

Trabajo Práctico

TPL 4 - Correo Electrónico SMTP - POP3 - IMAP4 - MIME

Fecha de entrega: 09/05/2021

Franco, Juan Martín 149.615

juanmartin_franco@hotmail.com

1. Describa el objetivo y como opera la aplicación correo electrónico, indicando los elementos involucrados: que son y cuál es la función de los agentes de usuario (user agents - UAs) y agentes de transferencia de mensajes (mail transfer agent - MTAs).

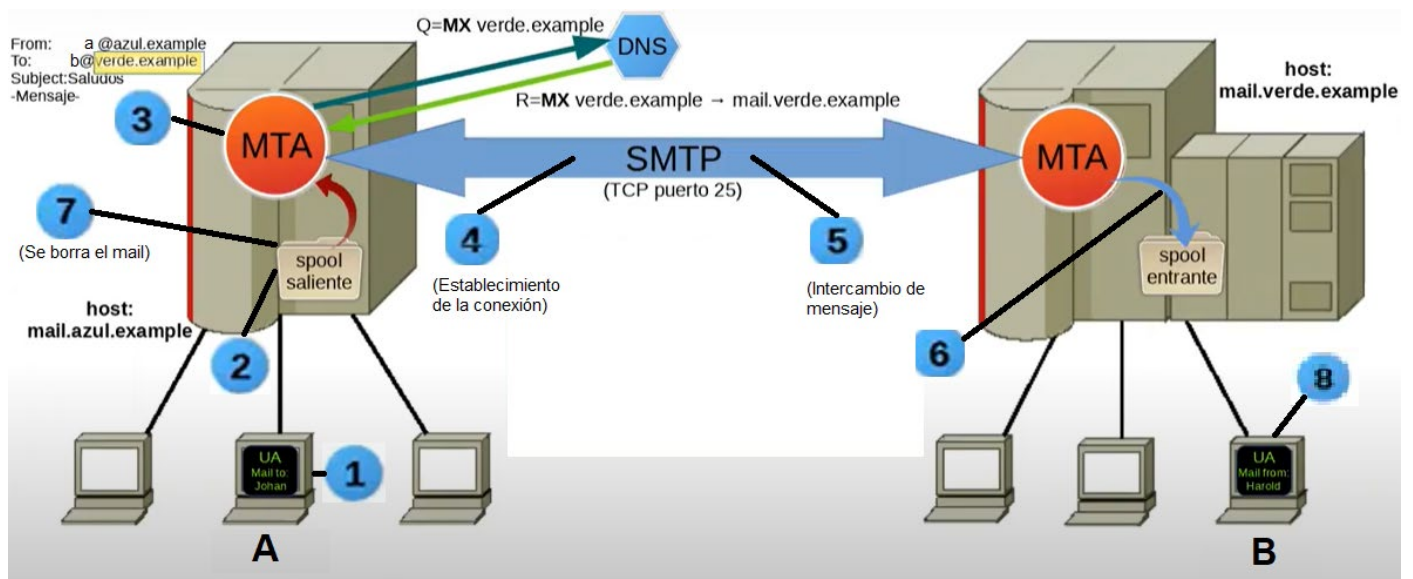
Objetivo del correo electrónico:

El objetivo del correo electrónico es el intercambio de mensajes entre un remitente y un destinatario, en cualquier parte del mundo, en el momento en el que se desee.

Como opera el correo electrónico:

Para explicar el funcionamiento del correo electrónico, me basé en la explicación dada en la clase, en la cual se detallaron todos los conceptos que intervienen (todos los elementos involucrados) y su función.

Para representarlo gráficamente utilizaré la imagen provista en la presentación de la clase, la cual resume bien donde interviene cada parte.



En el ejemplo, se supone que Harold quiere enviarle un mensaje a Johan, de aquí en adelante Harold y Johan serán A y B respectivamente (para simplificar la explicación).

A tiene acceso a un ordenador cuyo dominio pertenece a azul.example.

B tiene acceso a un ordenador cuyo dominio pertenece a verde.example.

1. Para enviar un mail, A utiliza un programa llamado User Agent (UA), que funciona como cliente en un protocolo de red y su funcionalidad se basa en permitirle al usuario ingresar el destinatario y el texto del mensaje.

Originalmente los UA se manejaban por terminal, pero en la actualidad existen alternativas como ThunderBird que es un UA pero con interfaz gráfica.

2. Bien, una vez completado el destinatario y el cuerpo del mensaje, el programa (UA) lo guarda en buffer (mejor conocido como Spool, que es un área de almacenamiento donde una aplicación puede guardar datos hasta que un periférico lento [Por ej., impresora] los pueda procesar).

El mensaje va a permanecer en el Spool saliente hasta que algún proceso (que constantemente está revisando dicho directorio) lo pueda procesar.

Al proceso que está revisando el directorio en busca de mensajes para procesar se lo conoce como MTA (Mail Transfer Agent, Agente de transferencia de mensajes).

3. El MTA una vez encuentra un mensaje, intentará enviarlo a destino utilizando el protocolo SMTP (Simple Mail Transfer Protocol).

Para poder enviarlo, deberá contactarse con el MTA de su par destino, es decir, el MTA de B (que pertenece al dominio verde.example), por lo que extrae el destino y por medio de una consulta DNS (será explicada en el punto 2), intenta obtener el valor del registro MX asociado al dominio verde.example.

4. Una vez obtenida la respuesta, el MTA de A conoce el host al cual le tiene que enviar los correos electrónicos.

Conocido ese host, lo que hace el MTA de A es establecer una conexión TCP en el puerto destino 25 (el puerto bien conocido relativo al protocolo SMTP).

5. Una vez establecida la conexión, los MTA comienzan a “conversar” (utilizando el protocolo que ambos conocen, el SMTP), y el MTA Origen (MTA de A) transmite al MTA Destino (MTA de B) el mensaje que había quedado en el Spool de A.

6. El MTA Destino lo recibe y lo almacena en su Spool (Spool de B), ya que es poco probable que el usuario B esté leyendo el mensaje en ese momento.

7. Cuando el MTA destino graba el archivo en un espacio de almacenamiento permanente, se confirma al MTA origen que el mensaje fue enviado, lo que implica que el MTA origen puede borrar de su Spool saliente el mensaje.

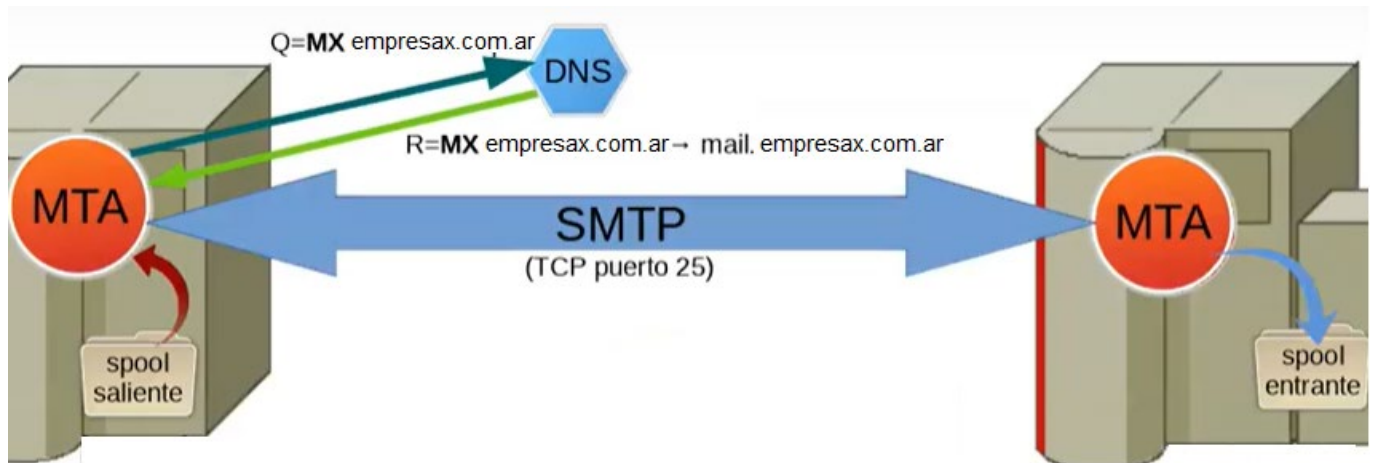
8. En algún momento, el usuario B va a acceder a la computadora utilizando su User Agent (UA), el cual le va a permitir leer el correo electrónico recibido.

