



# Servicio de Correo

T05 - SRI

# El servicio de correo electrónico

- El servicio de correo electrónico (email) es **uno de los más usados en Internet**.
- Permite el envío de **mensajes de entre usuarios de modo asíncrono**. El usuario puede no estar conectado a la red cuando el correo haya sido enviado, y recibirlo más tarde cuando se conecte.
- **Ha superado en gran medida al correo tradicional**, puesto que reduce los costes y agiliza las comunicaciones.
- Incluso puede sustituir al envío de **correo certificado**, ya que permite que el acuse de recibo certifique la recepción de un correo por un usuario.
- Se comenzó a utilizar **a partir de los años 60**, antes de que existiera Internet. En sus orígenes solo permitía mandar mensajes codificados en ASCII. Desde los años 90 se comenzaron a desarrollar los tipos **MIME**, para adjuntar **ficheros de varios tipos**.

# Ventajas frente al correo tradicional

- **Rapidez en la entrega:** Un mensaje de correo electrónico puede llegar a su destinatario prácticamente en el momento de ser enviado.
- **Coste económico:** Teniendo conexión a Internet se pueden enviar mensajes de correo electrónico sin ningún coste. Para las empresas supone un importante ahorro frente al envío tradicional de cartas a clientes, proveedores, trabajadores, etc.
- **Multidifusión:** Es posible enviar un correo a **varios destinatarios** a la vez.
- **Seguridad:** Se pueden establecer **medidas de seguridad** que eviten que puedan ser interpretados si son interceptados, que aseguran que **el receptor los ha leído**, y que aseguran la **identidad del autor del correo**.

# Elementos del servicio de correo electrónico

- **Mensaje:** Es la **información** que transmite el servicio de **correo electrónico**. Los mensajes son más conocidos como correos o **e-mails** y permiten adjuntar archivos de distintos tipos.
- **Cliente:** Es un programa de usuario que permite editar los mensajes, enviarlos hacia un servidor que se encargue de que lleguen al destinatario, o de descargar los recibidos desde un buzón de usuario.
- **Servidor:** Es un software que se encarga de **recibir los correos de un cliente o de otro servidor** funcionando como cliente y de **enviarlos hacia otro servidor o transferirlos hacia el usuario destino**.
  - Veremos que hay **dos tipos de servidores:** de **transferencia o de intercambio** y de **descarga o de entrega**.
  - Los servidores hacen lo que haría el cartero. Se encargan de recoger el correo, seleccionarlo, dirigirlo hacia su destino y dejarlo en los buzones de los domicilios de los destinatarios.
- **Cuentas de correo:** Son las **identificaciones para los remitentes del correo electrónico y los destinatarios**. Un usuario puede tener muchas **cuentas de correo**. Cada cuenta de correo está registrada en un servidor de correo. Cada cuenta de correo tiene asociado un **buzón de usuario**.

# Cuenta de correo

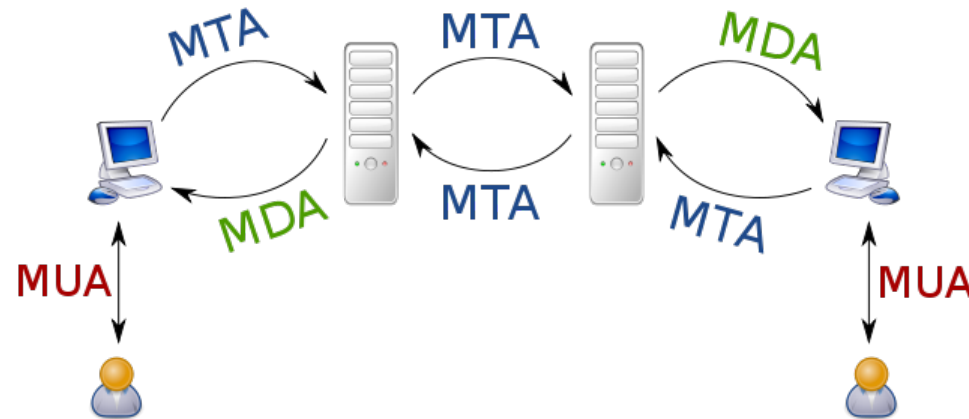
- **nombre@dominio**
  - Por ejemplo mario@asir.net
  - El nombre habrá sido elegido por el usuario.
  - El **dominio** es el que corresponde al **servidor que tiene registrada la cuenta de usuario**.
  - Cuando se va a enviar un correo electrónico, se consulta en DNS el nombre del dominio para buscar servidores de correo de ese dominio (registros **MX**).

# Cuenta de correo

- **Buzón:** Es un espacio de almacenamiento de un servidor donde éste guarda los mensajes destinados a usuarios para que éstos los puedan obtener.
  - Cada usuario tiene asignado un buzón de usuario.
  - Es como el buzón de nuestro domicilio, donde el cartero deja las cartas.
- **Cuentas de correo redirigidas:** Son cuentas a las que se asocia un buzón perteneciente a **otra cuenta de otro dominio**.
  - Por ejemplo, mario@blabla.com puede ser redirigida hacia mario@asir.net por lo que los correos enviados a mario@blabla.com se almacenarán en el buzón de mario@asir.net.
- **Alias:** Son cuentas a las que se asocia un buzón de **otra cuenta del mismo dominio**.
  - Por ejemplo, alumno23@asir.net puede ser redirigida a mario@asir.net y por lo tanto todos los correos enviados a alumno23@asir.net se almacenarán en el buzón de mario@asir.net.

# Agentes de correo

- En los **RFC** que describen el **funcionamiento del servicio de correo electrónico** se habla de los **agentes de correo**.
- Un agente de correo es **un software que se encarga de realizar alguna operación con los mensajes de correo**.
- En la transmisión de correo electrónico intervienen tres agentes: **MTA**, **MDA** y **MUA**.



# Agentes de correo: MTA

- **MTA (Mail Transfer Agent, Agente de transferencia de correo):** Es un software que se encarga de **recibir el correo o reenviarlo a otro agente MTA**, o de **dejarlo almacenado en el buzón de usuario**.
- Un **MTA** puede **recibir el correo desde otros agentes MTA** o desde un **agente MUA**.
- Un agente MTA debe tener **asociado un registro MX en una zona de dominio** para que pueda conocer a qué equipo tiene que enviar el correo destinado a las cuentas de ese dominio.
- Los agentes MTA usan el **protocolo SMTP** cuando **transfieren o reciben correo**.
- Muchos **clientes de correo modernos** pueden **actuar como MTA** cuando estén enviando correo, pero porque **el host en el que se ejecuta la aplicación no tiene su propio MTA**. No entregan el mensaje directamente al servidor de correos del recipiente, solo **envían el correo a un MTA que estén autorizados a usar**.

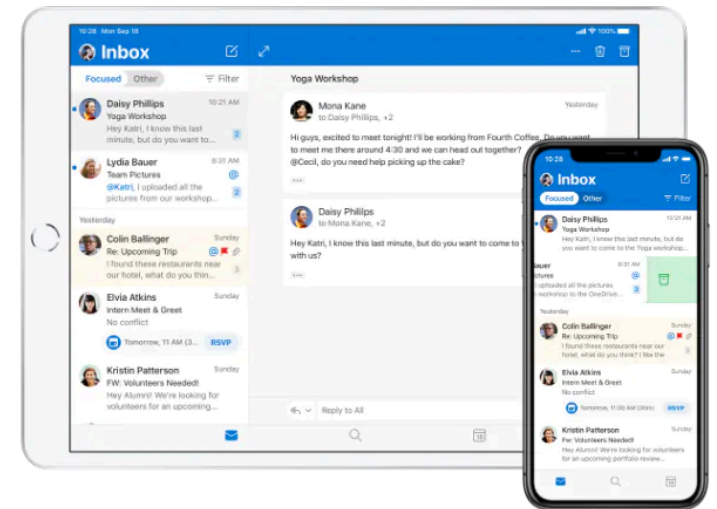
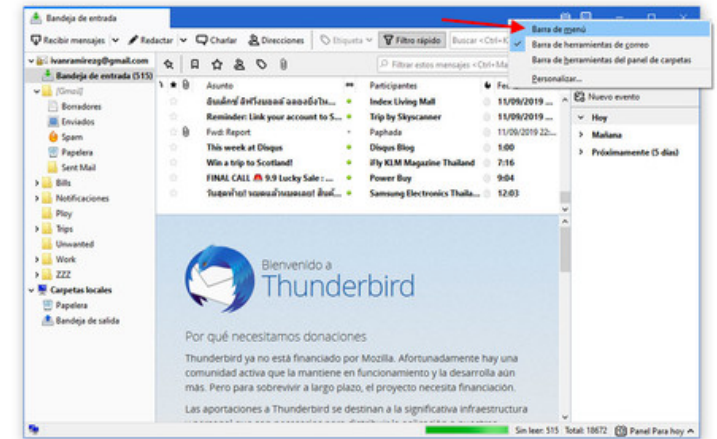


# Agentes de correo: MDA

- **MDA (Mail Delivery Agent, Agente de entrega de correo):** Es un software que **se comunica con los MUA para entregarles el correo almacenado en los buzones** de los usuarios que estén usando el MUA.
- Un MDA **accede a los buzones de los usuarios** y entrega los correos a los MUA utilizando el protocolo **POP3** o el protocolo **IMAP**.
- Cualquier programa que maneje la entrega de mensajes hasta el punto de que puede ser leído por una aplicación cliente de correos se puede considerar un MDA.
- En general los MDAs **no transportan mensajes entre sistemas y tampoco proporcionan una interfaz de usuario**. Los MDAs **distribuyen y clasifican mensajes** en la máquina local para que sean accedidos por una aplicación cliente de correo.

# Agentes de correo: MUA

- **MUA (Mail User Agent, Agente de usuario de correo):**  
Es un software donde un usuario puede enviar y recibir correos.
- El usuario puede iniciar sesión con **una o varias cuentas**.
- Para cada cuenta hay que:
  - **Configurar el MTA** que se encarga de la transferencia de correos que va a enviar
  - **Configurar el MDA** que se encarga de entregarle el correo almacenado en el buzón de usuario.
- A menudo se le denomina **cliente de correo**.



