

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

```
# 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 = "rlwd vrgm njlc bwyk" # 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)
```

creacion de variables para el mensaje

```
# Crear el mensaje
subject = "Saludos "
message_body = "Hola, espero que estés bien. ¡Saludos!"
```

envio de correos en forma masiva

```
# Enviar el correo a cada contacto
for index, row in contactos.iterrows():
    msg = MIME multipart()
    msg['From'] = sender_email
```

```

msg['To'] = row['email']
msg['Subject'] = subject

# Cuerpo del mensaje
msg.attach(MIMEText(message_body, 'plain'))

# Enviar el correo
server.sendmail(sender_email, row['email'], msg.as_string())
print(f"Correo enviado a {row['email']}")

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

```

para la programación de envío de correos se ha creado el archivo crontab.sh, el cual se encarga de crear una tarea automática que se ejecuta todos los lunes a las 9 de la mañana

```

# Define el trabajo de cron que quieres añadir
CRON_JOB="0 9 * * 1 /usr/bin/python3 $SCRIPT >>
/ruta/al/logs/salida.log 2>&1"

# Agregar el trabajo al crontab actual
(crontab -l 2>/dev/null; echo "$CRON_JOB") | crontab -

```

Conclusiones

El script desarrollado permite leer un archivo csv y enviar un correos