivo README.md o una licencia.
- Haga clic en Crear repositorio. Importante: Guarde la URL del repositorio remoto (HTTPS o SSH), ya que se necesitará para la conexión.

### 2. Clonación vs. Enlace:

- **Clonación (git clone):** Se usa cuando se quiere empezar a trabajar con un repositorio que ya existe en el servidor. Descarga una copia completa del repositorio (historial y archivos) en la máquina local. `$ git clone <URL_del_repositorio>`.
- **Enlace (git remote add):** Se usa cuando ya existe un repositorio local (creado con git init) y se necesita conectarlo con un repositorio remoto vacío (que se acaba de crear en GitHub).



Servicio Nacional de Aprendizaje – SENA

Centro de Tecnología de la Manufactura Avanzada – Regional Antioquia

**Paso a Paso de Enlace (Unir Local con Remoto):** (Asumiendo que ya creó el repositorio local y el remoto):

- En su terminal local, ubique la URL del repositorio remoto (ej. <https://github.com/usuario/estructura-datos.git>).
- Asigne un alias (nombre corto) a la URL remota. El alias estándar para el repositorio principal es origin. \$ git remote add origin <URL\_del\_repositorio>.
- Verifique el enlace: \$ git remote -v. 3. Transferencia de Cambios al Remoto: git push (Empujar): Este comando envía los commits confirmados localmente al repositorio remoto.

**Paso a Paso para el Primer Push (establecer upstream):**

- Una vez hecho el enlace, se sube el historial local a la rama principal remota (ahora llamada main o master): \$ git push --set-upstream origin master (si usa master).

**Nota:** Si la rama principal de su remoto es main, use: \$ git push -u origin main. \* git pull (Jalar): Este comando se usa para traer y fusionar las actualizaciones del servidor remoto a la rama local. Es equivalente a ejecutar git fetch (traer datos) seguido de git merge (fusionar los datos traídos). Es esencial usar git pull antes de git push si existe la posibilidad de que otros colaboradores hayan subido cambios, para evitar rechazos (rejections) del push.

## Ejercicios prácticos:

### Ejercicio 3: Conexión y Primer Push

1. Cree una cuenta en GitHub (si no la tiene).
2. Cree un nuevo repositorio remoto en GitHub llamado mi-proyecto-remoto. No marque la opción de añadir README o .gitignore.
3. En su máquina local, cree una carpeta, inicialice un repositorio en ella (git init) y haga un commit inicial (use la Guía 2 si es necesario).
4. Enlace el repositorio local con el remoto (usando la URL SSH o HTTPS proporcionada por GitHub): \$ git remote add origin <URL\_del\_repo>.
5. Suba su commit al repositorio remoto, asumiendo que la rama principal es master: \$ git push -u origin master
6. Verifique en la interfaz web de GitHub que sus archivos locales han sido subidos.



Servicio Nacional de Aprendizaje – SENA

Centro de Tecnología de la Manufactura Avanzada – Regional Antioquia

## Cuestionario de comprensión:

1. ¿Cuál es el propósito del alias origin en el comando git remote add origin <URL>?
2. Explique qué comandos combina internamente git pull.
3. Si un repositorio remoto ya contiene un historial y deseo trabajar en mi máquina local, ¿qué comando debo usar: git clone o git init seguido de git remote add?

## Materiales necesarios

- Computador
- <https://www.youtube.com/watch?v=Ei1y51K8jQk>
- <https://www.youtube.com/watch?v=HiVnGgYudLY>

## Evidencias de aprendizaje

- Mandar un PDF con la guía resuelta sin errores

## Resultados de aprendizaje

Adquirir conocimientos acerca de... tema (los que se evalúa en Sofía)