# Ejemplo de configuración de MUA

Configure su dirección de correo electrónico existente

Configure su dirección de correo electrónico existente  
Use su dirección de correo electrónico actual

Su nombre: Juana Pérez

Dirección de correo: testnewtest.es

Contraseña: .....

☒ Recordar contraseña

	ENTRANTE	SALIENTE
Protocolo:	IMAP	SMTP
Servidor:	imap.ionos.es	smtp.ionos.es
Puerto:	993	465
SSL:	SSL/TLS	SSL/TLS
Identificación:	Contraseña normal	Contraseña normal
Nombre de usuario:	testnewtest.es	testnewtest.es

[Config. avanzada](#)

Cancelar Volver a probar Hecho

- En el cliente de correo, por ejemplo el Thunderbird, debemos configurar:
  - Nombre, dirección de correo y contraseña.
  - **Servidor de correo entrante (MDA)**
    - Aquí podemos optar por POP3 o IMAP.
    - Configuramos el host, el puerto, la seguridad y la autenticación.
  - **Servidor de correo saliente (MTA)**
    - Aquí siempre será SMTP
    - Configuramos el host, el puerto, la seguridad y la autenticación.
- Es posible que muchos MUA ya **configuren automáticamente** los servidores MDA y MTA de los servicios de correo más populares: Gmail, Outlook, Yahoo...

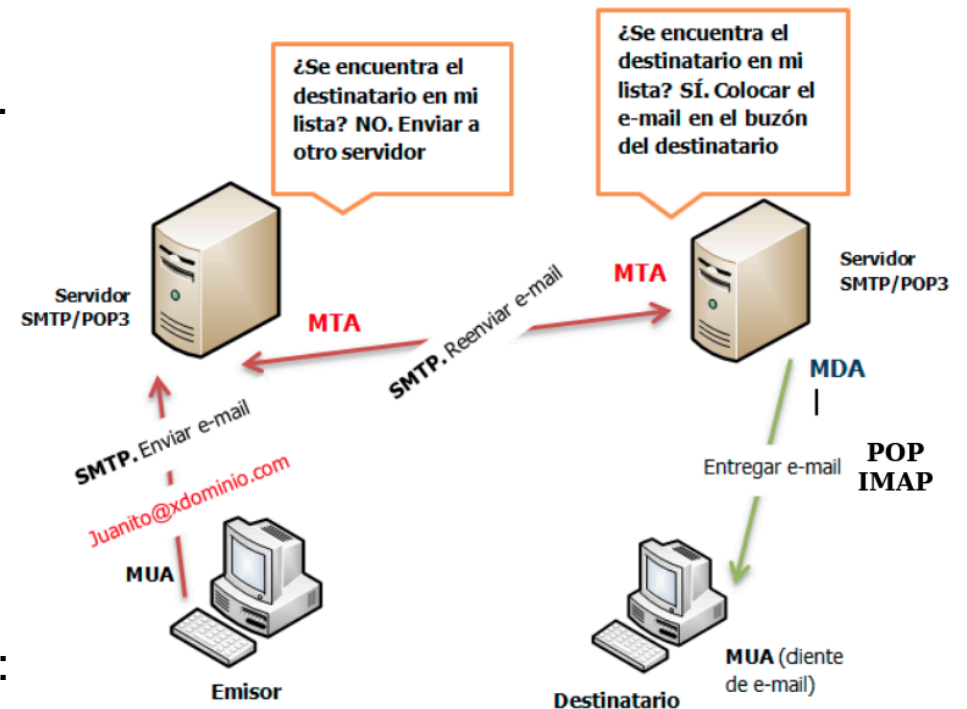
# Introducción a los protocolos de correo

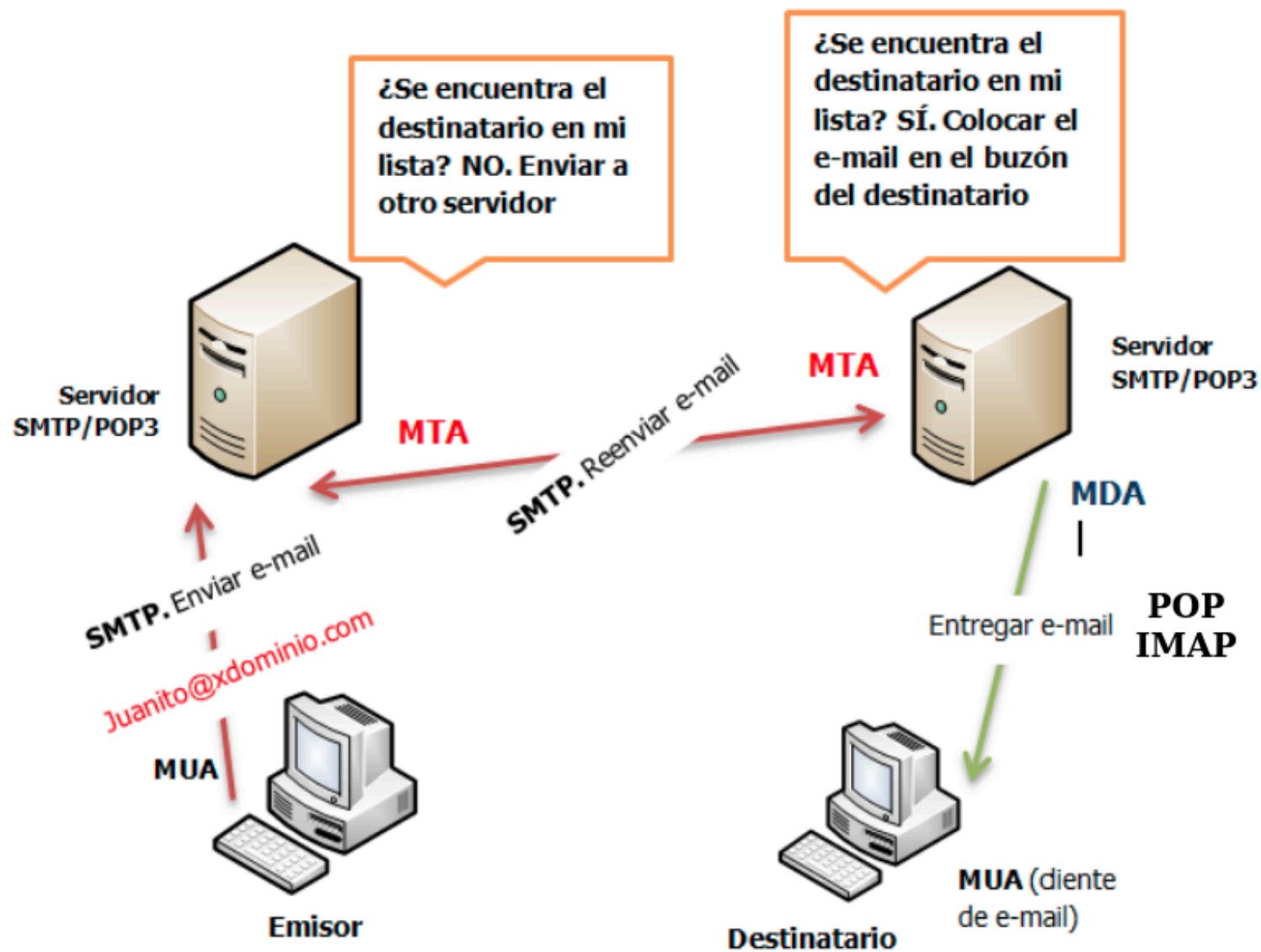
- **SMTP (Simple Mail Transfer Protocol**, Protocolo de transferencia simple de correo): Es el protocolo de red que se usa para **enviar mensajes de correo electrónico**.
  - Existe una mejora del protocolo llamada **ESMTP** (Enhanced SMTP).
- **POP (Post Office Protocol**, Protocolo de Oficina de Correos): Es el protocolo utilizado para **recuperar los correos electrónicos** de un MDA.
  - Su principal característica es que **los correos se descargan**.
- **IMAP (Internet Message Access Protocol**, Protocolo de acceso a mensajes de Internet): Es el protocolo utilizado para **recuperar correos electrónicos** de un MDA.
  - Su principal característica es que **el estado de los correos se sincronizan**.

# Envío y recepción de emails

1. Un usuario utiliza un **MUA** para **enviar el correo** electrónico a su servidor de correos (**MTA**).

- El envío se hace usando el protocolo **SMTP** (**25/TCP**).
- El nombre del servidor tendrá que estar definido en un **registro MX** de un servidor **DNS**.
- También debería declararse la zona inversa en DNS para autenticar servidores: comprobar si un servidor que envía un correo es quien dice ser.



