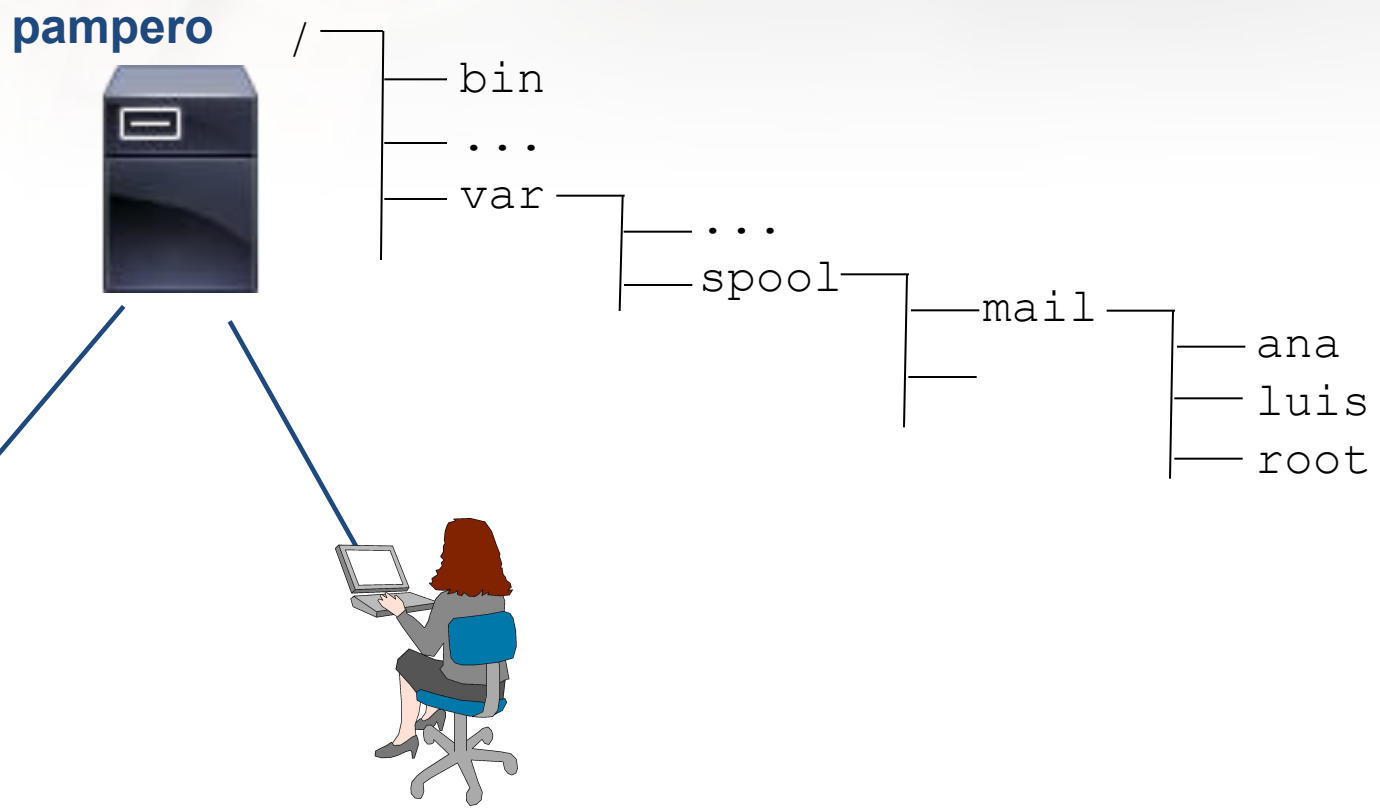
Correo electrónico

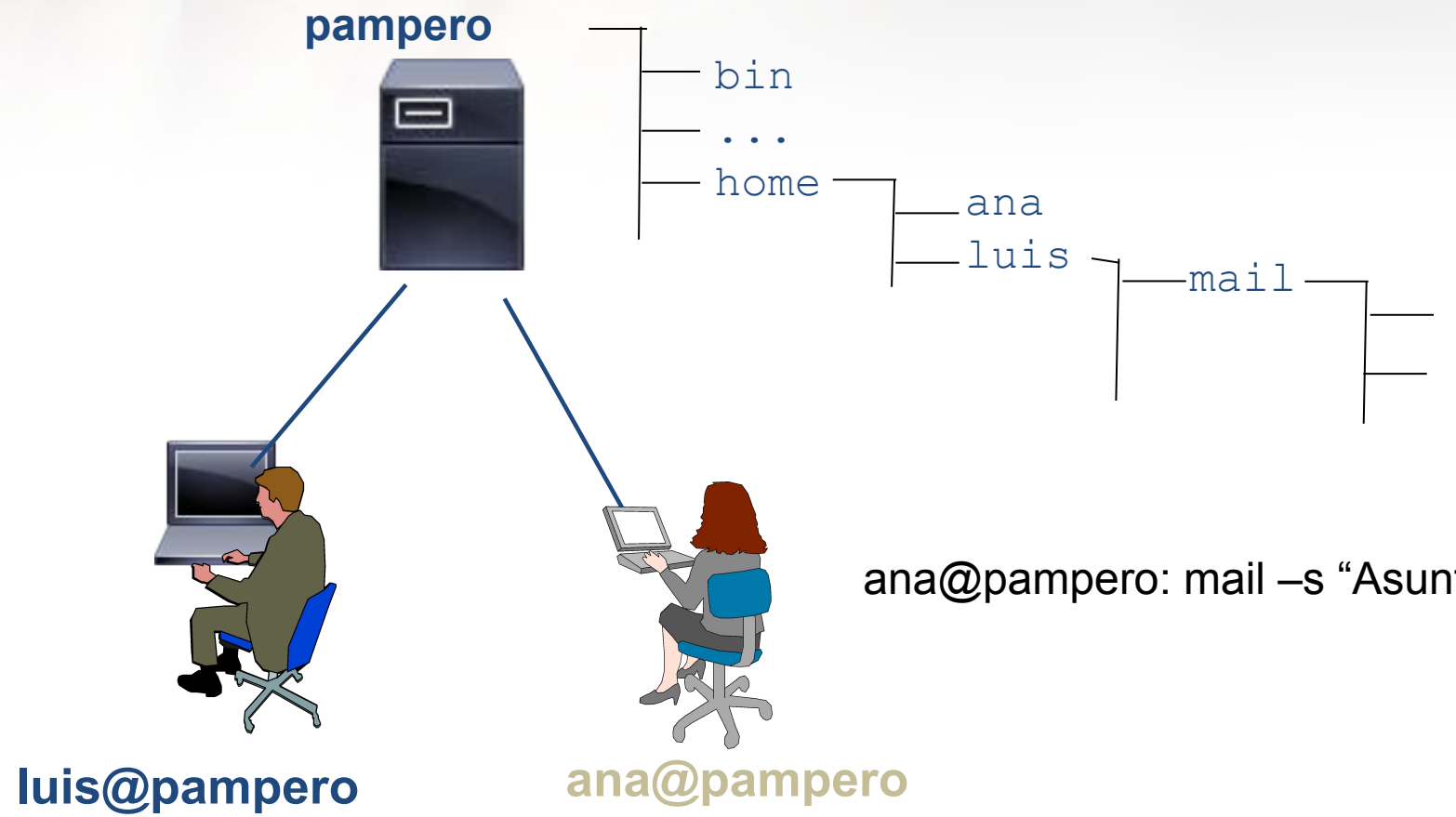
Email

En un sistema Unix/Linux, toda cuenta de usuario tiene también una cuenta de mail.



Email

En un sistema Unix/Linux, toda cuenta de usuario tiene también una cuenta de mail.



