



**Universidad Nacional Autónoma
de México**

Facultad de Ingeniería

Ingeniería en Computación



Tarea 01.

Envío de correo electrónico

Administración de Servicios de Internet

Profesor: Angel Brito Segura

Grupo: 02

Clave: 2945

Alumno:

Hernández Vázquez Daniela

318092867

Semestre: 2025-2

Fecha de realización: 16 de marzo de 2025

Fecha de entrega: 25 de marzo de 2025

Introducción

El presente documento mostrará evidencia de la implementación de un script en lenguaje de programación Python capaz de enviar un correo electrónico desde una cuenta de Gmail. Utilizando el protocolo SMTP a cualquier destinatario.

Del mismo modo se mostrará el funcionamiento del programa a través de la ejecución de una serie de comandos para permitir el envío de correos electrónicos de manera remota que probará su funcionamiento.

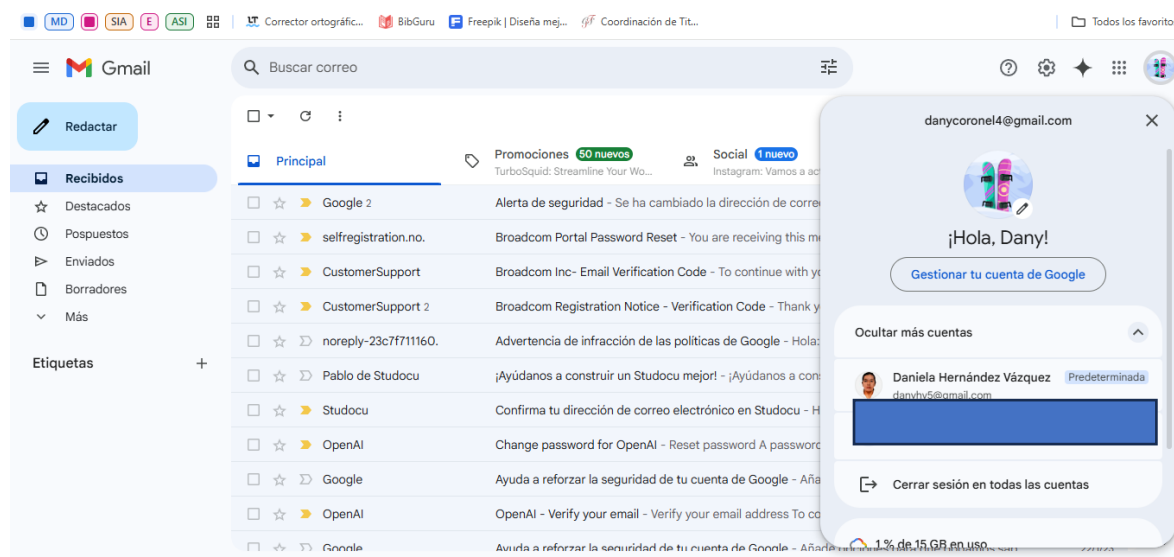
Objetivo

- Crear un script/programa en cualquier lenguaje de programación (Python, Java, Kotlin, Javascript, Typescript, Bash) que permita enviar un correo electrónico desde una cuenta Gmail/Outlook utilizando el protocolo SMTP a un destinatario específico: angel.brito@fi.unam.edu

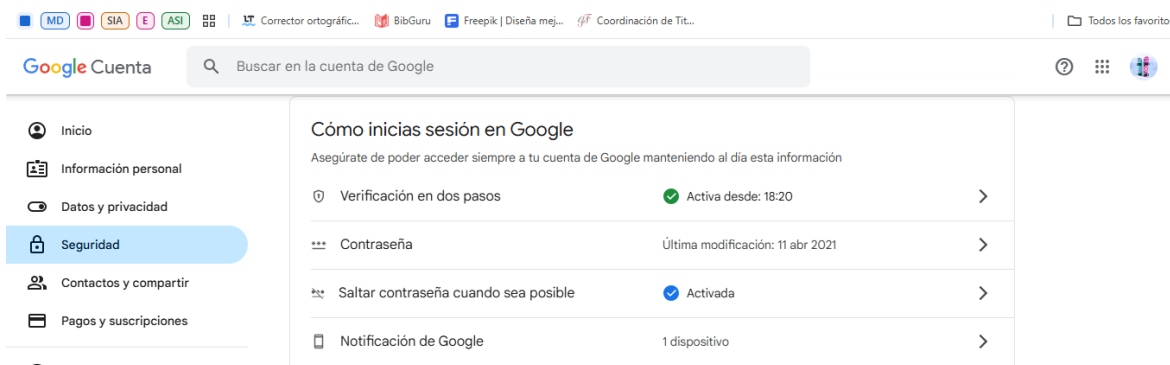
Contenido

Antes de realizar el programa debemos de habilitar en nuestro correo electrónico, ya sea desde nuestra cuenta personal o podemos crear un correo de prueba. En este caso se utiliza la cuenta danycoronel4@gmail.com

