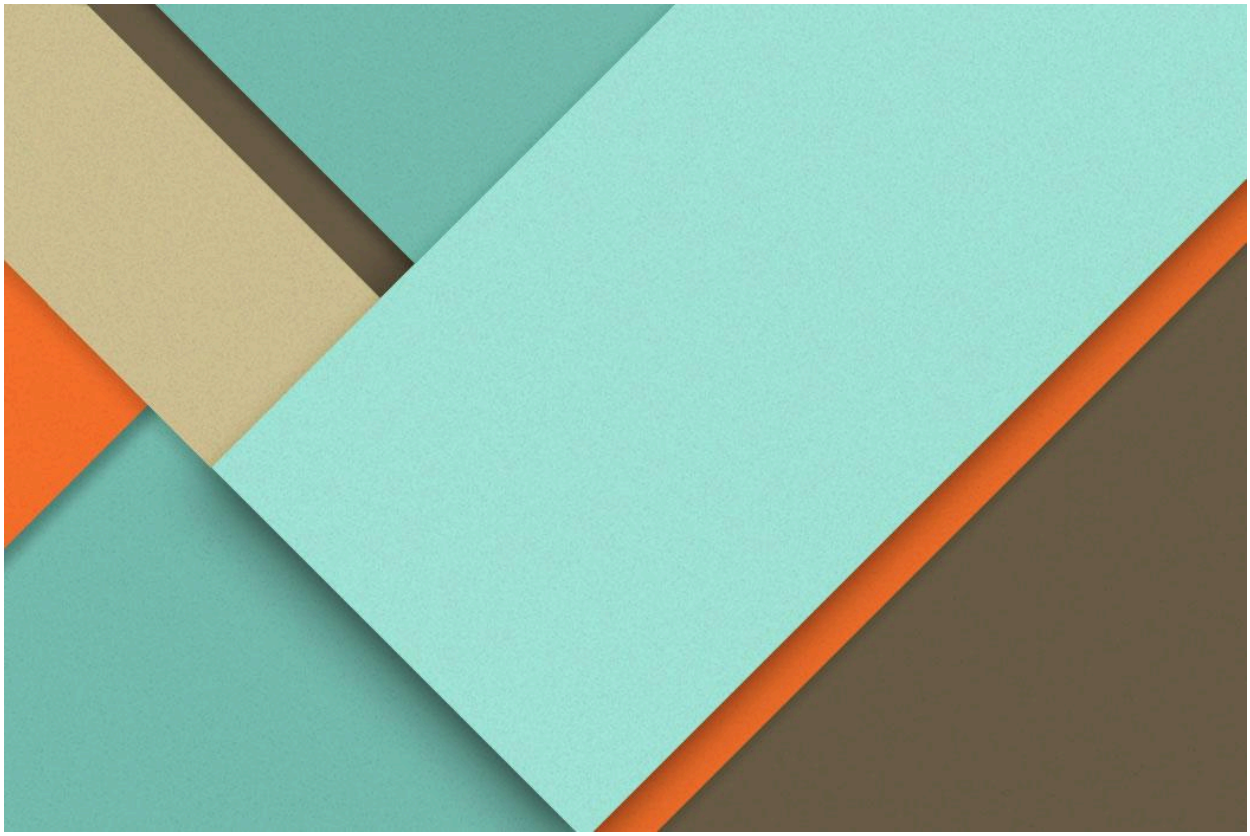


Reporte semanal - Semana 7

15/03/2024

Ismael Montoro Peñasco
Fundación GoodJob

Acciones llevadas a cabo

Seguiré un poco más con los encriptados, para ir en serio con Scapy (Para realizar OSINT a servidores públicos), y voy a ir poco a poco con Excel (Para ver qué es lo que quieren las empresas concretamente de aquellos que manejes Excel). Pero lo importante es finalizar con los encriptados para poder empezar con el curso de Inteligencia Artificial de la universidad de Harvard.

