TEC | Tecnológico de Costa Rica

Tarea 1 de Redes

Jesús Gabriel Cordero Díaz

Carnet: 2020081049

March 21, 2025

Redes

Profesor: Ing. Kevin Moraga, MSc.

Contents

1 Introducción

El correo electrónico es uno de los protocolos de comunicación más antiguos y aún vigentes en Internet, siendo el Protocolo Simple de Transferencia de Correo (SMTP) una pieza clave en su funcionamiento. Este proyecto tiene como objetivo la implementación de un servidor de correo que permita el envío, recepción y consulta de correos electrónicos utilizando el protocolo SMTP, así como la configuración de un servidor IMAP para la lectura de correos y la integración de un sistema de notificación basado en NNTP.

Durante el desarrollo del proyecto, se implementó un servidor SMTP (‘smtpserver.py‘) y un cliente SMTP (‘smtpcliente.py‘) utilizando la biblioteca Twisted. También se creó un servidor IMAP (‘imapserver.py‘) para permitir la lectura de correos desde un cliente externo. Para mejorar la experiencia del usuario, se diseñó una interfaz gráfica sencilla para el cliente SMTP, facilitando la interacción y el envío de correos electrónicos.

Adicionalmente, se configuró un dominio en Cloudflare para permitir el envío y recepción de correos desde un dominio personalizado. La implementación se realizó sobre una máquina virtual EC2 de Amazon Web Services (AWS), proporcionando un entorno de desarrollo estable y seguro para la configuración de los servicios de correo electrónico.

Este proyecto permitió explorar en profundidad los protocolos de correo electrónico (SMTP, IMAP y NNTP) y los desafíos técnicos relacionados con la autenticación, el cifrado SSL/TLS y la configuración de servidores en un entorno en la nube. La combinación de estas tecnologías garantiza una solución robusta y funcional para el manejo de correo electrónico.

2 Ambiente de Desarrollo

El desarrollo del proyecto se realizó utilizando un conjunto de herramientas y tecnologías modernas para garantizar estabilidad, seguridad y facilidad de implementación:

- **Lenguaje de programación:** Python 3.12, debido a su versatilidad y al soporte robusto para la biblioteca Twisted.
- **Biblioteca principal:** Twisted 24.11, utilizada para la implementación de los protocolos SMTP e IMAP, aprovechando su modelo basado en eventos y su compatibilidad con servicios de red.
- **Entorno de desarrollo:** Visual Studio Code, seleccionado por su entorno intuitivo, sus herramientas de depuración integradas y su compatibilidad con Git.
- **Sistema operativo:** Ubuntu, elegido tanto para el desarrollo local como para la implementación en la máquina virtual EC2 de AWS, debido a su estabilidad y soporte extendido para servicios de red.
- **Control de versiones:** Git, para el manejo eficiente del código fuente y la colaboración a través de un repositorio remoto.
- **Entorno de despliegue:** Máquina virtual EC2 de Amazon Web Services (AWS), configurada para la ejecución de los servicios de correo en un entorno de producción. Además, se configuró el dominio utilizando Cloudflare para gestionar los registros DNS (A y MX), asegurando la correcta redirección y recepción de correos electrónicos.

Este conjunto de herramientas proporcionó un entorno de desarrollo robusto y escalable, permitiendo una implementación eficiente y una configuración segura para los servicios de correo electrónico.

3 Estructuras de datos y funciones

En el desarrollo del proyecto se utilizaron varias estructuras de datos y clases para implementar los protocolos SMTP e IMAP, así como la funcionalidad del cliente SMTP. A continuación, se describen las principales estructuras y funciones utilizadas:

3.1 Servidor SMTP (smtpserver.py)

Para el servidor SMTP, se implementaron las siguientes clases principales:

- **ConsoleMessage**: Esta clase implementa la interfaz `IMessage` de Twisted para manejar mensajes entrantes. - Recibe líneas de un mensaje SMTP y las almacena en una lista interna. - Guarda el mensaje recibido en el sistema de archivos en formato '.eml', organizándolo por dominio y usuario.
- **ConsoleMessageDelivery**: Clase que implementa la interfaz `IMessageDelivery`. - Valida los remitentes y destinatarios de los mensajes. - Si el destinatario pertenece a un dominio permitido, crea una instancia de `ConsoleMessage` para almacenar el mensaje.
- **ConsoleSMTPFactory**: Clase que extiende `SMTPFactory` de Twisted. - Configura el comportamiento del servidor SMTP, incluyendo la autenticación y el manejo de mensajes. - Utiliza `ConsoleMessageDelivery` para procesar los mensajes entrantes.

