



Universidad de Jaén

Escuela Politécnica Superior de Linares

Tema 1. Protocolos de aplicación de Internet

Autor: Juan Carlos Cuevas Martínez

Fecha: 10/2024

Asignatura: **Protocolos de transporte**



Tema 1. Protocolos de Aplicación de Internet

Correo electrónico, alojamiento de ficheros, terminal virtual y servicio de nombres de dominio

Asignatura: Protocolos de Transporte
Grado en Ingeniería Telemática

Autor:

Juan Carlos Cuevas Martínez



Competencias

Código	Denominación de la competencia
CB2	Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.
CB3	Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.
CB4	Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
CB5	Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.
CG4	Capacidad de resolver problemas con iniciativa, toma de decisiones, creatividad, y de comunicar y transmitir conocimientos, habilidades y destrezas, comprendiendo la responsabilidad ética y profesional de la actividad del Ingeniero Técnico de Telecomunicación
TEL1	Capacidad de construir, explotar y gestionar las redes, servicios, procesos y aplicaciones de telecomunicaciones, entendidas éstas como sistemas de captación, transporte, representación, procesado, almacenamiento, gestión y presentación de información multimedia, desde el punto de vista de los servicios telemáticos.
TEL4	Capacidad de describir, programar, validar y optimizar protocolos e interfaces de comunicación en los diferentes niveles de una arquitectura de redes
TEL7	Capacidad de programación de servicios y aplicaciones telemáticas, en red y distribuidas

Resultados

Código	Descripción
2	El alumno podrá abordar la resolución del problema de intercomunicación entre procesos que se ejecutan en máquinas conectadas utilizando una red de comunicaciones.
3	Se aprenderá a diferenciar, considerando las características de un protocolo de transporte, cuál resulta más conveniente utilizar según los servicios de telecomunicación habituales.
4	El alumno dominará los protocolos más utilizados en la actualidad para resolver la interconexión de redes de diferente naturaleza, estableciendo la diferenciación entre los protocolos vinculados al transporte de información, los relativos al intercambio de información para el encaminamiento y los auxiliares a los dos tipos descritos anteriormente.
26	Adquirir facilidad para el manejo de especificaciones, reglamentos y normas de obligado cumplimiento.
27	Resolver problemas con iniciativa, toma de decisiones y creatividad.
30	Adquirir facilidad para el manejo de especificaciones, reglamentos y normas de obligado cumplimiento.
35	Comunicar, tanto por escrito como de forma oral, conocimientos, procedimientos, resultados e ideas relacionadas con las telecomunicaciones y la electrónica.

Contenido

1. Introducción
2. Protocolos y Aplicaciones en Internet
3. Correo Electrónico
4. Protocolo Telnet
5. Protocolo FTP
6. Sistema de Nombres de Dominio (DNS)



Bibliografía

Bibliografía básica

- Tanenbaum, A., **«Redes de computadores»**, 4ª y 5ª Ed, Prentice Hall.
- Forouzan, B., **«TCP IP Protocol Suite»**, 4ª Ed, Mc Graw Hill.
- Kozierok, C., **«The TCP IP guide: a comprehensive , illustrated internet protocols reference»**, No Starch Press.
- Reference For Comments (**RFC**)

Bibliografía

Bibliografía complementaria

- Kurose, J., Ross, K., «**Redes de Computadores. Un enfoque descendente basado en Internet**». 4ª Edición, Pearson Education.
- Comer, D., «**Computer networks and internets**», Prentice Hall.
- Stallings W. «**Comunicaciones y redes de computadores**», 7ª Edición. Prentice Hall.
- Cuevas Martínez, J. C., “Programación de aplicaciones de red. Protocolos de Internet cliente-servidor”, Altaria, 2016

1. Introducción

- El objetivo de las redes de ordenadores es **interconectar máquinas distantes entre sí en las que los usuarios ejecutan aplicaciones**, aunque más concretamente se debería decir que interconectan procesos.
 - La RFC6250 define de una manera muy sencilla el modelo de servicio de las redes: **mover bytes entre programas que corren en diferentes ordenadores** (o en el mismo ordenador ocasionalmente).
- Existen muchas aplicaciones que intercambian información sobre una red, pero solamente unas pocas han llegado a ser tan importantes que son usadas hoy en día, y desde hace más de 40 años.

1. Introducción

Resumen de la torre de protocolos OSI

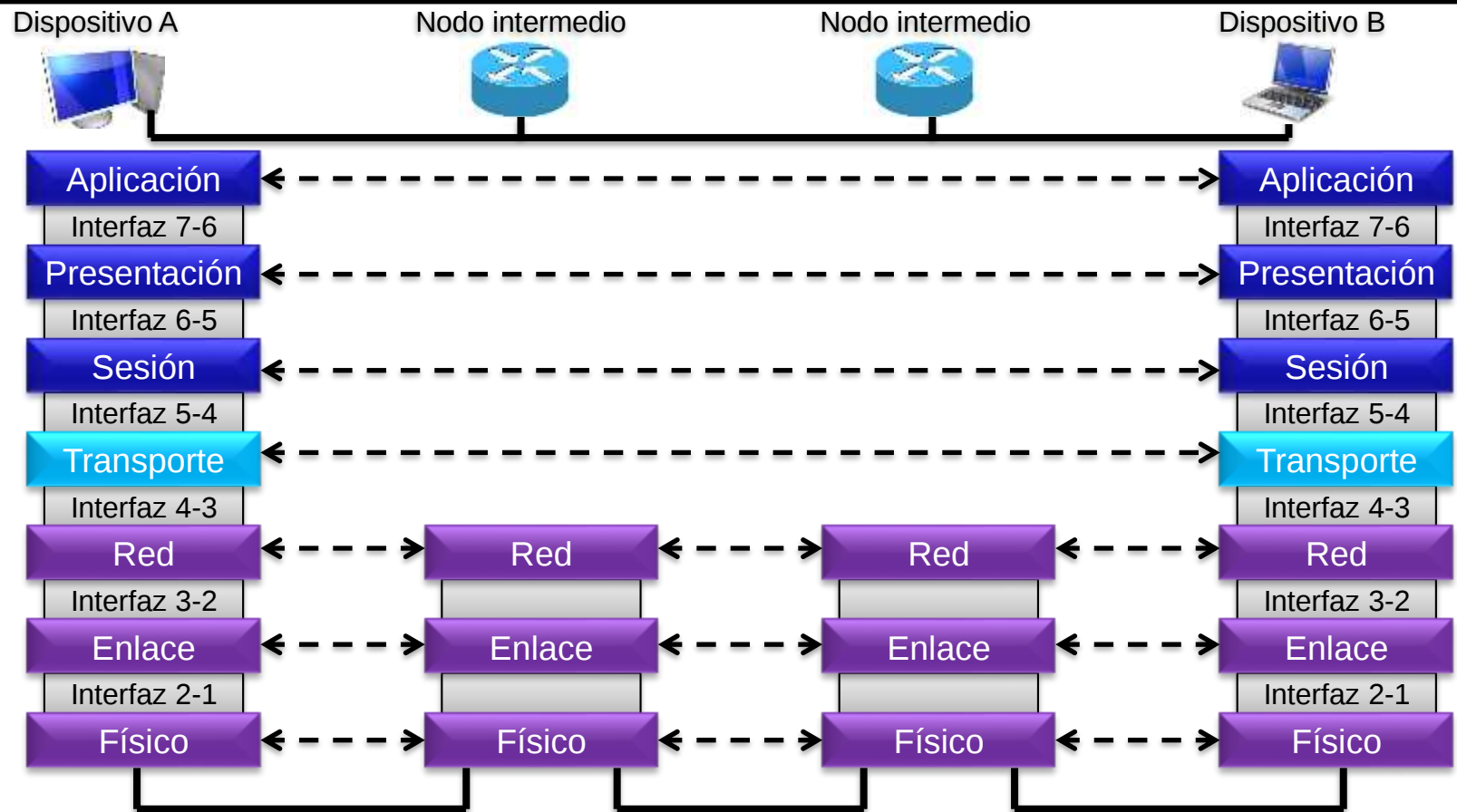
Figura 1.1. Torre de protocolos OSI (*Open System Interconnection*)

Capa 7	Aplicación	Permitir el acceso a los recursos de la red.
Capa 6	Presentación	Traducir, cifrar y comprimir los datos.
Capa 5	Sesión	Establecer, gestionar y terminar sesiones.
Capa 4	Transporte	Proporcionar una forma confiable de comunicación entre procesos, así como la recuperación de errores.
Capa 3	Red	Mover los paquetes desde su origen a su destino para proporcionar un servicio de conmutación virtual entre dispositivos y redes.
Capa 2	Enlace	Organizar los bits en tramas para proporcionar una distribución de los datos tramo a tramo (salto a salto).
Capa 1	Físico	Transmitir los bits sobre un medio y proporcionar las especificaciones mecánicas y eléctricas para poder hacerlo.

1. Introducción

Resumen de la torre de protocolos OSI

Figura 1.2.
Interfaces y comunicación entre entidades con el modelo de capas OSI

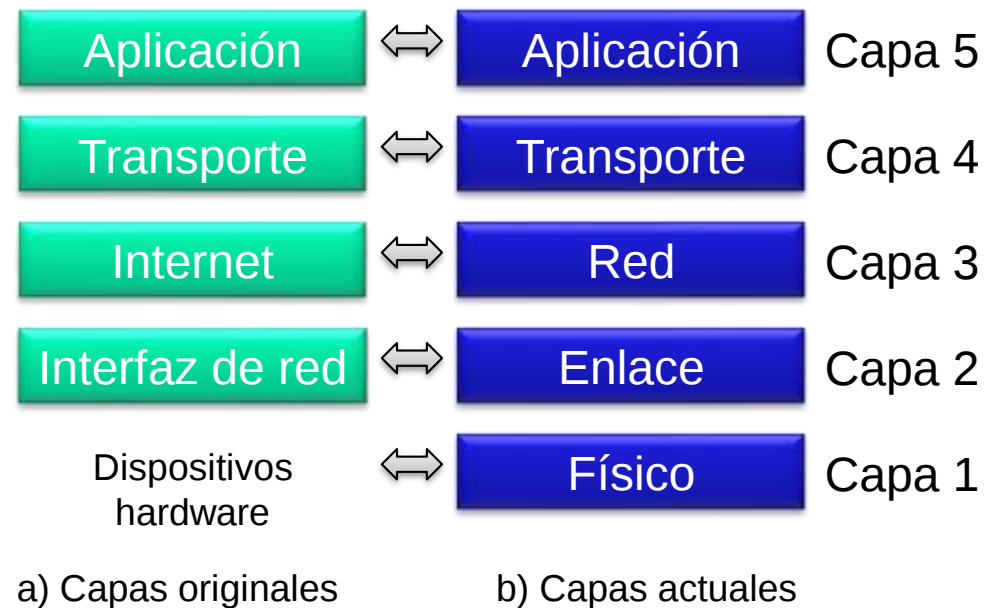


1. Introducción

Protocolos TCP/IP

- La torre de protocolos de TCP/IP fue desarrollada antes de la aparición del modelo OSI y por eso no coincide exactamente con ese modelo.
- En un principio tenía solo cuatro niveles, pero actualmente se definen cinco.
- Como se puede apreciar no aparecen las capas de sesión y presentación, las cuales se incorporan a la capa de aplicación en TCP/IP.