Avances realizados

Labor 1: Encriptaciones

He realizado encriptaciones también está semana

Labor 1.1: Cifrado César con Cambio

El cifrado César con Cambio es como una reimplementación del cifrado César tradicional, en este cifrado se utiliza una rueda de cambio para actualizar la tabla de caracteres para el cifrado, haciendo que roten los caracteres como si se tratase de una rueda.

```
322     def cifrado_cesar(self, texto: str, con_cambio: bool = False, clave_de_cambio: int = 7):
323         # https://es.wikipedia.org/wiki/Cifrado_C%C3%A9sar
324         texto_plano = texto.upper()
325         texto_cifrado = ""
326 
327         if con_cambio: # Aqui tenemos el cifrado cesar
328             # https://www.youtube.com/watch?v=kciDyNZblsU
329 
330             abecedario_original = "ABCDEFGHIJKLMNÑOPQRSTUVWXYZ"
331             abecedario_movil = "ABCDEFGHIJKLMNÑOPQRSTUVWXYZ"
332 
333             for _ in range(clave_de_cambio):
334                 abecedario_movil = abecedario_movil[1:] + abecedario_movil[0]
335 
336             for caracter in texto_plano:
337                 busqueda = abecedario_original.index(caracter)
338                 texto_cifrado += abecedario_movil[busqueda]
339                 abecedario_movil = abecedario_movil[1:] + abecedario_movil[0]
340 
```

```
PROBLEMAS   SALIDA   CONSOLA DE DEPURACIÓN   TERMINAL   PUERTOS

Cifrado Cesar
        Cifrado de la cadena Hola - KRÑD
        Descifrado de la cadena Hola - HOLA

Cifrado Cesar con Cambio
        Cifrado de la cadena Hola - JROF
        Cifrado de la cadena Hola - ÑWTK
        Descifrado de la cadena Hola - HOLA
        Descifrado de la cadena Hola - HOLA
```

Labor 1.2: Cifrado Hill

He conseguido implementar gracias a las matemáticas, el algoritmo de cifrado hill.

```
558     def cifrado_hill_3_3(self, texto: str, matriz_clave: list):
559         # https://es.wikipedia.org/wiki/Cifrado_Hill
560         ####### Preparar el mensaje, tenemos que poner el texto en mayusculas #######
561         mensaje = texto.upper()
562 
563         ####### Crear el diccionario con el alfabeto, asignando un numero a cada letra #######
564         text_abecedario = " ABCDEFGHIJKLMNOPQRSTUVWXYZ"
565         abecedario = dict()
566         for id, caracter in enumerate(list(text_abecedario)):
567             abecedario[caracter] = id
568 
569         abecedario_claves = list(abecedario.keys())
570         abecedario_valores = list(abecedario.values())
```

Labor 1.1.1: Matemáticas

Mientras realizaba el algoritmo hill he aprendido algunos conceptos matemáticos interesantes.

