

PROYECTO 0769

José Miguel Choc Samayoa, 202300795

Escuela de Mecánica Eléctrica, Facultad de Ingeniería, Universidad de San Carlos de Guatemala

Pues bueno tenemos un programa tipo Python en el cual se utilizaron librerías para tipos de letras, algunos protocolos de seguridad para envíos de correo, también para encriptación de contraseñas, escritura de texto en archivos .txt, entre otros

1. OBJETIVOS

- Aplicar los conocimientos de programación para construir un sistema funcional.
- Crear un modelo de base de datos empleando documentos de texto con los que realizar las operaciones de crear, leer, actualizar y eliminar datos de él.
- Utilizar distintas librerías de Python.
- Integrar un sistema de control de versiones en un proyecto de programación

2. MARCO TEÓRICO

En este proyecto para la Academia usac se utilizaron varias librerías en las primeras que podríamos mencionar serían las de tkinter las cuales son para interfaz gráfico para el usuario en el cual el usuario puede ingresar a distintas ventanas, manipular archivos a través de las mismas y también ejecutar las demás funciones que tiene dicho programa.

También se sí cortaron las librerías Font para manipular textos, la librería messagebox la cual sirve para mostrar mensajes de información, de alerta o de error al usuario. Tenemos algunas otras como smtplib y ssl las cuales nos ayudan a los protocolos de seguridad en el envío de correos electrónicos, y también tenemos otras para widget con barra de desplazamiento, también para números random en el caso de las notas de los estudiantes, y por último pon las librerías las cuales ayuda a la creación del recertificado PDF.

3. MARCO PRÁCTICO

En primer punto tenemos la generación de la ventana principal en la cual tenemos las diferentes opciones para iniciar sesión tales como iniciar sesión como estudiante cómo catedrático o cómo administrador también está la opción de registrarse esta es únicamente para los estudiantes podemos ver que esta es la función Font para el tamaño de letra y también la asignación de una imagen para una mejor presentación.



Fig.1 Ventana principal donde están las opciones para el ingreso sesión.

Por otra parte tenemos el "hashlib" en ".sha256" el cual realiza la encriptación de contraseñas y luego las escribe siempre y cuando ambas contraseñas ingresadas sean igual, genera un archivo .txt en el cual serán interesados todos los datos qué al estudiante se le piden.

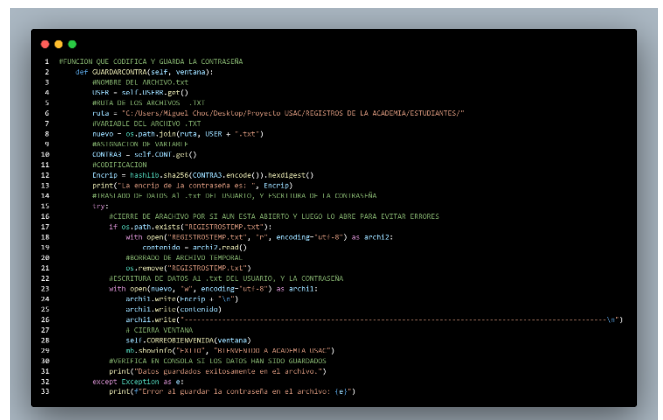



Fig.2 Encriptación de contraseña, comprobación de que las contraseñas sean iguales y la escritura de todos los datos junto con las contraseñas.

El otro segmento importante de código que podemos ver sería en el envío de correos ya sea al estudiante o al administrador los cuales cuentan con una biblioteca que es únicamente para protocolos de seguridad para el envío exitoso y seguro de dicha información por este medio, el correo por lo general llevará al estudiante su contraseña por si llega a olvidarla, avisos al

administrador de intentos de ingreso a otras cuentas y también la notificación de la asignación y des asignación de cursos del estudiante.