Figura 1.3. Evolución de la torre de protocolos de TCP/IP



1. Introducción

Estructura administrativa de Internet



Internet Corporation for
Assigned Names and Numbers
www.icann.org

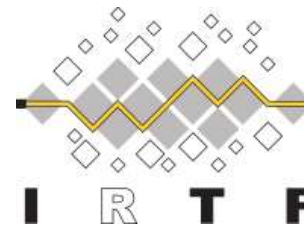
Internet Assigned Numbers
Authority
www.iana.org



Internet Society
www.isoc.org



Internet Architecture Board
www.iab.org



The Internet
Research
Task Force
irtf.org



The Internet
Engineering
Task Force
www.ietf.org

1. Introducción

Notación ABNF

- En la especificación de muchos de los protocolos y estándares de Internet se utiliza una sintaxis normalizada denominada **Augmented Backus-Naur Form** (ABNF), la cual se define en la RFC 5234.
- Esta notación define reglas a través de una sintaxis sencilla y jerárquica. En la antigua versión era común enmarcar los nombres de las reglas entre los símbolos ‘<’ y ‘>’, sin embargo, esta notación omite estos caracteres en la definición, aunque sí se emplean cuando se hace referencia a una regla dentro de un texto.
- Ejemplo de una regla:

`name1 = elements crlf`

1. Si nos refiriéramos al nombre de la regla dentro de un texto, se encerraría entre los símbolos de “menor que” y “mayor que” de la siguiente forma: “La regla <name> se ha definido justo antes”.

1. Introducción

Notación ABNF

Guía breve

- Las reglas se alinean a la izquierda, y si necesitan más de una línea, éstas deben indentarse con respecto a la primera línea.
- Las reglas están formadas por cadenas de **terminales** (caracteres) que son simplemente enteros no negativos, los cuales suelen responder con los valores de algún código concreto como ASCII (es la codificación recomendada).
- Los terminales se definen por uno o más caracteres numéricos con la denominación explícita de la base.
- La especificación de los elementos del núcleo de ABNF están en el Anexo B de la RFC 5234.

Bases definidas en ABNF

b = binario

d = decimal

x = hexadecimal

Ejemplo:

CR = %d13

CR = %x0D

Para concatenar se usa el
'.'

CRLF = %d13.10

1. Introducción

Notación ABNF

Guía breve

- ABNF permite la definición de literales directamente usando las comillas.
 - Para diferenciar entre literales en los que se distinga entre mayúsculas y minúsculas se hace con un prefijo¹:
 - `%s` = case-sensitive
 - `%i` = case-insensitive
 - Si no se especifica nada, para mantener la compatibilidad, se considera que es case-insensitive, como si tuviera puesto `%i`.
1. En la RFC 7405 se introdujo una modificación que permite poner literales que distinguen entre mayúsculas y minúsculas (“*case sensitive*”).

```
Regla1 = %s"abc"
```

```
Regla2 = %s"aBc"
```

Regla1 y Regla2 son diferentes.

`regla = "abc"` es igual que:

`regla = "aBc"`

O que: `regla = %i"aBC"`

Literales con código ASCII:

```
regla = %d97.98.99
```

o

```
regla = %d97 %d98 %d99
```

1. Introducción

Notación ABNF

Operadores

- Concatenación: Regla1 Regla2

uno = %x61

dos = %x62 ; b

regla = uno dos uno ; "aba"

Esto es un comentario.
Su inicio se marca con “;”

- La especificación de ABNF no proporciona una manera de especificar espacios en blanco lineales, ya sean espacios en blanco o tabuladores.
 - Normalmente se soluciona definiendo reglas con el código correspondiente al espacio en blanco o tabulador de un juego de caracteres como el ASCII:
SP = %x20 ; espacio en blanco

1. Introducción

Notación ABNF

Operadores

- Alternativa: Regla1 / Regla2

uno = %x61 ; a

dos = %x62 ; b

regla = uno / dos ; será admitido tanto <uno> como <dos>

- Alternativa incremental: Regla1 =/ Regla2

uno = %x61 ; a

dos = %x62 ; b

regla = uno / dos ; será admitido tanto <uno> como <dos>

tres = %x63 ; c

regla =/ tres ; será igual que regla = uno / dos / tres

1. Introducción

Notación ABNF

Operadores

- Rangos de valores: %c##-##

`DIGIT = %x30-39`

Es equivalente a:

`DIGIT = "0" / "1" / "2" / "3" / "4" / "5" / "6" / "7" / "8" / "9"`

- Grupo secuencial: (Regla 1 Regla 2)
 - Se tratan como un solo elemento y su contenido está ordenado estrictamente.
uno (uno / dos) tres ; podría ser (uno uno tres) o (uno dos tres)
Y
uno uno / dos tres ; sería (uno uno) / (dos tres)

1. Introducción

Notación ABNF

Operadores

- Repetición de variables: $\langle a \rangle^* \langle b \rangle$ elemento
 - $\langle a \rangle$ y $\langle b \rangle$ son valores decimales opcionales que indican al menos $\langle a \rangle$ repeticiones y como máximo $\langle b \rangle$. Los valores por defecto son 0 e infinito respectivamente:
 - $*\langle \text{elemento} \rangle$; indica cualquier número de elementos, incluso cero.
 - $1*\langle \text{elemento} \rangle$; requiere al menos uno
 - $3*3\langle \text{elemento} \rangle$; permite exactamente 3
 - $1*2\langle \text{elemento} \rangle$; permite uno o dos
- Repetición específica: $\langle n \rangle$ elemento (es equivalente a $\langle n \rangle^* \langle n \rangle$ element)
 - Son concretamente $\langle n \rangle$ ocurrencias del elemento, por lo que 2DIGIT son exactamente dos dígitos numéricos y 3ALPHA, tres caracteres alfabéticos.
- Secuencia opcional: [regla]
 - [uno dos] ; es equivalente a $*1$ (uno dos)

1. Introducción

Notación ABNF

Precedencia de operadores

- De mayor a menor:
 - Nombre de regla, cadenas entre comillas, valor terminal
 - Comentarios
 - Rango de valores
 - Repetición
 - Agrupación, Opcional
 - Concatenación
 - Alternativa
- Se recomienda el uso del operador de agrupación para el caso de mezclar concatenaciones y alternativas.

1. Introducción

Notación ABNF

Reglas del núcleo de ABNF (Anexo B RFC 5234)

```
ALPHA = %x41-5A / %x61-7A ; A-Z / a-z
BIT = "0" / "1"
CHAR = %x01-7F ; any 7-bit US-ASCII character, excluding NUL
CR = %x0D ; carriage return
CRLF = CR LF ; Internet standard newline
CTL = %x00-1F / %x7F ; controls
DIGIT = %x30-39 ; 0-9
DQUOTE = %x22 ; " (Double Quote)
HEXDIG = DIGIT / "A" / "B" / "C" / "D" / "E" / "F"
HTAB = %x09 ; horizontal tab
LF = %x0A ; linefeed
LWSP = *(WSP / CRLF WSP) ; this linear-white-space rule (ver RFC 5234)
OCTET = %x00-FF ; 8 bits of data
SP = %x20
VCHAR = %x21-7E ; visible (printing) characters
WSP = SP / HTAB ; white space
```

2. Protocolos y Aplicaciones en Internet

Aplicaciones en Internet

- Hay que diferenciar entre **aplicación de red** y un **protocolo de la capa de aplicación**.
 - Una **aplicación de red** es aquella que proporciona un determinado servicio a los usuarios, ya sean estos personas, máquinas u otras aplicaciones.
 - Un **protocolo de la capa de aplicación** son las normas que rigen el intercambio de información entre los procesos que pertenecen a la aplicación.
- Hay que tener en cuenta que una aplicación puede tener diferentes procesos, y que algunos de ellos pueden usar diferentes protocolos de la capa de aplicación (incluso diferentes protocolos de transporte).

2. Protocolos y Aplicaciones en Internet

Aplicaciones en Internet

Paradigma Cliente/Servidor

- Para proporcionar un servicio se necesitan dos tipos de aplicaciones, una que lo proporcione, o **SERVIDOR**, y otra que permita conectarse a éste y solicitar sus **SERVICIOS**, denominada **CLIENTE**.
- Características del servicio:
 - Suele implicar una relación uno (servidor), a muchos (clientes).
 - Un servicio, normalmente, requiere que el servidor esté continuamente ejecutándose en una máquina conectada a una red, para que así sea accesible a los clientes siempre que estos lo necesiten.
 - Habitualmente, un servicio es ofrecido por una aplicación servidora, pero puede que dicha aplicación ofrezca varios servicios diferentes.
 - También es posible que para ofrecer un servicio adecuadamente sean necesarios varios servidores.

2. Protocolos y Aplicaciones en Internet

Aplicaciones en Internet

Paradigma Cliente/Servidor

- **Servidor:** Es un programa que se ejecuta en una máquina remota (normalmente) y que proporciona un determinado servicio a los clientes.
 - Cuando se inicia, establece un mecanismo para recibir peticiones de los clientes y atenderlas.
 - Un servidor no suele iniciarse hasta que se le es requerido.
- **Cliente:** Es una programa ejecutado en la máquina local que permite solicitar un servicio determinado a un servidor remoto.
 - Un cliente es un programa finito: cuando un usuario, u otra aplicación, lo inicia, éste estará activo hasta que el servicio solicitado se haya completado o sea apagado por otro motivo.