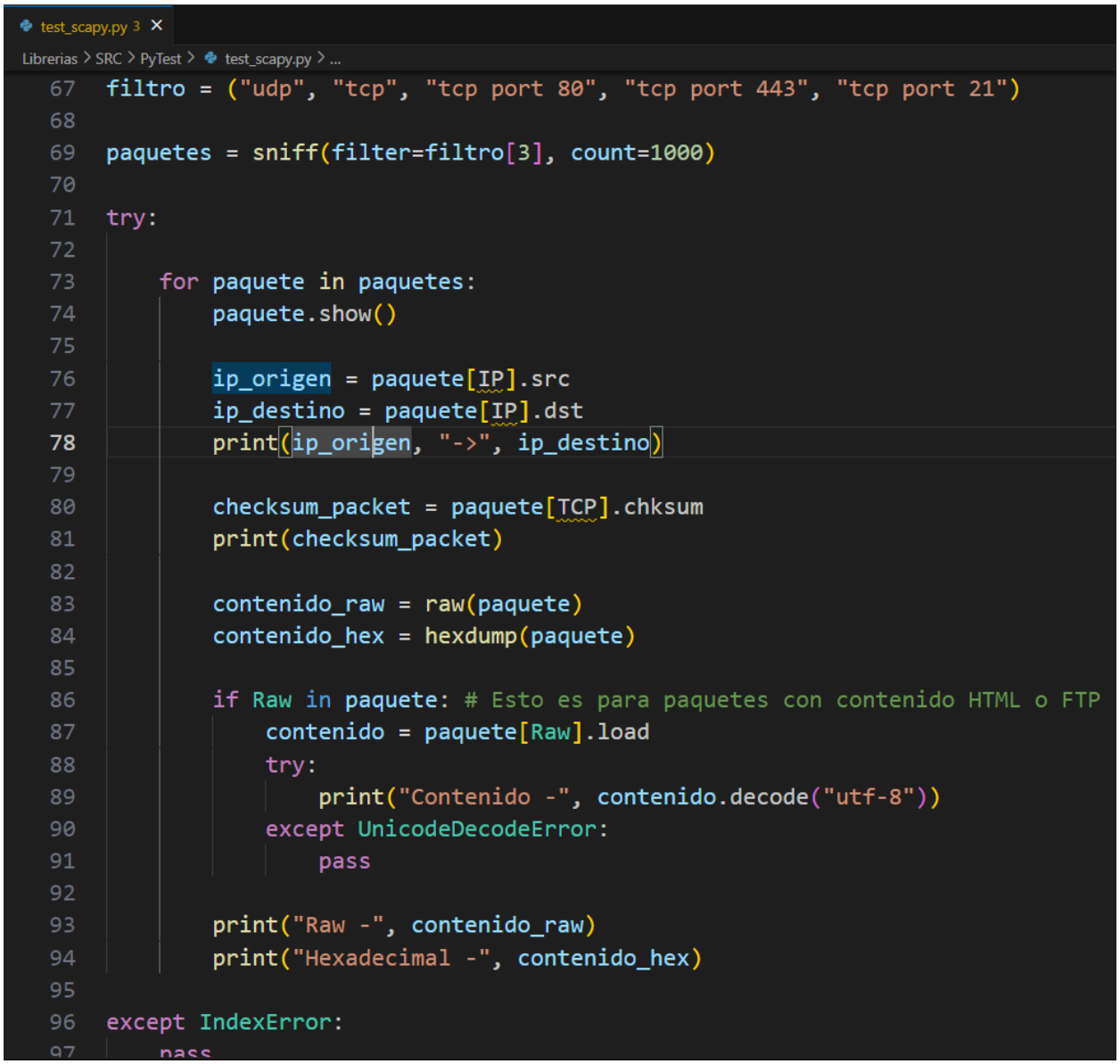
- Cálculo de Matrices.
- Hallar el determinante.
- Hallar la matriz transpuesta.
- Hallar la matriz adjunta.
- Hallar el cofactor de una matriz.

```
## Requisito 1 ##
def calcular_determinante(matriz: list):
    matriz_sarrus = matriz.copy()[0:2]
    matriz_calculo = list()
    matriz_calculo.extend(matriz)
    matriz_calculo.extend(matriz_sarrus)
    resultado_diagonales_izquierda = (matriz_calculo[0][0] * matriz_calculo[1][1] * matriz_
    resultado_diagonales_derecha = (matriz_calculo[0][2] * matriz_calculo[1][1] * matriz_ca
    resultado = resultado_diagonales_izquierda - (resultado_diagonales_derecha)
    return resultado

determinante = calcular_determinante(matriz_clave)
determinante_mod = determinante % modulo
```

```
707    def matriz_adjunta(matriz: list):
708        result_posicion_1_1 = (matriz[1][1] * matriz[2][2]) - (matriz[2][1] * matriz[1][2])
709        result_posicion_1_2 = -1 * ((matriz[1][0] * matriz[2][2]) - (matriz[2][0] * matriz[1][2]))
710        result_posicion_1_3 = (matriz[1][0] * matriz[2][1]) - (matriz[2][0] * matriz[1][1])
711        result_posicion_2_1 = -1 * ((matriz[0][1] * matriz[2][2]) - (matriz[2][1] * matriz[0][2]))
712        result_posicion_2_2 = (matriz[0][0] * matriz[2][2]) - (matriz[2][0] * matriz[0][2])
713        result_posicion_2_3 = -1 * ((matriz[0][0] * matriz[2][1]) - (matriz[2][0] * matriz[0][1]))
714        result_posicion_3_1 = (matriz[0][1] * matriz[1][2]) - (matriz[1][1] * matriz[0][2])
715        result_posicion_3_2 = -1 * ((matriz[0][0] * matriz[1][2]) - (matriz[1][0] * matriz[0][2]))
716        result_posicion_3_3 = (matriz[0][0] * matriz[1][1]) - (matriz[1][0] * matriz[0][1])
717
718        resultado = [[result_posicion_1_1, result_posicion_2_1, result_posicion_3_1],
719                     [result_posicion_1_2, result_posicion_2_2, result_posicion_3_2],
720                     [result_posicion_1_3, result_posicion_2_3, result_posicion_3_3]]
721        return resultado
```