ana@pampero: mail –s “Asunto del mensaje” luis

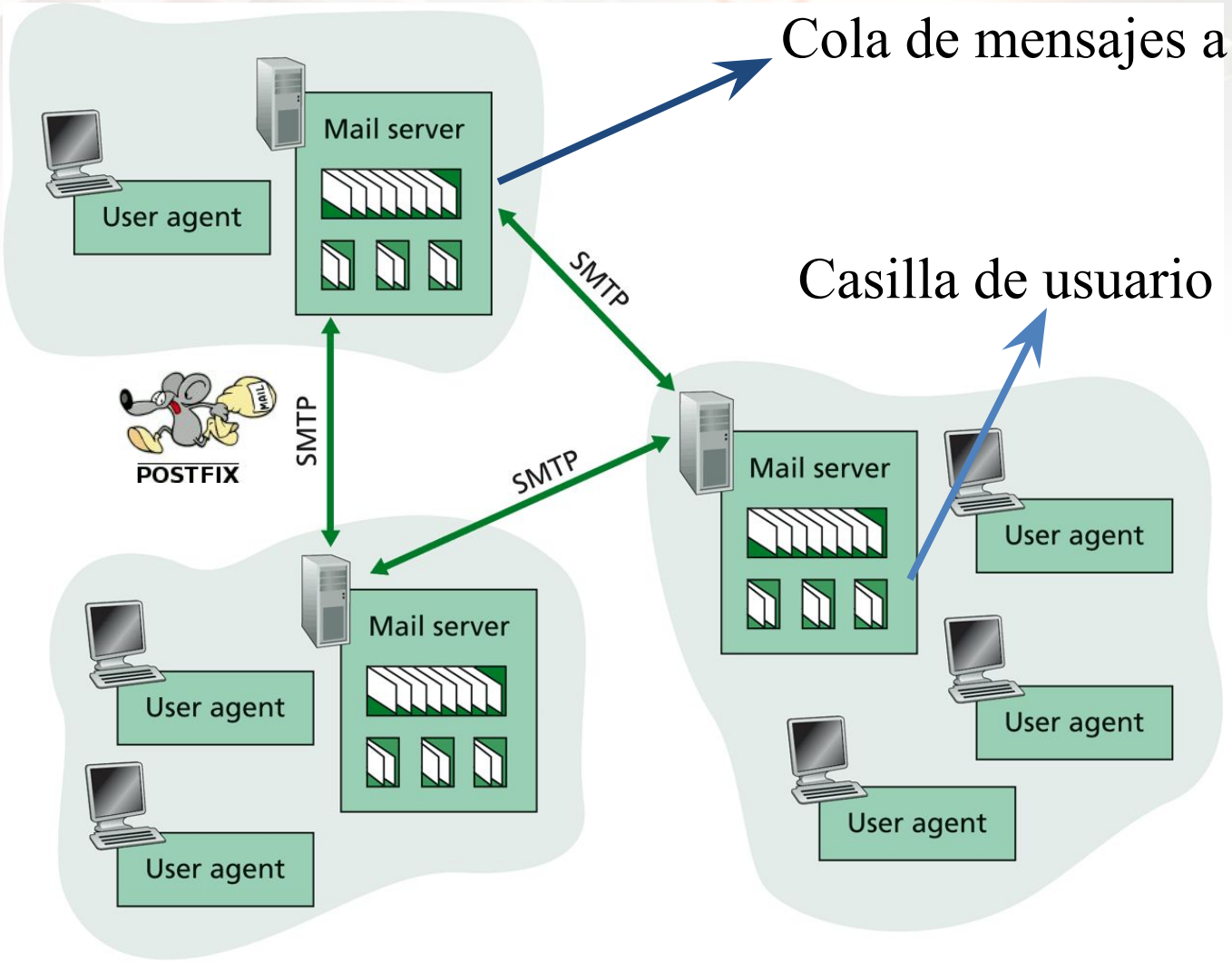
Email: ejemplos

Enviar y leer mails por línea de comando

```
$ mail –s "Subject" user
$ mail –a attachFile –s "Subject" user,user2
```

```
$ mail
"/var/spool/mail/mgarbe": 333 messages 332 new
> N 1 ……
? t 1
 texto del mensaje
? q
```

Cómo enviar mails de un host a otro



- **MTA**: Mail Transfer Agent
  - Transfiere el mensaje entre hosts
- **MUA**: Mail User Agent
  - permite al usuario leer y enviar mensajes
- **MDA**: Mail Delivery Agent
  - coloca el mensaje en la casilla de correo

SMTP

El protocolo por el cual se transmiten los mails por Internet es SMTP (Simple Mail Transport Protocol), definido en RFC821(1982) y RFC5321(2008).

From: luis@itba.edu.ar

conectarse a puerto SMTP

to: alumna@fibertel.com.ar

64.233.190.26

ASPMX.L.GOOGLE.COM

alt1.aspmx.l.google.com

alt2.aspmx.l.google.com

…

conectarse a puerto SMTP

mx1.fibertel.com.ar

SMTP

- Usa TCP para transferir mensajes desde el cliente al servidor (puerto 25)
- Transferencia directa: desde "sending server" a "receiving server"
- Tres fases
  - handshaking
  - transferencia de mensajes
  - cierre
- Interacción comando / respuesta
  - comando: texto ASCII
  - respuesta: código de status y comentario
- Los mensajes son US-ASCII: **ASCII-7 bits**

SMTP

Pasos en una transacción SMTP

MAIL <SP> FROM: <reverse-path> <CRLF>

250 OK

RCPT <SP> TO: <forward-path> <CRLF>

250 OK

DATA <CRLF>

354

…

.

<CRLF>

250 OK

SMTP: ejemplo

S: MAIL FROM:<Smith@Alpha.ARPA>

R: 250 OK

S: RCPT TO:<Jones@Beta.ARPA>

R: 250 OK

S: RCPT TO:<Green@Beta.ARPA>

R: 550 No such user here

S: RCPT TO:<Brown@Beta.ARPA>

R: 250 OK

S: DATA

R: 354 Start mail input; end with <CRLF>.<CRLF>

S: Blah blah blah...

S: <CRLF>.<CRLF>

R: 250 OK

Undeliverable mail notification

S: MAIL FROM:<>

R: 250 ok

S: RCPT TO:<@HOSTX.ARPA:JOE@HOSTW.ARPA>

R: 250 ok

S: DATA

R: 354 send the mail data, end with .

S: From: SMTP@HOSTY.ARPA

S: To: JOE@HOSTW.ARPA

S: Subject: Mail System Problem

S: Sorry JOE, your message to SAM@HOSTZ.ARPA lost.