2. Protocolos y Aplicaciones en Internet

Aplicaciones en Internet

Paradigma Cliente/Servidor

- **Proceso de solicitud de un servicio en Internet:**

- Un cliente establece un canal de comunicación con un servidor usando la dirección IP de éste último y un puerto “bien conocido” (**well-known**), que está asociado al servicio y en el cuál está escuchando el servidor.
- Una vez establecido este canal, el cliente enviará su petición y recibirá una respuesta, repitiéndose este proceso hasta que el servicio quede completado.
- Normalmente, este proceso es finito.

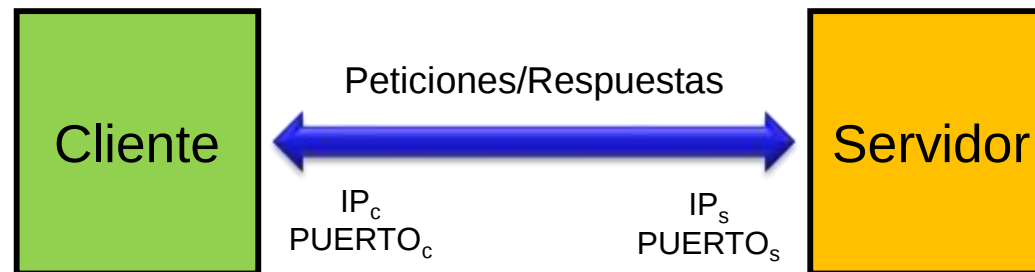


Figura 1.4. Modelo funcional de solicitud de servicio

2. Protocolos y Aplicaciones en Internet

Aplicaciones en Internet

Peer to Peer (P2P)

- Existen aplicaciones que en sí no son solamente clientes o servidores, sino que por su carácter distribuido **es necesario que se comporten como ambos tipos a la vez.**
- Estas se denominan aplicaciones **Peer to Peer o P2P (Igual a Igual o entre pares).**
- Todos los participantes ofrecen y demandan el servicio.
- Servicios habituales: **compartición de archivos, multi-conferencia.**
- Ejemplos: Skype, eDonkey, eMule, Ares o BitTorrent

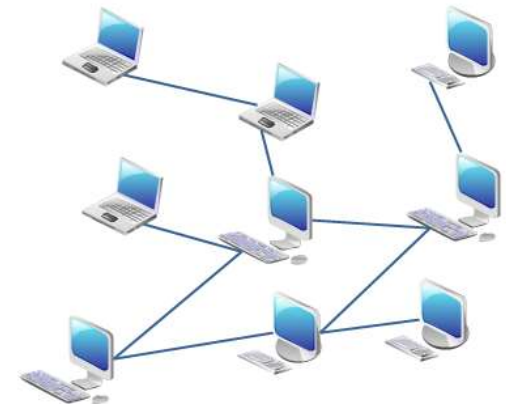


2. Protocolos y Aplicaciones en Internet

Aplicaciones en Internet

Peer to Peer (P2P)

- El conjunto de aplicaciones P2P y usuarios conforma una nueva red que proporciona solamente servicios a sus miembros.
- El principal problema de las aplicaciones P2P es el descubrimiento de los demás miembros.
- A veces se apoyan en:
 - Servidores que proporcionan el acceso a otros miembros.
 - Listas iniciales de miembros a partir de los cuales ir creciendo.



2. Protocolos y Aplicaciones en Internet

Aplicaciones en Internet

Procesado de peticiones

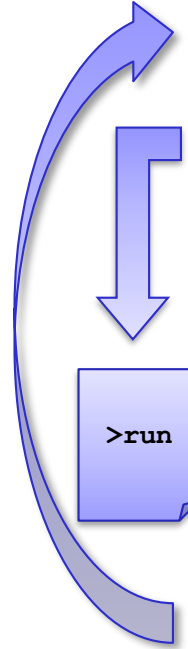
- **Tipo de ejecución:** tanto los clientes como los servidores pueden ejecutar sus procesos de dos maneras: iterativa o concurrente.

ITERATIVA

- 
1. Esperar a recibir una petición de un cliente.
 2. Procesar la petición de cliente.
 3. Devolver la respuesta al cliente que envió la petición.
 4. Volver al paso 1.

Si el paso 2 se alarga en el tiempo otros clientes se verán perjudicados por el retraso.

CONCURRENTE

- 
1. Esperar a recibir una petición de un cliente
 2. Iniciar una nueva instancia del proceso servidor para atender la petición de un nuevo cliente. Esta nueva instancia se ejecuta concurrentemente y no es necesario esperar a que termine el servicio a clientes que estén siendo atendidos.
 3. Volver al paso 1.

2. Protocolos y Aplicaciones en Internet

Normalización de los protocolos en Internet

- La mayoría de los protocolos de aplicación usados en Internet, y en concreto todos los que se verán en la asignatura, han sido desarrollados en el seno de los grupos de trabajo del **Internet Engineering Task Force (IETF)**.
- Los documentos que describen todos los aspectos de Internet se denominan **Request for Comments** o **RFC**.
- Para consultar el estado de cualquier documento lo más útil es acudir al índice mantenido por el IETF.

<http://www.rfc-editor.org/>

2. Protocolos y Aplicaciones en Internet

Servicios, protocolos y aplicaciones de Internet

- Las aplicaciones como Outlook, Thunderbird, Edge, Firefox, Chrome, Safari, FileZilla, etc.:
 - Permiten el acceso a uno, o varios, servicios de comunicación, información y datos.
 - Usan protocolos como SMTP, POP3, IMAP, HTTP o FTP para el intercambio de información.
 - Utilizan a nivel de transporte TCP (HTTP/3 usa QUIC). Aunque son independientes de la capa de transporte, sí delegan en ésta la entrega confiable de los datos.



2. Protocolos y Aplicaciones en Internet

Servicios, protocolos y aplicaciones de Internet

- Correo electrónico: se usan los protocolos Simple Mail Transfer Protocol (**SMTP**) para enviar los correos y para leerlos del buzón: Post Office Protocol versión 3 (**POP3**) o Internet Message Access Protocol (**IMAP**).
- Servicio de transferencia de ficheros: ofrecido por el protocolo File Transfer Protocol (**FTP**) o el Trivial File Transfer Protocol (**TFTP**) que se usa sobre UDP.
- Servicio de nombres de dominio: o Domain Name System (**DNS**) que usa el protocolo del mismo nombre (suele funcionar sobre UDP pero también emplea en ciertos casos TCP).
- Servicio *world wide web*: basado fundamentalmente en el protocolo Hyper-Text Transfer Protocol (**HTTP**). Emplea TCP o QUIC (HTTP/3) Se verá en el tema 2.

2. Protocolos y Aplicaciones en Internet

- Un **protocolo de la capa de aplicación** define la manera en la que los procesos que forman una determinada aplicación o servicio, intercambian mensajes entre sí.
- Concretamente, un protocolo de la capa de aplicación define:
 - El tipo de mensajes intercambiados, por ejemplo, de petición, respuesta, error, etc.
 - La sintaxis de los distintos tipos de mensaje, especificando los campos, la longitud de los mismos, delimitadores, etc.
 - La semántica de los campos, es decir, lo que significa la información que lleva cada campo y lo que implica en la comunicación.
 - Las reglas que rigen cuando un proceso debe enviar y/o responder mensajes, así como la manera en la que estos deben ser tratados.

3. Correo electrónico

- El correo electrónico es uno de los servicios más populares y usados en Internet.
- En sus primeros días fue tan solamente una forma de comunicarse entre los usuarios de los grandes ordenadores (mainframes).
- Un poco más tarde, antes de TCP/IP e Internet (comienzos de los años 70 del siglo XX), dentro de ARPAnet, ya comenzó a ser un servicio de intercambio de mensajes entre usuarios de diferentes máquinas (RFCs 95 y 155).
- Durante 10 años sufrió continuas mejoras y cambios (1971 Mail Box Protocol RFC 196, 1980 Mail Transfer Protocol RFC 772), estando incluso apoyado en el protocolo FTP.
- Esta evolución llevó en el año 1981 a la especificación del **Simple Mail Transfer Protocol** (RFC 788, en noviembre, aunque se toma siempre la 821 de agosto de 1982 como la inicial), o **SMTP** como es más conocido.

SMTP
POP3
IMAP



3. Correo electrónico

Modelo funcional

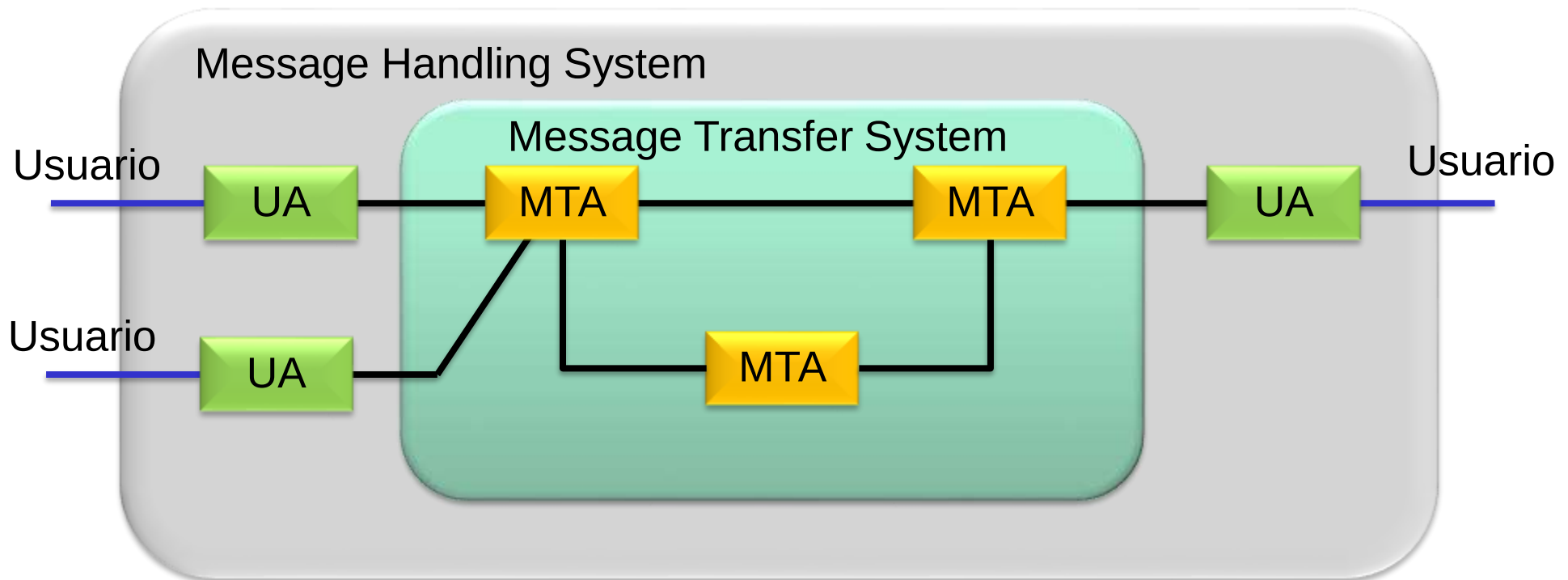


Figura 3.1. Modelo funcional del Sistema de Manejo de Mensajes

3. Correo electrónico

Modelo funcional

Sistema de tratamiento de mensajes (MHS)

