



# PROGRAMACION I

“PASSWORD LAB”

- Cristian Arroyo Garibay
- Juan Antonio Cano Mercado
- Jose Manuel Gomez Saldaña
- Paul Johan Neufeld Coria

# ¿QUE ES?

"Este es un programa en Python que actúa como un filtro de seguridad para contraseñas. Su trabajo consiste en:

1. **Revisar una lista de claves:** Separa las seguras (largas) de las inseguras (cortas).
2. **Proteger las buenas:** Las contraseñas seguras se encriptan usando tecnología SHA-256 para que sean **ilegibles**.
3. **Corregir las malas:** A las contraseñas débiles no las guarda, sino que te sugiere cómo mejorarlas.
4. **Buscador inteligente:** Puedes escribir una contraseña y el programa te dirá si existe en la lista, en qué archivo está guardada y cuánto tiempo exacto le tomó encontrarla."

# **Funciones**

**y**

## **Proposito principal**



HASHEAR  
()

LEE CONTRASEÑAS DE ARCHIVOS\_LIMPIOS.TXT, LAS CLASIFICA POR LONGITUD [8 CARACTERES]: LAS CONTRASEÑAS SEGURAS [>= 8] SE HASHEAN (SHA-256) Y SE ESCRIBEN EN ARCHIVOS\_ENcriptados.TXT, Y LAS DÉBILES (< 8) SE ESCRIBEN SIN HASHEAR EN SIN\_SEGURIDAD.TXT6

VER\_USUARIO\_Y\_SU\_CO  
NTRASEÑA\_YA\_ENcript  
ADA

LEE CONTRASEÑAS DE ARCHIVOS\_LIMPIOS.TXT , LAS HASHEA CON SHA-256 , Y LAS IMPRIME EN LA TERMINAL JUNTO CON UN NÚMERO DE USUARIO CORRELATIVO

VER\_CANTIDAD\_DE  
\_CONTRASEÑAS\_EN  
\_CRIPTADAS ()

MUESTRA EN LA TERMINAL EL NÚMERO TOTAL DE CONTRASEÑAS QUE FUERON HASHEADAS Y GUARDADAS EN ARCHIVOS\_ENcriptados.TXT

VER\_CANTIDAD\_DE\_CONTRA  
SEÑAS\_QUE\_NÓ\_FUERON  
\_ENcriptadas ()

MUESTRA EN LA TERMINAL EL NÚMERO TOTAL DE CONTRASEÑAS QUE SE CONSIDERARON DÉBILES Y SE GUARDARON EN SIN\_SEGURIDAD.TXT

RECOMENDACIONES\_PA  
RA CONTRASEÑAS\_NO  
\_ENcriptadas ()

LEE LAS CONTRASEÑAS DEL ARCHIVO SIN\_SEGURIDAD.TXT E IMPRIME EN LA TERMINAL RECOMENDACIONES ESPECÍFICAS PARA CADA UNA SI NO CUMPLE CON LA LONGITUD MÍNIMA, NO INCLUYE NÚMEROS, O NO INCLUYE CARACTERES ESPECIALES

COMPARAR  
()