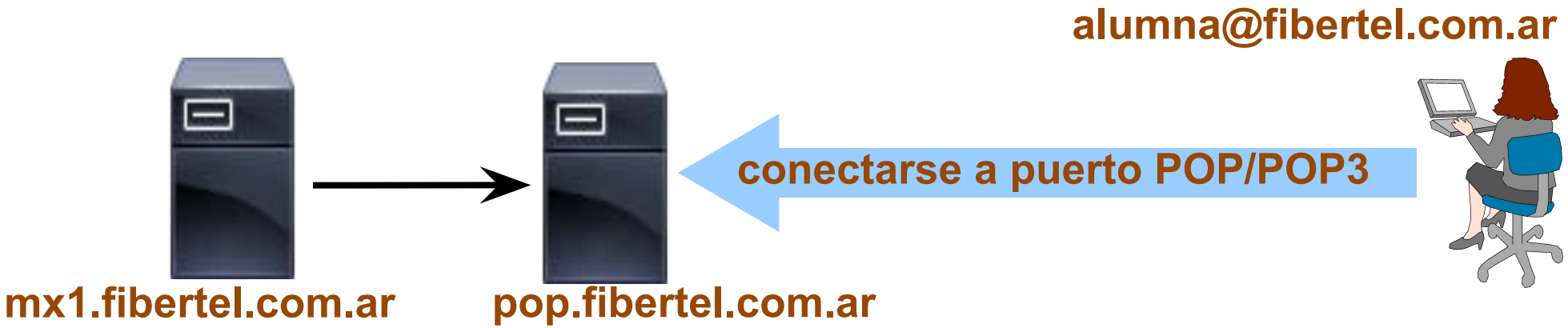
S: HOSTZ.ARPA said this: "550 No Such User"

S: .

R: 250 ok

Leer mail en forma remota

En caso de querer acceder a los mensajes en forma remota, el sistema tiene que tener habilitado algún **protocolo de entrega final de usuario**, por ejemplo POP2 (puerto 109), POP3 (puerto 110), IMAP (puerto 143) o IMAP3 (puerto 220).



Protocolos de entrega final a usuario

- POP3 (Post Office Protocol)
  - El objetivo de POP es obtener los mensajes del servidor y almacenarlos en el host local.
  - Existen versiones que permiten dejar una copia en el servidor.
- IMAP (Interactive Mail Access Protocol)
  - Pensado para que un usuario consulte su correo desde varios hosts
  - El servidor de correo mantiene un almacenamiento central accesible desde cualquier host.
  - El cliente IMAP no copia el correo en el host local
  - El cliente IMAP distingue si los mensajes ya han sido leídos
  - Permite crear carpetas en el servidor
  - El servidor realiza backups de los mails

POP3

Fase de autorizacion

- Comandos del cliente:
  - `user`: declara usuario
  - `pass`: password
- Respuestas del servidor
  - `+OK`
  - `-ERR`

Transacciones, cliente:

- `list`: lista mensajes
- `retr`: obtiene mensajes por numero
- `dele`: borra mensajes
- `quit`

```
S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on
```