- **Agente de usuario (UA: User Agent):**
Es una aplicación cliente cuya función es:
 - Ayudar al remitente-destinatario a gestionar sus mensajes.
 - Interaccionar con el MTS, entregando y recibiendo mensajes.
 - Puede proporcionar servicios como procesamiento de textos o interfaz de usuario.
- Ejemplos: Microsoft Outlook, Mozilla Thunderbird o Apple Mail.

Sistema de transferencia de mensajes (MTS)

- **Agente de transferencia de mensajes (MTA: Mail Transfer Agent o Servidor de correo).** Un MTA tiene como misión:
 - Aceptar mensajes que remiten con agentes de usuario y almacenarlos, en lo que comúnmente se denomina “buzón” del usuario.
 - Encaminar los mensajes hacia los MTA destino.
 - Entregar los mensajes a los agentes de usuario.

3. Correo electrónico

Modelo funcional

Agente de usuario

- Aspectos configurables de un Agente de Usuario:
 - MTAs (Servidores) con los que conectar.
 - Cada cuanto recoge o envía mensajes.
 - Formato de los mensajes.
 - Firma de los mensajes.
 - Cifrado de Mensajes.
 - Libreta de Direcciones.



3. Correo electrónico

Protocolos de Correo Electrónico

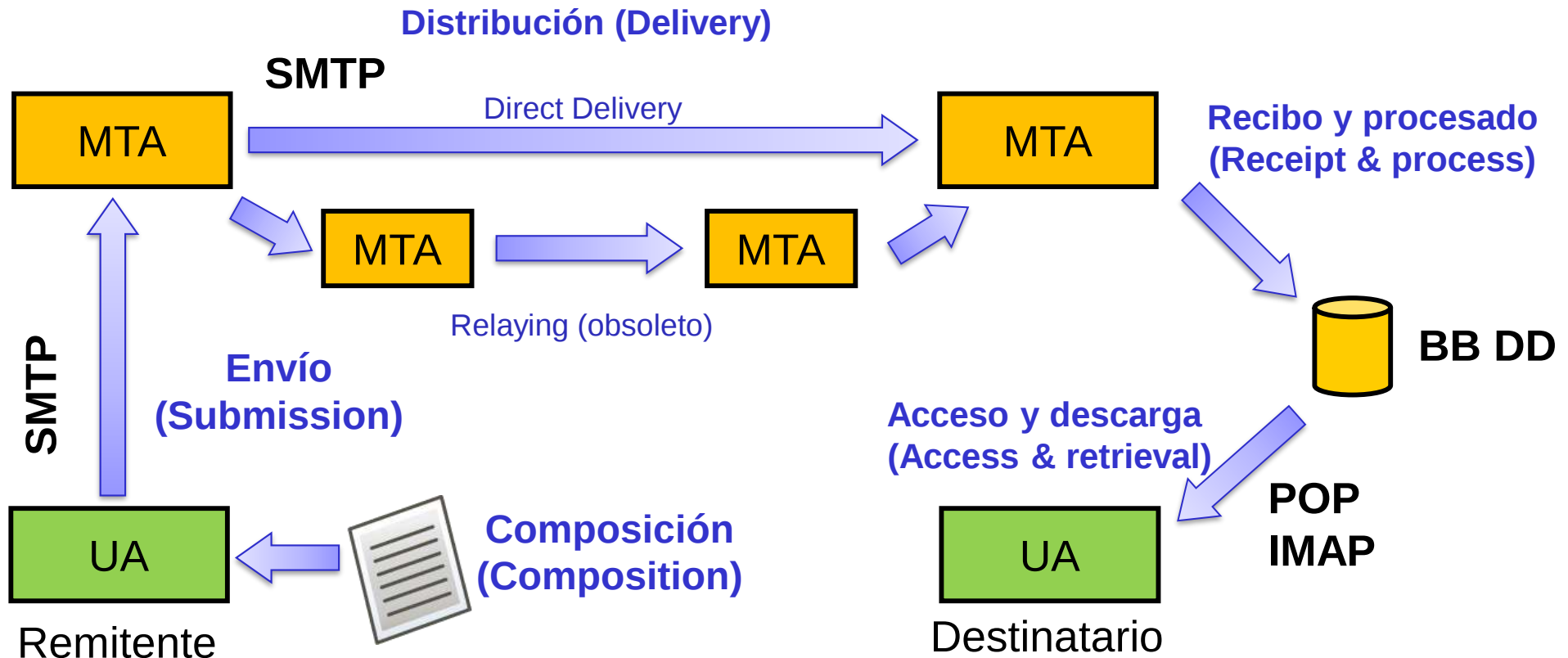


Figura 3.2. Ciclo de vida de un correo electrónico

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

- Definido inicialmente en la RFC 821 en el año 1982, (después RFC 2821 año 2001 y actualmente RFC 5321, año 2008).
- Hace uso de conexiones TCP en el puerto 25.
- La potencia y el éxito de SMTP se basa en su simplicidad, y es el estándar a emplear.
- En un principio se diseñó para aportar ciertos servicios que hoy no se implementan sobre SMTP, tales como la mensajería instantánea.
- *NOTA: La RFC 8314 obliga al uso implícito de TLS para POP3, SMTP e IMAP.*
 - *En SMTP establece el puerto 465 (el mecanismo STARTTLS establece el 587)*

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

- Se basa en un diálogo alternativo de preguntas (peticiones) y respuestas:
 - El cliente envía comandos y el servidor responde con un mensaje de estado que indica si el comando se ha podido realizar satisfactoriamente o ha habido algún problema.
 - El orden de los comandos necesarios para enviar un correo es importante.
 - Los comandos y mensajes de estado están formateados en ASCII de 7 bits.
 - Todos los comandos y los mensajes de estado están terminados por los caracteres retorno de carro (CR: ASCII 0Dh) y avance de línea (LF: ASCII 0Ah), apareciendo en la RFC como `CRLF` (ABNF).
 - Los mensajes de estado están formados por un código numérico de estado de tres dígitos y una cadena de texto.

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Modelo funcional básico

- SMTP transporta **Objetos de Correo** (*Mail objects*): Un Objeto de correo contiene un sobre (*envelope*) y contenido (*content*).

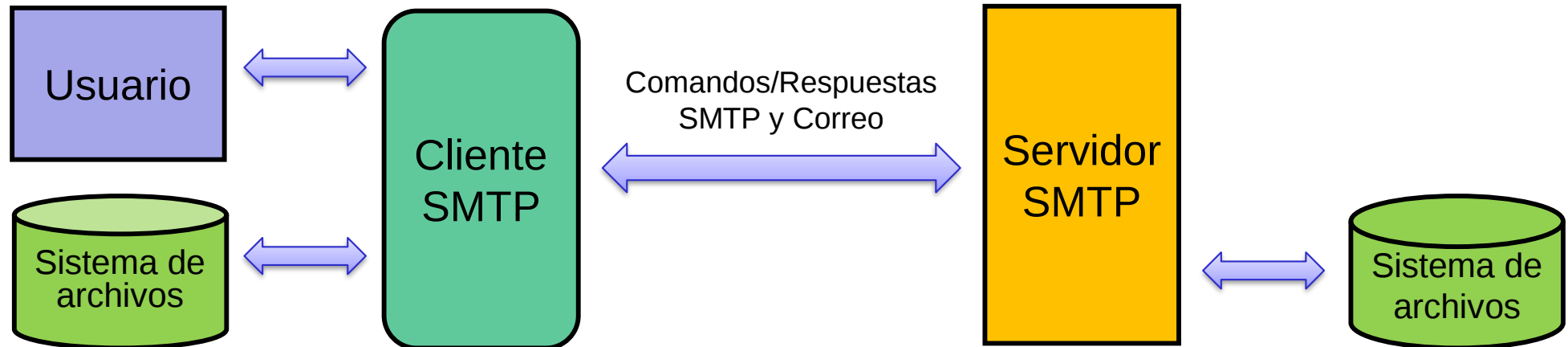


Figura 3.3. Modelo funcional básico del protocolo SMTP

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Objetos de correo

- **Sobre:** El sobre se envía como una serie de unidades de protocolo SMTP y consiste en:
 - La dirección del remitente (al que deben enviarse los informes de error).
 - La dirección de uno o más destinatarios.
 - Extensiones opcionales del protocolo.
- **Contenido:** éste es enviado en la unidad SMTP DATA, y tiene dos partes, la sección de cabecera y el cuerpo.
 - La **cabecera** se codifica según la RFC 5322.
 - El **cuerpo**, si se estructura, se hace según el formato marcado por las **extensiones MIME** (RFC 2045).
 - El **contenido es texto por naturaleza** (US-ASCII) pero la extensión **8BITMIME** (RFC 6152) puede hacer que el cuerpo no siga esta restricción, pero eso nunca podrá afectar a la cabecera.