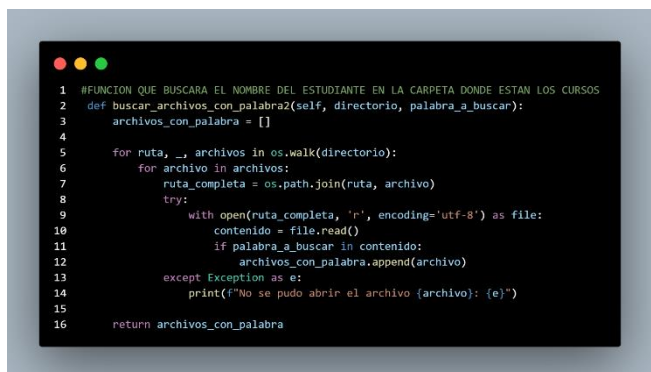
```

1 def CORREOBENVENIDA(self, ventana):
2     #DIRECCION DE CORREO ELECTRONICO
3     CORREO = self.EMAIL.get()
4     #INFORMACION PARA EL CORREO ELECTRONICO
5     usuario = self.USER.get()
6     #CORREO ELECTRONICO DEL EMISOR
7     email_emisor = 'usacggg@gmail.com'
8     email_contra = 'yehp mcneg yass ibsp'
9     #CORREO ELECTRONICO DEL RECEPTOR
10    email_receptor2 = CORREO
11    #LISTA DE DESTINATARIOS PARA EL CORREO
12    destinatarios = [email_receptor2]
13    #ASUNTO Y CUERPO DEL CORREO ELECTRONICO
14    asunto = 'INFORMACION ACADEMIA USAC'
15    cuerpo = ""BIENVENIDO A LA ACADEMIA USAC QUERIDO "" + (usuario)
16    #ASIGNACIONES DE LA LOCALIDAD EN EL CORREO ELECTRONICO PARA EL ENVIO
17    em=EmailMessage()
18    em['from'] = email_emisor
19    em['to'] = ", ".join(destinatarios)
20    em['Subject'] = asunto
21    em.set_content(cuerpo)
22    #SEGURIDAD PARA EL ENVIO DE CORREO
23    contexto = ssl.create_default_context()
24    with smtplib.SMTP_SSL('smtp.gmail.com', 465, context=contexto) as ADVERTENCIA:
25        ADVERTENCIA.login(email_emisor, email_contra)
26        ADVERTENCIA.sendmail(email_emisor, destinatarios, em.as_string())
27    #CIERRE DE VENTANA
28    ventana.destroy()
29    #INFORMACION EN CONSOLA
30    print("CORREO ENVIADO")
31    #-----

```

Fig.3 Función donde se puede observar la construcción de un correo de bienvenida

Y por último tenemos el segmento qué tiene la función de buscar una palabra en todos los archivos qué hay en una carpeta en este caso, lo que va a realizar el segmento es buscar el nombre del estudiante en todos los archivos .txt qué hay en una carpeta, y luego le asignará a cada archivo un botón para que el estudiante pueda revisar subnotas en dado caso son estas sea mayor a 61 se generará un certificado. PDF y un correo de felicitación por haber ganado el curso.



```

1 #FUNCION QUE BUSCARA EL NOMBRE DEL ESTUDIANTE EN LA CARPETA DONDE ESTAN LOS CURSOS
2 def buscar_archivos_con_palabra2(self, directorio, palabra_a_buscar):
3     archivos_con_palabra = []
4
5     for ruta, _, archivos in os.walk(directorio):
6         for archivo in archivos:
7             ruta_completa = os.path.join(ruta, archivo)
8             try:
9                 with open(ruta_completa, 'r', encoding='utf-8') as file:
10                    contenido = file.read()
11                    if palabra_a_buscar in contenido:
12                        archivos_con_palabra.append(archivo)
13            except Exception as e:
14                print(f"No se pudo abrir el archivo {archivo}: {e}")
15
16    return archivos_con_palabra

