



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA



Universidad de las Fuerzas Armadas

## “Informe: Como subir archivos a GitHub”

**Nombre:**

1. Brayan Stehp Mendoza Márquez

**NRC: 1323**

**Asignatura de la Programación Orientada a Objetos**

**Docente:**

LUIS ENRIQUE JARAMILLO MONTAÑO

21 de noviembre de 2024

## 1. Introducción

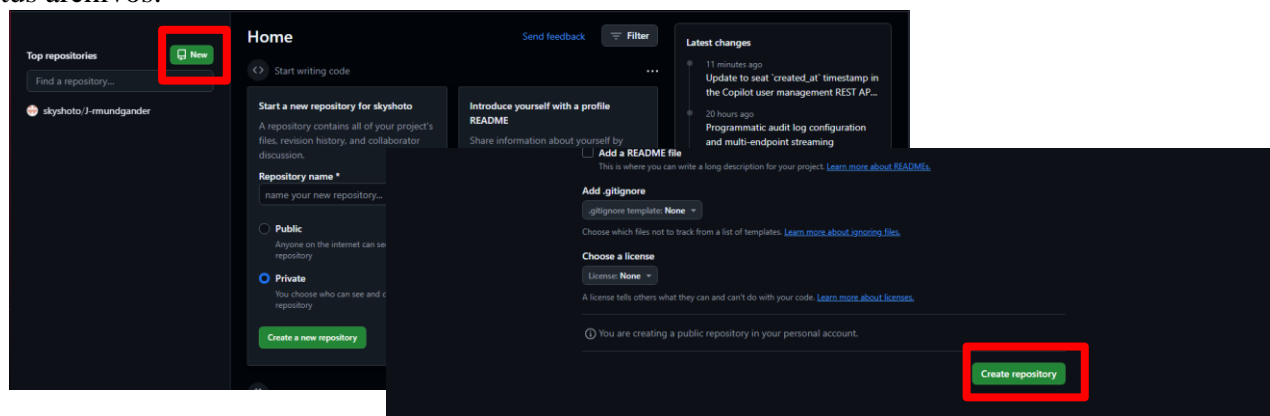
En el desarrollo de proyectos académicos o profesionales, es común trabajar con repositorios que permiten organizar y compartir archivos de manera eficiente. GitHub es una plataforma ampliamente utilizada para este propósito, ya que facilita el control de versiones y la colaboración en equipo. Pero antes de comenzar tengo que recalcar que para subir un archivo vamos a usar comandos con la denominación “git” que para usarlos necesitamos instalar un programa con el mismo nombre.

### 1.1 Instalación de git

- Accede al sitio oficial: <https://git-scm.com/>.
- Descarga la versión compatible con tu sistema operativo (Windows, macOS o Linux), en mi caso es Windows 64 bits.
- Abre el archivo descargado y sigue las instrucciones del asistente.
- Durante la instalación, selecciona las opciones recomendadas, como la integración de Git en la línea de comandos y la configuración del editor de texto (En mi caso las opciones predeterminadas me sirvieron).

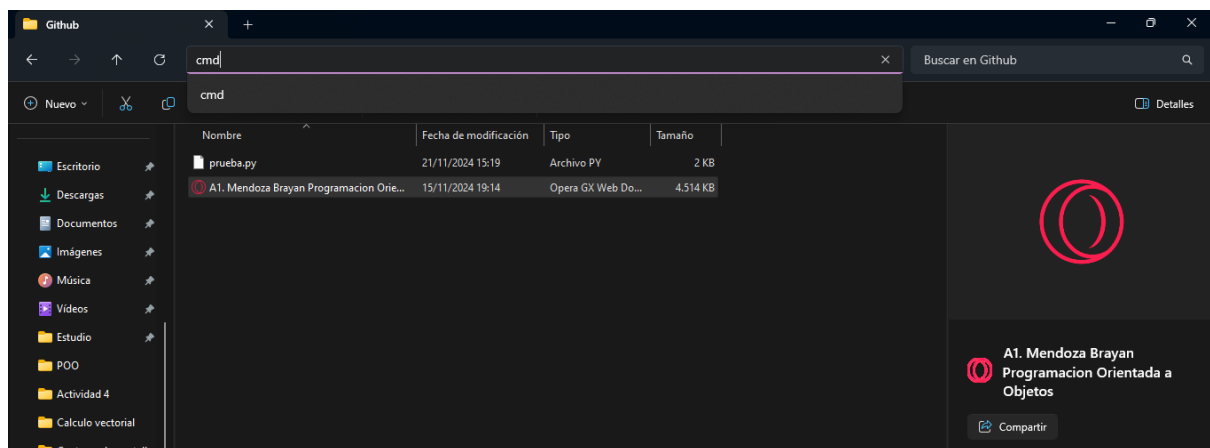
## 2. Creación de un repositorio en GitHub