3.2 Cliente SMTP (smtpclient.py)

El cliente SMTP permite enviar correos electrónicos de manera automatizada y personalizada:

- **send_email**: Función que toma los datos del destinatario desde un archivo CSV y envía el correo mediante el protocolo SMTP de Twisted. - Crea un mensaje MIME personalizado y permite añadir archivos adjuntos. - Utiliza la función `sendmail` de Twisted para enviar el correo.
- **main**: Función principal que gestiona la lectura de la plantilla de correo y la lista de destinatarios. - Procesa los argumentos de línea de comandos (como servidor SMTP, archivo CSV y plantilla de mensaje). - Llama a la función `send_email` para cada registro del CSV.

3.3 Cliente SMTP con Interfaz Gráfica (smtpclient_gui.py)

Se creó una interfaz gráfica con Tkinter para facilitar el envío de correos electrónicos:

- **send_emails_callback**: Función que se activa al presionar el botón de "Enviar Correos" en la interfaz. - Lee los datos de los campos de la interfaz y los valida. - Utiliza la función `send_email` para enviar el correo mediante Twisted.
- **Interfaz gráfica**: - La interfaz gráfica fue creada con Tkinter. - Permite seleccionar el archivo CSV, la plantilla de mensaje y un archivo adjunto de manera intuitiva.

3.4 Servidor IMAP (imapserver.py)

El servidor IMAP permite a los clientes consultar y descargar correos electrónicos:

- **IMAPProtocol**: Clase que implementa las operaciones del protocolo IMAP. - Maneja comandos IMAP como LOGIN, FETCH, STORE, y LOGOUT. - Permite la autenticación y la recuperación de correos electrónicos desde el sistema de archivos.
- **IMAPFactory**: Clase que crea las instancias de `IMAPProtocol` y gestiona el almacenamiento de correos. - Configura el directorio de almacenamiento de los correos recibidos.

Las estructuras de datos y clases mencionadas garantizan una implementación eficiente y modular del servidor y cliente SMTP e IMAP, facilitando la escalabilidad y el mantenimiento del código.

4 Instrucciones para ejecutar el programa

Para ejecutar los servicios SMTP e IMAP, así como el cliente SMTP, se deben seguir los siguientes pasos. El servidor SMTP y el servidor IMAP están configurados en una máquina virtual EC2 de AWS, por lo que será necesario tener acceso a la clave SSH para poder conectarse y ejecutar los comandos directamente en la instancia EC2.

4.1 Servidor SMTP

Para ejecutar el servidor SMTP, se debe ejecutar el siguiente comando desde la máquina virtual EC2:

```
sudo python3.12 smtpserver.py -d example.com,example.org,localhost,
   polonorte.com,corderodiaz.com -s ./var/mail -p 25
```

4.2 Servidor IMAP

Para ejecutar el servidor IMAP, se debe ejecutar el siguiente comando desde la máquina virtual EC2:

```
sudo python3.12 imapserver.py -s ./var/mail -p 465
```

4.3 Cliente SMTP (sin interfaz gráfica)

El cliente SMTP puede ejecutarse desde la terminal con el siguiente comando:

```
sudo python3.12 smtpclient.py -H localhost -P 25 -c doc/destinatarios.csv
   -m doc/mensaje.txt -f smtpserver.py
```

4.4 Cliente SMTP con Interfaz Gráfica

Para ejecutar el cliente SMTP con la interfaz gráfica (Tkinter), el comando es el siguiente:

```
sudo python3.12 smtpclient_gui.py
```

4.5 Configuración del dominio en Cloudflare y la máquina EC2

Para que el servicio de correo funcione correctamente en el dominio `corderodiaz.com`, se deben realizar las siguientes configuraciones:

4.5.1 Desplegar el Código en la Instancia EC2

Para subir los archivos a la máquina EC2, se utilizó el siguiente comando `scp` desde la máquina local:

```
scp -i "llave.pem" <archivo> <maquina_ec2>:<ruta_destino>
```

Una vez subidos los archivos, el código puede ejecutarse directamente en la máquina EC2 mediante los comandos descritos anteriormente.

4.5.2 Configurar la Seguridad en AWS

- Asegúrese de que el grupo de seguridad de la instancia EC2 permita conexiones entrantes en los siguientes puertos:

- **25, 587**: Para SMTP.
- **143, 993**: Para IMAP.