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Comandos SMTP

NEMÓNICO	SINTAXIS ¹	DESCRIPCIÓN
HELO	"HELO" SP Domain CRLF	Identifica al remitente, actualmente solamente se usa si no se requiere la funcionalidad extendida del servidor. Se mantiene por compatibilidad. Los clientes deben usar siempre EHLO (ver a continuación)
EHLO	"EHLO" SP (Domain / address-literal) CRLF	Permite usar las extensiones de SMTP definidas a partir de la RFC 1425 y subsecuentes estándares. Todos los servidores deben soportar este comando aunque no implementen ninguna extensión. El dominio debe ser un nombre primario del host como el que se obtiene de una petición a un registro de recursos de DNS o una dirección IP literal si no dispone de dominio.

1: Sigue la notación ABNF.

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Comandos SMTP

NEMÓNICO	SINTAXIS	DESCRIPCIÓN
MAIL	"MAIL FROM:" Reverse-path [SP <mail-parameters>] CRLF	Comienza la transacción de correo e identifica al remitente
RCPT	"RCPT TO:" ("<Postmaster@" Domain ">" / "<Postmaster>" / Forward-path) [SP Rcpt-parameters] CRLF	Identifica al destinatario. Pueden enviarse múltiples comandos RTCP, permitiendo el envío del mismo correo a los destinatarios indicados con este comando
DATA	"DATA" CRLF	Indica que el remitente está listo para transmitir una serie de líneas de texto, cada una finalizada con CRLF. Una línea que únicamente contiene .CRLF indica el fin de datos
QUIT	"QUIT" CRLF	Finaliza la sesión SMTP
RSET	"RSET" CRLF	Aborta la transacción en curso y reinicia la sesión

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Comandos SMTP

NEMÓNICO	SINTAXIS	DESCRIPCIÓN
VRFY	"VRFY" SP String CRLF	(Verify) Confirma que el nombre es un destinatario válido
EXPN	"EXPN" SP String CRLF	Lista los componentes de una lista de correo
NOOP	"NOOP" [SP String] CRLF	Responde con un código de asentimiento positivo (250 OK)
HELP	"HELP" [SP String] CRLF	Muestra ayuda sobre el servidor o sobre más específicamente sobre lo solicitado en la cadena si fuera posible (esta opción no es obligatoria)

Implementación mínima

Cualquier receptor del protocolo SMTP **debe** implementar los siguientes comandos: EHLO, HELO, MAIL, RCPT, DATA, RSET, NOOP, QUIT y VRFY

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Respuestas y códigos de respuesta

En SMTP cada respuesta está formada por un código de tres dígitos, un texto opcional y CRLF, herencia de FTP. Al primer dígito se le denomina 'x', 'y' al segundo y 'z' al tercero. El primer dígito informa del éxito o fallo de la operación en términos generales, el segundo divide a los mensajes entre diferentes grupos funcionales y el tercer dígito indica un mensaje específico dentro de ese grupo funcional.

FORMATO	SIGNIFICADO	DESCRIPCIÓN
1yz	Respuesta Positiva Preliminar	El comando ha sido aceptado y su proceso está en curso. SMTP realmente no responde con estos códigos
2yz	Respuesta de Cumplimentación Positiva	El comando ha sido procesado y completado satisfactoriamente
3yz	Respuesta Positiva Intermedia	El comando se ha aceptado pero su proceso se ha retrasado a la espera de más información
4yz	Respuesta de Completitud Negativa Transitoria	El comando no fue aceptado y ninguna acción fue llevada a cabo, pero el error es temporal y puede ser intentado de nuevo
5yz	Respuesta de Completitud Negativa Permanente	El comando no fue aceptado y ninguna acción fue llevada a cabo, intentar de nuevo el comando probablemente provocará el mismo error

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Respuestas multi-línea

- Algunos comandos como EHLO o EXPN tienen más de una línea de respuesta.
- En ese caso, cada línea de respuesta tiene un guion después del código de tres dígitos para indicar que le sigue otra línea. Así pues, para indicar el fin de líneas de respuestas, la última línea no lleva el guion.

```
220 150.214.179.156 ESMTP server ready.  
EHLO host  
250-150.214.179.156 Hello host; ESMTPs are:  
250-TIME  
250-SIZE 0  
250-HELP
```

Figura 3.4. Ejemplo de respuesta multi-línea al comando EHLO

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Ejemplo de comandos SMTP



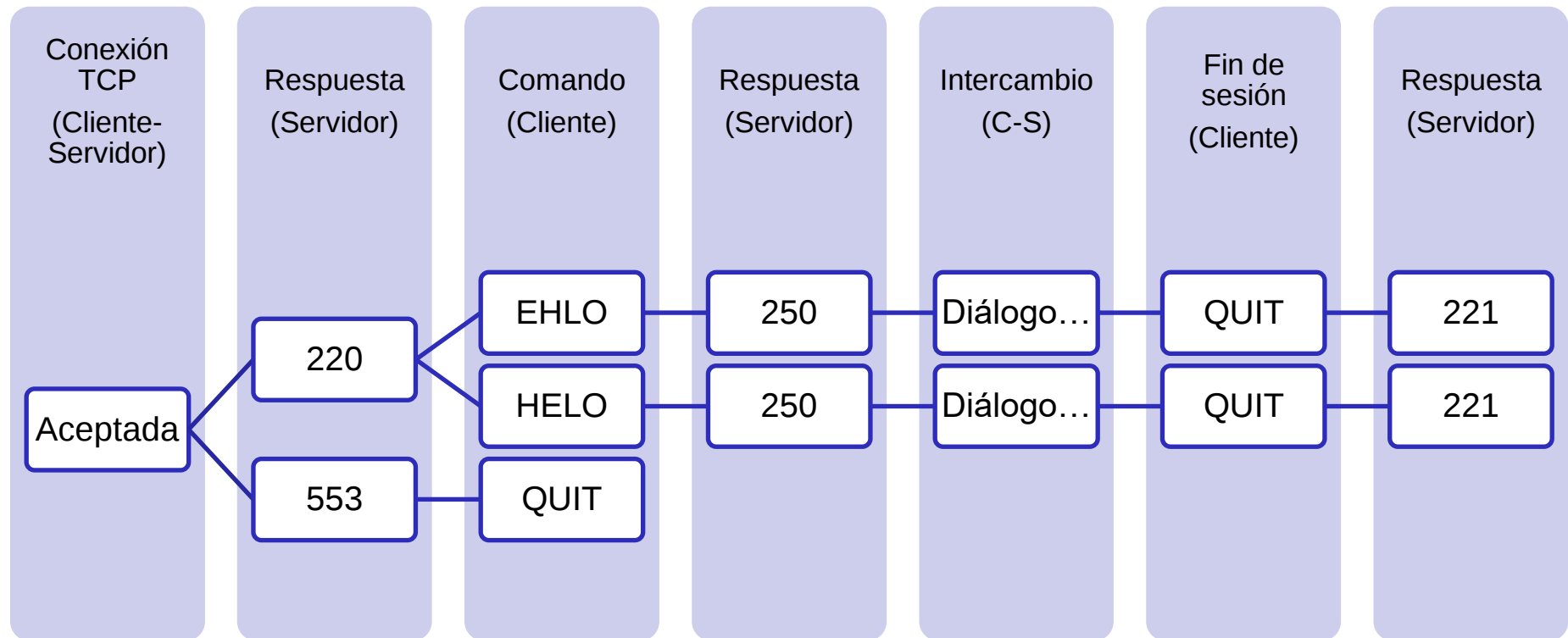
```
220 MERCUR SMTP-Server (v3.20.01 Unregistered) for Windows 95/98 ready at Mon,
8 Mar 2002 01:07:59 +0100
VRFY uno@telematica.net
250 <uno@telematica.net>
EXPN lista@telematica.net
250- <dos@telematica.net>
250 <uno@telematica.net>
NOOP
250 OK
HELO
250 jose Hello 192.168.1.202
MAIL FROM: uno@telematica.net
250 <uno@telematica.net>, sender ok
RSET
250 Reset State
RCPT TO: dos@telematica.net
501 invalid Command in this state
█
```

Figura 3.5. Ejemplo de diálogo para los comandos VRFY, EXPN, NOOP y RSET

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Ciclo de vida de una sesión SMTP



3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Diálogo para el envío de un correo electrónico

S: <wait for open connection>
C: <open connection to server>
S: 220 Innosoft.com SMTP service ready
C: HELO dbc.mtview.ca.us
S: 250 Innosoft.com
C: MAIL FROM:<mrose@dbc.mtview.ca.us>
S: 250 sender <mrose@dbc.mtview.ca.us> OK
C: RCPT TO:<ned@innosoft.com>
S: 250 recipient <ned@innosoft.com> OK
C: RCPT TO:<dan@innosoft.com>
S: 250 recipient <dan@innosoft.com> OK
C: DATA
S: 354 enter mail, end with line containing only "."
...
C: .
S: 250 message sent
C: QUIT
S: 221 goodbye

HELO/EHLO solamente una vez al inicio de la comunicación.

Envío de correo. Este proceso se puede repetir todas las veces que se desee:

- Sobre:
 - MAIL: remitente
 - RCPT: destinatarios (uno o varios).
- Contenido. Se inicia con DATA y se termina con <CRLF>.<CRLF>. Entre estos dos elementos va todo el correo del usuario.