2. Un usuario redacta un mensaje destinado a consultas@empresax.com.ar en su cliente de correo y lo envía mediante su propio MTA. Detalle paso a paso el procedimiento que sigue el MTA del usuario para conocer la dirección IP del MTA remoto con el que debe conectarse para entregar el mensaje al destinatario.

Como se explicó en el ejercicio 1), el MTA es un proceso que está constantemente revisando el Spool en busca de nuevos mails.

Una vez el MTA encuentra un mail en el Spool saliente, extrae del mismo mail el host de destino (en este caso empresax.com.ar) y realiza una consulta DNS preguntando por el valor del registro MX de dicho host extraído.

Se puede visualizar con claridad en la siguiente imagen:



Primero el MTA extrajo desde el Spool saliente el mail, el cual contiene la cabecera y el cuerpo. Dentro de la cabecera está contenida la dirección de destino, en este caso consultas@empresax.com.ar.

Una vez obtenida dicha dirección, el MTA realiza la siguiente query DNS:

Q = MX empresax.com.ar

Y, como en cualquier consulta DNS, se irá recorriendo de manera iterativa (la consulta recursiva solo la hace con el resolver, quien le devolverá la respuesta final) un camino hasta llegar a algún dominio que conozca a empresax.com.ar.

Una vez encontrado dicho dominio, obtiene como respuesta algo similar a lo siguiente:

R = MX empresax.com.ar

Authority Section

MX empresax.com.ar → mail.empresax.com.ar

Additional Section

mail.empresax.com.ar → 210.32.3.21 (Por ejemplo)

Obtenida esta respuesta, el MTA Origen ya está en condiciones de establecer una conexión TCP en el puerto destino 25 para poder intercambiar el mail.

3. ¿Cuáles son los comandos SMTP de una implementación mínima? Describa someramente cada uno.

Para una implementación mínima, es necesario contar con los siguientes comandos SMTP:

1. HELO = El cliente se conecta con el nombre de su ordenador e inicia la sesión con él.
2. MAIL FROM = El cliente nombra al remitente del correo electrónico.
3. RCPT TO = El cliente nombra al destinatario del correo electrónico.

4. DATA = El cliente inicia la transmisión del correo electrónico, en donde escribe el cuerpo del mensaje y luego mediante una combinación de caracteres pondrá fin al mismo.

5. QUIT = El cliente termina la sesión.

4. Comente los problemas que plantea el uso de SMTP en cuanto a que el protocolo no requiere obligatoriamente la autenticación por parte del usuario que envía correo y el abuso que esto puede acarrear.

Un inconveniente relacionado al uso del protocolo SMTP es que los usuarios no se autentican cuando se establece una conexión y, por lo tanto, el remitente de un correo electrónico es poco fiable.

Esto puede generar que varios problemas, como por ejemplo:

1. Envío masivo de SPAM.
2. Direcciones de remitente falsas.

Por poner un ejemplo, un usuario puede enviar un mail haciéndose pasar por cualquier persona (por ejemplo Bill Gates), y, como no requiere autenticación, quien lo reciba no sabrá si realmente se trata de dicha persona o simplemente es alguien que está abusando de la carencia de dicho protocolo.

5. ¿Cuál es el propósito de los protocolos POP e IMAP? Describa brevemente los comandos disponibles para el protocolo POP3. ¿Qué ventajas ofrece el protocolo IMAP4 sobre POP3?

El propósito de los protocolos POP e IMAP es el de facilitar el acceso a mensajes de correo electrónico, es decir, proveer acceso al buzón de correo almacenado en el servidor.

No todo el mundo se conecta directamente al host donde está el MTA destino, si no que puede ser que un usuario acceda a través de una red local o a través de internet a ese servidor de correo.

Ahora, el usuario puede leer el mensaje no directamente desde el directorio (desde el Spool).

El UA (User Agent) va a correr de manera remota, por lo que va a necesitar algún protocolo para acceder a ese Spool.

Estos protocolos son: POP (Post Office Protocol, cuya versión más actual es la denominada POP3) e IMAP (Internet Message Access Protocol, cuya versión más actual es la denominada IMAP4).

Comandos del POP3:

USER [nombre] = Permite identificar a un usuario. [nombre] será una cadena que identificará un buzón que solo tiene significado para el servidor.

PASS [cadena] = Permite al cliente indicar la contraseña de dicho usuario (del usuario especificado con anterioridad utilizando USER).

QUIT = Permite cerrar la conexión (en fase de autorización, es decir, cuando el usuario se está "loggeando"). Permite volver a la fase de autorización (en caso de estar en la fase de transacción, o sea, ya estando loggeado).

STAT = Permite obtener información acerca del buzón. El servidor devuelve la cantidad de mensajes y el tamaño total de los mismos.

LIST = Devuelve la lista de mensajes recibidos y el tamaño perteneciente a cada mensaje.

RETR [numeroDeMensaje] = Permite descargar el mensaje indicado como parámetro. La respuesta obtenida será el código de estado y luego el mensaje descargado (no se borra del Spool, solo se transmite).

DELE [numeroDeMensaje] = Suponiendo que descargué el mensaje y quiero eliminar el espacio ocupado en el spool, el comando DELE me permite eliminar mensajes dentro del servidor.

Ventajas del protocolo IMAP4 sobre POP3:

Una de las ventajas de IMAP4 por sobre POP3 es que, además de permitirme acceder al correo almacenado sobre el servidor y recuperar los mensajes, me permite administrar de mejor manera esos mensajes que están en el servidor (puedo armar carpetas y organizarlos), por lo que no hace falta que tenga las carpetas localmente, si no que puedo mantener las carpetas en el servidor.

6. ¿Para qué se definió la extensión MIME? Describa cómo se implementa y los diferentes tipos de contenidos y codificación MIME.

Originalmente, solo se permitían mensajes escritos utilizando un conjunto de caracteres US-ASCII de 7 bits.

La extensión MIME (Multipurpose Internet Mail Extensions) se definió para poder enviar mensajes no textuales, o sea, permite incluir imágenes, audios, videos, PDFs, etc.

Básicamente, MIME permite codificar los mensajes de alguna manera y, a la vez, indicarle al destino como tiene que interpretar esos mensajes.

Cuando el User Agent recibe dentro del cuerpo del mensaje (body) algo que no es texto, tiene que saber cómo representarlo (ya sea HTML, una imagen, etc.), si puede representarlo dentro del visualizador de correo electrónico o si tiene que invocar una aplicación externa.

¿Cómo sabe todo esto el User Agent? A través de nuevos encabezados definidos a través de la extensión MIME.

Los encabezados de MIME serán similares a los siguientes:

- MIME-Version: Indica que el mensaje se ha creado según el estándar MIME. Se decidió mantener la versión a 1.0 aunque se realizaron muchas actualizaciones a la versión de MIME.
- Content-Type: Indica el tipo de contenido: texto, audio, video.

