# “Como subir archivos a GitHub”


**Nombre:**

Renata Alejandra Salazar Caiza

**NRC:**

**1323**


**Asignatura de la Programación Orientada a Objetos**


**Docente:**

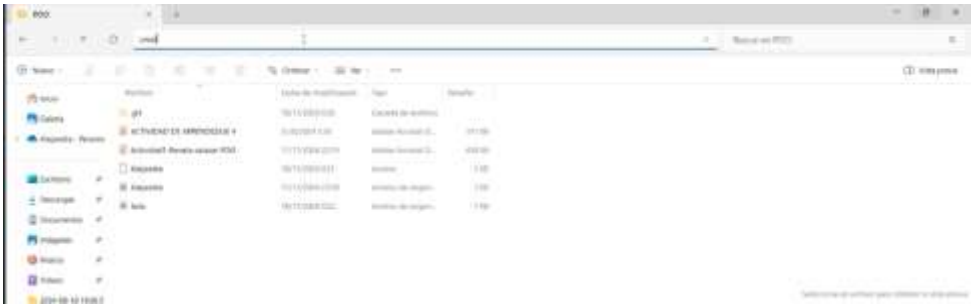
LUIS ENRIQUE JARAMILLO MONTAÑO

1. **Como subir archivos de nuestra carpeta a GitHub**

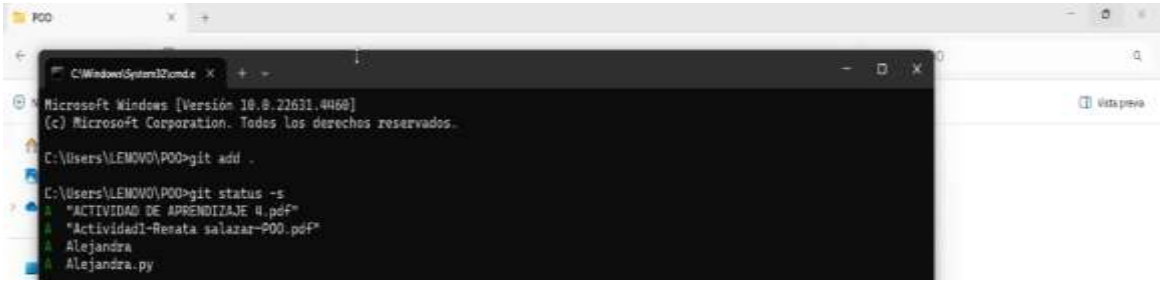
### Paso 3.1: Abrir el Símbolo del sistema en la carpeta del repositorio

Para comenzar, debemos abrir el **Símbolo del sistema (cmd)** directamente en la carpeta donde hemos configurado la conexión con nuestro repositorio de GitHub.

- Navega hasta la carpeta del repositorio en tu explorador de archivos.
- Haz clic en la barra de direcciones de la parte superior de la ventana (donde se muestra la ruta actual del directorio).
- Escribe cmd en esa barra y presiona **Enter**.



### Paso 3.2: Agregar archivos al área de preparación




- **git add . :** Este comando agrega todos los archivos y cambios detectados en el directorio actual al área de preparación
- El punto (.) al final indica que se incluirán **todos los cambios** realizados en los archivos del proyecto.

Después de ejecutar este comando, es recomendable verificar el estado de los archivos que están en el área de preparación. Para esto, usa el siguiente comando: Este comando muestra el estado actual de los archivos en formato resumido. Los archivos nuevos o modificados aparecerán con la letra A (en color verde), indicando que están listos para ser confirmados

**Paso 3.3: Confirmar los cambios en el repositorio local**


Ahora que los archivos están en el área de preparación, es momento de confirmar los cambios en el repositorio local. Para esto:

- **git commit:** Este comando guarda los cambios en el repositorio local.
- **-m "Mensaje descriptivo":** Este parámetro permite añadir un mensaje que describa los cambios realizados.

**Paso 3.4: Sincronizar con el repositorio remoto en GitHub**

Para finalizar, debemos sincronizar los cambios del repositorio local con el repositorio remoto en GitHub. Esto se realiza en dos pasos fundamentales:

Antes de enviar los cambios, es buena práctica actualizar el repositorio local con las modificaciones que puedan haberse realizado en el repositorio remoto. Para ello, utiliza: “**git pull”** Este comando descarga y aplica los cambios más recientes desde el repositorio remoto al local. Así se evitan posibles conflictos.

Después de haber confirmado los cambios localmente y asegurarte de que tu repositorio está actualizado, utiliza el siguiente comando para enviarlos: “**Git push**” Este comando sube los cambios desde el repositorio local al remoto, asegurando que todos los archivos estén sincronizados.