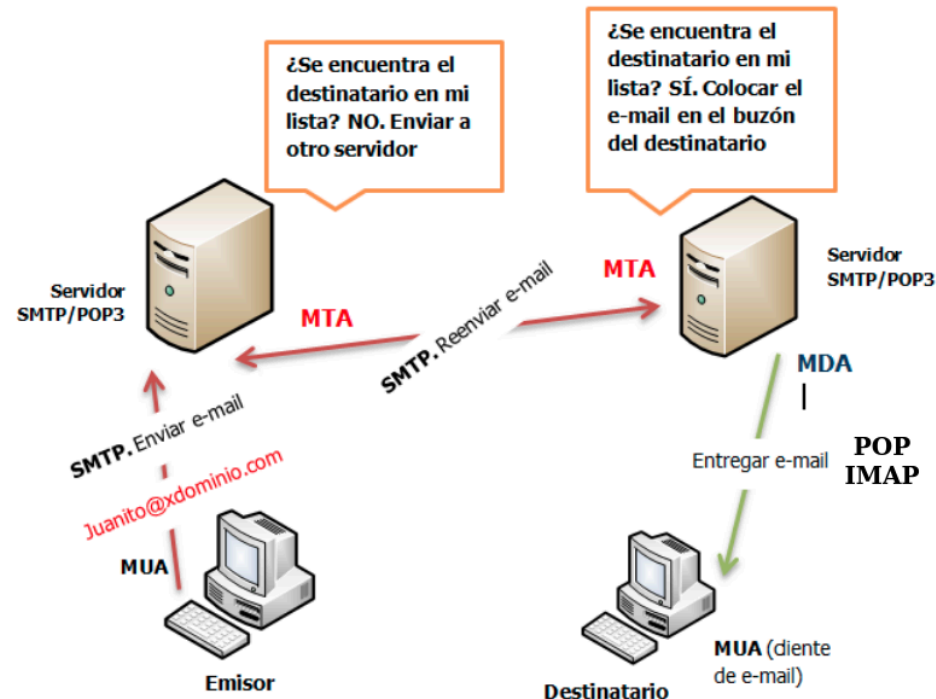


# Envío y recepción de emails

## 2. El **MTA** recibe el correo desde el MUA.

- Si la dirección del destinatario es la misma que controla este servidor, el correo no se envía a ningún **MTA** y se le da al **MDA** para que lo guarde en el buzón del usuario destinatario.
- Si la dirección es distinta a la que controla el servidor, el correo se mandará al **MTA** correspondiente a la dirección del destinatario.
  - Si el **MTA** por cualquier razón **no puede enviar el correo directamente al MTA destino**, puede tener **configurado un MTA intermediario (relay)**. Le mandará el correo a este **intermediario** y el intermediario a su vez al MTA destino.
  - Si el **MTA** no tiene configurado un servidor relay, tendrá que averiguar la dirección IP del servidor correspondiente al nombre de correo del destinatario haciendo una **consulta MX al sistema DNS**.
    - Si recibe **varios servidores de correo**, intentará utilizar el **más prioritario** (el que tiene el número más pequeño).

# Envío y recepción de emails



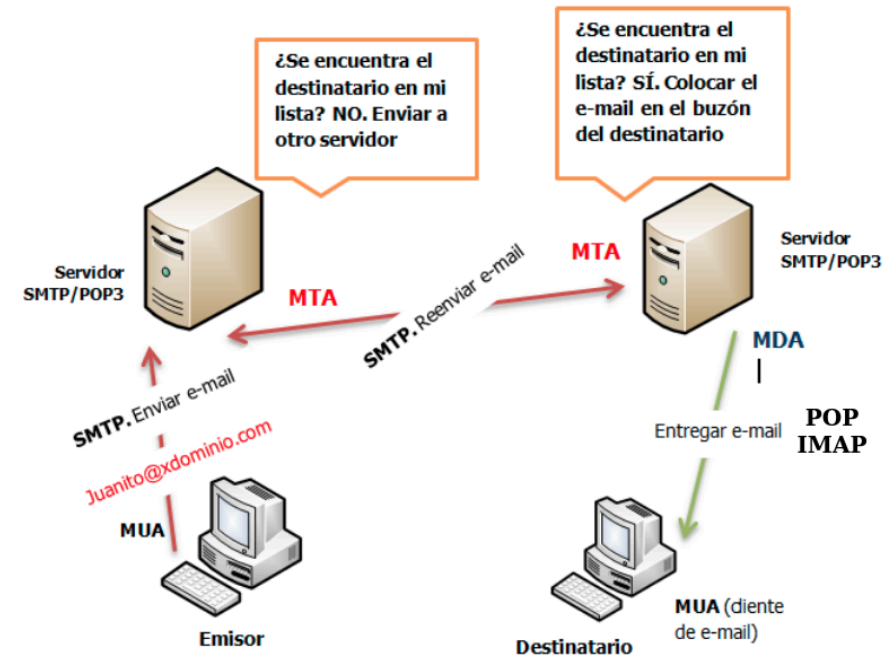
3. Cuando el correo llega al **MTA destino**, se pasa el correo al **MDA** que lo guardará en el **buzón del usuario destinatario**.



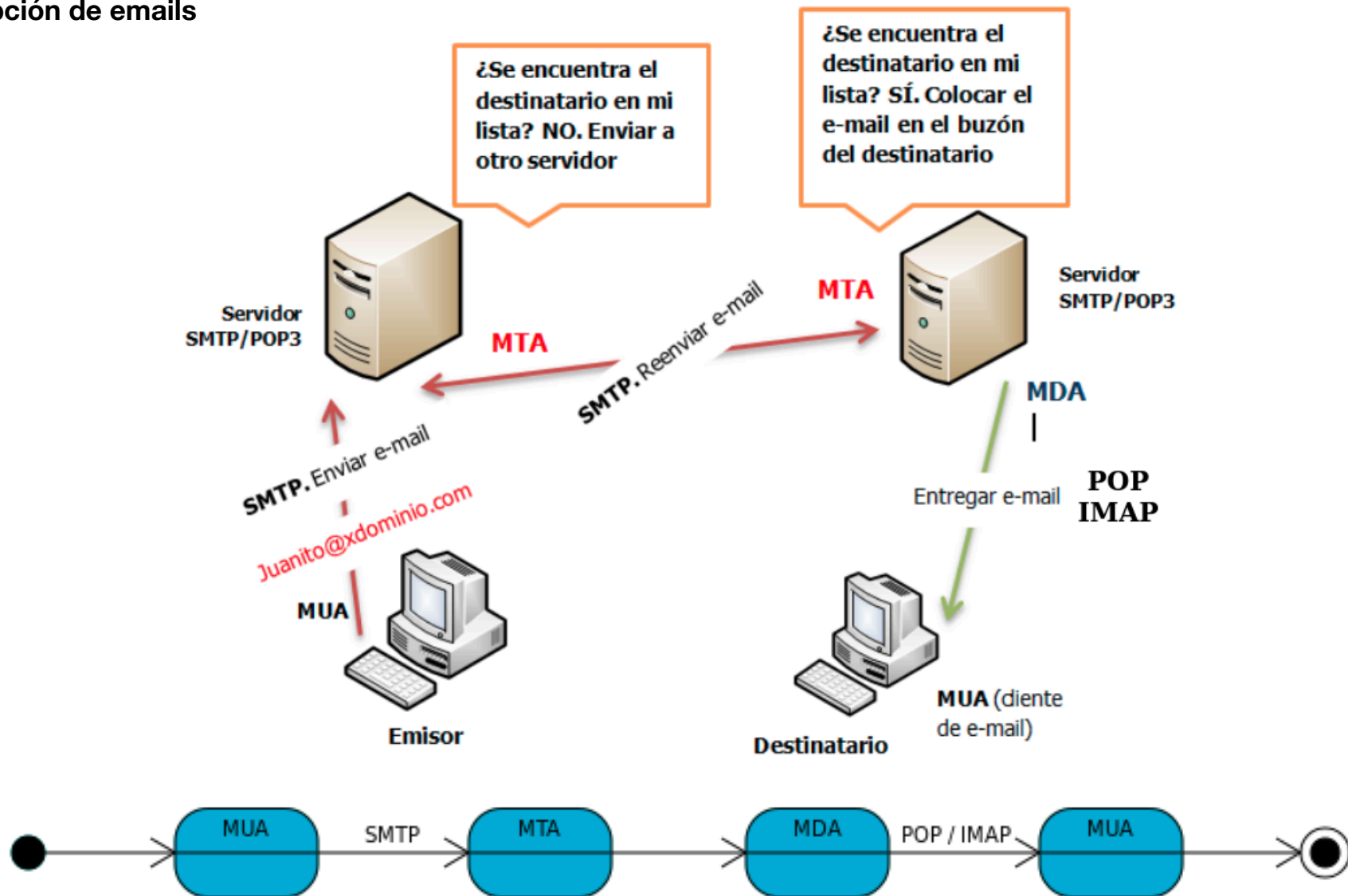
# Envío y recepción de emails

4. El usuario destinatario usará una **MUA** para conectarse al **servidor MDA** y recuperar el correo.

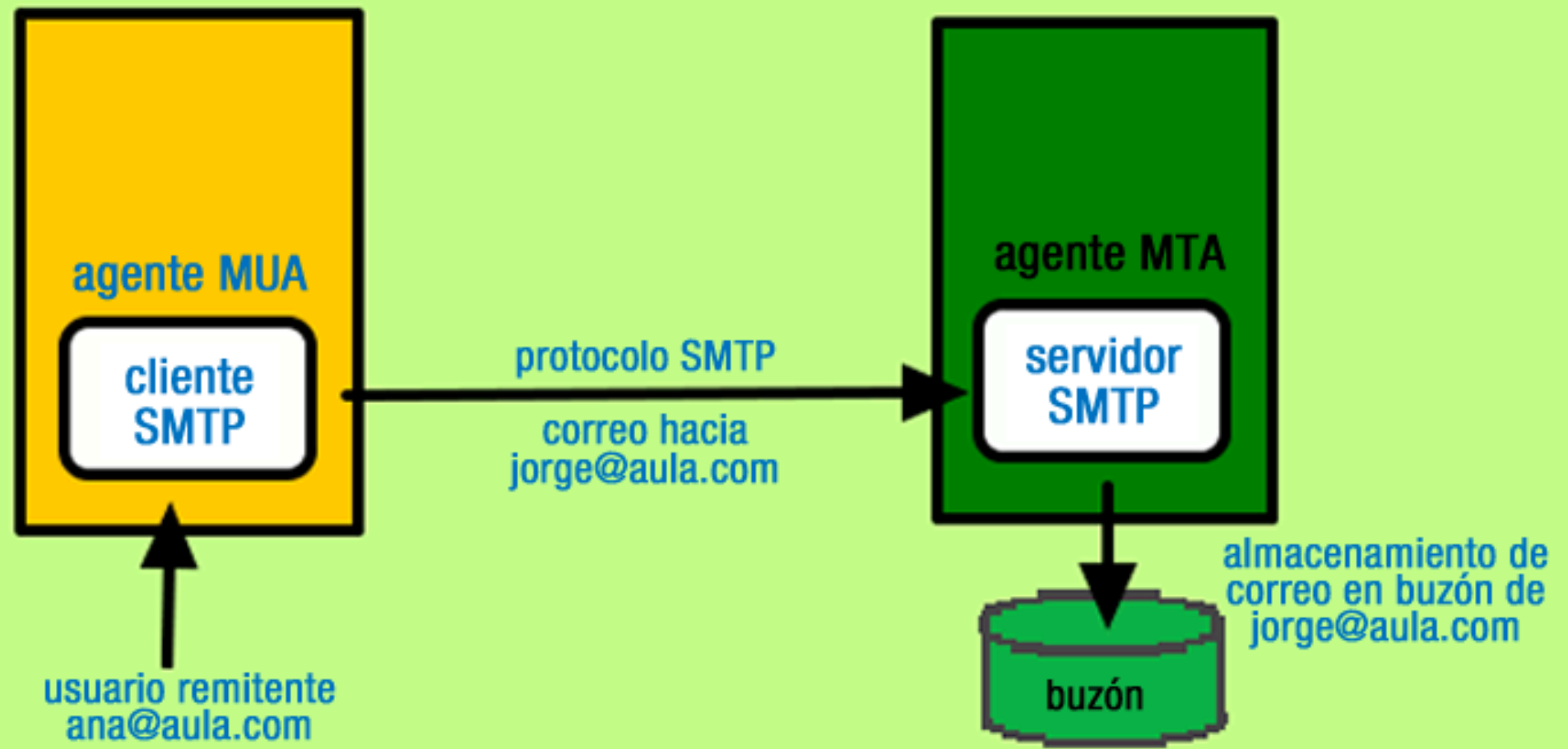
- Puede usar el protocolo **POP3** (110/TCP), autenticarse y **descargar** los correos desde el buzón remoto al **MUA**.
- Puede usar el protocolo **IMAP** (143/TCP), autenticarse y **sincronizar** el estado de los correos desde el buzón remoto al **MUA**.
- De esta manera podemos acceder desde **distintos MUA** y obtener el **mismo estado de los correos** en todos ellos.