Tipo	Descripción	Ejemplo de subtipos típicos
text	Representa cualquier documento que contenga texto y es teóricamente legible por humanos	text/plain, text/html, text/css, text/javascript
image	Representa cualquier tipo de imagen. Los videos no están incluidos, aunque las imágenes animadas (como el gif animado) se describen con un tipo de imagen.	image/gif, image/png, image/jpeg, image/bmp, image/webp
audio	Representa cualquier tipo de archivos de audio	audio/midi, audio/mpeg, audio/webm, audio/ogg, audio/wav
video	Representa cualquier tipo de archivos de video	video/webm, video/ogg
application	Representa cualquier tipo de datos binarios.	application/octet-stream, application/pkcs12, application/vnd.ms-powerpoint, application/xhtml+xml, application/xml, application/pdf

- Content-Transfer-Encoding: Indica que transformación se ha aplicado.

7bit	The data are all represented by short lines of ASCII characters.
8bit	The lines are short, but there may be non-ASCII characters (octets with the high-order bit set).
binary	Not only may non-ASCII characters be present, but the lines are not necessarily short enough for SMTP transport.
quoted-printable	Encodes the data in such a way that if the data being encoded are mostly ASCII text, the encoded form of the data remains largely recognizable by humans.
base64	Encodes data by mapping 6-bit blocks of input to 8-bit blocks of output, all of which are printable ASCII characters.
x-token	A named nonstandard encoding.

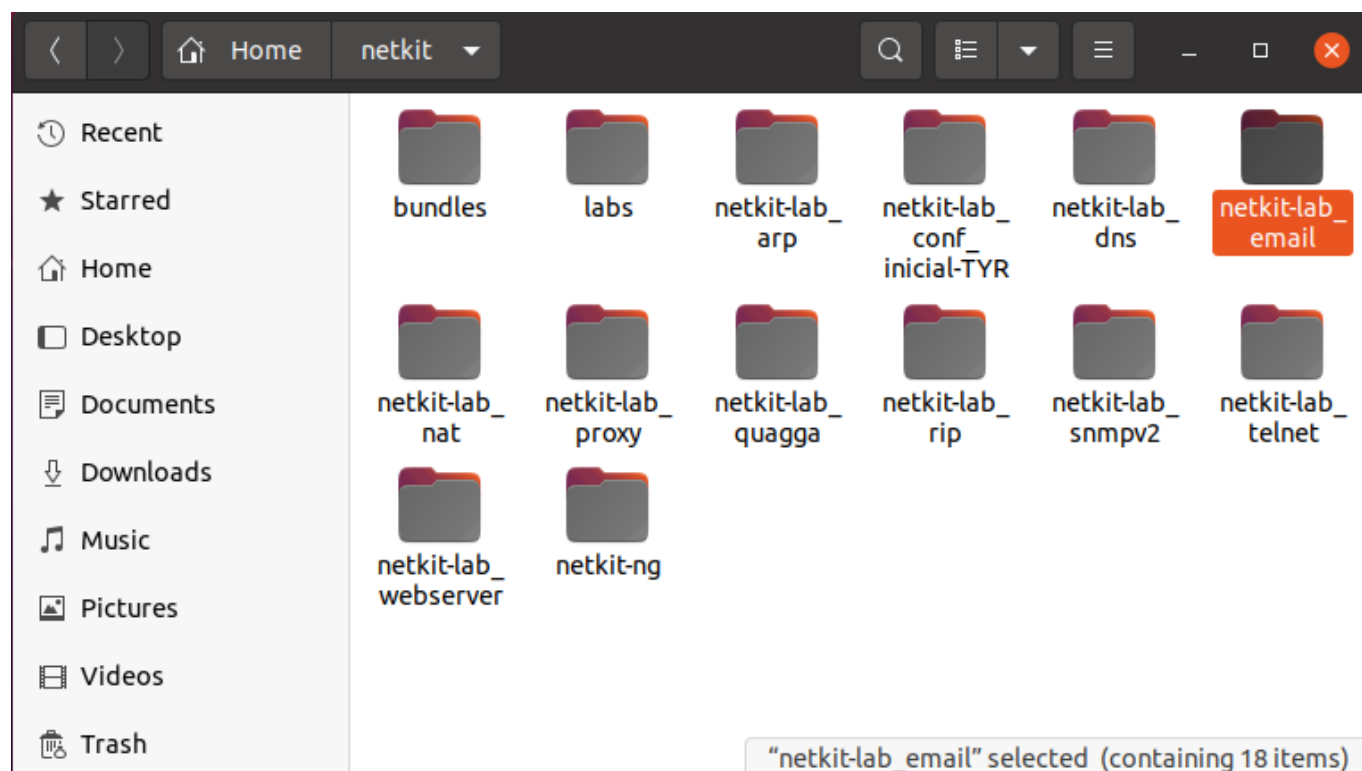
- Content-Id: Permite identificar entidades MIME en diversos contextos (generalmente es opcional).
- Content-Description: Permite asociar información que describa el cuerpo del mensaje.
- Content-Disposition: Permite indicar al UA receptor como representar los contenidos.

Aclaración: Opté por adjuntar imágenes en lugar de nombrarlos ya que me parece que queda mucho más prolijo y presentable.

7. Instale e inicie en el entorno netkit el laboratorio de email provisto por los docentes, disponible en https://github.com/redesunlu/netkit-labs/raw/master/tarballs/netkit-lab_email.tar.gz y realice las siguientes actividades:

Bien, para la resolución del punto 7, primero ingresé al link nombrado en la consigna.

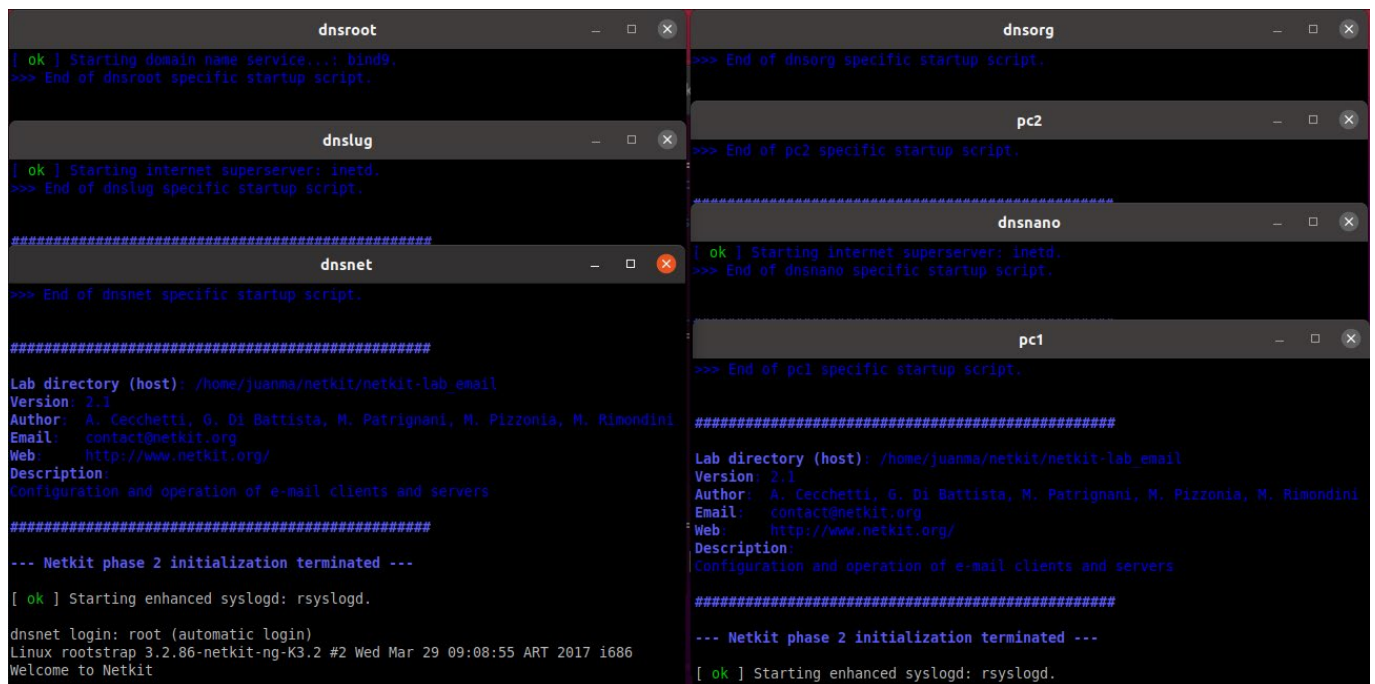
Una vez finalizada la descarga, opté por extraer el laboratorio en la carpeta de netkit, en donde se encuentra el resto de los laboratorios:



