

Reporte de Desarrollo

Introducción

Este reporte describe el proceso de desarrollo de un script en Python que lee una lista de correos de un .csv y que envíe un saludo en un correo electrónico

Se instalo las librerias pandas y smtplib para poder leer el archivo csv se utiliza la librería pandas y smtp para el envio de correos a continuación se describe el codigo realizado:

primero se configura variables necesarias para la integracion con gmail
el password se recupera de una variable de entorno por seguridad

```
# Configuración del correo
smtp_server = "smtp.gmail.com" # Servidor SMTP de Gmail
smtp_port = 587                # Puerto de Gmail para TLS
sender_email = "crilobra@gmail.com" # Correo del remitente
sender_password = os.getenv('password') # Contraseña del remitente
```

lectura de datos usando libreria pandas

```
# Leer la lista de correos electrónicos desde el archivo CSV
contactos = pd.read_csv("lista.csv")
```

conexion servidor de correos

```
# Iniciar la conexión con el servidor SMTP
server = smtplib.SMTP(smtp_server, smtp_port)
server.starttls() # Protocolo de encriptación
server.login(sender_email, sender_password)
```

envio de correo personalizado creando una plantill html

```
# Enviar el correo a cada contacto
for index, row in contactos.iterrows():
    msg = MIMEText()
    msg['From'] = sender_email
    msg['To'] = row['email']
    msg['Subject'] = subject

    # Contenido del correo en formato HTML
```

```

html_content = f"""
<html>
<head></head>
<body style="background-color:CCFFCC;">
<center>
    <h1 style="color: blue;">Feliz Cumpleaños!</h1>
    <p> <b>{row['nombre']}</b>
    en es dia tan especial </p>
    <p>;Espero que lo disfrutes!</p>
</center>
</body>
</html>
"""

# Adjuntar el contenido HTML al mensaje
msg.attach(MIMEText(html_content, 'html'))

fecha = datetime.strptime(row['fecha'], "%Y/%m/%d")

if fecha.day == fechaActual.day and fecha.month ==
fechaActual.month:
    # Enviar el correo
    server.sendmail(sender_email, row['email'], msg.as_string())
    print(f"Correo enviado a {row['email']}")
    with open("registro_envios.csv", "a") as log:
        log.write(f"{row['email']}, Enviado\n")

# Cerrar la conexión
server.quit()
print("Todos los correos han sido enviados.")

```

Conclusiones

El script desarrollado permite leer un archivo csv y enviar un correos y registrar un log de envios