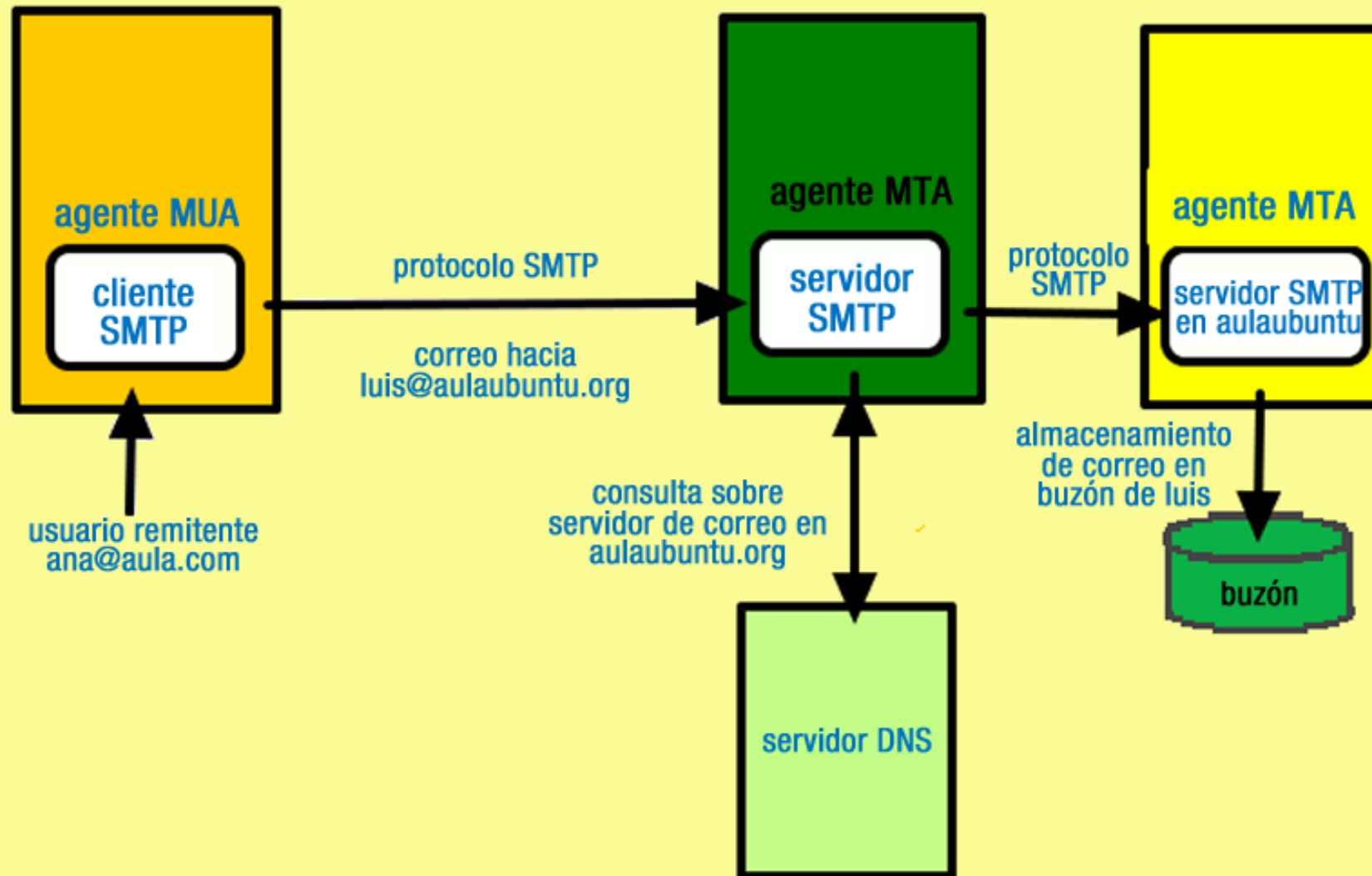
## Envío y recepción de emails



remitente y destinatario en el mismo dominio

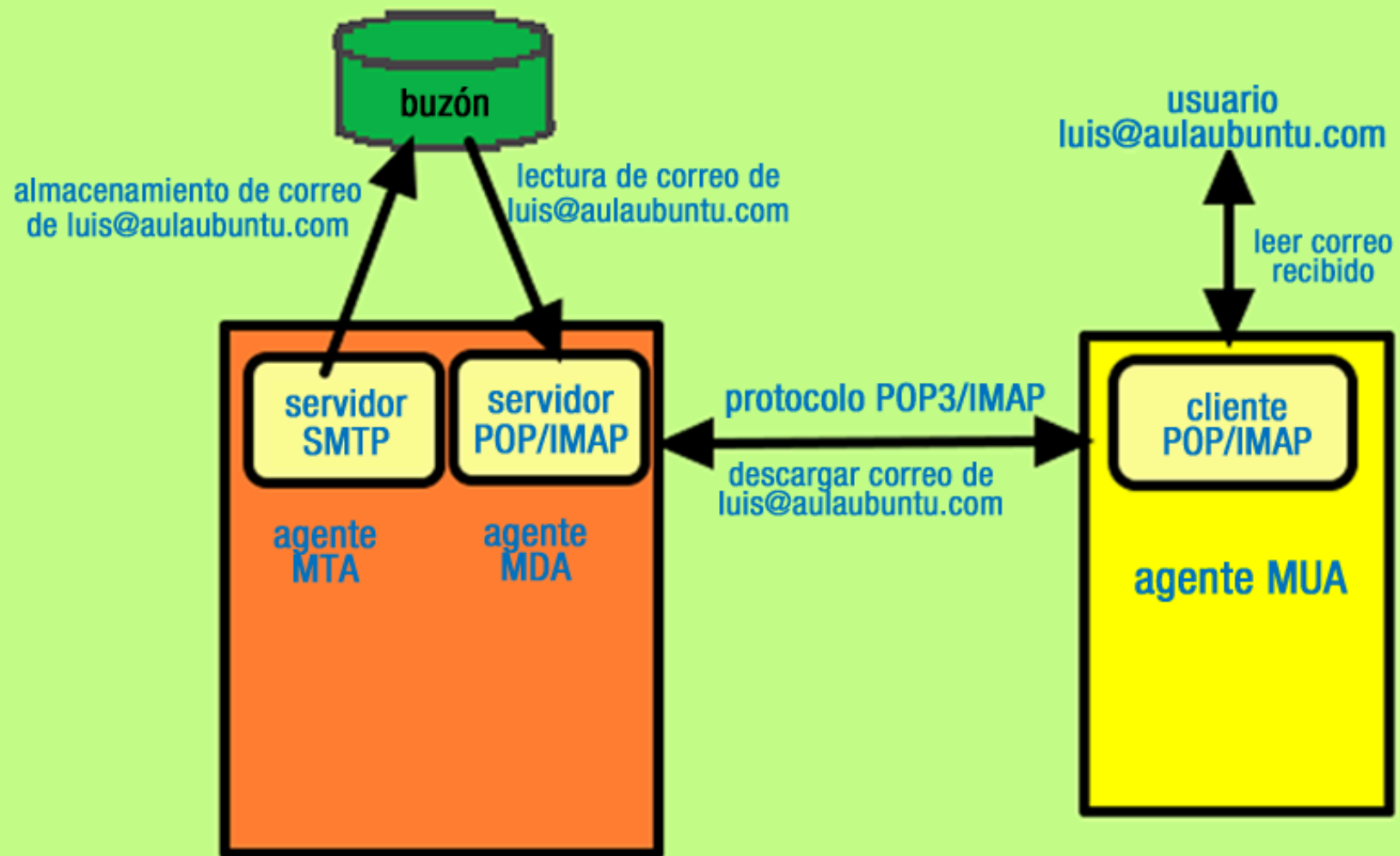


## Remitente y destinatario en distintos dominios



# Proceso de descarga de un correo

- Hemos visto cómo llega un correo a través de uno o pasando por varios MTA a un **buzón del usuario destinatario** y queda allí almacenado.
  - Un buzón de usuario en sistemas UNIX es una carpeta en la que se almacenan los emails. Normalmente los usuarios no tienen acceso al servidor, por lo que debe de haber un mecanismo que les permita obtener los correos que les han llegado.
- Para que el usuario destinatario pueda leer el mensaje tendrá que acceder a **consultar su buzón**.
- Hay dos protocolos que son utilizados para que un **MUA** pueda **obtener el correo almacenado en los buzones de usuario** y entregárselo a los destinatarios: **POP3** e **IMAP**.