Ahora, para inicializarlo, me posiciono en su carpeta desde la terminal y utilizo el comando:

`lstart`

Obteniendo el siguiente resultado :



```
dnsroot
[ ok ] Starting domain name service...: bind9.
>>> End of dnsroot specific startup script.

dnslug
[ ok ] Starting internet superserver: inetd.
>>> End of dnslug specific startup script.

dnsnet
>>> End of dnsnet specific startup script.

Lab directory (host): /home/juanma/netkit/netkit-lab_email
Version: 2.1
Author: A. Cecchetti, G. Di Battista, M. Patrignani, M. Pizzonia, M. Rimondini
Email: contact@netkit.org
Web: http://www.netkit.org/
Description:
Configuration and operation of e-mail clients and servers
--- Netkit phase 2 initialization terminated ---
[ ok ] Starting enhanced syslogd: rsyslogd.
dnsnet login: root (automatic login)
Linux rootstrap 3.2.86-netkit-ng-K3.2 #2 Wed Mar 29 09:08:55 ART 2017 i686
Welcome to Netkit

dnsorg
>>> End of dnsorg specific startup script.

pc2
>>> End of pc2 specific startup script.

dnsgano
[ ok ] Starting internet superserver: inetd.
>>> End of dnsgano specific startup script.

pc1
>>> End of pc1 specific startup script.

Lab directory (host): /home/juanma/netkit/netkit-lab_email
Version: 2.1
Author: A. Cecchetti, G. Di Battista, M. Patrignani, M. Pizzonia, M. Rimondini
Email: contact@netkit.org
Web: http://www.netkit.org/
Description:
Configuration and operation of e-mail clients and servers
--- Netkit phase 2 initialization terminated ---
[ ok ] Starting enhanced syslogd: rsyslogd.
```

Como se puede apreciar en la imagen, el laboratorio de email cuenta con 7 máquinas virtuales:

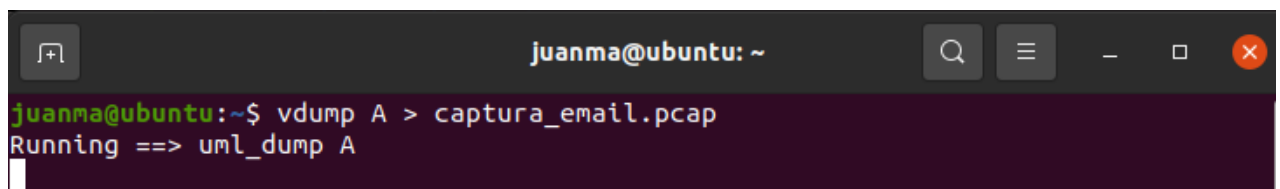
- dnsnet
- dnslug
- dnsroot
- dnsorg
- dnsgano
- pc1
- pc2

Las mismas máquinas virtuales que tenía el laboratorio de DNS, con el cual se trabajó en el TP3.

a. Inicie una captura desde el host.

Para iniciar la captura, utilizo el comando:

`vdump A > captura_email.pcap`



```
juanma@ubuntu: ~
juanma@ubuntu:~$ vdump A > captura_email.pcap
Running ==> uml_dump A
```

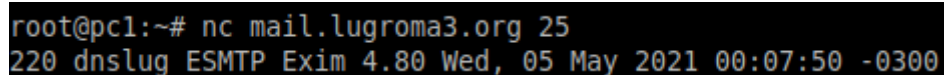
b. Desde la pc1, utilizando nc , conéctese al servidor SMTP mail.lugroma3.org (TCP puerto 25) y envíe un mensaje cuyo remitente sea su-nombre@lugroma3.org destinado a la cuenta de correo guest@nanoinside.net.

- Indique en el encabezado Subject: “Resolución del ejercicio 8”. Escriba un cuerpo de mensaje de al menos 3 líneas, incluyendo su nombre y su legajo.

- **Finalice el mensaje escribiendo un punto en una línea en blanco. Deberá ver la respuesta 250 OK id=... indicando que el mensaje fue procesado correctamente.**

Para conectarme al servidor SMTP mail.lugroma3.org (TCP puerto 25), debo utilizar el comando:

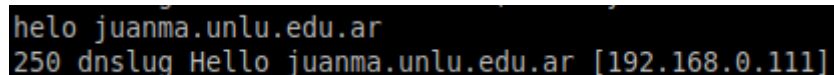
```
nc mail.lugroma3.org 25
```



```
root@pc1:~# nc mail.lugroma3.org 25
220 dnslug ESMTExim 4.80 Wed, 05 May 2021 00:07:50 -0300
```

Como se aprecia en la imagen, se recibe una respuesta exitosa (código 220)

Ahora, para iniciar sesión con el nombre del ordenador, utilizo el comando HELO seguido del nombre, en mi caso juanma.unlu.edu.ar (ejemplo)



```
helo juanma.unlu.edu.ar
250 dnslug Hello juanma.unlu.edu.ar [192.168.0.111]
```

Ahora, una vez recibida la respuesta exitosa (código 250), procedo a definir el remitente y el destino del mensaje.

Para esto utilizo:

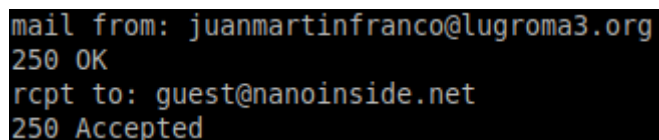
```
mail from: [remitente]
```

```
rcpt to: [destino]
```

En este caso:

```
mail from: juanmartinfranco@lugroma3.org
```

```
rcpt to: guest@nanoinside.net
```



```
mail from: juanmartinfranco@lugroma3.org
250 OK
rcpt to: guest@nanoinside.net
250 Accepted
```

Una vez definido esto, utilizo el comando:

```
data
```

Hecho esto, recibo una respuesta la cual me notifica de que estoy listo para escribir el mensaje, y que para finalizar el mismo debo escribir un "." en una línea nueva.

Ahora, ingreso lo siguiente:

From: "Juan Martin Franco" <juanmartinfranco@lugroma3.org>

To: "TyR" <guest@nanoinside.net>

Subject: Resolución del ejercicio 8

Probando desde pc1

Juan Martín Franco

Legajo: 149615

Este es el fin de mi mensaje.


```
pc1
root@pc1:~# nc mail.lugroma3.org 25
220 dnslug ESMTP Exim 4.80 Wed, 05 May 2021 00:13:04 -0300
helo juanma.unlu.edu.ar
250 dnslug Hello juanma.unlu.edu.ar [192.168.0.111]
mail from: juanmartinfranco@lugroma3.org
250 OK
rcpt to: guest@nanoinside.net
250 Accepted
data
354 Enter message, ending with "." on a line by itself
From: "Juan Martin Franco" <juanmartinfranco@lugroma3.org>
To: "TyR" <guest@nanoinside.net>
Subject: Resolucion del ejercicio 8
Probando desde pc1
Juan Martin Franco
Legajo: 149615
Este es el fin de mi mensaje.
.
250 OK id=1le810-0000Q3-Vm
```

Una vez ingresado todo, me posiciono en una línea vacía e ingreso un . para finalizar el mensaje, recibiendo como respuesta un código 250 OK.