```

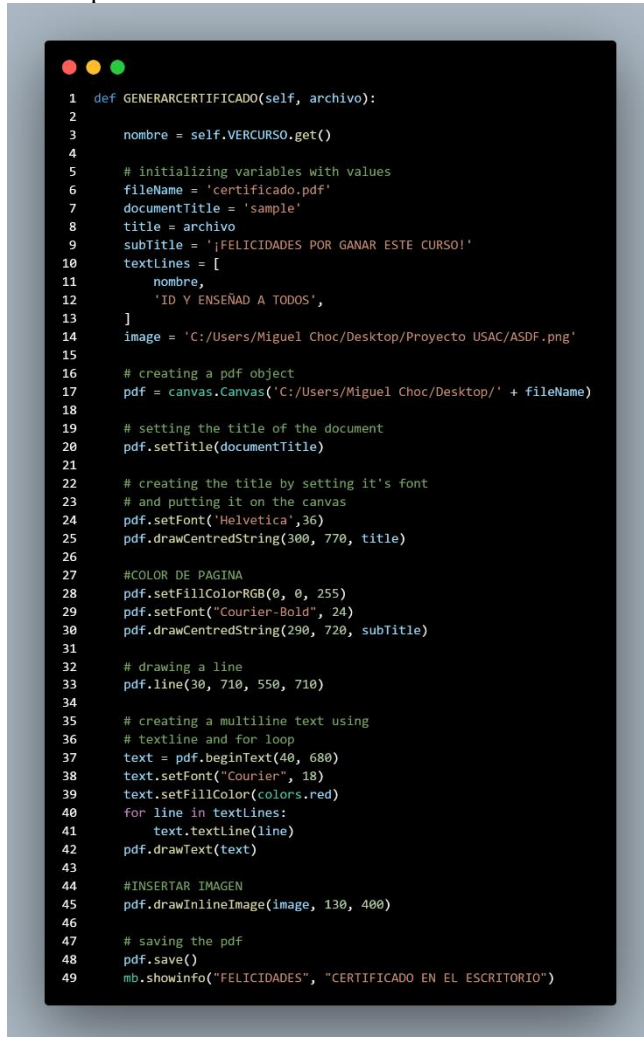
Fig.4 Función que busca el nombre del estudiante en los cursos para mostrar en cuales esta asignado.

4. CÓDIGO

Aquí tenemos el código que genera un certificado. PDF en el cual podemos ver la asignación de variables tales como: el nombre del curso el nombre del estudiante y un mensaje, lema De la Academia usac.

Podemos ver que en el código se tienen las distintas opciones como color de letra una imagen insertada en el PDF yo también un último mensaje de información en el cual dice felicidades

certificado en el escritorio esto se refiere aquí el certificado. PDF lo podrá encontrar el estudiante en su escritorio.



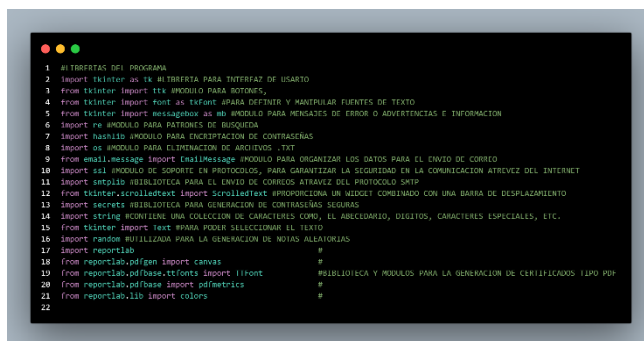
```

1 def GENERARCERTIFICADO(self, archivo):
2
3     nombre = self.VERCURSO.get()
4
5     # initializing variables with values
6     fileName = 'certificado.pdf'
7     documentTitle = 'sample'
8     title = archivo
9     subTitle = '¡FELICIDADES POR GANAR ESTE CURSO!'
10    textlines = [
11        nombre,
12        'ID Y ENSEÑAD A TODOS',
13    ]
14    image = 'C:/Users/Miguel Choc/Desktop/Proyecto USAC/ASDF.png'
15
16    # creating a pdf object
17    pdf = canvas.Canvas('C:/Users/Miguel Choc/Desktop/' + fileName)
18
19    # setting the title of the document
20    pdf.setTitle(documentTitle)
21
22    # creating the title by setting it's font
23    # and putting it on the canvas
24    pdf.setFont('Helvetica', 36)
25    pdf.drawCentredString(300, 770, title)
26
27    #COLOR DE PAGINA
28    pdf.setFillRGB(0, 0, 255)
29    pdf.setFont("Courier-Bold", 24)
30    pdf.drawCentredString(290, 720, subTitle)
31
32    # drawing a line
33    pdf.line(30, 710, 550, 710)
34
35    # creating a multiline text using
36    # textline and for loop
37    text = pdf.beginText(40, 680)
38    text.setFont("Courier", 18)
39    text.setFillRGB(colors.red)
40    for line in textlines:
41        text.textLine(line)
42    pdf.drawText(text)
43
44    #INSERTAR IMAGEN
45    pdf.drawImage(image, 130, 400)
46
47    # saving the pdf
48    pdf.save()
49    mb.showinfo("FELICIDADES", "CERTIFICADO EN EL ESCRITORIO")