Descarga de correo  
desde un buzón de usuario

# Proceso de descarga

1. Un usuario luis@aulaubuntu.com inicia sesión en el **MUA** de su equipo.
2. El **MUA** actúa como **cliente POP3 y/o IMAP** y **contacta con el MDA** o servidor POP3/IMAP que tenga configurado utilizando el protocolo correspondiente.
3. **El MUA solicita descargar el correo** de luis@aulaubuntu.com
  1. **El MDA** del dominio aulaubuntu.com **lee el buzón del usuario** luis@aulaubuntu.com, **obtiene los mensajes almacenados** y **se los entrega al MUA**.
4. El **MUA muestra los mensajes recibidos** al usuario.

# Formato de los correos electrónicos

- Aunque **SMTP** solo utilizaba **ASCII**, actualmente usa **MIME** (extensiones multipropósito de correo de internet):
  - Texto en **conjuntos de caracteres distintos a US-ASCII**
  - **Adjuntos** que no son de tipo texto
  - Cuerpos de mensaje con **multi partes** (multi-part)



# Estructura de los mensajes de correo electrónico

- El documento **RFC 5322** establece cuál debe de ser la estructura de los mensajes de correo electrónico.
- Un **mensaje** está dividido en **dos partes**:
  - **Cabeceras o heads**: Contienen varios campos con la información necesaria para que se pueda transmitir el mensaje.
  - **Cuerpo o body**: es el texto del mensaje.

# Cabeceras de correo de un e-mail

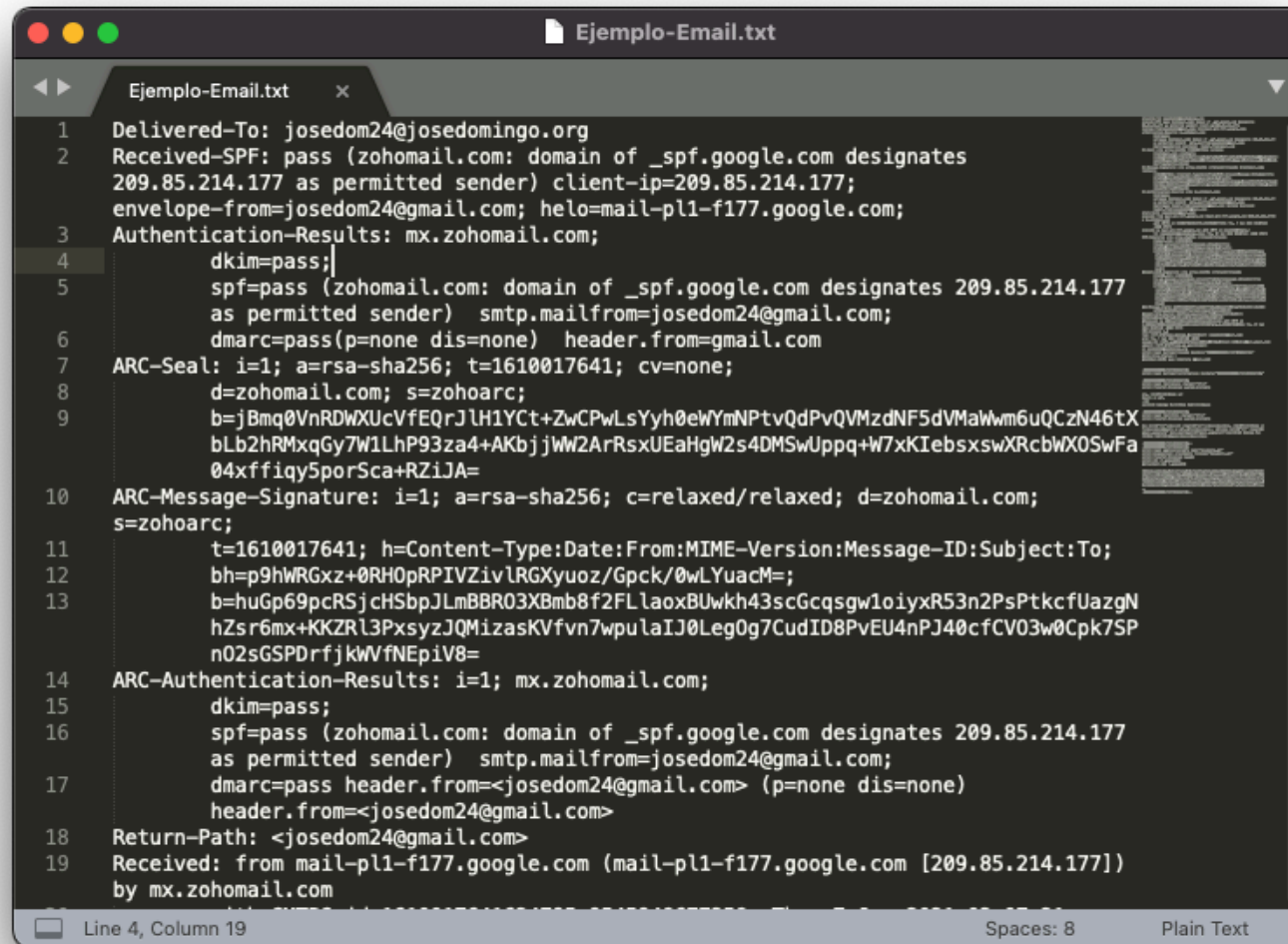
Es importante saber que **cualquier línea puede ser falsificada**, y solo las líneas Received: que son generadas por el servidor del correo, pueden ser **fiables**.

- **From:** Identifica al **remitente** del mensaje.
- **Subject:** Asunto.
- **To:** Muestra a **quién va dirigido el mensaje**, pero podría no contener la dirección del destinatario.
- **CC:** Copia de carbón. Identifica los **destinatarios secundarios** del correo. Todos los destinatarios podrán ver, al recibir un correo con varios destinatarios, las direcciones de las cuentas de correo de otros destinatarios.
- **CCO o BCC:** Copia de carbón oculta. Identifica los **destinatarios ocultos** del correo. En este caso los destinatarios no pueden ver las direcciones de correos de otros destinatarios BCC. Este campo es **eliminado por el servidor al transmitir el mensaje**.
- **Date:** Indica la fecha y hora en la que se ha enviado desde el ordenador remitente el mensaje. Detrás de la hora se indica cuántas horas está desfasado el sistema horario del remitente respecto al horario GMT.
- **Reply-To:** Especifica la dirección del correo electrónico en la que el remitente indica al destinatario que quiere recibir la respuesta al mensaje enviado.

# Otras cabeceras del email

- Las cabeceras anteriores son las que vamos a leer en un correo de forma normal.
- Hay **otras cabeceras** que son añadidas por los servidores por donde pasa el correo:
  - **Return-Path:** La cuenta de correo electrónico para devolver el correo.
  - **Delivered-To:** Usuario final en cuyo buzón se entrega el correo. Podría no coincidir con la dirección del campo To: si esta era una cuenta virtual.
  - **Received:** Información de cada servidor de correo por la que ha pasado el correo.
- Más cabeceras:
  - **MIME-Version:** La presencia de este encabezado indica que el mensaje utiliza el formato MIME.
  - **Content-Type:** Indica el tipo de medio que representa el contenido del mensaje. A través del uso del tipo multiparte, MIME da la oportunidad de crear **mensajes con partes de distintos tipos**.
- Otras cabeceras de seguridad: **Received-SPF, DKIM-Signature**.
- Cabeceras no estándar que empiezan por **X-**

# Ejemplo de mensaje de correo

A screenshot of a text editor window titled "Ejemplo-Email.txt". The editor shows an email header with various fields like "Delivered-To", "Received-SPF", "Authentication-Results", "ARC-Seal", "ARC-Message-Signature", "ARC-Authentication-Results", "Return-Path", and "Received". The text is formatted with line numbers on the left and a status bar at the bottom indicating "Line 4, Column 19", "Spaces: 8", and "Plain Text".

```
1 Delivered-To: josedom24@josedomingo.org
2 Received-SPF: pass (zohomail.com: domain of _spf.google.com designates
  209.85.214.177 as permitted sender) client-ip=209.85.214.177;
  envelope-from=josedom24@gmail.com; helo=mail-pl1-f177.google.com;
3 Authentication-Results: mx.zohomail.com;
4   dkim=pass;
5   spf=pass (zohomail.com: domain of _spf.google.com designates 209.85.214.177
  as permitted sender) smtp.mailfrom=josedom24@gmail.com;
6   dmarc=pass(p=none dis=none) header.from=gmail.com
7 ARC-Seal: i=1; a=rsa-sha256; t=1610017641; cv=none;
8   d=zohomail.com; s=zohoarc;
9   b=jBmq0VnRDWXUcVfEQrJlH1Yct+ZwCPwLsYyh0ewYmNPtvQdPvQVMzdNF5dVMawm6uQCzN46tX
  bLb2hRMxqGy7W1LhP93za4+AKbjjWw2ArSxUEaHgW2s4DMSwUppq+W7xKIebsxswXRcbWX0SwFa
  04xffiqy5porSca+RZiJA=
10 ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed; d=zohomail.com;
  s=zohoarc;
11   t=1610017641; h=Content-Type:Date:From:MIME-Version:Message-ID:Subject:To;
12   bh=p9hWRGxz+0RH0pRPIVZivlRGXyuoZ/Gpck/0wLYuacM=;
13   b=huGp69pcRSjchSbpJLmBBR03XBmb8f2FLlaoxBUwkh43scGcqsgw1oiyxR53n2PsPtkcfUazgN
  hZsr6mx+KKZRL3PxsyzJQMizaskVfvn7wpulaIJ0Leg0g7CudID8PvEU4nPJ40cfCV03w0Cpk7SP
  n02sGSPDrfjkwVfNEpiV8=
14 ARC-Authentication-Results: i=1; mx.zohomail.com;
15   dkim=pass;
16   spf=pass (zohomail.com: domain of _spf.google.com designates 209.85.214.177
  as permitted sender) smtp.mailfrom=josedom24@gmail.com;
17   dmarc=pass header.from=<josedom24@gmail.com> (p=none dis=none)
  header.from=<josedom24@gmail.com>
18 Return-Path: <josedom24@gmail.com>
19 Received: from mail-pl1-f177.google.com (mail-pl1-f177.google.com [209.85.214.177])
  by mx.zohomail.com
```

# Cientes de correo

- Un **cliente de correo** es un programa de usuario que permite que los **usuarios editen y envíen mensajes de correo** y/u obtengan el correo recibido leyendo los buzones del usuario.
- Existen muchos clientes de correo electrónico.
- La mayoría pueden **actuar como clientes SMTP, POP3 e IMAP**, aunque algunos clientes solo pueden actuar como clientes de un protocolo.
- Hay **clientes en modo texto**, pero lo normal es que los usuarios utilicen clientes gráficos.

