

Laboratorio #03
SMTP RFC 5321
Ciencias de la Computación VIII

Resumen

En este laboratorio, implementaremos un servidor y cliente SMTP en Java para comprender el funcionamiento del protocolo SMTP y los comandos principales que se utilizan en la transmisión de correos electrónicos. Los comandos a implementar incluyen: HELO, MAIL FROM:, RCPT TO:, DATA, ".", y QUIT. Al finalizar este laboratorio, los estudiantes tendrán una mejor comprensión de cómo se maneja la comunicación de correos electrónicos a nivel de protocolo.

Protocolo SMTP

Es un protocolo de red estándar utilizado para la transmisión de correos electrónicos a través de Internet. Fue definido por primera vez en 1982 (RFC 821) y se ha convertido en el principal protocolo para el envío de correos electrónicos. SMTP opera sobre el puerto 25 por defecto, aunque los puertos 587 y 465 también se utilizan para conexiones autenticadas y seguras. En el laboratorio vamos a entender el flujo de comunicación entre un cliente y un servidor SMTP, recuerde que este protocolo es de la Capa de Aplicación.

Implementar un Servidor SMTP

- Desarrollar un servidor que pueda manejar múltiples conexiones de clientes.
- Procesar y responder adecuadamente a los comandos SMTP enviados por el cliente.

Implementar un Cliente SMTP

- Desarrollar un cliente que pueda enviar correos electrónicos utilizando comandos SMTP.
- Apóyese en el protocolo HTTP para crear una interfaz gráfica.

2.1. Basic Structure

The SMTP design can be pictured as:



Comandos

HELO Inicia una sesión SMTP con el servidor. HELO es el comando básico

1. HELO

- a. Inicia una sesión SMTP con el servidor. HELO es el comando básico. Un saludo simple que verifica el protocolo SMTP este activo.

- b. **Ejemplo:**

- **HELO** ejemplo.com

2. MAIL FROM:

- a. Especifica la dirección del remitente del correo electrónico. Este contiene el dominio del servidor SMTP del que se está enviando.

- b. **Ejemplo:**

- **MAIL FROM:** <remitente@ejemplo.com>

3. RCPT TO:

- a. Especifica la dirección del destinatario del correo electrónico. Este puede ser del mismo servidor o de otro servidor.

- b. **Ejemplo:**

- **RCPT TO:** <destinatario@gmail.com>

4. DATA (.)

- a. Inicia la entrada del contenido del correo electrónico. El cuerpo del mensaje se termina con una línea que contiene solo un punto (.), se recalca que simplemente es un punto en la última línea.

- b. **Ejemplo:**

- **DATA**

- Hola Mensaje

- .

5. QUIT:

- a. Termina la sesión SMTP, cierra conexión el cliente con el servidor.

- b. **Ejemplo:**

- **QUIT**

Recuerda leer el RFC para tener los códigos de respuesta del Servidor en la sección 4.2.2 del RFC 5321

Ejemplo Completo

El ejemplo es de la vista del Cliente, el color Azul es la línea enviada por el SMTP Server y el rojo enviado por el SMTP Client

```
220 Lab03.com SMTP Server
```

```
HELO galileo.edu
```

```
250 Ok Hello Galileo.edu, pleased to meet you
```

```
MAIL FROM:<tu-correo@Lab03.com>
```

```
250 Ok MAIL FROM
```

```
RCPT TO:<destinatario_1@gmail.com>
```

```
250 Ok RCPT TO
```

```
RCPT TO:<destinatario_2@gmail.com>
```

```
250 Ok RCPT TO
```

```
DATA
```

```
354 Send DATA
```

```
Subject: Prueba de correo con múltiples destinatarios
```

```
From: tu-correo@ejemplo.com
```

```
To: destinatario1@ejemplo.com, destinatario2@ejemplo.com
```

```
Cc: destinatario3@ejemplo.com
```

Este es un correo de prueba enviado usando comandos SMTP simples.

Incluye múltiples destinatarios y una línea de copia (Cc).