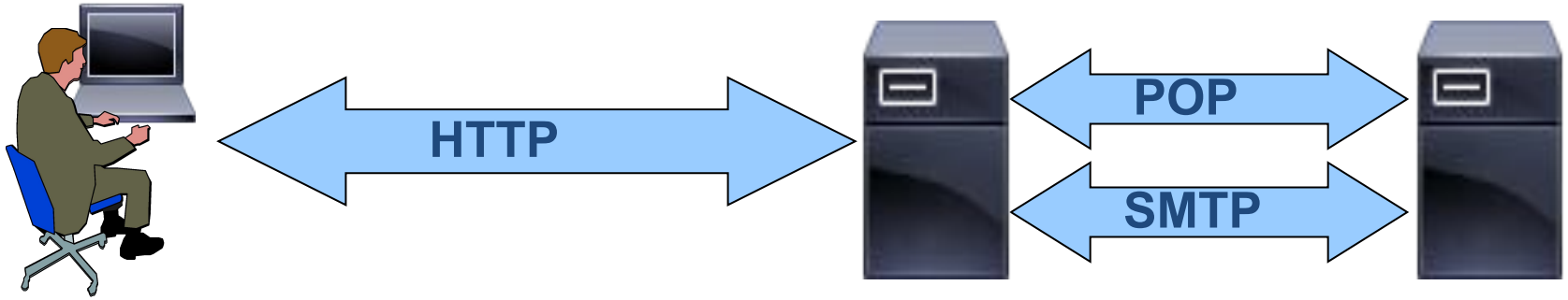
```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

POP3: comandos

Comando	Función
USER [*username*]	indicar el user name
PASS [*password*]	password del usuario
QUIT	cierra la conexión
STAT	retornar cantidad de mensajes y bytes
LIST	listar todos los mensajes
RETR *msgNumber*	pedir un mensaje
DELE *msgNumber*	marcar para borrar un mensaje
NOOP	
RSET	recuperar los mensajes marcados para borrado
TOP [*msg*] [*lines*]	header y primeras líneas de un mensaje

Webmail

Es un cliente de correo electrónico (MUA) con una interface web, que permite acceder al mailbox de un usuario.



IMAP: comandos

Comando	Función
LOGIN *user key*	Login
CAPABILITY	Lista las funcionalidades
LOGOUT	Cierra la conexión
SELECT *mailbox*	Selecciona un buzón
EXAMINE *mailbox*	Abre un mailbox como solo lectura
CREATE *mailbox*	Crea un nuevo buzón
DELETE *mailbox*	Elimina un buzón
RENAME *mbx1 mbx2*	Cambia el nombre de un buzón
CLOSE	Cierra el buzón y elimina los mensajes marcados para borrar
EXPUNGE	Elimina los mensajes marcados para borrar
FETCH *n mbx*	Muestra un mensaje del mailbox

Formato de un mensaje

El formato de un mail, según el RFC 822, consta de

- Una envoltura primitiva (ver RFC 821).
- Campos de cabecera (From: .., To: ..., etc.)
  (los usuarios pueden definir cabeceras que comiencen con X)
- Una línea en blanco
- Cuerpo del mensaje

El E-mail es probablemente la aplicación TCP/IP más usada. Sin embargo, SMTP y RFC 822 se limitan a texto ASCII de 7 bits con una longitud de línea máxima de 100 caracteres.

