

Tarea de GitHub

Objetivo

Crea dos repositorios remotos, **repoA** y **repoB**, conectados a un único repositorio local y realiza las siguientes operaciones utilizando ramas, commits y comandos avanzados de Git.

Este ejercicio cubre una serie de conceptos fundamentales y avanzados de Git, como el trabajo con ramas, la manipulación del historial de commits, el uso de **.gitignore**, la eliminación de commits innecesarios, la edición de commits y la configuración de claves SSH para la subida a repositorios remotos.

Pasos a seguir

1. Crear dos repositorios remotos (**repoA** y **repoB**):

- Configura los dos repositorios remotos que estarán conectados a un único repositorio local.

2. Crear un **README.md** en la rama principal:

- En la rama principal, crea un archivo **README.md** con el siguiente contenido: **Tarea git**
- Añade un archivo adicional llamado **texto.txt**.

3. Crear dos ramas locales, **ramaA** y **ramaB**:

- Crea dos ramas locales (**ramaA** y **ramaB**) a partir de la rama principal.
- Las ramas serán utilizadas para subir contenido a cada repositorio remoto.

4. Trabajar en la **ramaA**:

- En la **ramaA**, realiza los siguientes cambios:
 - Crea una carpeta llamada **carpetaOculto**.
 - Dentro de **carpetaOculto**, crea un archivo llamado **archivoOculto.txt**.
 - Fuera de la carpeta, crea un archivo adicional llamado **archivoOculto2.txt**.
- Crea un archivo **.gitignore** para evitar la subida de los archivos **archivoOculto.txt** y **archivoOculto2.txt** al repositorio **repoA**.

5. Trabajar en la **ramaB**:

- En la **ramaB**, replica la misma estructura de carpetas y archivos que en la **ramaA**, pero **no** crees un archivo **.gitignore**.
- Sube el contenido al repositorio **repoB**.

6. Eliminar la **ramaB**:

- Dado que no se utilizará más el repositorio **repoB**, elimina la **ramaB** con el siguiente comando:

```
git branch -D ramaB
```

- Ahora solo trabajaremos en la **ramaA**.

7. Realizar tres commits adicionales en la **ramaA**:

- Realiza tres commits en la **ramaA** que se añadan a los ya existentes.

8. Navegar entre commits:

- Utiliza el comando **git checkout** para moverte al **primer commit** de los tres que acabas de crear. Esto demostrará la capacidad de navegar entre diferentes puntos de la historia del proyecto.
- Luego, regresa al último commit utilizando **git checkout** nuevamente.

9. Eliminar commits innecesarios:

- En el tercer commit, supongamos que los **dos últimos commits no son necesarios**.
- Utiliza el siguiente comando para eliminarlos del historial, manteniendo solo el primero:

```
git reset --hard <commit_hash_del_primer_commit>
```

10. Crear un nuevo repositorio local y remoto:

- Crea un nuevo repositorio local y un nuevo repositorio remoto.
- Genera un par de claves SSH y configura la autenticación SSH.
- Finalmente, realiza una subida del archivo **README.md** al nuevo repositorio remoto utilizando SSH.