Labor 2: Scapy

He realizado una extracción básica de la información de los paquetes capturados.



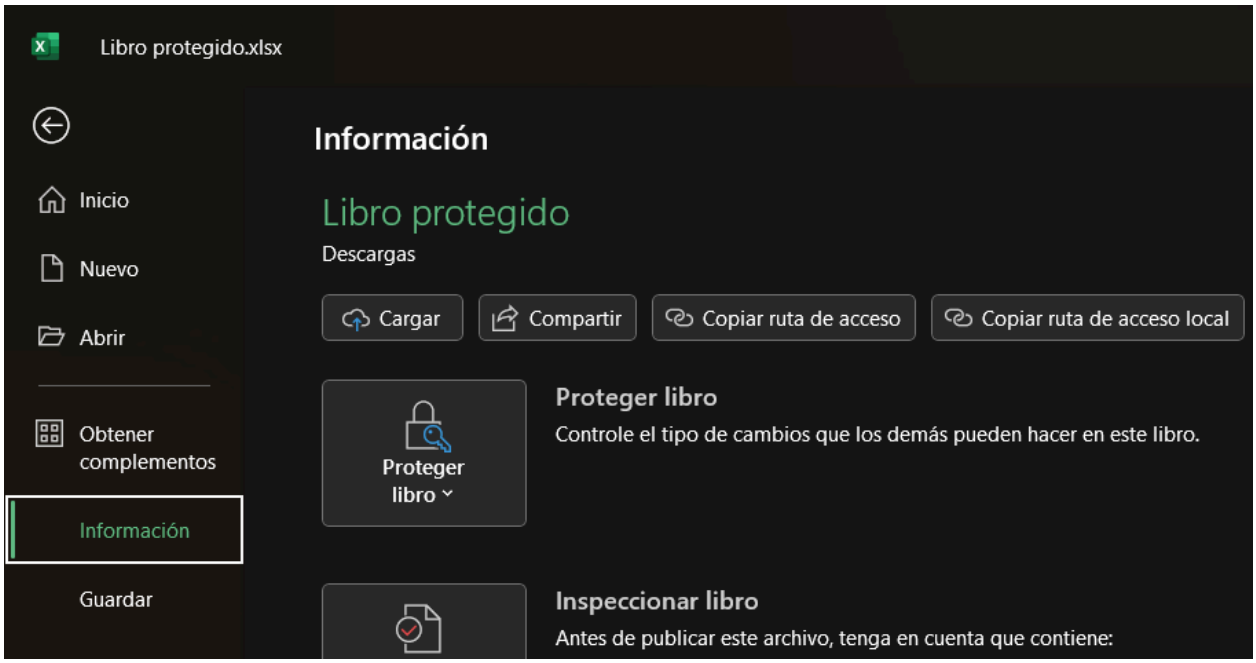
He conseguido modificar los paquetes de la máquina que me envia la maquina de destino

```
142 ip_destino = ("80.58.61.250", "8.8.8.8")
143 filtro = ("udp", "tcp", "icmp", "tcp port 80", "tcp port 443", "tcp port 21","host 192.168.1.34")
144 paquete = IP(src="192.168.1.34", dst=ip_destino[1])/ICMP()
145
146 for i in range(1, 4):
147     send(paquete)
148     paquete = sniff(filter=filtro[6], count=1)
149     paquete[IP].dst = ip_destino[i % 2]
150     paquete.show()
```