# Cientes de correo: Modo texto

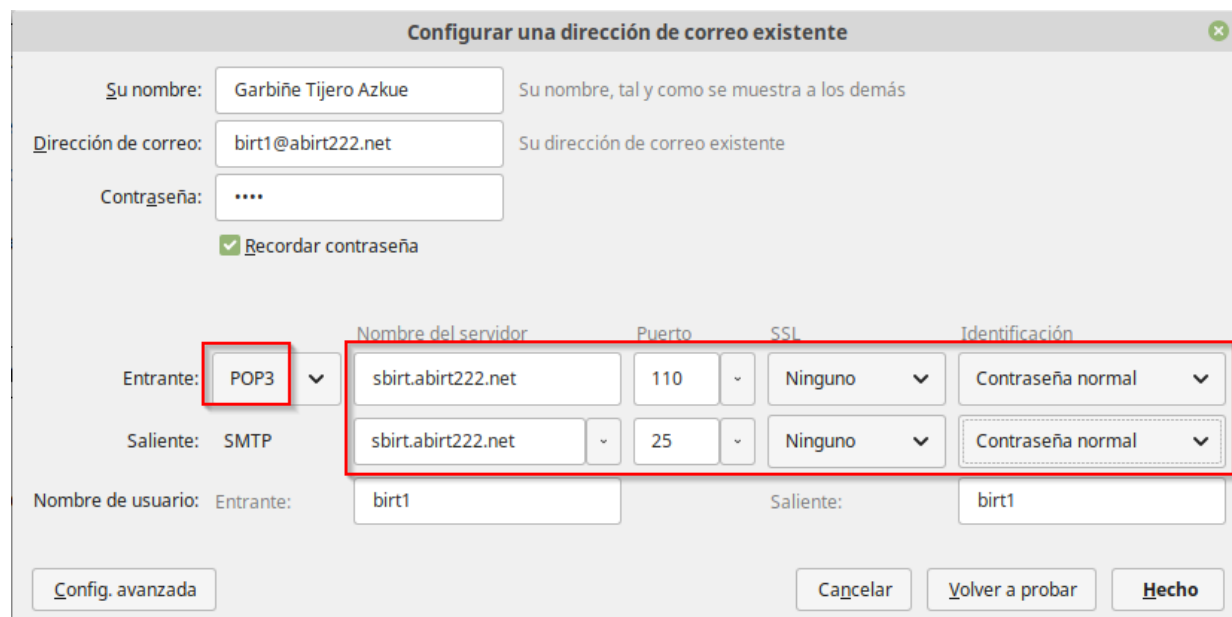
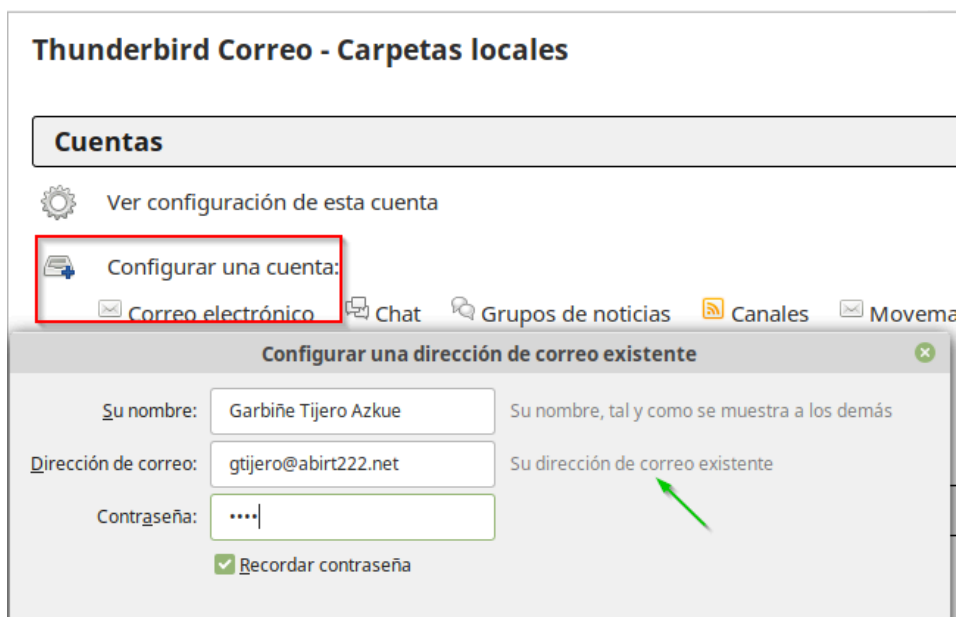
- **mail:** Es un cliente para Linux que solo puede trabajar con SMTP, por lo tanto no puede descargar correo. Al ejecutarse se inicia una interfaz de comandos mail.
- **pine:** Es un cliente para Linux muy fácil de usar. Permite enviar y descargar correo. Al ejecutarlo se inicia una interfaz de ventana con menús pero de tipo texto (no se puede usar ratón).
- **mutt:** Es un cliente para Linux muy potente. Permite enviar y descargar correo. Al ejecutarlo se inicia una interfaz de ventana con menús pero de tipo texto (no se puede usar el ratón).
- **fetchmail:** Es un cliente para Linux que solo permite leer mensajes de correos recibidos en buzones.

# Cientes de correo: Modo gráfico

- **Mozilla Thunderbird:** Es un cliente multiplataforma para Linux, Windows y Mac OS. Facilita la configuración de cuentas de usuario, pues obtiene la configuración desde los servidores que tienen registradas las cuentas. Permite utilizar extensiones.
- **Windows Live Mail:** Es un cliente muy fácil de usar, disponible solo para windows en sus nuevos sistemas.
- **Evolution:** Cuenta con distribuciones para Linux y Windows. Es fácil de usar aunque no tiene muchas opciones de configuración. Está integrado en los sistemas Debian/Ubuntu.

# Cientes de correo: Configuración de cuentas

- En **Thunderbird** o en cualquier cliente **no se crean cuentas, se agregan** cuentas de correos ya creadas en servidores de correo.



- El usuario y la clave, a veces, es lo único que necesita Thunderbird para configurar la cuenta. En otras ocasiones tendremos que especificar los servidores a mano.



# Clientes de correo: Envío y recepción

- Los clientes de correo electrónico disponen de un **editor de textos** desde el que redactar el mensaje.
- En la ventana de edición y envío se pueden especificar los destinatarios del mensaje.
- Es posible insertar imágenes y tablas en el mensaje, solicitar acuse de recibo, notificar estado de entrega del mensaje, guardar una copia del mensaje, adjuntar ficheros.
- Thunderbird permite gestionar y organizar una **libreta de direcciones** de otros usuarios para enviarles mensajes sin tener que recordar su dirección de correo.
- Thunderbird permite **cifrar los mensajes** o **firmarlos** digitalmente.

Inbox

Get Messages

Write

Chat

Address Book

Tag

Quick Filter

Search <Ctrl+K>

Account

Inbox (2)

Sent

Trash

Local Folders

Trash

Outbox

★	Subject	Correspondents	Date
☆	Welcome to the "daily-image-l" mailing list	daily-image-l-request@lists.wikimedia....	8:49 PM
☆	Welcome to the "WikimediaAnnounce-l" mailing ...	wikimediaannounce-l-request@lists.wi...	8:49 PM
☆	Welcome to the "daily-article-l" mailing list	daily-article-l-request@lists.wikimedia....	8:49 PM
☆	Welcome to the Mozilla Information Trust Initiati...	Mozilla	8:49 PM
☆	Welcome to the Inhuman Ads newsletter	Mozilla	8:49 PM
☆	<b>We're glad to meet you</b>	<b>Mozilla</b>	<b>8:50 PM</b>
☆	<b>Welcome to the Mozilla Developer Newsletter!</b>	<b>Mozilla</b>	<b>8:51 PM</b>

From Mozilla <Mozilla@e.mozilla.org> ☆

Subject **We're glad to meet you**

To Me ☆

8:50 PM

Unread: 2

Total: 7

Today Pane

# SMTP: Protocolo de transferencia de mensajes

- Mediante el **SMTP** (Simple Mail Transfer Protocol) se produce el **transporte de los mensajes de correo electrónico desde el ordenador del remitente hasta que se deposita en el buzón del usuario destinatario**.
- El protocolo SMTP establece **conexiones cliente servidor**, en las que el cliente solicita el envío de un mensaje de correo al servidor, y éste se encarga de transportarlo hacia otro servidor o bien de almacenarlo en buzones de usuario.
- El servidor utiliza el puerto **TCP 25** y los clientes utilizan cualquier puerto superior a 1024.
- Se trata de un **protocolo inseguro**: transmite la información en **texto plano y no requiere autenticación**.
- Ahora es bastante **normal que SMTP trabaje sobre conexiones seguras** en las que se cifra la información.
- Al no requerir autenticación puede ocasionar graves problemas ya que **cualquier usuario podría solicitar el envío de correos** a través de un servidor SMTP (Spam, emails falsos...).
- Se define en el **RFC 2821** y se han publicado RFC posteriores cambios para mejorar la seguridad.
- El protocolo establece un **conjunto de comandos** que los clientes pueden enviar a los servidores y los formatos de las posibles respuestas.