Encoding: Base64

- Método de codificación simple
- Cada grupo de 3 bytes es codificado como 4 bytes, cada uno conteniendo sólo 6 bits de datos
- Estos son enviados como “7-bit ASCII”
- Base64
  - 6 bits -> [0,63]
  - Se asigna un carácter a cada número

Encoding: Base64

Alfabeto Base64

0 A	17 R	34 i	51 z
1 B	18 S	35 j	52 0
2 C	19 T	36 k	53 1
3 D	20 U	37 l	54 2
4 E	21 V	38 m	55 3
5 F	22 W	39 n	56 4
6 G	23 X	40 o	57 5
7 H	24 Y	41 p	58 6
8 I	25 Z	42 q	59 7
9 J	26 a	43 r	60 8
10 K	27 b	44 s	61 9
11 L	28 c	45 t	62 +
12 M	29 d	46 u	63 /
13 N	30 e	47 v	
14 O	31 f	48 w	(pad) =
15 P	32 g	49 x	
16 Q	33 h	50 y	

Base64: ejemplo

3 bytes a codificar: 10101111 11001010 11101010

Stream de 24 bits: 101011111100101011101010

Agrupamos de a 6 bits:	**101011**	**111100**	**101011**	**101010**
Valor decimal	**43**	**60**	**43**	**42**
Caracter Base64	**r**	**8**	**r**	**q**
Se transmite:	**01110010**	**00111000**	**01110010**	**01110001**

Base64: relleno

¿Qué sucede si la cantidad de bits no es múltiplo de 6?

- Se agregan ceros al final hasta un múltiplo de 6.
- Uno o dos caracteres de relleno (‘=‘) son agregados para que sea múltiplo de 6 bytes.
- En general se agrega un solo carácter de relleno

Base64: relleno

4 bytes a codificar: 10101111 11001010 11101010 00100011

Stream de 32 bits: 10101111110010101110101000100011

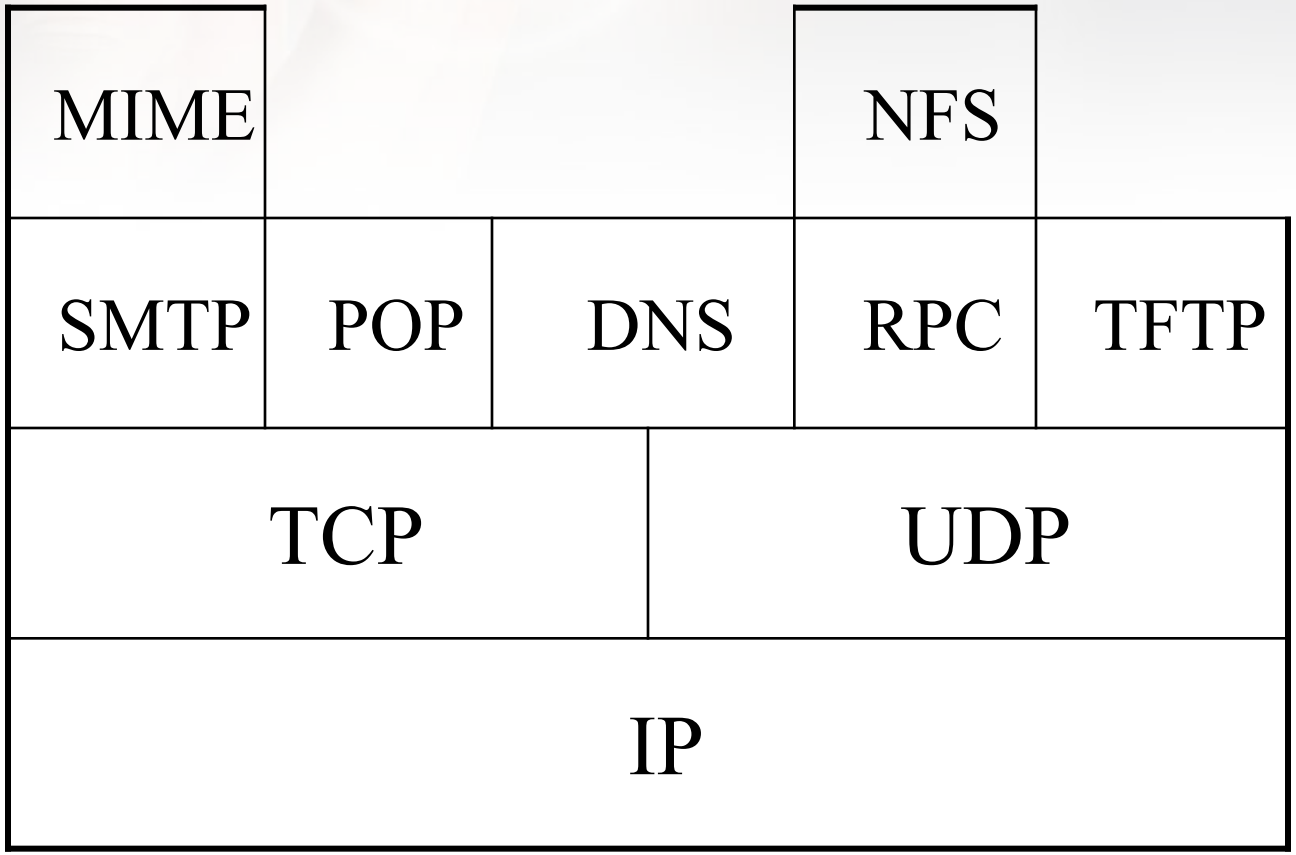
Agrupado de a 6 bits:	**101011**	**111100**	**101011**	**101010**	**001000**	**110000**
