

Proyecto final Ciberseguridad

Nombre: Cifrado de Archivos

Nivel: Explorador

Autor: Edwin Hair Hidalgo Narvaez

Video Explicativo: <https://youtu.be/C226631iDEk>

Descripción del Proyecto

Se desarrolla un algoritmo para cifrar archivos ubicados en Google Drive utilizando el lenguaje de programación Python y el entorno de desarrollo Google Colab el cual se basa sobre el algoritmo de Fernet el cual es un cifrado simétrico.

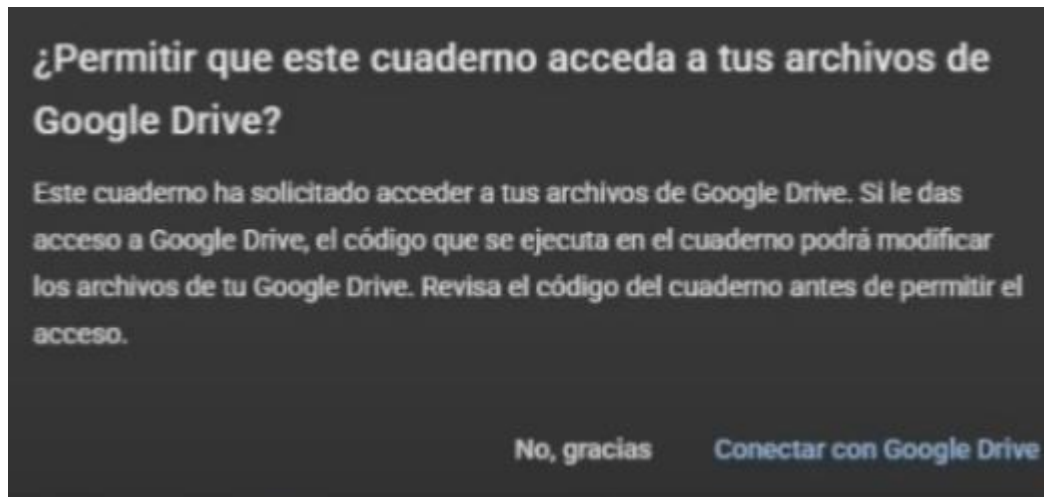
Detalles del código

- Se importan las librerías correspondientes las cuales permitirán la manipulación de los archivos ubicados en Google Drive y el uso del algoritmo de Fernet el cual se utiliza para realizar el cifrado de los archivos.

```
import os
from cryptography.fernet import Fernet
from google.colab import files
from google.colab import drive
```

- Teniendo en cuenta que se van a manejar archivos de Google Drive es necesario solicitar acceso a dichos archivos, por lo cual, al ejecutar el programa, este solicitará acceso el cual permitirá conectarse con los archivos de Google Drive con Google Colab.

```
#Brindar acceso a la unidad de Drive
drive.mount('/content/drive')
# Ruta donde está montado tu Drive
drive_path = '/content/drive/My Drive/'
```



- Una vez se den acceso a los archivos de Drive es necesario digitar la ruta donde se encuentran los archivos a cifrar o descifrar.

La variable que guarda la ruta en folder.

```
# Lista de archivos de una carpeta específica
folder = 'DOCUMENTOS/' # IMPORTANTE digitar en que carpeta se encuentra los archivos a cifrar o descifrar
folder_path = os.path.join(drive_path, folder)
```

**En caso que los archivos se encuentren en la raíz de Google Drive (Mi Unidad) no es necesario digitar la ruta.*

- Ya definida la ruta donde se encuentran los archivos, se procede a listar dichos archivos.

```
def listar_archivos():
    # Muestra los archivos disponibles
    files = os.listdir(folder_path)
    print("\nLista de Archivos:")
    for i, f in enumerate(files):
        print(f"{i}: {f}")

    # Selección del archivo a cifrar
    file_index = int(input("Selecciona el número del archivo: "))
    if file_index < 0 or file_index >= len(files):
        print("Índice de archivo inválido.")
        return None
    else:
        selected_file = os.path.join(folder_path, files[file_index])
        return selected_file
```

- El cifrado de Fernet se base en utilizar una llave para cifrar o descifrar los archivos, por lo tanto, se ejecutan el código para generar dicha clave, también para cargar la clave ya que será necesaria para descifrar los archivos.

```
def generar_clave():
    clave = Fernet.generate_key()
    with open(drive_path+folder+"clave.key", "wb") as archivo_clave:
        archivo_clave.write(clave)
    print("Clave generada y guardada")
    return clave

def cargar_clave(ruta):
    if not os.path.exists(ruta):
        print("El archivo 'clave.key' no existe. Generando una nueva clave...")
        return generar_clave()
    else:
        with open(ruta, "rb") as archivo_clave:
            clave = archivo_clave.read()
        print("Clave cargada")
        return clave
```