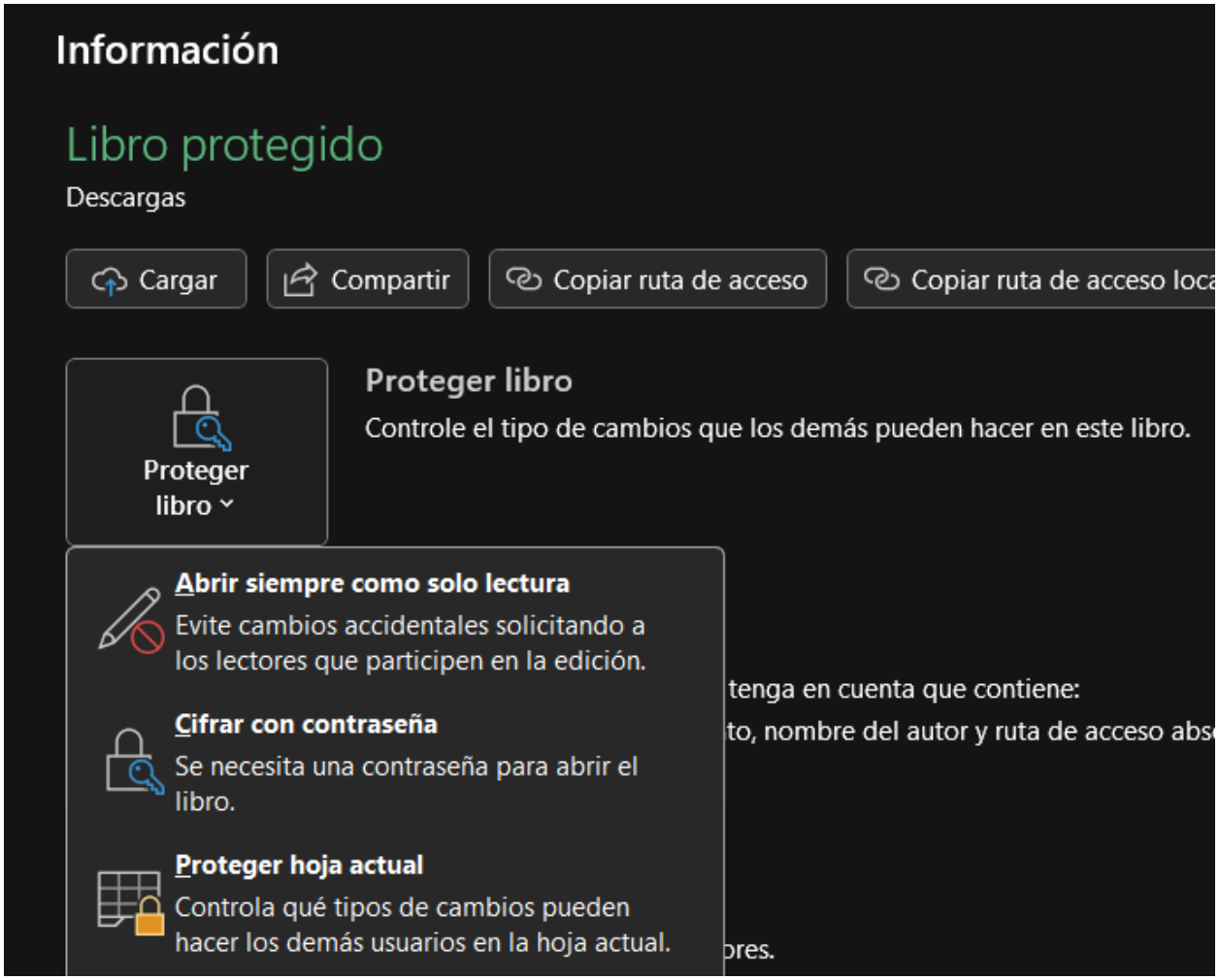
```
PROBLEMAS 3  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

PS C:\Users\ismae\Downloads\EstudiosPython> & c:/Users/ismae/Downloads/EstudiosPython/Librerias/Scripts/python.exe c:/Users/ismae/Downloads/EstudiosPython/Li
.
Sent 1 packets.
0000 Ether / IP / ICMP 8.8.8.8 > 192.168.1.34 echo-reply 0 / Padding
.
Sent 1 packets.
0000 Ether / 192.168.1.34 > 239.255.255.250 2 / Raw
.
Sent 1 packets.
0000 Ether / IP / UDP / DNS Qry "b'mobile.events.data.microsoft.com.'"
PS C:\Users\ismae\Downloads\EstudiosPython>
```