Valor decimal	**43**	**60**	**43**	**42**	**08**	**48**
Caracter Base64	**r**	**8**	**r**	**q**	**I**	**w**
Relleno	**r**	**8**	**r**	**q**	**I**	**w = =**

Ver https://www.base64encode.net

MIME (RFC1521 y RFC1522)

- SMTP no puede transmitir objetos binarios.
- SMTP no puede transmitir texto que incluya caracteres nacionales
- Los servidores SMTP pueden rechazar los mensajes que superen un tamaño concreto.
- Algunas implementaciones de SMTP u otros MTAs de Internet no respetan por completo el estándar SMTP. Algunos problemas son:
  - Eliminación de espacios al final de la línea
  - Relleno de todas las líneas de un mensaje para que tengan la misma longitud.
  - Conversión de caracteres TAB a múltiples caracteres SPACE.

MIME



MIME

MIME (*Multipurpose Internet Mail Extensions Encoding*) respeta el RFC 822. Agrega una estructura al cuerpo del mensaje, por lo que es transpartente para SMTP. Permite:

- Normalizar intercambio de diferentes tipos de contenido (texto simple, texto formateado, imágenes, sonido, video y documentos HTML).
- Solucionar el problema de enviar texto internacional por mail.

MIME: encabezados

- MIME-version: 1.0 (*comentario*)
- Content-type
- Content-Transfer-Encoding
  - 7 bits
  - 8 bits
  - binario
- Content-Description
- Content-ID

MIME:content-type

(https://www.w3.org/Protocols/rfc1341/0_TableOfContents.html)

`Content-Type:` *type/subtype ;parameter=value ;parameter=value*

Type	subtype
text	html, plain, richtext