```

Fig.5 Función donde se construye un pdf con distintas asignaciones de variable, insertar imagen y mensaje de felicitaciones.

Cómo podemos observar aquí están todas las librerías, también algunas clases y protocolos de seguridad ya antes mencionados.



```

1 #LIBRERIAS DEL PROGRAMA
2 import tkinter as tk #LIBRERIA PARA INTERFAZ DE USUARIO
3 from tkinter import ttk #MODULO PARA BOTONES,
4 from tkinter import font as tkFont #PARA DEFINIR Y MANIPULAR FUENTES DE TEXTO
5 from tkinter import messagebox as mb #MODULO PARA MENSAJES DE ERROR O ADVERTENCIAS E INFORMACION
6 import re #MODULO PARA PATRONES DE BUSQUEDA
7 import hashlib #MODULO PARA ENCRIPCIÓN DE CONTRASEÑAS
8 import os #MODULO PARA LISTADOS DE ARCHIVOS - TXT
9 from email.message import EmailMessage #MODULO PARA ORGANIZAR LOS DATOS PARA EL ENVIO DE CORREO
10 import ssl #MODULO DE SOPORTE EN PROTOCOLOS, PARA GARANTIZAR LA SEGURIDAD EN LA COMUNICACION ATRAVES DEL INTERNET
11 import random #LIBRERIA PARA EL ENVIO DE CORREOS ATRAVES DEL PROTOCOLO SMTP
12 from tkinter.scrolledtext import ScrolledText #PROPORCIONA UN MIDOLET COMBINADO CON UNA BARRA DE DESPLAZAMIENTO
13 import secrets #LIBRERIA PARA GENERACION DE CONTRASEÑAS SEGURAS
14 import string #CONTIENE UNA COLECCION DE CARACTERES COMO, EL ALFABETO, DIGITOS, CARACTERES ESPECIALES, ETC.
15 from tkinter import Text #PARA PODER SELECCIONAR EL TEXTO
16 import random #UTILIZADA PARA LA GENERACION DE NOTAS ALEATORIAS
17 import reportlab
18 from reportlab.pdfgen import canvas #
19 from reportlab.pdfbase.ttfonts import TTFont #
20 from reportlab.pdfbase import pdfmetrics #
21 from reportlab.lib import colors #
22

```

Fig.6 Librerías e importaciones de las mismas para diferentes aplicaciones en el código.

Por último, también tenemos a la función pedir contraseña esta función lo que hace es asignar a una variable el nombre del usuario luego busca en un archivo .txt, el cual tiene el nombre del usuario cómo, en este archivo lo que hace es buscar el correo electrónico para luego asignárselo a una variable y con