[https://es.wikipedia.org/wiki/Protocolo\\_para\\_transferencia\\_simple\\_de\\_correo](https://es.wikipedia.org/wiki/Protocolo_para_transferencia_simple_de_correo)

# Puertos que se usan en SMTP

- **25/TCP**: Es el puerto predeterminado para la transmisión de correos por SMTP.
- **587/TCP**: Es una opción alternativa al puerto 25 para el envío de correos desde clientes (MUA) a servidores salientes (MTA). es **más seguro** ya que **requiere autenticación del cliente y conexiones cifradas** usando STARTTLS.
- **465/TCP**: Este puerto se ha usado históricamente para el cifrado SSL/TLS de conexiones SMTP, aunque su uso ha sido desalentado en favor del uso del puerto 587 con STARTTLS. El puerto 465 se considera obsoleto y no se recomienda su uso actualmente.
- La configuración de puertos puede variar según la configuración específica del servidor y las políticas de seguridad. Si se va a configurar un cliente es importante leer la documentación.

# SMTP: Comandos

- Tras establecer una conexión TCP cliente/servidor en el puerto 25, el cliente puede comenzar a enviar comandos SMTP al servidor.
- Algunos de los comandos más importantes son:
  - **HELO nombre\_maquina:** Lo envía el cliente para identificarse, aunque es válido cualquier valor en nombre\_maquina, se debería escribir el nombre real.
  - **MAIL FROM:dirección:** Para identificar al remitente del correo.
  - **RCPT TO:dirección:** Para identificar al destinatario del correo.
  - **DATA:** para indicar que se va a comenzar a enviar el mensaje.
  - **SUBJECT:Asunto:** Para especificar el asunto. Es obligatorio que este comando se envíe a continuación de DATA, que después se envíe el contenido o cuerpo del mensaje. El mensaje tiene que terminar con una línea que tenga un solo punto y pulsando después ENTER.
  - **QUIT:** Cierra la conexión cliente servidor.

[https://es.wikipedia.org/wiki/Protocolo\\_para\\_transferencia\\_simple\\_de\\_correo](https://es.wikipedia.org/wiki/Protocolo_para_transferencia_simple_de_correo)

# SMTP: Comandos

```
HELO pc1.aula.com  
MAIL FROM:ana@aula.com  
RCPT TO:jorge@aula.com  
RCPT TO:luis@aulalinux.com  
DATA  
SUBJECT:saludo
```

```
Hola,  
por la gran ayuda que me has prestado,  
un abrazo y muchísimas gracias.
```

```
.
```

```
QUIT
```

# SMTP: Códigos de respuesta

- El servidor SMTP envía respuestas al cliente para indicarle cómo ha procesado un comando con un código numérico de 3 dígitos.

Código	Descripción
2xy	La acción se realizó con <b>éxito</b> . Puede enviarse otro comando.
3xy	Se está esperando que el cliente envíe <b>información adicional necesaria</b> para poder completar la acción.
4xy	Se indica que se ha producido un <b>error temporal</b> y que podría enviarse el comando nuevamente para posiblemente procesarse con éxito.
5xy	Se indica que se ha producido un <b>error permanente</b> , se seguirá produciendo aunque se siga enviando el comando.

# SMTP: Códigos de respuesta

## Código Significado

200 (Respuesta no-standard success response, ver rfc876)

211 System status, or system help reply (estado del sistema)

214 Mensaje de Ayuda

220 Servicio Listo

221 Service closing transmission channel

250 Requested mail action okay, completed

251 User not local; will forward to

354 Start mail input; end with .

421 Service not available, closing transmission channel

450 Requested mail action not taken: mailbox unavailable

451 Requested action aborted: local error in processing

452 Requested action not taken: insufficient system storage

## Código Significado

500 Syntax error, command unrecognised

501 Syntax error in parameters or arguments

502 Command not implemented

503 Bad sequence of commands

504 Command parameter not implemented

521 does not accept mail (see rfc1846)

530 Access denied (???a Sendmailism)

550 Requested action not taken: mailbox unavailable

551 User not local; please try

552 Requested mail action aborted: exceeded storage allocation

553 Requested action not taken: mailbox name not allowed

554 Transaction failed

<https://americandominios.com/conta/knowledgebase.php?action=displayarticle&id=303>



# Protocolos y servicios de descarga de correo

- Los **servidores SMTP** se encargan de **transportar los mensajes desde el ordenador cliente SMTP del remitente hasta el buzón del destinatario** final.
  - En este buzón quedan almacenados los mensajes como archivos.
  - Un usuario podría consultar los mensajes recibidos accediendo a su buzón, una carpeta en el disco del servidor donde el usuario tiene la cuenta de correo...
    - Pero evidentemente esto **no es realizable en la práctica** puesto que los usuarios no tienen acceso al ordenador servidor.
- Además del protocolo SMTP, existen dos protocolos para la descarga de correo por los usuarios receptores desde sus buzones de usuario: **POP3** e **IMAP**.
- Para ambos protocolos, un cliente establece una **conexión autenticada** y así puede acceder a su buzón, solicitar la entrega de correos y gestionar su cuenta.

# POP: Características y funcionamiento

- El protocolo **POP** (Post Office Protocol) es un **protocolo** cliente servidor para la **entrega o descarga de correo** desde los buzones de usuario en los servidores a los programas clientes de usuario.
- Actualmente se usa **POP3 (RFC 1939)**. Puerto **110/TCP**.
- El protocolo **POP3** es un protocolo **inseguro** ya que, aunque utiliza autenticación, todo el intercambio de información se realiza en **texto plano** sin cifrar (incluyendo la contraseña y la dirección de correo).
- El protocolo **POP3S** mejora la seguridad encriptando la comunicación: mediante **SSL** (puerto **995/TCP**).
- También se puede usar **TLS** en el puerto **110/TCP**.

# POP: Fases de una conexión

- En una conexión POP3 el cliente envía comandos y el servidor devuelve respuestas.
  - Los comandos POP3 se describen en el **RFC 1989**.
- 1.**Conexión:** El cliente solicita establecer una conexión de transporte con el puerto **110/TCP**.
  - 2.**Autorización:** Tras establecerse la conexión, el cliente se **autentifica** enviando la dirección de correo y la contraseña del usuario mediante los comandos POP3 **USER** y **PASS**.
  - 3.**Transacción:** Desde el cliente ya se pueden enviar comandos al servidor para **gestionar el correo** y listarlo, descargarlo, eliminarlo, etc.
  - 4.**Actualización:** Se inicia cuando el cliente solicita cerrar la conexión con el comando **QUIT**. En esta fase es **cuando el servidor elimina realmente mensajes** que habían sido marcados para ser borrados.

# IMAP: Características y funcionamiento

- Al igual que POP3, el protocolo **IMAP** (Internet Message Access Protocol) es un **protocolo de descarga de correos** desde los buzones de usuario a los clientes de servicio.
- Actualmente se trabaja con la versión 4 (**IMAP4**), desarrollada en el **RFC 3501**.
- Un servidor IMAP establece las conexiones con los clientes en el puerto **143/TCP** y si se utiliza el protocolo **IMAP seguro** se utiliza el puerto **993/TCP**.

# POP vs IMAP

- El protocolo **POP** se creó **para recibir correo**, no para enviarlo.
  - Su objetivo principal era poder **gestionar los correos sin estar conectado**.
  - Era una época en la que los **tiempos de conexión eran críticos y los tiempos de descarga, importantes**.
- El protocolo **IMAP** es el protocolo **ideal cuando se cambia de ordenador habitualmente**. Es más flexible, pero tiene las siguientes desventajas:
  - Con IMAP tienes que estar **conectado a Internet todo el tiempo** que leas tu correo y contestes tus mensajes.
  - **Si pierdes la conexión** a Internet **no podrás acceder a tu correo recibido**, porque está almacenado en el servidor de correo y no en tu ordenador.
- ¿Qué pasa si yo uso IMAP para gestionar mi correo, y tengo que acceder (un día, de casualidad) por POP?
  - **POP** solo lee la carpeta de entrada **INBOX**.
  - Esto ocasiona que todas **las carpetas** (y por lo tanto el contenido que tengan) **creadas con IMAP, no puedan ser leídas por POP a excepción de la carpeta INBOX**.

# IMAP vs POP

Protocolo	Ventajas	Desventajas
<b>IMAP4</b>	<ul style="list-style-type: none"><li>• Trabaja en modo de conexión permanente, por lo que avisa inmediatamente de la llegada de nuevo correo</li><li>• Transmite solo las cabeceras por lo que el usuario puede decidir su borrado inmediato</li><li>• La bajada del mensaje se produce solo cuando el usuario quiere leerlo</li><li>• El almacenamiento local del mensaje es opcional (una opción del cliente de correo)</li><li>• Gestiona carpetas, plantillas y borradores en el servidor</li><li>• El almacenamiento de mensajes y carpetas en el servidor permite su uso desde múltiples dispositivos y de forma simultánea</li><li>• Permite la búsqueda de mensajes por medio de palabras claves</li><li>• Los mensajes se pueden etiquetar. El marcado queda en el servidor</li><li>• Se pueden crear carpetas compartidas con otros usuarios (depende del servidor)</li></ul>	<ul style="list-style-type: none"><li>• No todos los clientes de correo soportan la extensión IMAP IDLE (aviso de nuevos correos)</li><li>• Necesita una transacción por cada correo que se quiera leer</li><li>• Hay un retraso en la aparición del mensaje en la pantalla del usuario, mientras se descarga</li><li>• Si se pierde la conexión, no se podrá ver el mensaje salvo si el cliente de correo lo haya almacenado en local</li><li>• Las carpetas, plantillas y borradores no podrán ser leídos usando POP (excepto la Bandeja de entrada)</li><li>• Todos los mensajes ocupan espacio en disco del servidor.</li></ul>
<b>POP3</b>	<ul style="list-style-type: none"><li>• Los correos aparecen inmediatamente porque quedan residentes en el dispositivo (una vez descargados)</li><li>• Permite ahorrar espacio en el disco del servidor Si se configura para eliminar los mensajes descargados al cliente.</li></ul>	<ul style="list-style-type: none"><li>• Sólo se conecta periódicamente cada X minutos para buscar por nuevo correo</li><li>• La conexión periódica provoca un aumento del tráfico y un retraso en la respuesta del cliente (esperar la descarga completa)</li><li>• En cada conexión, se baja todos los correos nuevos, vayan a ser después leídos o no</li><li>• Los correos ocupan espacio local del dispositivo</li><li>• Por defecto, elimina los mensajes del servidor, haciendo imposible el acceso a ellos desde otro dispositivo.</li></ul>

# IMAP vs POP

- **Modos de operación conectado y desconectado**
  - Al utilizar **POP**, los clientes **se conectan** brevemente al servidor de correo, solo **el tiempo que tardan en bajar los mensajes**.
  - Al utilizar **IMAP4**, los clientes **permanecen conectados el tiempo que su interfaz permanezca activa** y descargan **mensajes bajo demanda**.
  - Esta manera de trabajar puede dar **tiempos de respuesta más rápidos** para usuarios que tienen una **gran cantidad de mensajes o mensajes grandes**.