multipart	mixed, alternative, digest
message	Partial, external-body
image	gif, jpeg, tiff
video	mpeg, quicktime
audio	basic, x-aiff, x-wav
application	pdf, rtf, postscript

MIME: ejemplo parcial

...lo dicho, Industrial and Commercial Bank of China (Argentina) S.A. no resulta responsable por dichas modificaciones ni por los eventuales perjuicios que tales circunstancias puedan ocasionar.

------=_NextPart_000_00A5_01C0F442.AAB436D0 Content-Type: application/pdf; name=" Adjunto_1_OP.pdf" Content-Disposition: attachment; filename="Adjunto_1_OP.pdf" Content-Transfer-Encoding: base64
JVBERi0xLjMKJaqrrK0KNCAwIG9iago8PCAvVHlwZSAvSW5mbwovUHJvZHVjZXIgKG51bGwpID4+
CmVuZG9iago1IDAgb2JqCjw8IC9MZW5ndGggMjQ3NiAvRmlsdGVyIC9GbGF0ZURlY29kZSAKID4+
CnN0cmVhbQp4nJ1bW3PaShJ+96/gMalaK3O/5E3G2IddG3sxJy/rfSCg+LBFkBdwqs7++u2RhGYE
uvScpCqR5U9fX6a7p3sEN4uryeLqv1dk9HbFVUJGWjD4VwoC/7qrfTb6cfXPq5sQJ+A+pzyRBUJw
AlctOKkVDsg9rl+wLhSkDl1wd+GkwxF6kkvbxQqQiMGVYgtYm9gvd3xELdxa/Lhyyi9WxVN0ROAv
HVECUrQeaSYTy8Vo8XP0r0+Pk9uHp/vPo2umDTzyKZ3fT2aL6Sz1t16SNPk8+vdo8fdQY1l7gKoe
T2FNkxrrKo9rE1z4gJQ+OBkuuEk4A7t1Ik52P81vJzNv4+3EXz+n909fK4O/3NELOslUolST7quy
jNbPtKjASMKkHGn4Re37u8n4t7Rdh8nj9GX6NOtVA5ZakzNKChGiE0YYOV+xgdhphqzip1TRpG9d
h3GF2BLWFbL6zFUk0XKkjA6CNFtv87cgSPdv2e642S1bg/SMSDNPNF5ut9nX8R+bbeYfNYS2PykD
Fcb5evOWB0GSH47L7VdKrPE3bz6yXX4I9Nzss0M7twi0CsxpzbLOYI9ejTLLMKvrcW2Ca2OouVis
8e/TRRi358sLESEoaz7DyTWEsWXG6GsRpNH5s5W4cEln+2BV1sGyTndv4P1wOW72H8f80KebNYnR
...

SPAM (https://www.youtube.com/watch?v=zLih-WQwBSc)