Para crear un nuevo repositorio en GitHub, haz clic en el botón "New". Esto abrirá la página titulada "Create a new repository", donde deberás asignar un nombre al repositorio. Finalmente, haz clic en "Create repository" para completarlo. Una vez creado, podrás subir tus archivos.



### 3. Como subir archivos de nuestra carpeta a GitHub

**Paso 3.1** Como primer paso, debemos abrir la ventana del **Símbolo del sistema** (**cmd**) en la carpeta donde hemos configurado la conexión de Git con nuestro repositorio de GitHub. Para hacerlo, ve a la barra de direcciones de la carpeta en cuestión, haz clic en ella y escribe **cmd**, seguido de **Enter**. Esto abrirá una ventana del símbolo del sistema directamente en la ubicación de la carpeta seleccionada.



**Paso 3.2** A continuación vamos a escribir **“git add .”** este comando agrega todos los archivos y cambios realizados en el directorio actual al área de preparación. El punto (.) indica que se agregan todos los cambios detectados.

Después veremos el estado con **“git status -s”** de los archivos que hemos subido que deberán aparecer con la letra **“A”** en color verde y el nombre del archivo que estamos observando el estado

```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [Versión 10.0.22631.4460]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\RYZEN\Github>git add .

C:\Users\RYZEN\Github>git status -s
A "A1. Mendoza Brayan Programacion Orientada a Objetos.pdf"
M prueba.py
```

**Paso 3.3** En este paso, utilizaremos el comando **git commit -m**, seguido de un mensaje entre comillas que describa los cambios realizados. Este mensaje servirá como referencia en el historial de GitHub. Por ejemplo:

```
C:\Users\RYZEN\Github>git commit -m "Agrega el informe sobre como subir un archivo"
[main 27d2a74] Agrega el informe sobre como subir un archivo
2 files changed, 47 insertions(+), 1 deletion(-)
create mode 100644 A1. Mendoza Brayan Programacion Orientada a Objetos.pdf
```

Después de ejecutar este comando, el **cmd** mostrará líneas de confirmación indicando que el comentario se ha guardado correctamente y los cambios han sido registrados en el repositorio local.

**Paso 3.4** En esta parte, se realizan dos comandos fundamentales para la sincronización de archivos con el repositorio remoto en GitHub:

**git pull:** Este comando actualiza el repositorio local con los cambios que puedan existir en el repositorio remoto. En el ejemplo, se muestra cómo Git descarga y aplica actualizaciones:

**git push:** Después de realizar un commit, este comando envía los cambios desde el repositorio local al remoto. En el ejemplo:

```
C:\Users\RYZEN\Github>git pull
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (2/2), 891 bytes | 81.00 KiB/s, done.
From https://github.com/skyshot0/J-rmundgander
1aa58ed..41f820c main -> origin/main
Successfully rebased and updated refs/heads/main.

C:\Users\RYZEN\Github>git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 4.20 MiB | 1.38 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/skyshot0/J-rmundgander.git
41f820c..5892e52 main -> main
```

Con esto, hemos finalizado los pasos necesarios para llevar a cabo la subida de archivos y la sincronización con un repositorio en GitHub. Siguiendo estas instrucciones, aseguramos una gestión eficiente de los cambios en nuestros proyectos y mantenemos actualizado nuestro repositorio tanto local como remoto.