PIDE AL USUARIO QUE INGRESE UNA CONTRASEÑA Y LA BUSCA: PRIMERO SIN HASHEAR EN SIN\_SEGURIDAD.TXT , EN CASO DE NO ESTAR LA HASHEA PARA PODER BUSCARLA EN ARCHIVOS\_ENcriptados.TXT. INFORMA EN QUE ARCHIVO LA ENCONTRO Y EL TIEMPO QUE TARDÓ EN LA BÚSQUEDA PARA POSTERIORMENTE GUARDARLAS COMO UN HISTORIAL EN BUSCADOS.TXT INCLUSO SI NO LA ENCUENTR

```
1 import hashlib
2 import time
3
4 # con usuarios totales vamos a ir aumento el numero de usuario que se encuentra en el diccionario de persona
5 usuarios_totales = 0
6
7 # cada contraseña que sea mandada a sin_seguridad.txt aumentara este contador
8 cantidad_sin_seguridad = 0
9
10 #cada contraseña que se enviada a archivos_encriptados.txt aumentara este contador
11 cantidad_encriptadas = 0
12
13
14 # creamos una funcion en donde se mostrara en terminal el numero de usuario y su contraseña encriptada
15 def hashear():
16
17 #con el with open vamos a abrir 3 carpetas, archivos limpios.txt en donde "r" es para que lea la carpeta y
18 #archivos encriptados.txt con "w" para que escriba las contraseñas ya encriptadas que si tiene una longitud mayor a 8
19 #y sin seguridad.txt con "w" para que escriba las contraseñas con longitud menor a 8
20
21 with open("archivos_limpios.txt", "r")as archivos, open("archivos_encriptados.txt", "w") as encriptados, open("sin_seguridad.txt", "w")as sin_seguridad:
22     for linea in archivos:
23         global cantidad_encriptadas # llamamos a cantidad_de_encriptadas
24
25         global cantidad_sin_seguridad # llamamos a cantidad_sin_encriptar
26
27         longitud = 8 #definimos nuestra longitud para saber si las contraseñas se iran a sin_seguridad.txt
28             #o a archivos_encriptados.txt
29
30         nueva_linea = linea.strip() #evitamos saltos de linea
31         encriptar =hashlib.sha256(nueva_linea.encode()).hexdigest() # encriptamos las contraseñas
32
33         if len(nueva_linea) < longitud: # si la cantidad de caracteres es menor que longitud la envia a sin_seguridad.txt
34             sin_seguridad.write(nueva_linea + "\n")
35             cantidad_sin_seguridad +=1 # el responsable de aumentar el contador cantidad_sin_seguridad
36         else:
37             encriptados.write(encriptar+"\n")
```

```
1 #funcion para ver el numero de usuarios y sus contraseñas ya hasheadas
2 def ver_usuario_y_su_contraseña_ya_encriptada():
3
4     #abrimos el archivo_limpios.txt con r para leerlo y le accipnamos archivo como valor
5     with open("archivos_limpios.txt", "r")as archivos:
6
7         for linea in archivos: # agregamos un for para apoyarno en saber que queremos que la impresion sea linea por linea
8             global usuarios_totales # llamamos a usuarios_totales
9             nueva_linea = linea.strip() #evitamos saltos
10            encriptar =hashlib.sha256(nueva_linea.encode()).hexdigest() #hasheamos
11
12            persona = { #creamos nuestro diccionario
13                "Usuario": usuarios_totales,
14                "Contraseña": encriptar,
15            }
16
17            print(persona)
18            print("*"*30)
19            usuarios_totales += 1 #aumentamos el numero de usuarios
20
21
22
23 #funcion para ver la cantidad de contraseñas encriptadas
24 def ver_cantidad_de_contraseñas_encriptadas():
25     global cantidad_encriptadas # llamamos a cantidad_encriptadas que se aumentaba en la funcion hashear
26     print("*"*60)
27     print("\nLa cantidad de contraseñas encriptadas es de: ",cantidad_encriptadas,"\n")
28
29     print("*"*60)
30 #funcion para ver la cantidad de contraseñas sin encriptar
31 def ver_cantidad_de_contraseñas_que_no_fueron_encriptadas():
32     global cantidad_sin_seguridad # llamamos a cantidad_sin_seguridad que se aumentaba en la funcion hashear
33     print("*"*60)
34     print("\nLa cantidad de contras
```

```
1 #funcion para recomendar, para las contraseñas enviadas al archivo sin seguridad
2 def recomendaciones_para_contraseñas_no_encriptadas():
3     #abrimos el archivo sin_seguridad.txt con r para leer y se asipnamos sin_seguridad como valor
4     with open("sin_seguridad.txt","r")as sin_seguridad:
5         for recomendaciones in sin_seguridad: #dado que la funcion es para recomendaciones cambiamos el valor de sin_seguridad