- c. Desde la pc2, utilizando nc , conéctese al servidor POP3 pop.nanoinside.net (TCP puerto 110). Acceda a la cuenta de usuario guest (contraseña guest), recupere el mensaje almacenado en la casilla, bórralo y finalice adecuadamente la sesión POP.

Para conectarme al servidor POP3 pop.nanoinside.net en el puerto 110, utilicé el siguiente comando:

```
nc pop.nanoinside.net 110
```

Recibiendo un +OK como respuesta.

Luego, procedí a acceder como guest utilizando los siguientes comandos:

```
USER guest
```

```
PASS guest
```

Una vez aceptado mi login, recibo la respuesta:

```
+OK Mailbox open, 1 messages
```

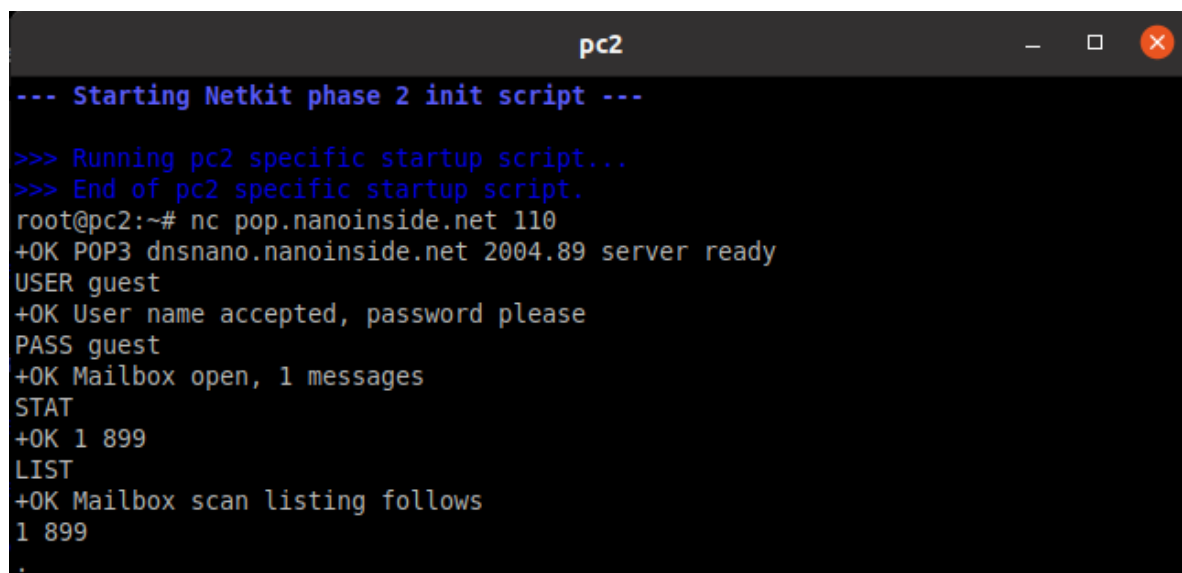
Lo que me indica que tengo 1 mensaje en el buzón.

Luego, simplemente para probar utilicé los comandos:

STAT, para ver la cantidad de mensajes y el tamaño total de los mismos.

LIST, para ver la lista de mensajes y el peso de c/u (en este caso solo había 1 mensaje).

Todo esto se puede apreciar en la siguiente imagen:



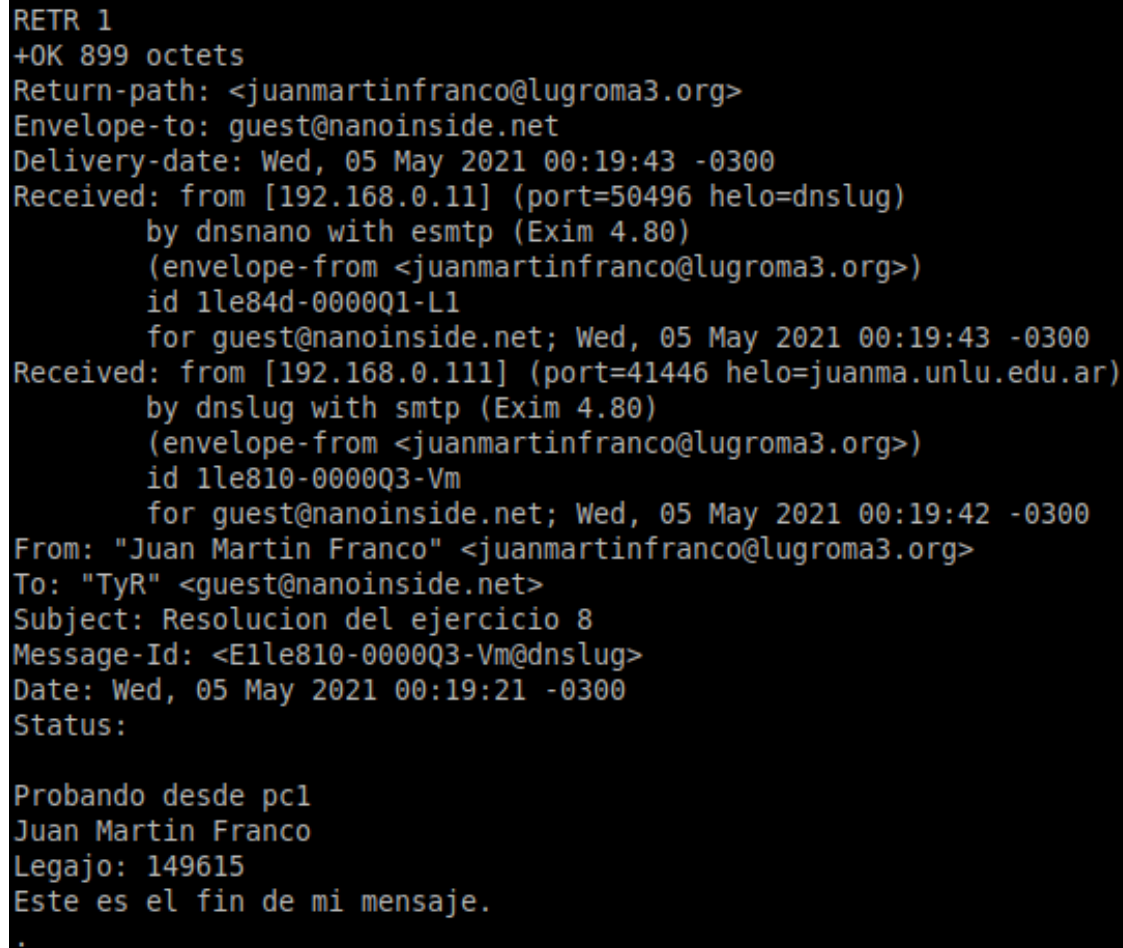
```
pc2
--- Starting Netkit phase 2 init script ---
>>> Running pc2 specific startup script...
>>> End of pc2 specific startup script.
root@pc2:~# nc pop.nanoinside.net 110
+OK POP3 dnsnano.nanoinside.net 2004.89 server ready
USER guest
+OK User name accepted, password please
PASS guest
+OK Mailbox open, 1 messages
STAT
+OK 1 899
LIST
+OK Mailbox scan listing follows
1 899
.
```

Ahora bien, como se ve en la imagen anterior, tengo 1 mensaje.

Para recuperarlo utilicé el comando:

RETR 1, siendo 1 el número de mensaje a recuperar.