Saludos,

Tu Nombre

.

```
250 OK DATA #1234 <-- ID ROW
```

```
QUIT
```

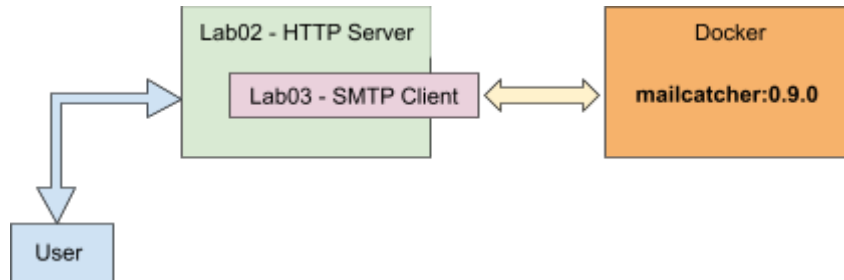
```
221 Ok QUIT
```

NOTA: Para facilitar la implementación debe de utilizar el Servidor HTTP del Lab 02, utilizando su implementación POST, creando el thread con un socket TCP para crear el protocolo SMTP Client para abrir comunicación con el SMTP Server.

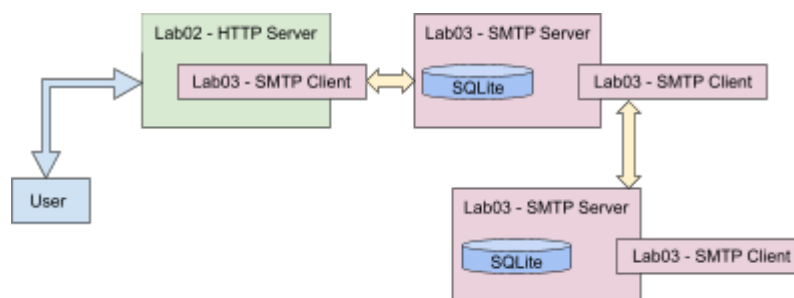
Realizar Pruebas

Probar la comunicación entre el cliente y el servidor SMTP.

- Verificar que los correos electrónicos se envían correctamente siguiendo el protocolo y se almacenan correctamente.
- Pueden utilizar mailcatcher en Docker para probar con un SMTP Server muy simple.



- Recuerde que el Servicio SMTP (Server/Client), va a tener el buzón de un dominio de correo por ejemplo, galileo.edu, si es de otro conocido Servicio conocido en la RED (Alumno de la Clase) se reenviará a él.
 - Ejemplo: Si nuestro Servicio SMTP se llama Lab03.com y sus correos son Usuario1@Lab03.com, si desde el servicio HTTP/SMTPClient se envía un correo con destinatario Usuario2@Lab03.com este se almacena en el Servidor, pero si existe un destinatario que sea UsuarioExterno@yahoo.com este se debe de enviar al Servicio SMTP correspondiente, en este caso al de su compañero de clase.
- Haga la prueba con otro Servicio SMTP en la clase para realizar Relay del Correo.



Notaciones de código entregado:

Se entrega una idea del POST y HTML dentro del código de **Lab02** para el servidor HTTP, se entrega un ejemplo de **SQLiteJDBC.java** para que pueda implementar una base de datos para gestionar los datos, también existe en el Zip un **SMTPClient.java** con una implementación ejemplo del Protocolo, también se entrega un **SMTPServer.java** con lo mínimo funcional.

El código entregado está pensado para que funcionen como una sola unidad separada SMTPClient ----> SMTPServer, pero la implementación debe ser unificada creando [SMTPServer-SMTPClient] -> Otro-ServicioSMTP, este código lo puede modificar o hacer una implementación de Clases.

Entrega

- Suba un archivo ZIP al GES con todos los archivos fuente.
- La calificación se realizará de manera presencial, en dado caso que no pueda presentarse enviar un video Explicando su Código, mostrando la compilación y ejecución, hacer varios ejemplos de su Funcionalidad leyendo los LOGs.
- Use Java 21 o 22 para su implementación.
- El laboratorio puede tener una calificación de cero si:
 - Si no compila con el Makefile (en java 21 o 22)
 - Servicio HTTP/SMTPClient, Servicio SMTP (Relay)
 - Si los LOGS no tienen el formato establecido, establecido en el Lab02.
 - Si no siguió las instrucciones del funcionamiento.
 - Si no cumple con los comandos SMTP.

Referencias:

- https://es.wikipedia.org/wiki/Protocolo_para_transferencia_simple_de_correo
- <https://datatracker.ietf.org/doc/html/rfc5321>
- <https://www.linkedin.com/pulse/email-marketing-101-what-smtp-relay-why-do-we-use-amir-jirbandey>

Ejemplo amplio sobre el Protocolo SMTP y Protocolos IMAP/POP