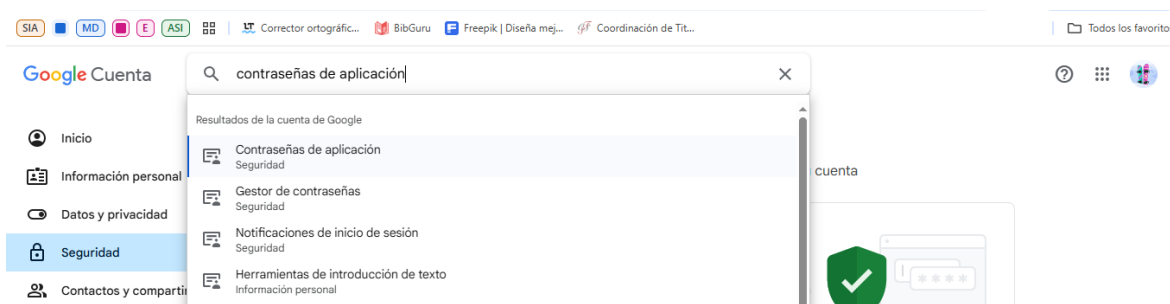
En nuestra cuenta desde la cual enviaremos correos ingresamos a la sección de “cuenta” o “Gestionar tu cuenta de Google”



Posteriormente, en la sección de “seguridad” debemos activar la verificación en dos pasos de la cuenta, en caso de que esta aún no se encuentre activa.



Buscamos en nuestra cuenta de Google “Contraseñas de aplicación”, esto nos llevará a una ventana nueva donde se ingresará el nombre de la aplicación a la cual le deseamos permitir acceso. En este caso se ingresa el nombre de Python lo que generará una contraseña de 16 caracteres, dicha cuenta solo saldrá una vez por lo que es necesario apuntarla correctamente.



← Contraseñas de aplicación

[Más información](#)

Tus contraseñas de aplicación

Python

Creada: 15 mar; utilizada por última vez: 15 mar

Para crear una contraseña específica de la aplicación, escribe el nombre de la aplicación a continuación...

Nombre de la aplicación

Una vez modificado todo lo anterior realizamos en nuestro computador un script en cualquier lenguaje de programación para enviar correos, en este caso Python.

```
1. # Administración de servicios de internet Gpo:2
2. # Hernández Vázquez Daniela - Mar,2025
3.
4. import smtplib
5. import ssl
6. import getpass          #Para contraseña segura
7. from email.mime.text import MIMEText
8. from email.mime.multipart import MIMEMultipart
9.
10. def send_email():
11.     sender_email = "danycoronel4@gmail.com"
12.     password = getpass.getpass("Ingrese su contraseña: ")
13.     receiver_email = "angel.brito@fi.unam.edu"
14.
15.     subject = "Prueba de envío con SMTP"
16.     body = "Esta es una prueba de envio SMTP"
17.
18.     # Crear el mensaje
19.     message = MIMEMultipart("alternativo") #estandar
20.     message["From"] = sender_email
21.     message["To"] = receiver_email
22.     message["Subject"] = subject
23.     message.attach(MIMEText(body, "plain"))
24.
25.     # Configurar conexión SMTP segura por SSL
26.     context = ssl.create_default_context() #conexion segura
27.     with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as server:
28.         server.login(sender_email, password)
29.         print("Inició sesión!")
30.         server.sendmail(sender_email, receiver_email, message.as_string())
31.         print("Mensaje enviado exitosamente")
32.
33. # Llamar a la función
34. send_email()
```

Observamos el programa anterior, y nos damos cuenta como las primeras líneas se encargan de la importación de módulos:

- **smtplib** se usa para enviar correos a través del protocolo SMTP.
- **ssl** se usará para establecer conexiones seguras.
- **getpass** solicitará la contraseña de manera oculta (sin mostrar en pantalla).
- **MIMEText** y **MIMEMultipart** se utilizarán para formatear los correos con texto enriquecido.