Labor 3: Excel

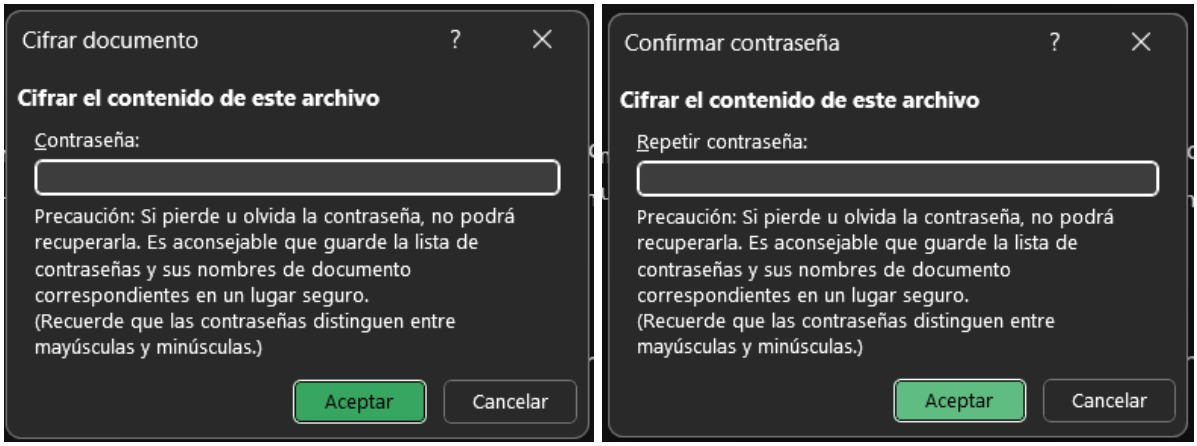
He aplicado protección a un archivo de Excel, para ello seleccionaremos la pestaña Archivo y la sección de Información.