4.5.3 Configurar los registros DNS en Cloudflare

En el panel de configuración de DNS de Cloudflare para el dominio `corderodiaz.com`, se configuraron los siguientes registros:

- **Registro A:** - Nombre: mail - Tipo: A - Valor: 34.227.16.51 (IP de la máquina EC2) - Modo: DNS Only
- **Registro MX:** - Nombre: corderodiaz.com - Tipo: MX - Valor: mail.corderodiaz.com - Prioridad: 10

4.5.4 Verificación de la Configuración DNS

Para verificar que la configuración DNS es correcta, se puede hacer un ping al dominio configurado:

```
ping mail.corderodiaz.com
```

Ejemplo de respuesta:

```
PING mail.corderodiaz.com (34.227.16.51) 56(84) bytes of data.
64 bytes from ec2-34-227-16-51.compute-1.amazonaws.com (34.227.16.51):
   icmp_seq=1 ttl=52 time=90.7 ms
```

Este resultado indica que el DNS está correctamente configurado y que el tráfico está redirigiendo a la instancia EC2 correctamente.

5 Actividades realizadas por el estudiante

A continuación, se presenta un resumen de las actividades realizadas durante el desarrollo del proyecto, ordenadas cronológicamente y con las horas aproximadas invertidas:

Fecha	**Actividad realizada**	**Horas**
05/03/2025	Initial commit, configuración inicial del repositorio y entorno de desarrollo.	2
08/03/2025	Comienzo de la implementación del servidor SMTP (sin SSL/TLS).	4
12/03/2025	Primera implementación de SMTP con recepción de archivos adjuntos usando MIME.	5
14/03/2025	Implementación inicial de `imapserver.py`.	4
15/03/2025	Validaciones y soporte para archivos adjuntos en `smtpclient.py`.	3
17/03/2025	Implementación de la interfaz gráfica para el cliente SMTP.	4
20/03/2025	Correcciones en `imapserver.py` y ajustes en el cliente SMTP.	5
21/03/2025	Implementación de cifrado SSL/TLS en el servidor SMTP e IMAP.	6
21/03/2025	Versión final de SMTP e IMAP (sin NNTP).	4
Total		**50**

Table 1: Resumen de actividades realizadas

La implementación del proyecto abarcó desde el 8 de marzo hasta el 21 de marzo de 2025, con una inversión total de aproximadamente 50 horas de trabajo. Las actividades incluyeron la implementación de los protocolos SMTP e IMAP, la creación de una interfaz gráfica para el cliente SMTP y la configuración de seguridad mediante SSL/TLS.

6 Autoevaluación

En esta sección se presenta una evaluación del estado final del proyecto, los problemas encontrados y las limitaciones técnicas, un resumen de los commits realizados y una calificación personal sobre el desempeño en los diferentes aspectos del proyecto.

6.1 Estado final del programa

El estado final del programa fue el siguiente:

- **SMTP Server**: - Funcional con SSL/TLS, pero debido a problemas con el cliente SMTP, se dejó sin cifrado para asegurar el envío de correos correctamente.
- **IMAP Server**: - Funcional con soporte para SSL/TLS, pero finalmente se descartó debido a problemas de tiempo y compatibilidad.
- **SMTP Client**: - Funcional y compatible con el servidor SMTP sin cifrado SSL/TLS.
- **SMTP Client GUI**: - Funcional con el servidor SMTP sin cifrado SSL/TLS.
- **Configuración del dominio**: - Se configuró correctamente utilizando Cloudflare y una instancia EC2 de AWS para alojar el servidor SMTP e IMAP.
- **NNTP-Notifier**: - No se logró implementar debido a limitaciones de tiempo.

  Aunque la idea de usar SSL/TLS fue descartada debido a errores en el cliente SMTP al intentar enviar correos, en pruebas realizadas con Thunderbird el cifrado SSL/TLS funcionó correctamente. Los problemas solo ocurrieron al implementar SSL/TLS con el cliente SMTP, lo que llevó a la decisión de desactivarlo para garantizar el envío adecuado de correos desde el cliente.

6.2 Problemas encontrados y limitaciones

- **Problemas encontrados**: - Problemas en el cliente SMTP al intentar enviar correos con cifrado SSL/TLS, lo que obligó a desactivar el cifrado en el servidor SMTP. - No se logró implementar el servicio NNTP-Notifier debido a la complejidad de la configuración y la falta de tiempo.
- **Limitaciones**: - Información limitada sobre cómo Thunderbird maneja IMAP, lo que dificultó la implementación. - La mayoría de los ejemplos encontrados estaban relacionados con clientes IMAP y no con servidores IMAP, lo que complicó el desarrollo del servidor IMAP.