estos datos construir un correo electrónico en el cual habrá una variable a la cual se le asigna un valor random cómo contraseña.

```

1 #FUNCION PARA PEDIR CONTRASEÑA
2 def PEDIRCONTRA(usuario, ventana):
3
4     #ASTROCCIONES DE VARIABLES INGRESADAS
5     usuario = self.IDM.get()
6     archiver = usuario
7     #PARA LA CONTRASEÑA
8     caracteres = string.ascii_letters + string.digits + string.punctuation
9     #CANTIDAD DE DIGITOS
10    longitud = 3
11    #GENERACION DE UNA CONTRASEÑA RANDOM
12    contraseña_aleatoria = ''.join(random.choice(caracteres) for _ in range(longitud))
13    #ASIGNACION DE CONTRASEÑA A UNA VARIABLE
14    contraseña = contraseña_aleatoria
15
16    #ENCRIPCIÓN DE NUEVA CONTRASEÑA
17    Encrip = hashlib.sha256(contraseña.encode()).hexdigest()
18
19    #RUTA DE LOS ARCHIVOS
20    rutauser = "C:/Users/Miguel One/Desktop/Proyecto USAC/REGISTROS DE LA ACADEMIA/ESTUDIANTES/"
21    ruta_completa = os.path.join(rutauser, archiver + ".txt")
22
23    try:
24        with open(ruta_completa, "r", encoding="utf-8") as archivo:
25            contenido = archivo.read()
26            correoemail = re.findall(r'@gmail.com', contenido)
27            CORREO = correoemail[0]
28            print(CORREO)
29        try:
30            nombre_archivo = ruta_completa
31            with open(nombre_archivo, "r") as archivo:
32                lineas = archivo.readlines()
33                lineas[0] = Encrip
34            with open(nombre_archivo, "w") as archivo:
35                lineas = [line + "\n" for line in lineas]
36                archivo.writelines(lineas)
37        except:
38            #MENSAJES PARA EL CORREO ELECTRONICO
39            usuario = self.IDM.get()
40            RECORDAR_HISTORICO_DEL_EMAIL =
41            email_emisor = "usuario@gmail.com"
42            email_contra = "your new pass 1234"
43            RECORDAR_HISTORICO_DEL_RECEPION =
44            email_receptor = CORREO
45            #LISTA DE DESTINATARIOS PARA EL CORREO
46            destinatarios = [email_receptor]
47            #ASUNTO Y CUERPO DEL CORREO ELECTRONICO
48            asunto = "¡SEGURO! ACADEMIA USAC"
49            cuerpo = (usuario) + "" "SU CONTRASEÑA A CAPTADO A: \n "" + (contraseña)
50            #MENSAJES PARA EL CORREO ELECTRONICO
51            em = smtplib.SMTP()
52            em["from"] = email_emisor
53            em["to"] = ", ".join(destinatarios)
54            em["subject"] = asunto
55            em.set_content(cuerpo)
56            #MENSAJES PARA EL CORREO ELECTRONICO
57            contexto = ssl.create_default_context()
58            with smtplib.SMTP_SSL(smtp.gethostbyname(host), context=context) as ADMINISTRACION:
59                ADMINISTRACION.login(email_emisor, email_contra)
60                ADMINISTRACION.sendmail(email_emisor, destinatarios, em.as_string())
61
62    except:
63        ventana.destroy()
64        #MENSAJES PARA EL CORREO ELECTRONICO
65        #MENSAJES PARA EL CORREO ELECTRONICO
66
67    except KeyboardInterrupt:
68        #MENSAJES PARA EL CORREO ELECTRONICO
69        #MENSAJES PARA EL CORREO ELECTRONICO
70
71    except KeyboardInterrupt:
72        #MENSAJES PARA EL CORREO ELECTRONICO
73        #MENSAJES PARA EL CORREO ELECTRONICO

```

Fig.7 Función de pedido de contraseña, la cual es enviada por correo electrónico

Repositorio en GitHub del código:

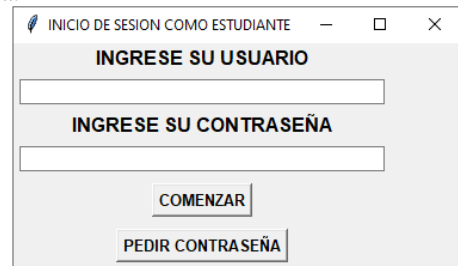
<https://github.com/JMCS373/PROYECTO-0769.git>

5. RESULTADOS

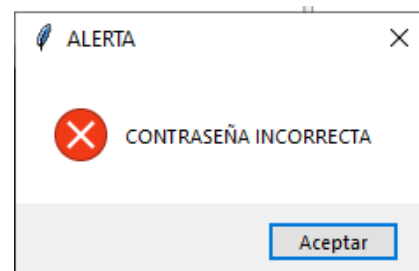
Ventana principal del estudiante, con las distintas opciones para iniciar sesión.