# IMAP vs POP

- **Conexión de múltiples clientes de forma simultánea a un mismo buzón**
- El protocolo **POP** requiere que el cliente conectado sea **el único conectado al buzón**.
- **IMAP4** permite **accesos múltiples y simultáneos** y proporciona mecanismos para detectar los **cambios realizados a un buzón** por parte de otro cliente conectado. (RFC3501, sección 5.2)



# IMAP vs POP

- **Acceso a partes MIME de los mensajes y obtención parcial**
  - Casi todo el correo de Internet se transmite en **formato MIME**.
  - El protocolo **IMAP4** permite a los clientes obtener separadamente **cualquier parte MIME individual** y también obtener porciones de las partes individuales o los mensajes completos.
  - Estos mecanismos permiten a los clientes **obtener la porción de texto** de un mensaje **sin tener que descargar los archivos adjuntos o en flujos**.

# IMAP vs POP

- **Información del estado del mensaje**
  - A través de las **señales definidas en el protocolo IMAP4**, los clientes pueden seguir el estado del mensaje: si ha sido leído, respondido, eliminado...
  - Estas **señales se almacenan en el servidor**, de manera que varios clientes conectados al mismo buzón pueden **detectar los cambios realizados por otros clientes**.
  - **POP no ofrece ningún mecanismo para esta información de estado**. Así que si un usuario accede a un buzón con dos clientes POP diferentes, en tiempos diferentes, la información de estado (si el email ha sido accedido) **no puede sincronizarse entre clientes**.
  - **IMAP** soporta que el usuario utilice **palabras clave o etiquetas**, cuyo significado depende del cliente.

# IMAP vs POP

- **Múltiples buzones en el servidor**
  - Los **clientes de IMAP4** pueden **crear, renombrar y/o eliminar buzones** (por lo general presentadas como carpetas de usuario) en el servidor, y copiar mensajes entre buzones.
  - El soporte para **múltiples buzones** también le permite a los servidores proporcionar acceso a carpetas públicas y compartidas.
  - La **IMAP4 Access Control List (ACL) Extension** (RFC4314) se puede usar para regular los derechos de acceso.

# IMAP vs POP

- **Búsquedas de parte del servidor**
  - **IMAP4** proporciona un mecanismo para que los clientes **pidan al servidor que busque** mensajes de acuerdo a ciertos criterios.
  - Este mecanismo **evita** que los clientes **descarguen todos los mensajes**, agilizando las búsquedas.

# IMAP vs POP

- **Mecanismo de extensión**
  - Como reflejo de la experiencia en los protocolos de Internet, IMAP define un mecanismo mediante el cual se puede extender.
  - Se han propuesto **muchas extensiones de IMAP4** y son de uso común.
  - IMAP2 bis no tiene ese mecanismo, y POP tiene uno definido en RFC 2449.
  - Un ejemplo de extensión es **IMAP IDLE**, que sirve para que **el servidor avise al cliente cuando ha llegado un nuevo mensaje de correo** y éstos se sincronicen.
  - Sin esta extensión el cliente debería de contactar de forma periódica con el servidor para ver si hay mensajes nuevos.

# Servidores de correo

- **Servidores SMTP (Agentes MTA)**

- Postfix: Linux. IBM Public License
- Sendmail: Linux. GNU GPL
- Qmail: Linux. GNU GPL.



- **Servidores POP3/IMAP (Agentes MDA)**

- Dovecot: Linux. MIT y LGPL
- Cyrus IMAP Linux. server: BSD

- **Servidores SMTP y POP3/IMAP**

- hMailServer: Windows. GPL
- Zimbra: Linux, Windows, Mac OS. Yahoo Public License.

Para elegir un software servidor de correo hay que seleccionar uno que sea **fácil de configurar** y que tenga muchas **opciones** de configuración, el **consumo** de recursos, la **rapidez** de respuesta, etc.

# Integración de servidores en DNS (I)

- Para enviar un mensaje y que llegue al destinatario, es necesario **conocer el servidor de correo** encargado del buzón de usuario y su dirección IP.
- Será el **servidor de correo del dominio** al que pertenece la **cuenta de correo del destinatario** el que deberá estar integrado en la **zona DNS**.
- Antes de enviar un mensaje a ana@aula.com desde una cuenta de correo de un dominio diferente, se hace una consulta al servidor DNS responsable de la zona aula.com consultando:
  - **Registro MX:** Indica el nombre del **un servidor MTA** de correo para el dominio.
  - **Registro A:** Indica la dirección **IP del servidor de correo**.
- Obtenida la dirección de correo, se establecerá una **conexión SMTP** con el equipo que tiene esa dirección y se le enviará el mensaje de correo.

# Integración de servidores en DNS (II)

- **No es obligatorio** añadir un registro **PTR** en la zona inversa para resolver la dirección IP del servidor de correo.
- Algunos servidores de correo, al recibir un correo de otro servidor, **verifican la dirección IP** correspondiente al servidor remitente mediante el registro PTR.
- Si **no hay registro PTR** o es **incoherente** con el registro A, **el mensaje de correo se rechaza**.



# Integración de servidores en DNS (III)

En la zona directa "aula.com"

```
aula.com.  IN  MX  1  s1.aula.com.  
s1.aula.com.  IN  A  88.88.88.88  
aula.com.  IN  MX  2  s2.aula.com.  
s2.aula.com.  IN  A  88.88.88.89
```

En la zona inversa "88.88.88.in-addr.arpa"

```
88  IN  PTR  s1.aula.com.  
89  IN  PTR  s2.aula.com.
```

# Servidores open relay y servidores smart host

- Un servidor de transferencia (MTA) es **open relay** cuando permite **reenviar a través de él cualquier correo recibido desde otro servidor MTA o desde un MUA**.
- Puede utilizarse para enviar indiscriminadamente **mensajes de spam**, virus... ya que no controlan el acceso de los remitentes.
  - Muchas veces los servidores de correo pueden estar **configurados como open relay por descuido**, desconocimiento, **error de configuración** o por la modificación de configuración por parte de un atacante.
  - Aunque en otros casos se instalan servidores en modo open relay para facilitar el envío de spam, virus y correo no deseado.
- Hay **listas negras de servidores de correo open relay** sobre los que se han detectado usos maliciosos. Estas listas sirven a los servidores para detectar el correo basura o spam.
- Si en un servidor open relay se establecen unas direcciones de **otros servidores SMTP autorizados a reenviar correo** y se **deniega el permiso para reenviar a cualquier otro servidor**, se dice que el servidor es un **smart host**.
  - En la autorización se pueden dar direcciones IP, nombres de servidores, direcciones de red, rangos de direcciones o nombres de dominios.



<https://derechodelared.com/breve-historia-del-primer-spam/>

# Servicio de correo electrónico vía web

- Cuando queremos tener un servidor de correo para nuestro dominio, podemos **instalar nuestro propio servidor** o **contratar el servicio de un proveedor externo**.
- Las **empresas de hosting** suelen proporcionar este servicio.
- El servicio se dará en base a ciertas condiciones:
  - **Espacio total de disco** disponible para los buzones de usuarios.
  - **Máximo número de cuentas de usuario**
  - **Tamaño máximo de los buzones** de usuario.
  - Sistemas de **seguridad**, tipo de **asistencia**, etc.
  - Acceso a **aplicaciones** por los clientes que accedan al servicio
- Google ofrece este servicio a través de **Google Workspace**.

# Desventajas de configurar un servicio de correo

- La **configuración** puede ser **compleja**.
- Los servicios requieren **mantenimiento** regular: ver si están actualizados, protegidos contra amenazas, funcionando correctamente...
  - Actualizaciones, monitoreo, copias de seguridad, resolución de problemas técnicos que puedan aparecer.
- Es **costoso**: Adquisición de hardware y software, ancho de banda, electricidad, salarios de quienes lo mantienen, etc.
- **Responsabilidad**: los servicios de correo son un objetivo atractivo para los atacantes puesto que pueden obtener información sensible y confidencial. Hay que asegurar el servicio y protegerlo contra amenazas: phishing, malware, spoofing, spam...
- **Regulaciones**: Dependiendo de la ubicación y la naturaleza puede haber requisitos legales que hay que cumplir, como la ley CAN-SPAM en EEUU o el Reglamento General de Protección de datos (GDPR) en la Unión europea.
- **Problemas de reputación**: Si el servidor de correo no está correctamente gestionado puede haber problemas de reputación en la entrega de correos. El servicio puede ser marcado como spam o los correos pueden ser rechazados por otros servidores. Por lo que esto puede afectar a la capacidad de entregar correos de manera confiable y profesional.
  - Por ejemplo, si se configura mal un servicio puede ser agregado a una lista negra de spam. Y para salir de ella hay que seguir el procedimiento que quien gestiona la lista imponga.
- En la mayoría de los casos **es mejor contratar este servicio a una empresa externa**.