6.3 Reporte de commits de Git

A continuación, se muestra un resumen de los commits más relevantes realizados durante el desarrollo del proyecto:

Fecha	Descripción del commit
05/03/2025	Initial commit, configuración inicial del repositorio.
08/03/2025	Comenzó la tarea 1, configuración inicial del entorno de desarrollo.
12/03/2025	Primera implementación de SMTP con soporte para archivos adjuntos usando MIME.
14/03/2025	Implementación inicial de `imapserver.py`.
15/03/2025	Validaciones y soporte para archivos adjuntos en `smtpclient.py`.
17/03/2025	Implementación de la interfaz gráfica para el cliente SMTP.
20/03/2025	Correcciones en `imapserver.py` y ajustes en el cliente SMTP.
21/03/2025	Implementación de cifrado SSL/TLS en el servidor SMTP e IMAP.
21/03/2025	Versión final sin soporte para NNTP y sin SSL/TLS en el cliente SMTP.

Table 2: Resumen de commits relevantes

6.4 Calificación personal

A continuación, se presenta una autoevaluación del proyecto, usando una escala de 0 a 5:

- **Aprendizaje del rotocolo SMTP**: 4
- **Aprendizaje del protocolo IMAP**: 4
- **Aprendizaje del protocolo NNTP**: 2
- **Aprendizaje de la capa SSL/TLS**: 4

El proyecto permitió adquirir experiencia práctica en la implementación de protocolos de correo electrónico y el uso de la biblioteca Twisted en Python. También se profundizó en la configuración de servidores en entornos basados en la nube y en la administración de seguridad mediante SSL/TLS.

7 Lecciones aprendidas

El desarrollo de este proyecto permitió obtener una comprensión más profunda sobre los protocolos de correo electrónico y la configuración de servidores en un entorno de red. Las principales lecciones aprendidas fueron las siguientes:

- **Protocolo SMTP**: Se adquirió un conocimiento sobre el funcionamiento del protocolo SMTP, incluyendo el proceso de autenticación, el envío de correos electrónicos y la gestión de archivos adjuntos utilizando el formato MIME. También se comprendió la importancia de la configuración adecuada de los registros DNS (MX y A) para asegurar la entrega correcta de los correos.
- **Protocolo IMAP**: Se exploró la estructura y funcionamiento del protocolo IMAP, comprendiendo el manejo de buzones, la autenticación de usuarios y la recuperación de correos desde el servidor. La implementación del servidor IMAP permitió experimentar con el manejo de estados y la gestión de sesiones de usuario.
- **Seguridad mediante SSL/TLS**: Se adquirió experiencia en la configuración de cifrado mediante SSL/TLS para proteger la comunicación entre el cliente y el servidor. Aunque se encontraron problemas al implementar SSL/TLS con el cliente SMTP, se comprendió el proceso de negociación de cifrado y el manejo de certificados.
- **Uso de la biblioteca Twisted**: La biblioteca Twisted permitió implementar servidores de red basados en eventos, facilitando la gestión de múltiples conexiones simultáneas. Se comprendió el modelo de programación asíncrona y la importancia de la estructura de callbacks y Deferreds en Twisted.
- **Configuración de servidores en la nube**: La configuración y despliegue de servicios en una máquina virtual EC2 de AWS proporcionó experiencia práctica en la administración de servidores remotos. Se comprendió el proceso de apertura de puertos, configuración de grupos de seguridad y monitoreo de conexiones entrantes y salientes.

Estas lecciones me permitirán abordar futuros proyectos de redes y servidores con mayor confianza, especialmente en la implementación de protocolos seguros y en la configuración de entornos en la nube.

8 Bibliografía

Durante el desarrollo del proyecto, se consultaron diversas fuentes para comprender e implementar los protocolos SMTP e IMAP, así como la configuración de servidores en AWS y Cloudflare. Las principales fuentes utilizadas fueron las siguientes:

[1] T. M. Labs, *Twisted documentation*, 2025. [Online]. Available: `https://twistedmatrix.com/documents/current/`.

[2] P. S. Foundation, *Python email library documentation*, 2025. [Online]. Available: `https://docs.python.org/3/library/email.html`.

[3] Cloudflare, *Cloudflare dns documentation*, 2025. [Online]. Available: `https://developers.cloudflare.com/dns/`.

[4] A. W. Services, *Aws ec2 documentation*, 2025. [Online]. Available: `https://docs.aws.amazon.com/`.

[5] Stack Overflow, *Solutions for ssl/tls configuration with twisted and python*, Accessed: 2025-03-21, 2025. [Online]. Available: `https://stackoverflow.com`.