ENVÍO DE CORREOS CON PYTHON

from tkinter import * import tkinter from tkinter import Label, ttk				
# Esta importacion es para manejar el filesystem. en este proyecto la u Envio Correo import os u Libretria para trabajar con archivos Json u Libretria para trabajar con archivos Json v load (cargar desde archi import json		-		×
# Esta importacion es para formar el headr. from email.mime.multipart import MIMEMultipart # Esta importacion es para formar el body del mail from email.mime.text import NINEText # Esta importacion es para establecer la comunicacion SNTP import smtplib # Solo para manejar error de conexion al vidor (dns mail ingresado N from socket import gaierror # Esta libreria nos permite trabajar con fecume # para luego grabarlo en el 35 on de enviados.ison # para luego grabarlo en el 35 on de enviados.ison Destinatario:	UNIVER	RSIDAD DE L	E	
# Lista que contendran informacion de Contactos y correos enviados, est # Cargando toda la informacion que tendran los Ison. ibreta = []				
<pre># Verificamos si el archivo libreta.json existe, si no existe lo creamo # con los datos incial en vacio file libreta = "clibreta.json" == False): file = open("libreta.json", "w") file.write("(\"libreta.json", "w") file.close()</pre>				
# Leo todo el contenido del Json y lo almaceno en la lista libreta # Libreta es una variable que contiene una lista de diccionarios, cada # es un diccionario que contiene los datos de cada contacto def cargarLibretaMemoria(): with open("libreta.json", "r") as file: data = json.load(file) # Verifico que el archivo contenta al menos un contacto grabado if (len(data ['ibreta"]) > 0):		impiar iar correo		
for contacto in data['tiereto']: libreta.append(contacto) # Funcion para ordenar la lista libreta utilizando como cri # el atributo nombre de cada diccionario libreta.sort(key=lambda x: x["nombre"]) cangarLibretaMemoria()				
# Grabo lo que contiene la libreta en el Json # Este proceso corre cada vez que modifico la lista				

Johanna Arteaga / Florencia Britos / Patricia Felipe / Diego Fynn / Thiago Mauri / Germán Medina / Fabiana Montes de Oca

Docente: Ing Pablo Ortiz

OBJETIVO

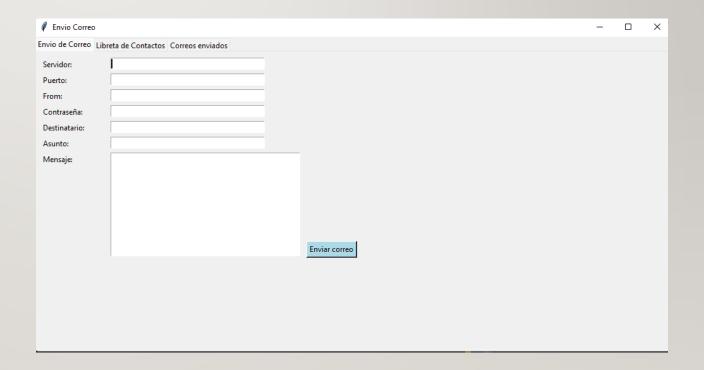
• Generar un simple GUI con Tkinter para gestionar un email con sus funciones básicas

UTILIZAMOS LOS SIGUIENTES MÓDULOS

- Tkinter: Usamos Tkinter para la generación de interfaz grafica GUI
- Os: Modulo que nos permite interactuar con el file system y acceder funcionalidades dependientes del sistema operativo
- Json: Modulo que nos permite manipular archivos JSON
- Socket: usamos gaierror de Socket para verificar errores en conexión TCP cuando establecemos la comunicación SMTP
- Datetime: Modulo que nos permite manipular fecha
- Smptlib: Modulo que nos permite establecer comunicación SMTP
- MIMEMultipar y MIMEText: Modulos que nos permites armar el header y body del correo

ESTRUCTURA DEL PROGRAMA

El programa está organizado en 3 pestañas, Envío de Correo, Libreta de Contactos y Correos enviados



ENVÍO DE CORREO

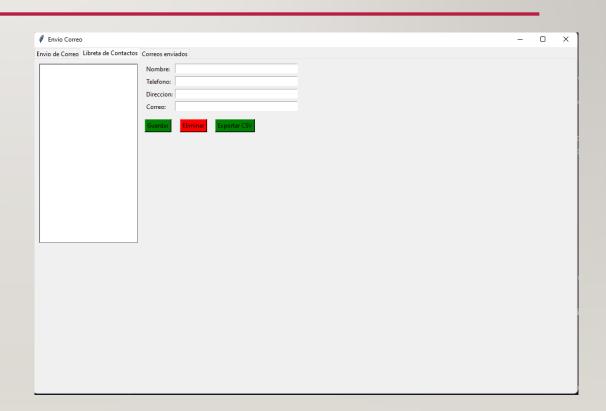
Esta pestaña constará de 6 requerimientos

- Servidor: Dirección del servidor de correo desde donde se enviará el mail
- Puerto: Puerto SMTP (25, 587, 465)
- From: El emisor del correo, aquí ingresaremos la dirección del mail
- Contraseña: Contraseña para validar el mail del emisor
- Destinatario: Ingresaremos el correo electrónico del destinatario
- Asunto: Aquí va el titular del mensaje
- Mensaje: Ingresaremos el cuerpo del correo electrónico

LIBRETA DE CONTACTOS

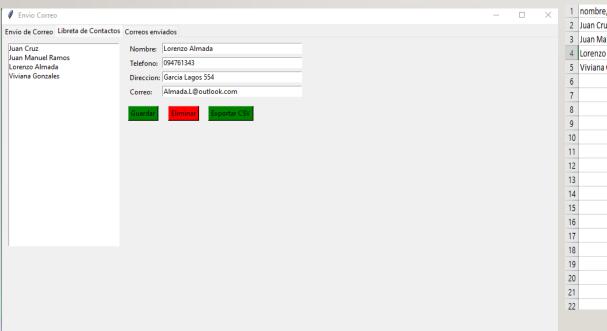
La libreta al principio es una lista de diccionarios vacía, luego a medida que se agregan nuevos contactos se va almacenando los contactos dentro de un archivo .json.

Cada vez que abrimos la aplicación la lista se ira rellenando a partir del contenido del json.

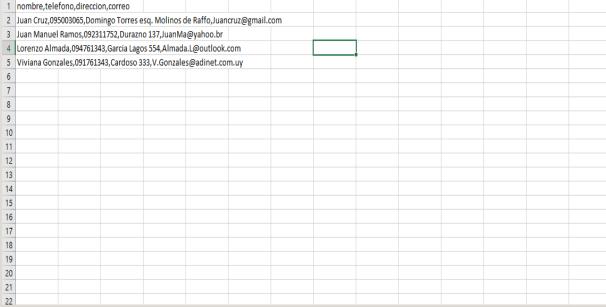


FUNCIONES DENTRO DE LA LIBRETA

Podemos agregar nuevos contactos, actualizar los que ya tenemos, o eliminarlos



Podemos también exportar los contactos con salida .csv



CORREOS ENVIADOS

Donde se almacenan los correos enviados.

Al comienzo Es una lista vacía, luego a medida que se mandan los correos, la lista se carga de datos y los almacena en un archivo .json