6             # por recomendaciones para tener un mejor apoyo visual
7             longitud = 8
8             numeros = "0","1","2","3","4","5","6","7","8","9"
9             caracteres_especiales = "@","!","#","$","%","&","/",(",,"),"="
10            recomendacion = recomendaciones.strip() #evitamos saltos
11
12
13    if len(recomendacion)< longitud: #verificamos si el numero de caracteres es menor que longitud
14        print( recomendaciones,"- Tuvo una longitud < 8\n" # si los caracteres de la contraseña son menores a longitud imprime esto
15        " - Se recomienda alargarla hasta superar los 8 caracteres")
16    if not (any(c in numeros for c in recomendaciones)): # si la contraseña no tiene numero imprimira esto
17        print(recomendaciones,"- No cuenta con ningun numero\n",
18        "- Se recomienda el uso de numeros para una mayor seguridad")
19    if not(any(c in caracteres_especiales for c in recomendaciones)):# si la contraseña no tiene caracteres especiales imprime esto
20        print(recomendaciones,"- No cuenta con caracteres especiales\n",
21        "- Se recomienda agregarlo para mayor seguridad")
22    print(_*30)#limita cuando acaban las recomendaciones para cada contraseña y sea facil de leer
23
24 #funcion para comparar en base a un input
25 def comparar():
26     comparacion = input("-Ingrese la contraseña a comparar: ")
27     nueva_comparacion =comparacion.strip() #quitamos espacios
28     encriptar =hashlib.sha256(nueva_comparacion.encode()).hexdigest() #hasheamos
29
30     #abrimos los archivos sin_seguridad.txt y archivos_encriptados.txt, ambos en modo lectura
31     with open("sin_seguridad.txt","r",encoding="utf-8")as sin_seguridad, \
32         open("archivos_encriptados.txt","r",encoding="utf-8")as encriptados,\n
33         open("busqueda.txt","a",encoding="utf-8")as buscado:
34
35
36     lineas_sin_seguridad =[linea.strip()for linea in sin_seguridad] #guardamos en una lista cada contraseña de sin-seguridad.txt
37     lineas_encriptadas = [linea.strip()for linea in encriptados] #guardamos en una lista cada contraseña de archivos_encriptados.txt
38
39     contraseñas =[lineas_encriptadas,lineas_sin_seguridad]
40     objetivo = nueva_comparacion
41     tiempo_inicio = time.time()
42     for tiempo in contraseñas:
43         if tiempo == objetivo:
44             break
45
46     tiempo_fin = time.time()
47     tiempo_encontrar_contraseña = tiempo_fin - tiempo_inicio
48
49
50
```

```
1 #donde empieza todo, de aqui se llamara a cada funcion dependiendo el numero ingresado
2 accion = input("¿Que quieres hacer?:\n"
3 "[1] Hashear [1] \n"
4 "[2] Ver usuario y su contraseña ya encriptada [2] \n"
5 "[3] Ver cantidad de contraseñas encriptadas [3] \n"
6 "[4] Ver cantidad de contraseñas que no fueron encriptadas [4] \n"
7 "[5] Recomendaciones para contraseñas no encriptadas [5] \n"
8 "[6] Comparar [6]\n"
9
10    "-- Escriba --salir-- para terminar el programa. --\n" \
11    "Elija su opcion: "
12
13 while accion != "salir":#mientras lo ingresado en el input es diferente de salir el programa continua
14     if accion == "1":
15         hashear()
16     elif accion == "2":
17         ver_usuario_y_su_contraseña_ya_encriptada()
18     elif accion == "3":
19         ver_cantidad_de_contraseñas_encriptadas()
20     elif accion == "4":
21         ver_cantidad_de_contraseñas_que_no_fueron_encriptadas()
22     elif accion == "5":
23         recomendaciones_para_contraseñas_no_encriptadas()
24     elif accion == "6":
25         comparar()
26     else:
27         print("Acción no válida")
28
29 accion = input("¿Qué más deseas hacer?:\n"      #una vez termine lo que hace cada función se repite un ciclo
30 "[1] Hashear [1] \n"
31 "[2] Ver usuario y su contraseña ya encriptada [2] \n"
32 "[3] Ver cantidad de contraseñas encriptadas [3] \n"
33 "[4] Ver cantidad de contraseñas que no fueron encriptadas [4] \n"
34 "[5] Recomendaciones para contraseñas no encriptadas [5] \n"
35 "[6] Comparar [6]\n"
36
37    "-- Escriba --salir-- para terminar el programa. --\n" \
38    "Elija su opción: "
39
40 print("-- Fue un honor servirle, despidase de su system32 <:)"")# en caso de escribir salir el accion finaliza
```



**iMUCHAS GRACIAS!**