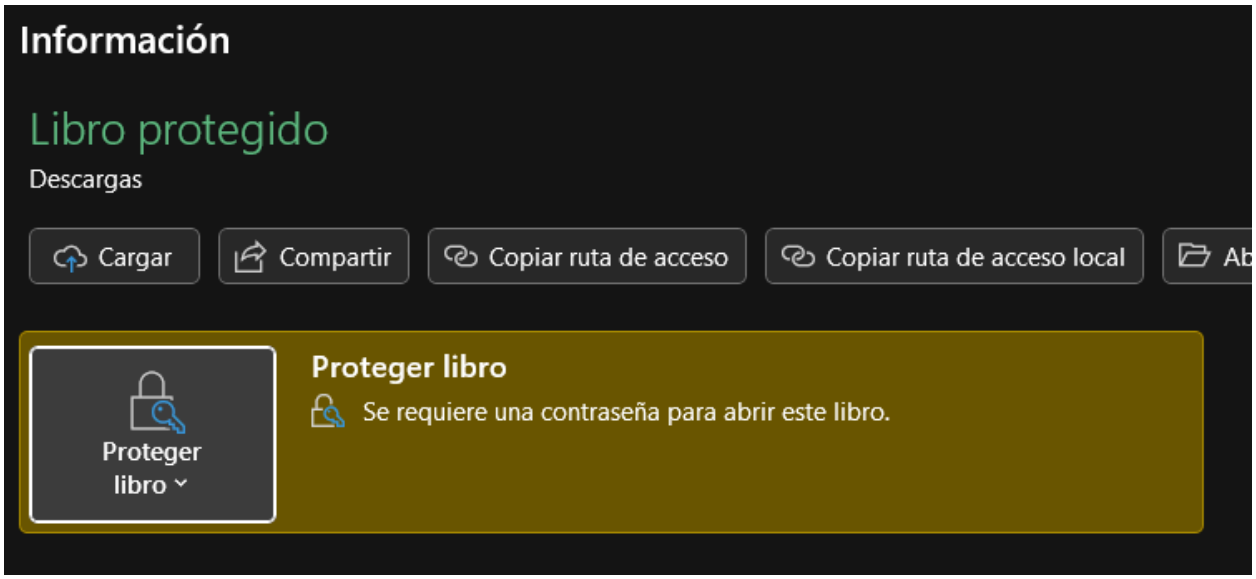
Pulsamos la opción de “Proteger libro” y dentro del menú pulsamos la opción de “Cifrar con contraseña”.



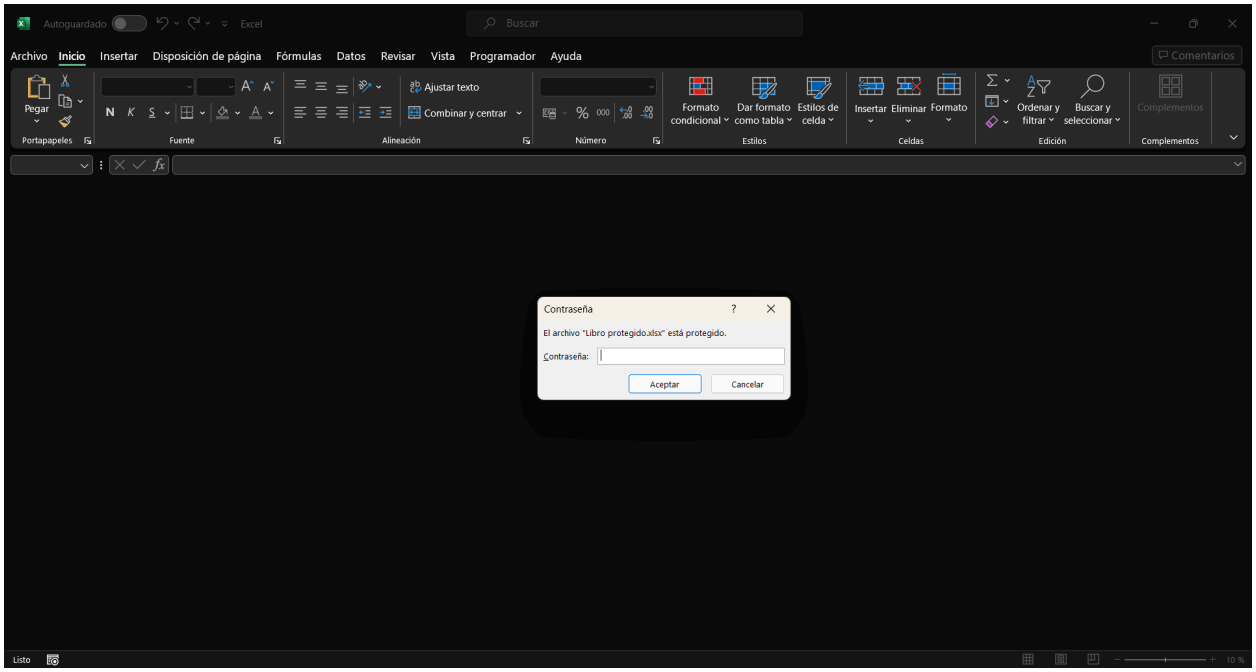
Aquí escribiremos la contraseña y confirmamos dicha contraseña.