En la siguiente imagen se muestra toda la información del mensaje, incluyendo destinatario y destino, fecha de envío, id, etc., seguido del cuerpo del mensaje.



```
RETR 1
+OK 899 octets
Return-path: <juanmartinfranco@lugroma3.org>
Envelope-to: guest@nanoinside.net
Delivery-date: Wed, 05 May 2021 00:19:43 -0300
Received: from [192.168.0.11] (port=50496 helo=dnslug)
    by dnsnano with esmtp (Exim 4.80)
    (envelope-from <juanmartinfranco@lugroma3.org>)
    id 1le84d-0000Q1-L1
    for guest@nanoinside.net; Wed, 05 May 2021 00:19:43 -0300
Received: from [192.168.0.111] (port=41446 helo=juanma.unlu.edu.ar)
    by dnslug with smtp (Exim 4.80)
    (envelope-from <juanmartinfranco@lugroma3.org>)
    id 1le810-0000Q3-Vm
    for guest@nanoinside.net; Wed, 05 May 2021 00:19:42 -0300
From: "Juan Martin Franco" <juanmartinfranco@lugroma3.org>
To: "TyR" <guest@nanoinside.net>
Subject: Resolucion del ejercicio 8
Message-Id: <E1le810-0000Q3-Vm@dnslug>
Date: Wed, 05 May 2021 00:19:21 -0300
Status:

Probando desde pc1
Juan Martin Franco
Legajo: 149615
Este es el fin de mi mensaje.
.
```

Una vez descargado y leído el mensaje, procedo a eliminarlo utilizando el comando:

DELE 1, siendo 1 el número de mensaje a eliminar.

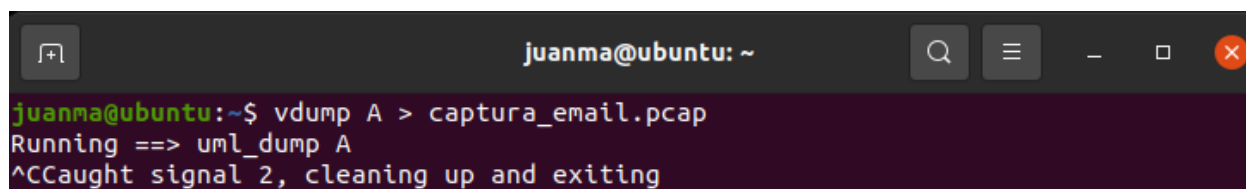
Por último, finalizo la sesión utilizando el comando:

QUIT

```
DELE 1
+OK Message deleted
QUIT
+OK Sayonara
```

d. Detenga el proceso de captura en el host.

Para detener el proceso de captura, simplemente me posiciono en la terminal donde la inicié y presiono la combinación de teclas CTRL + C.



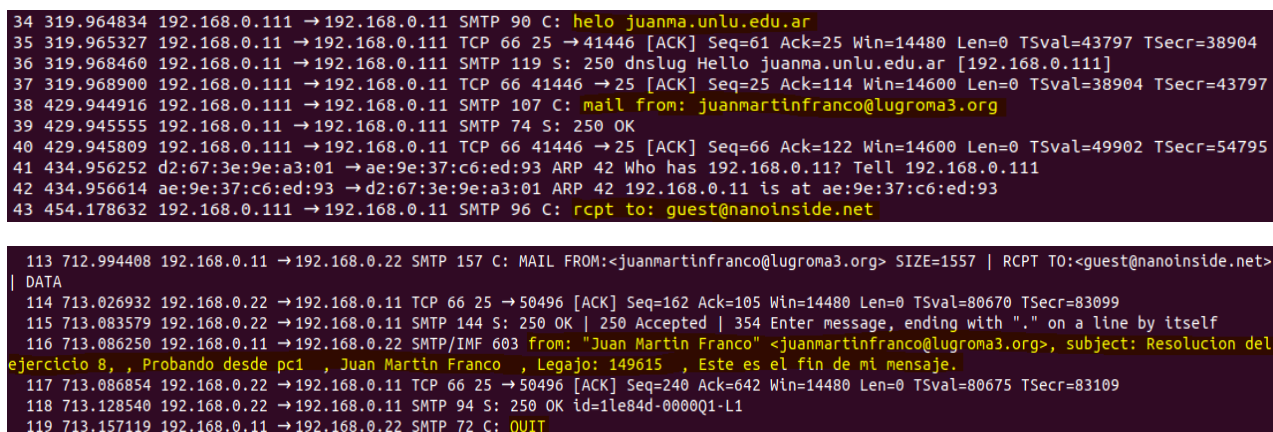
```
juanma@ubuntu: ~
juanma@ubuntu:~$ vdump A > captura_email.pcap
Running ==> uml_dump A
^CCaught signal 2, cleaning up and exiting
```

e. Analice la captura y discuta acerca de la confidencialidad de los datos transmitidos.

Para analizar la captura puedo utilizar el comando:

```
tshark -r captura_email.pcap
```

En el resultado de la ejecución de dicho comando pude obtener ciertas tramas interesantes como por ejemplo:



```
34 319.964834 192.168.0.111 → 192.168.0.11 SMTP 90 C: helo juanma.unlu.edu.ar
35 319.965327 192.168.0.11 → 192.168.0.111 TCP 66 25 → 41446 [ACK] Seq=61 Ack=25 Win=14480 Len=0 TSval=43797 TSecr=38904
36 319.968460 192.168.0.11 → 192.168.0.111 SMTP 119 S: 250 dnslog Hello juanma.unlu.edu.ar [192.168.0.111]
37 319.968900 192.168.0.111 → 192.168.0.11 TCP 66 41446 → 25 [ACK] Seq=25 Ack=114 Win=14600 Len=0 TSval=38904 TSecr=43797
38 429.944916 192.168.0.111 → 192.168.0.11 SMTP 107 C: mail from: juanmartinfranco@lugroma3.org
39 429.945555 192.168.0.11 → 192.168.0.111 SMTP 74 S: 250 OK
40 429.945809 192.168.0.111 → 192.168.0.11 TCP 66 41446 → 25 [ACK] Seq=66 Ack=122 Win=14600 Len=0 TSval=49902 TSecr=54795
41 434.956252 d2:67:3e:9e:a3:01 → ae:9e:37:c6:ed:93 ARP 42 Who has 192.168.0.11? Tell 192.168.0.111
42 434.956614 ae:9e:37:c6:ed:93 → d2:67:3e:9e:a3:01 ARP 42 192.168.0.11 is at ae:9e:37:c6:ed:93
43 454.178632 192.168.0.111 → 192.168.0.11 SMTP 96 C: rcpt to: guest@nanoinside.net

113 712.994408 192.168.0.11 → 192.168.0.22 SMTP 157 C: MAIL FROM:<juanmartinfranco@lugroma3.org> SIZE=1557 | RCPT TO:<guest@nanoinside.net>
| DATA
114 713.026932 192.168.0.22 → 192.168.0.11 TCP 66 25 → 50496 [ACK] Seq=162 Ack=105 Win=14480 Len=0 TSval=80670 TSecr=83099
115 713.083579 192.168.0.22 → 192.168.0.11 SMTP 144 S: 250 OK | 250 Accepted | 354 Enter message, ending with "." on a line by itself
116 713.086250 192.168.0.11 → 192.168.0.22 SMTP/IMF 603 from: "Juan Martin Franco" <juanmartinfranco@lugroma3.org>, subject: Resolucion del
ejercicio 8, , Probando desde pc1 , Juan Martin Franco , Legajo: 149615 , Este es el fin de mi mensaje.
117 713.086854 192.168.0.22 → 192.168.0.11 TCP 66 25 → 50496 [ACK] Seq=240 Ack=642 Win=14480 Len=0 TSval=80675 TSecr=83109
118 713.128540 192.168.0.22 → 192.168.0.11 SMTP 94 S: 250 OK id=1le84d-0000Q1-L1
119 713.157119 192.168.0.11 → 192.168.0.22 SMTP 72 C: QUIT
```

En ambas tramas, se pueden visualizar tanto los comandos ejecutados como además el cuerpo del mensaje, lo que significa que los datos pueden ser visualizados por cualquier persona que esté “sniffeando” la red, quedando expuesto a cualquier fuga de información.