QUIT, fin de la sesión

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Transparencia

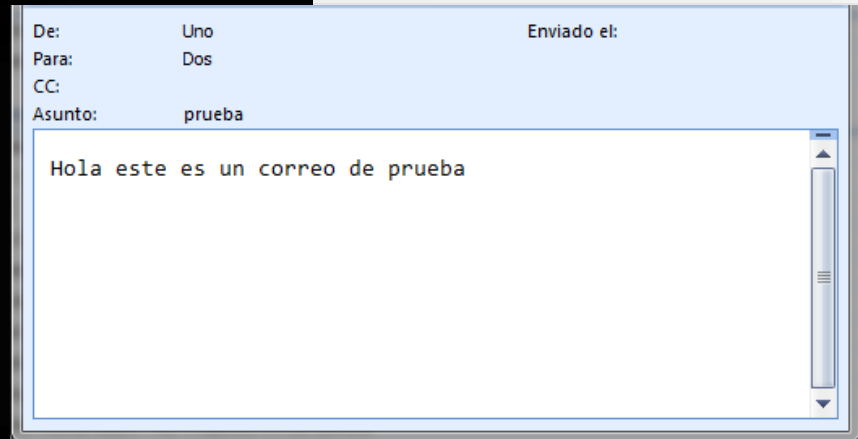
- Dado que la secuencia “<CRLF>.<CRLF>” termina le envío del correo, se debe evitar que aparezca en el cuerpo del mensaje.
 - Los usuarios no tienen por qué estar pendientes de esta coincidencia.
 - Por eso se implementa un mecanismo de transparencia.
- **Mecanismo de transparencia:**
 - Antes de enviar una línea de texto, el cliente SMTP comprueba si el primer carácter de la línea es un punto, si es así, se inserta un punto adicional al inicio de la línea.
 - Cuando una línea de correo es recibida por un servidor SMTP, se comprueba esa línea, si está compuesta de un solo punto, se trata como el indicador de fin de correo. Sin embargo, si el primer carácter es un punto, pero hay más caracteres en la línea, se borra dicho primer carácter.

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Ejemplo de intercambio SMTP (envío de un correo electrónico)

```
220 150.214.xxx.xxx ESMTP server ready.  
HELO host  
250 150.214.xxx.xxx Hello, host.  
MAIL FROM:<uno>  
250 Sender OK - send RCPTs.  
RCPT TO:<dos>  
250 Recipient OK - send RCPT or DATA.  
DATA  
354 OK, send data, end with CRLF.CRLF  
subject:prueba  
to:Dos  
from:Uno  
  
Hola este es un correo de prueba  
.  
250 Data received OK.  
QUIT  
221 150.214.xxx.xxx Service closing channel.
```



3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Extensiones de servicio de SMTP

- Surgieron a partir de 1990 con la intención de aportar nueva funcionalidad al servicio definido en la RFC 821.
- El modelo de **extensiones de servicio** proporciona al cliente y al servidor la capacidad de negociar nuevas capacidades para el servicio.
- Los servidores actuales **deben** dar soporte a **EHLO** (aunque no implementen ninguna extensión).
- Por compatibilidad hacia atrás, tanto clientes como servidores, **deben** dar soporte a **HELO**.
- Los clientes deben iniciar la comunicación preferiblemente con EHLO.

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Extensiones de servicio de SMTP

- Las extensiones que empiecen por “X” son solamente locales, y se pueden usar si ambos extremos están de acuerdo.
- Una extensión que no empiece por “X” debe haber sido registrada previamente (Standard, Standards-Track, o IESG-approved Experimental SMTP service extension) por el IANA.

Para poder hacer uso de las extensiones SMTP se debe iniciar la sesión con EHLO (Extended Hello), si el servidor las soporta devolverá 250, en otro caso devolverá 500.

```
220 smtp.ujaen.es ESMTP
EHLO host
250-smtp.ujaen.es
250-8BITMIME
250-SIZE 20971520
250 STARTTLS
```

Figura 3.6. Ejemplo de las extensiones de servicio del servidor de correo de la Universidad de Jaén.

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Extensiones de servicio de SMTP – respuesta comando EHLO

```
ehlo-ok-rsp = ( "250" SP Domain [ SP ehlo-greet ]  
CRLF )  
/ ( "250-" Domain [ SP ehlo-greet ] CRLF  
*( "250-" ehlo-line CRLF )  
"250" SP ehlo-line CRLF )  
  
ehlo-greet = 1*(%d0-9 / %d11-12 / %d14-127)  
; string of any characters other than CR or LF  
  
ehlo-line = ehlo-keyword *( SP ehlo-param )  
  
ehlo-keyword = (ALPHA / DIGIT) *(ALPHA / DIGIT /  
"-")  
; additional syntax of ehlo-params depends on  
; ehlo-keyword  
  
ehlo-param = 1*(%d33-126)  
; any CHAR excluding <SP> and all  
; control characters (US-ASCII 0-31 and 127  
; inclusive)
```

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Extensiones de servicio de SMTP

PALABRA CLAVE	EXTENSIÓN	DOCUMENTO
SIZE	Declaración del tamaño de máximo de los mensajes que admite el servidor	RFC 1870
8BITMIME	Soporte al MIME de 8 bits (se consideran todos los bits de un byte pero no es para contenido binario puro)	RFC 6152
AUTH	Autorización con Simple Authentication and Security Layer (SASL). Usa un nuevo comando AUTH.	RFC 4954
DSN	Notificación de estado de envío	RFC 3461
ENHANCEDSTATUSCODES	Códigos de estado mejorados	RFC 2034 RFC 3463
PIPELINING	Comandos en pipeline	RFC 2920
STARTTLS	Iniciar el uso de la capa de seguridad TLS (Transport Layer Security)	RFC 3207
SMTPUTF8	Direcciones de correo con internacionalización	RFC 6531

La lista completa es mantenida por el IANA puede verse en:
<http://www.iana.org/assignments/mail-parameters/mail-parameters.xhtml>

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Extensiones de servicio de SMTP – Códigos de estado mejorados

- **Notación:**

```
status-code ::= class "." subject "." detail
class ::= "2" / "4" / "5"
subject ::= 1*3digit
detail ::= 1*3digit
```

- Solo se aplican a los códigos que comienzan con “2”, “4” o “5”, y el nuevo código extendido también emplea ese mismo carácter de comienzo.
- Por ejemplo, el código 250, anteriormente enviado por el servidor tanto para un MAIL o RCPT correctos, con los códigos extendidos varía:

```
MAIL ...
250 2.1.0 ...
RCPT ...
250 2.1.5 ...
```

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Extensiones de servicio de SMTP - Ejemplos

```
S: <wait for connection on TCP port 25>
C: <open connection to server>
S: 220 dbc.mtview.ca.us SMTP service ready
C: EHLO ymir.claremont.edu
S: 250-dbc.mtview.ca.us says hello
S: 250 ENHANCEDSTATUSCODES
C: MAIL FROM:<ned@ymir.claremont.edu>
S: 250 2.1.0 Originator <ned@ymir.claremont.edu> ok
C: RCPT TO:<mrose@dbc.mtview.ca.us>
S: 250 2.1.5 Recipient <mrose@dbc.mtview.ca.us> ok
C: RCPT TO:<nosuchuser@dbc.mtview.ca.us>
S: 550 5.1.1 Mailbox "nosuchuser" does not exist
C: RCPT TO:<remoteuser@isi.edu>
S: 551-5.7.1 Forwarding to remote hosts disabled
S: 551 5.7.1 Select another host to act as your forwarder
C: DATA
S: 354 Send message, ending in CRLF.CRLF.
...
C: .
S: 250 2.6.0 Message accepted
C: QUIT
S: 221 2.0.0 Goodbye
```

Códigos de
estado
mejorados

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Extensiones de servicio de SMTP - Ejemplos

```
S: 220-smtp.example.com ESMTP Server
C: EHLO client.example.com
S: 250-smtp.example.com Hello client.example.com
S: 250-AUTH GSSAPI DIGEST-MD5
S: 250-ENHANCEDSTATUSCODES
S: 250 STARTTLS
C: STARTTLS
S: 220 Ready to start TLS
... TLS negotiation proceeds, further commands
protected by TLS layer ...
C: EHLO client.example.com
S: 250-smtp.example.com Hello client.example.com
S: 250 AUTH GSSAPI DIGEST-MD5 PLAIN
C: AUTH PLAIN dGVzdAB0ZXN0ADEyMzQ=
S: 235 2.7.0 Authentication successful
```

Autenticación

3. Correo electrónico

Simple Mail Transfer Protocol (SMTP)

Extensiones de servicio de SMTP - Ejemplos

```
S: <wait for open connection>
C: <open connection to server>
S: 220 innosoft.com SMTP service ready
C: EHLO dbc.mtview.ca.us
S: 250-innosoft.com
S: 250 PIPELINING
C: MAIL FROM:<mrose@dbc.mtview.ca.us>
C: RCPT TO:<ned@innosoft.com>
C: RCPT TO:<dan@innosoft.com>
C: RCPT TO:<kvc@innosoft.com>
C: DATA
S: 250 sender <mrose@dbc.mtview.ca.us> OK
S: 250 recipient <ned@innosoft.com> OK
S: 250 recipient <dan@innosoft.com> OK
S: 250 recipient <kvc@innosoft.com> OK
S: 354 enter mail, end with line containing only "."
...
C: .
C: QUIT
S: 250 message sent
S: 221 goodbye
```

Pipelining

3. Correo electrónico

Formato de los mensajes de Internet

Evolución

- El formato para los correos electrónicos se definió aparte, en la RFC 822, y no dentro de la especificación de SMTP, con objeto de que éste pudiera ser extendido y desarrollado por separado, a la vez que usado por otros protocolos, como de hecho así ocurre.
- La **RFC 822 Standard for the format of ARPA internet text messages** se revisó en 2001 dando la **RFC 2822 Internet Message Format (IMF)**.
- En 2008 se lanzó una nueva versión dando como resultado la **RFC 5322**.
 - La RFC 5322 se actualizó en 2013 por la **RFC 6854: Update to Internet Message Format to Allow Group Syntax in the "From:" and "Sender:" Header Fields**
- Aun así, al formato se le sigue denominando como «tipo RFC822».