SPAM

1. Obtener direcciones de correo
2. Envío de mensajes
   - Troyanos, bots
   - Servidores SMTP que aceptan rely
3. Verificación de la recepción (a veces)

SPAM: cómo defenderse

- Evitarlos
  - ~~Listas negras y listas blancas~~ *Deny list* y *Allow list*
  - GreyListing
- Clasificarlos
  - Filtros Bayesianos

Autenticación de mails

Según Verizon el 30% de los mails de phishing se abren, y en hasta un 12% de ellos se accede a los enlaces o archivos maliciosos

- Hay dos técnicas principales para autenticar mails
  - SPF (Sender Policy Framework)
  - DKIM (DomainKeys Identified Mail)

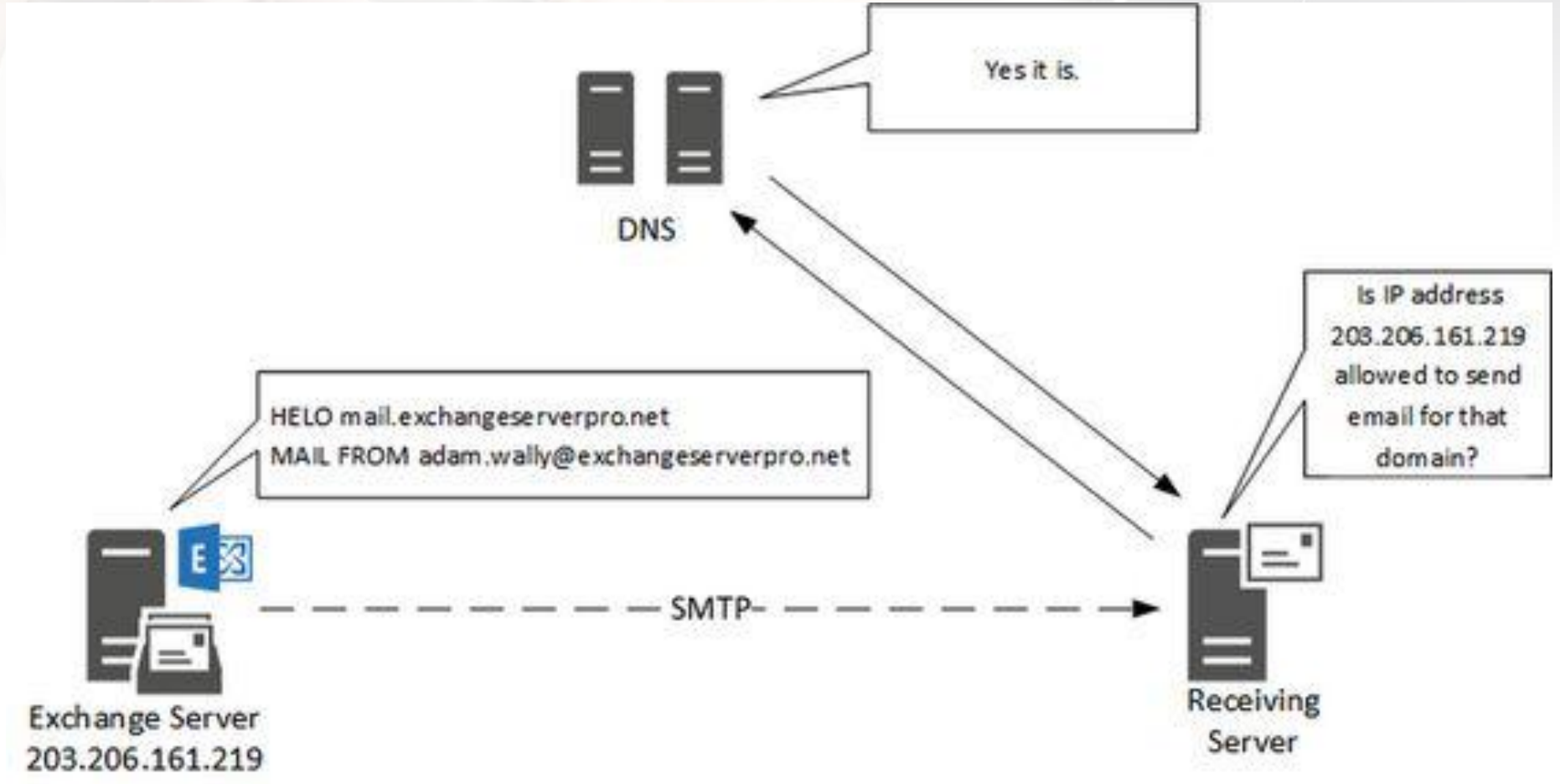
SPF

Permite detectar qué servidor o servidores tienen permiso para enviar mails en su nombre.

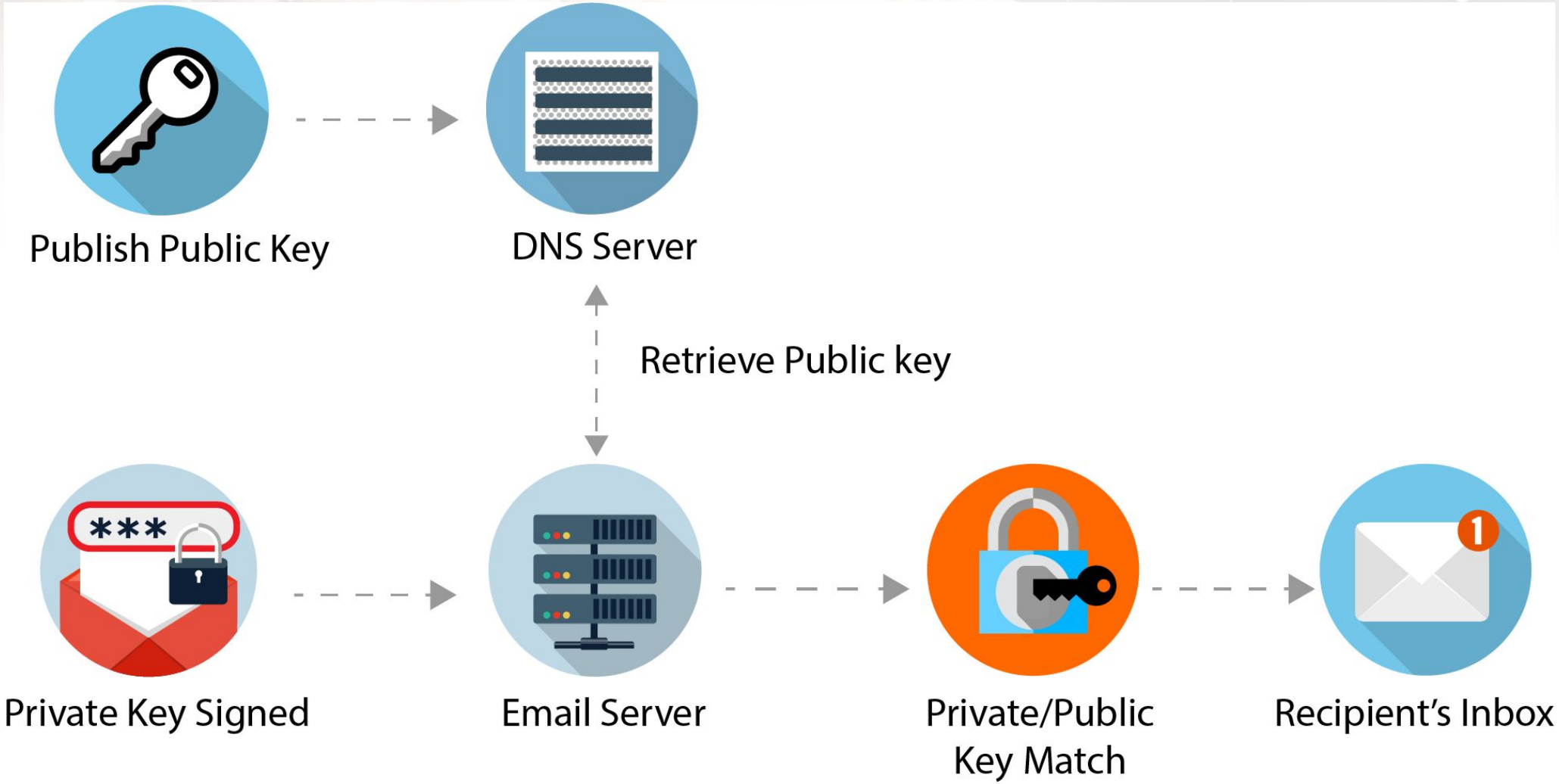
Su función es evitar que un mail sea enviado desde servidores que no estén relacionados con el dominio.

Se utilizan registros TXT y SPF en el servidor DNS

SPF


Yes it is.
DNS
Is IP address 203.206.161.219 allowed to send email for that domain?
HELO mail.exchangeserverpro.net
MAIL FROM adam.wally@exchangeserverpro.net
SMTP
Exchange Server 203.206.161.219
Receiving Server

DKIM


Publish Public Key
DNS Server
Retrieve Public key
***
Private Key Signed
Email Server
Private/Public
Key Match
1
Recipient's Inbox

Material de lectura

Capítulo 2.4 de la bibliografía

https://postmarkapp.com/guides/dkim

https://postmarkapp.com/guides/spf

https://www.sparkpost.com/resources/email-explained/dmarc-explained/