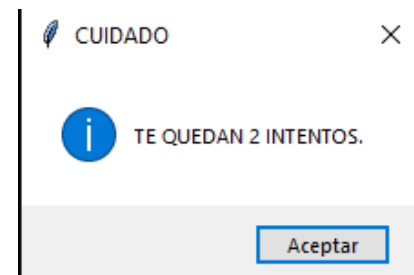
Solicitud de información para ingresar como estudiante, el botón comenzar para ingresar con credenciales correctas, o pedir contraseña donde se nos enviara un correo con una nueva contraseña.



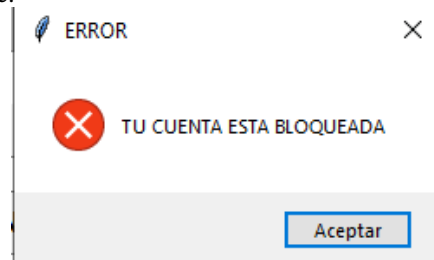
Mensaje de error por dato de contraseña erróneo.



Mensaje de información, indicando el numero de intentos que nos quedan, si se sobrepasan los intentos la cuenta se bloquea.



Mensaje de cuenta bloqueada, aunque se ingrese la contraseña correcta no se permitirá el ingreso, hasta que el administrador la desbloquee.



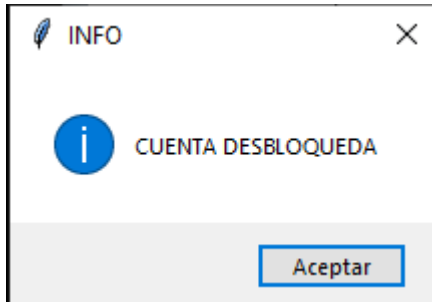
Correo Gmail de intento de ingreso y bloqueo de contraseña.

SEGURIDAD ACADEMIA USAC Recibidos x

usac66654@gmail.com
para mí ▾

SE INTENTÓ INGRESAR A LA CUENTA
ESTA CUENTA SERÁ BLOQUEADA
CUENTA DEL USUARIO:
MiguelC

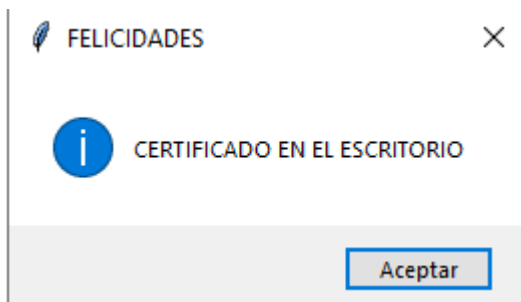
Mensaje de cuenta desbloqueada al ingresar las contraseñas de administrador.



Ventana principal del estudiante.



Mensaje de felicidades por ganar curso y la información de la locación del certificado.



Vista de la información que tiene el certificado.pdf

C++.txt

¡FELICIDADES POR GANAR ESTE CURSO!

Pincho
ID Y ENSEÑAD A TODOS



6. CONCLUSIONES

- Se comprendió el manejo de ventanas, para la interfaz del usuario.
- Se utilizó como medio de base de datos los archivos.txt, en los cuales se maneja la lectura, escritura y la incorporación de nueva información.
- Se comprendió el uso de librerías para protocolos de seguridad para el envío de correos a través de Gmail.
- Se entendió la importancia y funcionalidad de la comprobación de datos ingresados en el formato deseado.
- Se comprendió la composición y redacción de los segmentos de código donde se elabora el correo electrónico y el certificado.pdf.

7. REFERENCIAS

- Anónimo, ¿Leer Archivo txt en Python? (s. f.). Stack Overflow en español. <https://es.stackoverflow.com/questions/466161/leer-archivo-txt-en-python>
- Frank Andrade. (2022, 9 junio). Como automatizar envío de correos con Python [Vídeo]. YouTube. <https://www.youtube.com/watch?v=DDVpKvJXRz8>
- Moisset, D. (s. f.). Python ya. <https://www.tutorialesprogramacionya.com/pythonya/index.php?inicio=75>
- R CODER. (1d. C., enero 1). Colores en Matplotlib [Lista completa, conversor a RGB y selector de color] | PYTHON CHARTS. PYTHON CHARTS | Visualización de datos con Python. <https://python-charts.com/es/colores/>