3. Correo electrónico

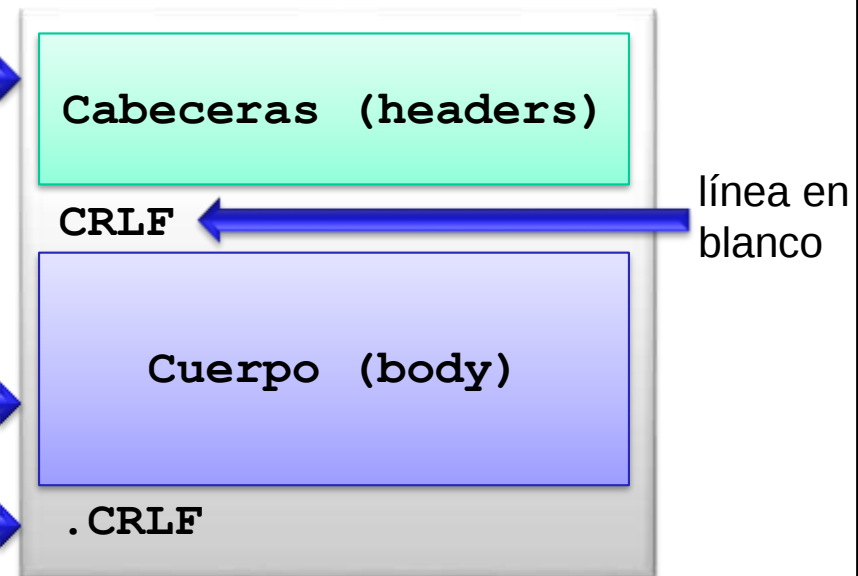
Formato de los mensajes (RFC 822)

- El formato descrito en la **RFC 822** propone una codificación distinta a los campos de bits vistos para otros protocolos. En vez de estos, usa **texto plano** en formato ASCII (concretamente de 7 bits de Estado Unidos, US-ASCII).
- Formato de una línea:
 - **Siempre** terminada por el retorno de carro y el avance de línea (**CRLF**).
 - **Deben** tener un **máximo de 998 caracteres** y **deberían** tener **no más de 78 caracteres** (excluidos **CRLF**).
- El IMF define solamente el formato para mensajes de texto, para el envío de otros tipos de contenido se crearon las **extensiones MIME** (RFCs 2045, 2046 y 2049), que se verán más adelante.

3. Correo electrónico

Formato de los mensajes (RFC 822)

- **Líneas de cabecera:** Información de control para los clientes:
"nombre cabecera:" valor CRLF
Ejemplos: To:, From:, Subject:
- **Cuerpo:** Datos que quiera el usuario, deben mantener las consideraciones de tamaño.
- **Fin del mensaje:** Línea que contiene solo ".CRLF". Esto lo marca el protocolo SMTP
- **TODAS** las líneas irán terminadas por CRLF.



3. Correo electrónico

Formato de los mensajes (RFC 822)

Definición según la RFC en ABNF

```
message      =      (fields / obs-fields)
                  [CRLF body]
body         =      (* (*998text CRLF) *998text) / obs-body
text         =      %d1-9 / ; Characters excluding CR
                  %d11 / ; and LF
                  %d12 /
                  %d14-127
```

NOTA: No se aconseja el uso de los caracteres de control del código US-ASCII (valores 1 al 8, 11, 12 y 14 al 31) porque la interpretación correcta en el terminal receptor no está garantizada.

3. Correo electrónico

Formato de los mensajes (RFC 822)

Cuerpo (body)

- Tanto el carácter CR como el LF solamente pueden aparecer en el cuerpo unido en la forma de CRLF.
- La longitud máxima de una línea será de 998 caracteres excluyendo el <CRLF> y debería estar limitada a 78 caracteres.
 - Las extensiones MIME pueden modificar esta limitación.

3. Correo electrónico

Formato de los mensajes (RFC 822)

Cabeceras

- Las cabeceras se estructuran de la siguiente manera (siempre hay que tener en cuenta la limitación 998 caracteres más el <CRLF>).
- Si fueran necesarias más de una línea para el valor de la cabecera, cada nueva línea se marcaría al inicio con un espacio en blanco (normalmente se usa el carácter tabulador para esto) denominado *folding white space* (<FWS>):

`"cabecera : " valor CRLF`

`WSP valor2 CRLF`

`WSP valor3 CRLF`

`FWS = ([*WSP CRLF] 1*WSP) / obs-FWS
WSP = SP/HTAB ; white space`

- Ejemplo:

```
Received: from siles.ujaen.es (siles.ujaen.es [150.214.170.33])  
by gamol.ujaen.es (ESMTP/RIUJA5.0) with ESMTP id 91F631408051  
for <@ujaen.es>; Wed, 12 Sep 2012 00:27:42 +0200 (CEST)
```

3. Correo electrónico

Formato de los mensajes (RFC 822)

Cabeceras

- La RFC 822 define varios tipos de cabeceras que pueden ser incluidas en un correo electrónico, algunas son obligatorias, otras suelen estar siempre, aunque son opcionales, y otras tan solamente son incluidas cuando son necesarias.
- En la RFC 2822 se clasificaron los tipos de cabeceras (ligeramente diferentes a los que había en la RFC 822).

3. Correo electrónico

Formato de los mensajes (RFC 822)

Cabeceras

- Los tipos son (según aparecen en la RFC):
 - **Origination Date Field:** Especifica la fecha y la hora desde que el mensaje está listo para enviar.
 - **Originator fields:** Información sobre el remitente del mensaje.
 - **Destination Address Fields:** Especifican el/los destinatarios del mensaje, que pueden ser de tres tipos, el principal (“To”), destinatarios copiados (“Cc”) y destinatarios con copia ciega (“Cco”).
 - **Indentification Fields:** Contienen información para ayudar a identificar el mensaje.
 - **Informational fields:** Información opcional para facilitar al destinatario el conocer acerca de qué se trata en el correo.
 - **Resent Fields:** Preservar el remitente original, destino y otros campos necesarios para ser reenviado.
 - **Trace Fields:** Muestran la ruta seguida por el correo mientras ha sido transportado por el MTS.

3. Correo electrónico

Formato de los mensajes (RFC 822)

Cabeceras

Grupo	Campo	Aparición	Ocurrencias por mensaje	Descripción
Origination Date Field	Date:	Obligatorio	1	Fecha y hora de cuando el mensaje estaba listo para enviar
Originator fields	From:	Obligatorio	1	Remitente del mensaje
	Sender:	Opcional	1	Quién lo envía en nombre del remitente
	Reply-to:	Opcional	1	A quién responder preferiblemente
Destination Address Fields	To:	Normalmente presente	1	Una lista de los destinatarios preferentes del mensaje
	Cc:	Opcional	1	Destinatarios de una copia del mensaje. No hay diferencia técnica en el contenido, sino en la interpretación de quien lo recibe
	Bcc:	Opcional	1	Los demás destinatarios no sabrán que los que van en este campo también han recibido el mensaje.

3. Correo electrónico

Formato de los mensajes (RFC 822)

Cabeceras

Grupo	Campo	Aparición	Ocurrencias por mensaje	Descripción
Identification Fields	Message-ID:	Debería estar presente	1	Código único para identificar al mensaje generado, normalmente, cuando se envía
	In-Reply-To:	Opcional, presente normalmente en las respuestas	1	Lleva el identificador de mensaje del cual es respuesta para que pueda ser identificado
	References:	Opcional	1	Identifica otros documentos relacionados con el mensaje, normalmente otros correos.
Informational fields	Subject:	Normalmente presente	1	Describe el asunto o tema del mensaje
	Comments:	Opcional	Sin límite	Comentarios acerca del mensaje
	Keywords:	Opcional	Sin límite	Lista separada por comas de palabras clave

3. Correo electrónico

Formato de los mensajes (RFC 822)

Cabeceras

Grupo	Campo	Aparición	Ocurrencias por mensaje	Descripción
Resent Fields	Resent-Date: Resent-From: Resent-Sender: Resent-To: Resent-Cc: Resent-Bcc: Resent-MessageID:	Cada vez que un correo es reenviado se necesita un bloque Resent	Resent-Date y Resent-Sender son obligatorios, los demás son opcionales	Se usan solamente cuando el destinatario principal reenvía el correo a otro destinatario.
Trace Fields	Received: Return-Path:	Insertados por el sistema de correo electrónico	Sin límite	Insertados por los gestores de correo cada vez que procesan un mensaje. Pueden servir para rastrear el correo.

Campos opcionales: se pueden añadir campos adicionales de cabecera siempre que estos no coincidan con alguno que exista en US-ASCII, y no debe tener ni caracteres de control, ni espacios en blanco, ni el carácter ':'. Ejemplo: <nombre>:<cadena sin estructura>CRLF

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

- Descrita desde la RFCs 2045 a la 2049, se crearon para evitar tener que cambiar los protocolos estándar que se habían venido usando durante décadas (SMTP, POP e IMAP) y poder cubrir las nuevas necesidades que surgieron con el boom de Internet.
- Así, fueron creadas para:
 - Permitir la inclusión de contenido multimedia.
 - Poder enviar en los correos electrónicos ficheros de cualquier tipo.
 - Para permitir mensajes en lenguajes con juegos de caracteres diferentes del ASCII.
 - Permitir enviar mensajes con estructuras complejas.

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

- Fueron desarrolladas para el correo electrónico, pero su utilidad las han llevado a ser usadas también por protocolos como HTTP
 - HTTP emplea algunas cabeceras MIME para describir las características de los contenidos que transfiere.
 - De hecho, algunas de estas cabeceras han sido creadas para HTTP y no para el correo.
 - Sin embargo hay que tener en cuenta que HTTP **NO CUMPLE Estrictamente** CON EL FORMATO MIME.

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Funcionamiento

- MIME define una estructura especial para el contenido ASCII de las RFC 822 para permitir el contenido no codificable con US-ASCII.
- Se usan caracteres ASCII para codificar los contenidos no-ASCII en bloques, normalmente denominados entidades MIME. Así los correos pueden llevar:
 - Información no-textual: ficheros gráficos, audio, clip multimedia, etc.
 - Ficheros binarios de cualquier tipo: Programas ejecutables y ficheros de almacenamiento en formatos propietarios (AutoCad, Adobe PDF, etc.).
 - Mensajes de texto que usan juegos de caracteres diferentes al ASCII: Gracias a las extensiones MIME se pueden usar caracteres no-ASCII en las cabeceras RFC 822 de los correos electrónicos.
- Además MIME permite, gracias a su estructura, la inclusión de varios ficheros en un solo correo.