Si vemos esto en la ventana



Si cierras y abres la hoja de Excel, se nos aparecerá la ventana sin el contenido de la hoja de forma visible y con una ventana con la contraseña.

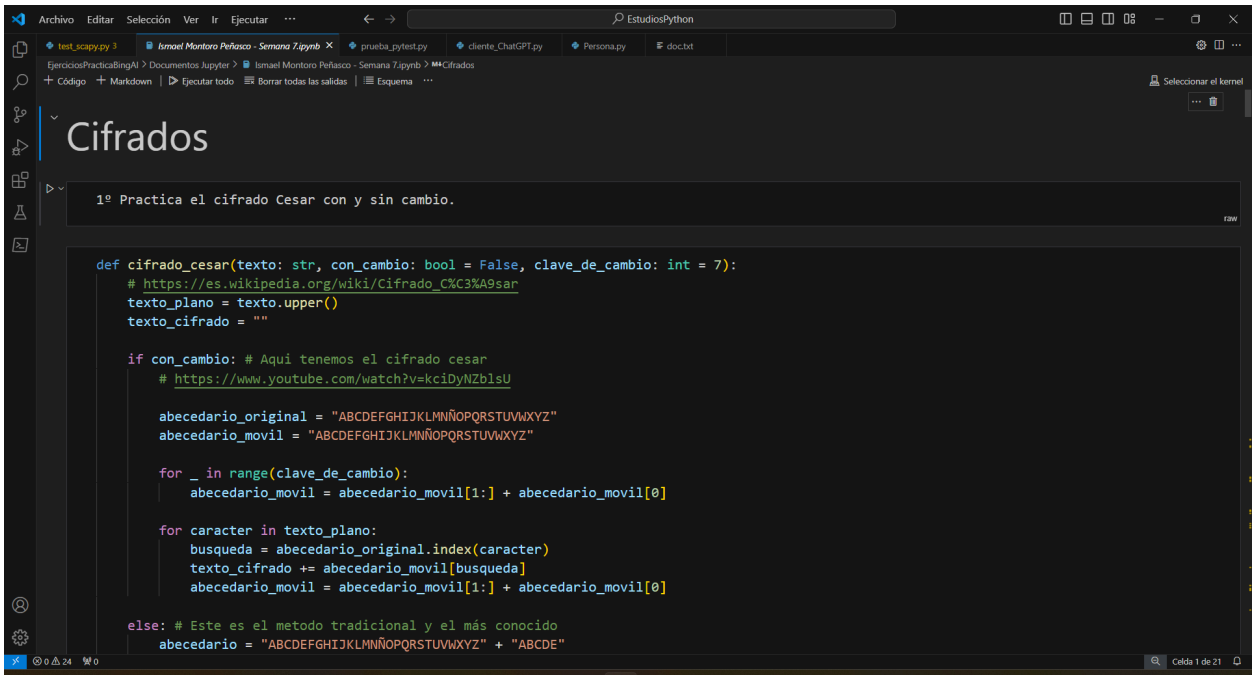


Si escribes la contraseña, desbloquearas el contenido del libro.

	A	B	C	D	E	F
1						
2						
3					hjg	jgjh
4					hjg	jgjh
5					jghj	jgjh
6					jghj	jgjh
7					jgj	jgjh
8					jgj	jgjh
9					jgj	jgjh
10						
11						

Labor 4: Los ejercicios de la semana

He realizado los diez ejercicios de la semana, acabando con el temario de los cifrados por el momento, y entrando de lleno en scapy.



Labor 5: Bienvenida al Curso de Inteligencia Artificial

He podido introducirme al curso de Inteligencia artificial, trataremos bastantes temas de ahora en adelante. Por lo que daré prioridad al curso para ir adelantando el temario.

Labor 5.1: Descargas

He realizado la descarga del material de la introducción y la lectura 0 (Tema 1).

Planes para la próxima semana

Seguiré estudiando Scapy e Inteligencia Artificial, dando prioridad a la Inteligencia Artificial.