- Una vez generada la clave y conociendo que archivos se desea cifrar se ejecuta la función que permite cifrar el archivo, la cual solicita la ruta del archivo y la clave generada con anterioridad.

```
def cifrar_archivo(nombre_archivo, clave):
    f = Fernet(clave)
    try:
        with open(nombre_archivo, "rb") as archivo_original:
            datos_originales = archivo_original.read()

        datos_cifrados = f.encrypt(datos_originales)

        with open(nombre_archivo + ".cifrado", "wb") as archivo_cifrado:
            archivo_cifrado.write(datos_cifrados)
        print(f"\nArchivo cifrado y guardado exitosamente")

    except FileNotFoundError:
        print(f"Error: El archivo '{nombre_archivo}' no fue encontrado.")
    except Exception as e:
        print(f"Ocurrió un error al cifrar el archivo: {e}")
```

- Posteriormente se desarrolla la función para descifrar los archivos la cual requiere la ruta del archivo cifrado y la clave generada con anterioridad.

```
def descifrar_archivo(nombre_archivo_cifrado, clave):  
    f = Fernet(clave)  
    try:  
        with open(nombre_archivo_cifrado, "rb") as archivo_cifrado:  
            datos_cifrados = archivo_cifrado.read()  
  
            datos_descifrados = f.decrypt(datos_cifrados)  
  
            # Eliminar la extensión ".cifrado" para el archivo descifrado  
            if nombre_archivo_cifrado.endswith(".cifrado"):  
                nombre_archivo_descifrado = nombre_archivo_cifrado[:-len(".cifrado")]  
            else:  
                nombre_archivo_descifrado = nombre_archivo_cifrado + ".descifrado"  
  
            with open(nombre_archivo_descifrado, "wb") as archivo_descifrado:  
                archivo_descifrado.write(datos_descifrados)  
  
            print(f"\nArchivo descifrado y guardado exitosamente")  
    except FileNotFoundError:  
        print(f"Error: El archivo '{nombre_archivo_cifrado}' no fue encontrado.")  
    except Exception as e:  
        print(f"Ocurrió un error al descifrar el archivo. Asegúrate de que la clave sea correcta y el archivo no esté corrupto: {e}")
```

Nota: La clave generada, los archivos cifrados y descifrados se almacenan en la ruta donde se encuentran los archivos de original, teniendo en cuenta que anteriormente digitamos la ruta donde se encontraban los archivos de origen (folder= 'ruta_de_los_archivos')

- Por ultimo se crear una función la cual ejecuta las funciones anteriores dependiendo de la opción que se elija.

```
def main():
    print("\n--- Cifrador y Descifrador de Archivos ---")

    # Cargar o generar la clave
    clave = cargar_clave(drive_path+folder+"clave.key")

    while True:
        print("\nOpciones:")
        print("1. Cifrar archivo")
        print("2. Descifrar archivo")
        print("3. Salir")

        opcion = input("Seleccione una opción (1/2/3): ")

        if opcion == '1':
            nombre_archivo = listar_archivos()
            cifrar_archivo(nombre_archivo, clave)
        elif opcion == '2':
            print("\n-----Seleccione el archivo a Descifrar(solo el número)-----")
            nombre_archivo_cifrado = listar_archivos()

            print("\n-----Seleccione la clave para descifrar el archivos(solo el número)-----")
            elegir_clave = listar_archivos()
            clave_privada = cargar_clave(elegir_clave)
            descifrar_archivo(nombre_archivo_cifrado, clave_privada)
            break
        elif opcion == '3':
            print("Saliendo del programa.")
            break
        else:
            print("Opción no válida. Por favor, intente de nuevo.")

if __name__ == "__main__":
    main()
```

Menú de la función main()

```
Mounted at /content/drive

--- Cifrador y Descifrador de Archivos ---
El archivo 'clave.key' no existe. Generando una nueva clave...
Clave generada y guardada

Opciones:
1. Cifrar archivo
2. Descifrar archivo
3. Salir
Seleccione una opción (1/2/3): 1
```

```
Lista de Archivos:
0: documento.pdf
1: clave.key
Selecciona el número del archivo: 0

Archivo cifrado y guardado exitosamente
```