Como conclusión se puede decir que los datos transmitidos no son confidenciales, quedando expuestos a cualquier persona que esté capturando el tráfico en el momento del intercambio de mensajes.

- f. Identifique la conexión TCP que se establece entre los MTA's. Utilice tshark para mostrar el contenido de dicho stream y adjúntelo.

Para visualizar el contenido del stream utilizo el comando:

```
tshark -r captura_email.pcap -nqz follow,tcp,hex,2
```

```

juanma@ubuntu:~/Desktop$ tshark -r captura_email.pcap -nqz follow,tcp,hex,2
=====
Follow: tcp,hex
Filter: tcp.stream eq 2
Node 0: 192.168.0.111:41446
Node 1: 192.168.0.11:25
00000000 32 32 30 20 64 6e 73 6c 75 67 20 45 53 4d 54 50 220 dnsl ug ESMTP
00000010 20 45 78 69 6d 20 34 2e 38 30 20 57 65 64 2c 20 Exim 4. 80 Wed,
00000020 30 35 20 4d 61 79 20 32 30 32 31 20 30 30 3a 31 05 May 2 021 00:1
00000030 33 3a 30 34 20 2d 30 33 30 30 0d 0a 3:04 -03 00..
00000000 68 65 6c 6f 20 6a 75 61 6e 6d 61 2e 75 6e 6c 75 helo jua nma.unlu
00000010 2e 65 64 75 2e 61 72 0a .edu.ar.
0000003C 32 35 30 20 64 6e 73 6c 75 67 20 48 65 6c 6c 6f 250 dnsl ug Hello
0000004C 20 6a 75 61 6e 6d 61 2e 75 6e 6c 75 2e 65 64 75 juanma. unlu.edu
0000005C 2e 61 72 20 5b 31 39 32 2e 31 36 38 2e 30 2e 31 .ar [192 .168.0.1
0000006C 31 31 5d 0d 0a 11]..
00000018 6d 61 69 6c 20 66 72 6f 6d 3a 20 6a 75 61 6e 6d mail fro m: juanm
00000028 61 72 74 69 6e 66 72 61 6e 63 6f 40 6c 75 67 72 artinfra nco@lugr
00000038 6f 6d 61 33 2e 6f 72 67 0a oma3.org .
00000071 32 35 30 20 4f 4b 0d 0a 250 OK..
00000041 72 63 70 74 20 74 6f 3a 20 67 75 65 73 74 40 6e rcpt to: guest@n
00000051 61 6e 6f 69 6e 73 69 64 65 2e 6e 65 74 0a anoinsid e.net.
00000079 32 35 30 20 41 63 63 65 70 74 65 64 0d 0a 250 Acce pted..
0000005F 64 61 74 61 0a data.
00000087 33 35 34 20 45 6e 74 65 72 20 6d 65 73 73 61 67 354 Ente r messag
00000097 65 2c 20 65 6e 64 69 6e 67 20 77 69 74 68 20 22 e, endin g with "
000000A7 2e 22 20 6f 6e 20 61 20 6c 69 6e 65 20 62 79 20 ." on a line by
000000B7 69 74 73 65 6c 66 0d 0a itself..
00000064 46 72 6f 6d 3a 20 22 4a 75 61 6e 20 4d 61 72 74 From: "J uan Mart
00000074 69 6e 20 46 72 61 6e 63 6f 22 20 3c 6a 75 61 6e in Franc o" <juan
00000084 6d 61 72 74 69 6e 66 72 61 6e 63 6f 40 6c 75 67 martinfr anco@lug
00000094 72 6f 6d 61 33 2e 6f 72 67 3e 0a roma3.or g>.
0000009F 54 6f 3a 20 22 54 79 52 22 20 3c 67 75 65 73 74 To: "TyR " <guest
000000AF 40 6e 61 6e 6f 69 6e 73 69 64 65 2e 6e 65 74 3e @nanoins ide.net>
000000BF 0a .
000000C0 53 75 62 6a 65 63 74 3a 20 52 65 73 6f 6c 75 63 Subject: Resoluc
000000D0 69 6f 6e 20 64 65 6c 20 65 6a 65 72 63 69 63 69 ion del ejercici
000000E0 6f 20 38 0a o 8.
000000E4 50 72 6f 62 61 6e 64 6f 20 64 65 73 64 65 20 70 Probando desde p
000000F4 63 31 0a cl.
000000F7 4a 75 61 6e 20 4d 61 72 74 69 6e 20 46 72 61 6e Juan Mar tin Fran
00000107 63 6f 0a co.
0000010A 4c 65 67 61 6a 6f 3a 20 31 34 39 36 31 35 0a Legajo: 149615.
00000119 45 73 74 65 20 65 73 20 65 6c 20 66 69 6e 20 64 Este es el fin d
00000129 65 20 6d 69 20 6d 65 6e 73 61 6a 65 2e 0a e mi men saje..
00000137 2e 0a ..
000000BF 32 35 30 20 4f 4b 20 69 64 3d 31 6c 65 38 31 30 250 OK i d=1le810
000000CF 2d 30 30 30 30 51 33 2d 56 6d 0d 0a -0000Q3- Vm..
000000DB 34 32 31 20 64 6e 73 6c 75 67 3a 20 53 4d 54 50 421 dnsl ug: SMTP
000000EB 20 63 6f 6d 6d 61 6e 64 20 74 69 6d 65 6f 75 74 command timeout
000000FB 20 2d 20 63 6c 6f 73 69 6e 67 20 63 6f 6e 6e 65 - closi ng conne
0000010B 63 74 69 6f 6e 0d 0a ction..
=====
juanma@ubuntu:~/Desktop$

```

Por ejemplo, se pueden visualizar los códigos de estado y la respuesta (220 dnslug ESMTP).

Luego, se puede visualizar los comandos enviados por mi (Seguido de la fecha y hora en la que se estableció la conexión, se ve el comando "helo juanma.unlu.edu.ar").

g. ¿Qué cosas adicionó al mensaje original el servidor mail.lugroma3.org?

El servidor mail.lugroma3.org adicionó al mensaje original el campo Message-Id, la fecha (el campo Date) y un servidor SMTP por el que pasó el mensaje (en este caso el resolver, cuya IP es 192.168.0.11, se puede visualizar en el mensaje por la presencia de "Received : from").

8. Utilizando el comando nc -C (el parámetro -C es requerido para este ejercicio), conéctese al servidor SMTP smtp.ethereal.email (puerto 25) y efectúe toda la transacción SMTP necesaria para enviar un mensaje a la dirección de correo jaiden.sipes59@ethereal.email.

Como remitente del mensaje utilice su propia cuenta de correo y como Asunto (Subject) especifique su nombre completo y legajo. Todo el mensaje debe cumplir con los requisitos de la RFC 5322 y ser de tipo MIME text/plain. Dentro del cuerpo del mensaje responda cuáles son los campos de encabezado obligatorios según RFC5322.

Como resolución de este ejercicio, copie y pegue los comandos enviados y las respuestas recibidas desde el servidor (es decir, toda la transacción efectuada).

Bien, para comenzar con la resolución de este ejercicio, primero debo conectarme al servidor SMTP smtp.ethereal.mail.

Para realizar esto, utilizo el comando:

```
nc -C smtp.ethereal.email 25
```

Una vez recibida la respuesta por parte del servidor (código 220), procedo a iniciar sesión con el comando

```
helo juanma.unlu.edu.ar
```

Como se aprecia en la imagen, obtengo una respuesta exitosa (código 250).