Por su parte, la función **send_email()** es la que se encarga de enviar el correo. Para ello es necesario definir en primer lugar todas las partes que lo componen:

- Remitente de correo

- sender_email = "danycoronel4@gmail.com"
- Contraseña del remitente, para esto el correo del remitente ya debe de estar previamente modificado y se debió de haber obtenido la clave de 16 caracteres.
 - password = getpass.getpass("Ingrese su contraseña: ")
- Destinatario del correo, en este caso se utiliza de destinatario a un correo fijo. En caso de querer ingresarlo manual, usar la función input(), lo mismo ocurre con el remitente y el mensaje en caso de querer hacerlo manual.
 - receiver_email = "angel.brito@fi.unam.edu"

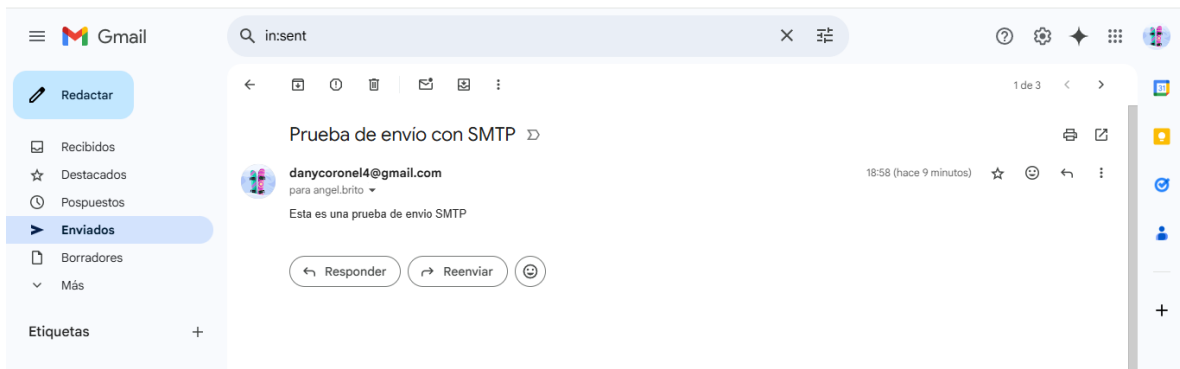
En segundo lugar, realizamos la configuración del correo electrónico como tal, agregando las variables de cadena de asunto “subject”, cuerpo “body” del mensaje o lo que queremos enviar en texto plano.

En tercer lugar, se crea el mensaje uniendo todas las partes anteriores con las líneas 19 a 23 en formato MIME con: Remitente (‘From’), Destinatario (‘To’), Asunto (‘Subject’), Cuerpo en texto plano (‘MIMEText(body, "plain")’) que equivaldrán a las variables previamente definidas.

En cuarto lugar, se crea la conexión al servidor SMTP de Gmail, creando una conexión segura con ssl (línea 26) para encriptar la comunicación con Gmail. Abre una conexión SMTP segura el servidor SMTP de Gmail con "smtp.gmail.com", puerto 465, usado para conexiones SSL de Gmail. En caso de que se quisiera una conexión desde un correo Outlook, el servidor para la conexión segura sería “smtp.office365.com” en el puerto 587. Se ingresa al servidor ya con esta conexión a través del usuario y la contraseña de remitente y se envía el correo imprimiendo na conexión exitosa.

Ahora bien, si ejecutamos el programa, este nos da un mensaje de conexión exitosa y adicionalmente a entrar a nuestro correo remitente observaremos como el mensaje ha sido enviado.

```
Ingrese su contraseña:
Inició sesión!
Mensaje enviado exitosamente|
>>>
```



Conclusiones

Con este ejercicio pudimos ver de manera práctica la implementación de un script en Python para el envío de correos electrónicos utilizando el protocolo SMTP. Así mismo, se ha visto el modo de establecer conexiones seguras mediante SSL y cómo se formatea correctamente un mensaje de correo utilizando las bibliotecas `smtplib`, `ssl`, `getpass` y `email.MIME`.

Es importante recalcar que para que el programa funcione es necesario tener un correo modificado para poder trabajar con la aplicación de Python, de lo contrario el envío no será exitoso. Al tener un destinatario fijo se optimiza el envío de correos a un destinatario, asegurando que los datos sean manejados de manera segura y eficiente, pero también implica que puede ser aplicable a diversas situaciones, como la automatización de notificaciones, el envío masivo de correos y la integración de servicios de comunicación en aplicaciones más complejas.

Bibliografía

Dani, Q. (jun 11, 2023). *Enviar correos con python [metodo 2023]*. Youtube. Recuperado el 20 de marzo de 2025, de <https://www.youtube.com/watch?v=oPAo8Hh8bj0&t=52s>

Luis Carlos Contreras – EL INGE. (may 9, 2020). *Cómo enviar correos electrónicos con Python [CLASE COMPLETA]*. Youtube. Recuperado el 20 de marzo de 2025, de <https://www.youtube.com/watch?v=mRXR8eO9igQ&t=448s>