Ahora, debo especificar el remitente y el receptor del mail, utilizando los comandos:

```
mail from: juanmartin_franco@hotmail.com (para especificar remitente)
```

```
rcpt to: jaiden.sipes59@ethereal.mail (para especificar receptor)
```

Ahora, para empezar a especificar el cuerpo del mensaje, utilizo el comando:

```
data
```

Y luego, escribo lo siguiente:

```
MIME-Version: 1.0
```

```
Content-Type: text/plain
```

```
From: Juan Martin Franco <juanmartin_franco@hotmail.com>
```

```
To: Jaiden Sipes <jaiden.sipes59@ethereal.email>
```

```
Subject: Juan Martin Franco 149615
```

```
Date: Sun, 09 May 2021 17:59 -0300
```

Los campos de encabezado obligatorios según RFC 5322 son MIME-Version, From, To, Date, Subject

(Content-Type no, ya que si no se especifica su valor por defecto es text/plain)

Luego, para finalizar el envío del email, dejo una línea en blanco y luego un ".",

```
juanma@ubuntu:~$ nc -C smtp.ethereal.email 25
220 mx.ethereal.email ESMTP Ethereal MX
helo juanma.unlu.edu.ar
250 mx.ethereal.email Hello 6-110-136-186.fibertel.com.ar [186.136.110.6]Haraka is at your service.
mail from: juanmartin_franco@hotmail.com
250 sender <juanmartin_franco@hotmail.com> OK
rcpt to: jaiden.sipes59@ethereal.email
250 recipient <jaiden.sipes59@ethereal.email> OK
data
354 go ahead, make my day
MIME-Version: 1.0
Content-Type: text/plain
From: Juan Martin Franco <juanmartin_franco@hotmail.com>
To: Jaiden Sipes <jaiden.sipes59@ethereal.email>
Subject: Juan Martin Franco 149615
Date: Sun, 09 May 2021 17:59 -0300
Los campos de encabezado obligatorios según RFC 5322 son MIME-Version, From, To, Date, Subject
(Content-Type no, ya que si no se especifica su valor por defecto es text/plain)
.
250 Message processed (CDCFE63E-FB5C-48B0-A19E-72341895B086.1)
```

Por último, para cerrar la conexión utilizo el comando:

quit

```
quit
221 mx.ethereal.email closing connection. Have a jolly good day.
```

Copia de toda la transacción efectuada (sin explicación de por medio):

```
juanma@ubuntu:~$ nc -C smtp.ethereal.email 25
220 mx.ethereal.email ESMTP Ethereal MX
helo juanma.unlu.edu.ar
250 mx.ethereal.email Hello 6-110-136-186.fibertel.com.ar [186.136.110.6]Haraka is at your
service.
mail from: juanmartin_franco@hotmail.com
250 sender <juanmartin_franco@hotmail.com> OK
rcpt to: jaiden.sipes59@ethereal.email
250 recipient <jaiden.sipes59@ethereal.email> OK
data
354 go ahead, make my day
MIME-Version: 1.0
Content-Type: text/plain
From: Juan Martin Franco <juanmartin_franco@hotmail.com>
```


To: Jaiden Sipes <jaiden.sipes59@ethereal.email>

Subject: Juan Martin Franco 149615

Date: Sun, 09 May 2021 17:59 -0300

Los campos de encabezado obligatorios según RFC 5322 son MIME-Version, From, To, Date, Subject

(Content-Type no, ya que si no se especifica su valor por defecto es text/plain)

250 Message processed (CDCFE63E-FB5C-48B0-A19E-72341895B086.1)

quit

221 mx.ethereal.email closing connection. Have a jolly good day.

9. Seleccione un mensaje dentro de la carpeta SPAM de su casilla de correo y, utilizando el menú "...", descargue el código RFC 822 del mismo (en Gmail corresponde a la opción Mostrar original, en Outlook a Ver origen del mensaje, en Yahoo a Ver mensaje original, etc.). Analice los encabezados del mensaje e indique:

- **La semántica y el valor de los campos de encabezado vistos en clase (From, To, CC, Date, Subject, Reply-To, MIME-Version, Content-Type).**

From: DataCamp <team@datacamp.com>

To: juanmartin_franco@hotmail.com

CC: [No posee]

Date: Thu, 22 Apr 2021 14:13:37 +0000

Subject: Congrats: Free Unlimited Access through April 30!

Reply-To: [No posee]

Content-Type: multipart/alternative; boundary="80127ae883014310a7686f4b69e86cca"

MIME-Version: 1.0

- **El valor del campo Return-Path y si coincide con el valor del campo From.**

Return-Path: postmaster@cio33540.datacamp.com

No coincide con el valor del campo From.

- **La lista de servidores SMTP por los que fue pasando el mensaje (encabezados que comienzan con Received: from), la hora en la que pasó por cada uno de ellos y qué protocolo se utilizó en la transferencia (indicado por with ...).**

Received: from m44-9.mailgun.net (69.72.44.9) by BN3NAM01FT012.mail.protection.outlook.com (10.152.67.126) with Microsoft SMTP Server (version=TLS1_2, cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.4065.21 via Frontend Transport;

FECHA Y HORA: Thu, 22 Apr 2021 14:13:39 +0000

Protocolo utilizado en la transferencia: Microsoft SMTP Server

Received: from BN3NAM01FT012.eop-nam01.prod.protection.outlook.com (10.152.66.58) by BN3NAM01HT055.eop-nam01.prod.protection.outlook.com (10.152.66.77) with Microsoft SMTP Server (version=TLS1_2, cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.4065.21;

FECHA Y HORA: Thu, 22 Apr 2021 14:13:40 +0000

Protocolo utilizado en la transferencia: Microsoft SMTP Server

Received: from BN3NAM01HT055.eop-nam01.prod.protection.outlook.com (2603:10b6:5:3b6::35) by DM6PR06MB5897.namprd06.prod.outlook.com with HTTPS via DS7PR03CA0210.NAMPRD03.PROD.OUTLOOK.COM;

FECHA Y HORA: Thu, 22 Apr 2021 14:13:41 +0000

Protocolo utilizado en la transferencia: HTTPS

• Si es MIME de tipo `_multipart/*_`, determinar para qué se utiliza el valor del dato `boundary`, cuantos bloques componen el mensaje, qué tipo de contenido (`Content-Type`) y qué codificación se utiliza (`Content-Transfer-Encoding`) en cada bloque.

El tipo de MIME es `mutipart/alternative`.

El valor del dato `boundary` se utiliza para delimitar los bloques de un email.

Para poner un ejemplo:

```
To: juanmartin_franco@hotmail.com
From: DataCamp <team@datacamp.com>
Subject: Congrats: Free Unlimited Access through April 30!
Content-Type: multipart/alternative; boundary="80127ae883014310a7686f4b69e86cca"
```

Es el código que se puede visualizar en el final del mail:

```
<img src=3D"https://links.datacamp.com/e/o/eyJlbWFpbF9pZCI6ImRnUGRod1RkaHdR=
REFBRjQtZTJwUHBTWVJQTY4RjFwOFd3PSJ9" style=3D"height: 1px !important; max-
height: 1px !important; max-width: 1px !important; width: 1px !important"> <=
/body> </html>=
--80127ae883014310a7686f4b69e86cca--
```

Por lo que se podría decir que el dato `boundary` es una especie de frontera que delimita cada bloque de correo.

El mensaje está compuesto por 3 bloques (identificados por la presencia del valor del campo `boundary`).

Primer bloque:

Content-Type: multipart/alternative

Content-Transfer-Encoding: 7BIT (Al no estar especificado, tomo su valor por defecto)

Content-Type: multipart/alternative; boundary="80127ae883014310a7686f4b69e86cca"

Segundo bloque:

Content-Type: text/plain

Content-Transfer-Encoding: quoted-printable

--80127ae883014310a7686f4b69e86cca

Content-Transfer-Encoding: quoted-printable

Content-Type: text/plain; charset="utf-8"

Tercer bloque:

Content-Type: text/html

Content-Transfer-Encoding: quoted-printable

--80127ae883014310a7686f4b69e86cca

Content-Transfer-Encoding: quoted-printable

Content-Type: text/html; charset="utf-8"