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Funcionamiento

- El soporte a MIME se consigue de una sencilla manera:
 - La inclusión de cabeceras adicionales en el formato RFC 822.
 - Reglas para codificar ficheros no-ASCII en texto ASCII.
- El soporte a las extensiones MIME es tan solamente necesario en el cliente origen y destino, no necesitando cambios en los servidores SMTP, POP3 o IMAP.
- Ejemplos de nuevas cabeceras:
 - `Content-Transfer-Encoding: método (de codificación).`
 - `Content-Type: tipo/subtipo (de contenido).`

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

RFCs más importantes que dan soporte a las extensiones MIME

RFC	Nombre	Descripción
2045	Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies	Describe los conceptos fundamentales y la estructura de los mensajes MIME. Actualizada por la RFC 2646, RFC 3798, RFC 5147 y RFC 6657
2046	Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types	Explica los conceptos de tipo y subtipo MIME y describe algunos tipos definidos en el estándar MIME. Actualizada por la RFC 2646, RFC 3798, RFC 5147 y RFC 6657
2047	MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text	Describe como se pueden modificar las cabeceras RFC 822 para llevar texto no-ASCII. Actualizada por la RFC 2184 y RFC 2231
4289	Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures	Define como las organizaciones pueden definir nuevos tipos. Actualizó la RFC 2048.
2049	Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples	Proporciona información adicional para la implementación y uso de los tipos MIME

También es de utilidad la RFC 6838 Media Type Specifications and Registration Procedures.

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Cabeceras con caracteres no ASCII

- La RFC 2047 permite la inserción de caracteres no ASCII en las cabeceras a través de la codificación de su valor y la especificación del código que se usa.

- Se delimitan de la siguiente manera:

`encoded-word = "=?" charset "?" encoding "?" encoded-text "=?"`

- Cada encoded-Word no debería tener una longitud mayor de 75 caracteres.
 - Los juegos de caracteres se definen en una cadena de texto independiente de mayúsculas. Algunos ejemplos son: ISO-8859-1 o UTF8.
 - La codificación puede ser Q (o q) (parecida a quoted-printable) o B (también b) (Base64).
 - Ejemplo:

`=?iso-8859-1?q?this=20is=20some=20text?="`

- La codificación UTF-8 no se rige por esta RFC sino por la RFC 6532

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Estructuras básicas MIME

- La manera exacta en la que los datos son codificados en el cuerpo y de cómo son incluidas las cabeceras MIME, dependen del tipo de estructura general del mensaje MIME:
 - **Estructura simple:** este tipo de mensajes lleva un único tipo de contenido, tal como un mensaje de texto o una imagen. En este caso, en el cuerpo del mensaje, tan solamente se encuentra una codificación de contenido.
 - **Estructura compleja:** el mensaje lleva un tipo de contenido compuesto, como un texto con una imagen, o un mensaje que encapsula a otro. Así pues, estos mensajes contienen en el cuerpo del mensaje varias entidades MIME.

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Entidades MIME

- Tanto los mensajes completos con MIME, como cada parte individual, se denominan **Entidades MIME**.
- Cuando un mensaje se recibe:
 - Primero se examinan las cabeceras RFC 822 que determinarán si el mensaje tiene una o varias entidades.
 - A continuación serán examinadas, una por una, las demás cabeceras de las entidades MIME restantes.

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Cabeceras MIME primarias

- **MIME-Version:** indica el uso de las extensiones y su versión para compatibilidad futura.
- **Content-Type:** tipo y subtipo de contenido de la entidad MIME.
- **Content-Transfer-Encoding:** codificación empleada para la entidad MIME.
- **Content-ID:** identificador de la entidad MIME.
- **Content-Description:** texto descriptivo de la entidad MIME.

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Ejemplo de las cabeceras MIME¹

Diagram illustrating the structure of a MIME email header with arrows pointing from labels to fields:

- Versión MIME → **MIME-Version: 1.0**
- Método usado para codificar los datos → **Content-Transfer-Encoding: base64**
- Tipo de datos multimedia, subtipo → **Content-Type: image/jpeg**
- Descripción del objeto (eje: audio) → **Content-Description:**
- Identificador de la entidad MIME → **Content-ID:**
- Datos codificados de la entidad MIME → **Datos codificados en base64**

```
From: uno@telematica.net
To: dos@telematica.net
Subject: extensiones MIME
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
Content-Description:
Content-ID:

Datos codificados en base64
.....
.....
.
```

1: No se incluyen los saltos de línea: CRLF

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Cabecera MIME-Version

- Es obligatoria en un mensaje y tiene dos propósitos:
 - Identificar al mensaje como codificado usando MIME.
 - Permitir la compatibilidad futura con otras versiones (Aunque solamente exista una versión en la actualidad).
- Es la única cabecera que se aplica a todo el mensaje en su conjunto, no repitiéndose en ninguna de las demás entidades del cuerpo (de hecho es la única que no comienza por Content).

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Cabecera Content-Type

- Especifica un tipo general y un subtipo, ambos separados por una barra.
 - Ejemplo: text/html cuando el contenido es un página web en formato html.
- Tipos MIME:
 - **Text:** información textual.
 - Ejemplos de subtipos: plain, html, richtext (comandos de formato sencillos).
 - **Image:** Imágenes, gráficos y fotografías.
 - subtipos: jpeg, gif.
 - **Audio:** audio digitalizado.
 - Ejemplos de subtipos: basic (codificación 8-bits ley-mu).
 - **Video:** imágenes en movimiento con o sin audio.
 - Ejemplos: mpeg, avi, fli.
 - **Application:** Datos que han de ser procesados por el cliente antes de ser visibles.
 - Ejemplo de subtipos: postscript, octet-stream (secuencia de bytes).
 - **Multipart:** Permite tener uno o más cuerpos del mensaje separados por un juego de caracteres frontera.
 - **Message:** Permite que un mensaje esté encapsulado dentro de otro.
 - Ejemplo de subtipos: Rfc822 (mensaje MIME RFC822)



3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Cabecera Content-Type - Tipos MIME (continua)

- **Multipart:** Permite que el mensaje tenga más de una parte.
 - Se delimitan por una frontera que es una cadena de caracteres definida en la propia cabecera con el parámetro `boundary="simple boundary"`, que más tarde se inserta al inicio de cada parte precedida de dos guiones: `--simple boundary` y marca el fin de la última entidad con el mismo delimitador con otros dos guiones al final: `--simple boundary--`
 - Ejemplos de tipos de contenido multiparte:
 - **Mixed:** partes independientes. Mostrar en el mismo orden del mensaje.
 - **Alternative:** mismo mensaje en distintos formatos.
 - **Digest:** está pensado para enviar colecciones de mensajes.
 - **Parallel:** las partes pueden verse al mismo tiempo.

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Cabecera Content-Type - Tipos MIME (continua)

- **Example:** Para ejemplos de uso de los demás tipos. No tiene subtipos registrados.
- **Model:** Para modelos en 3D.
 - Ejemplo de subtipos: mesh, vrml.

Lista completa del IANA: <http://www.iana.org/assignments/media-types>

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Cabecera Content-Transfer-Encoding

- Define la manera en la que el contenido MIME va codificado.
 - No es obligatoria.
 - Si no está presente se asume ASCII de 7 bits.
- Codificaciones:
 - **ASCII (7 bits)**
 - **8 bits o binario:** Sí la soportan los agentes, pero no soportado por SMTP, salvo que emplee la extensión correspondiente.
 - **Base64:** Es un método de representación de datos binarios a través de la conversión a un alfabeto de 64 caracteres imprimibles ASCII.
 - **Quoted-Printable:** Para cuando la mayor parte del texto es ASCII pero existen violaciones a las normas RFC 822.

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Codificación BASE 64

- La codificación BASE64 es ampliamente usada para representar datos binarios en entornos donde la información se debe enviar o mostrar como texto plano.
 - Certificados digitales (claves).
 - Ficheros incrustados en XML y HTML.
 - Peticiones GET de HTTP.
- Funcionamiento
 - Se dividen grupos de 24 bits (3 bytes) en unidades de 6 bits.
 - Para indicar que hay bytes de relleno se usa el carácter '='.
 - Estas unidades de 6 bits se envían como caracteres ASCII
 - $2^6 = 64$, codificándose según la tabla.

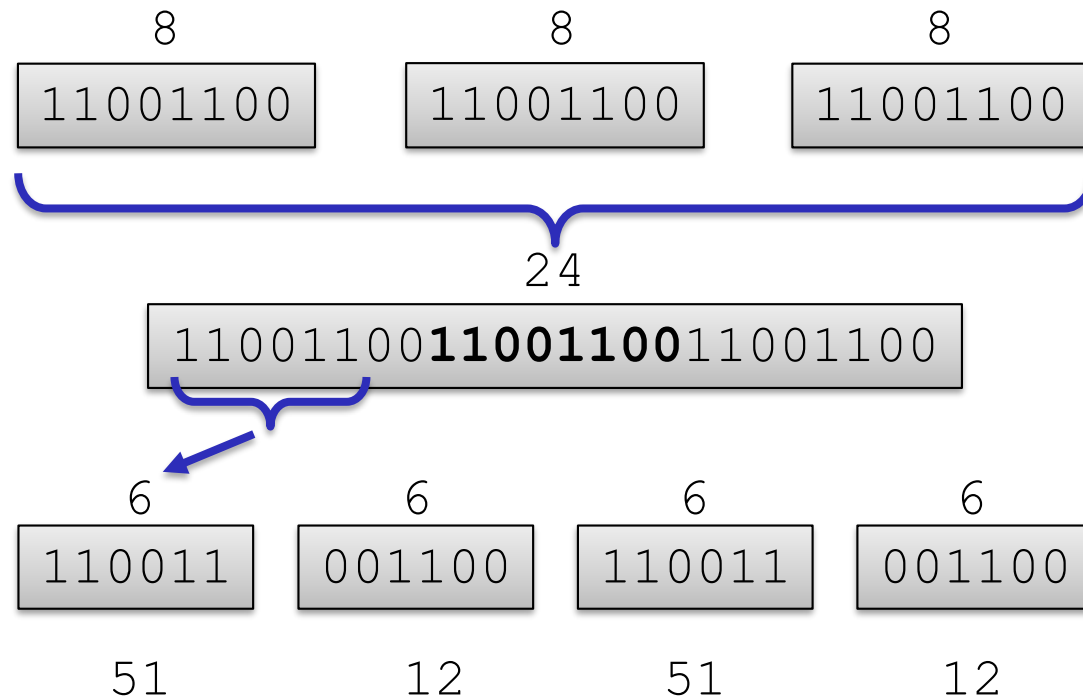
Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29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

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Codificación BASE 64.

Ejemplo de codificación para 3 bytes



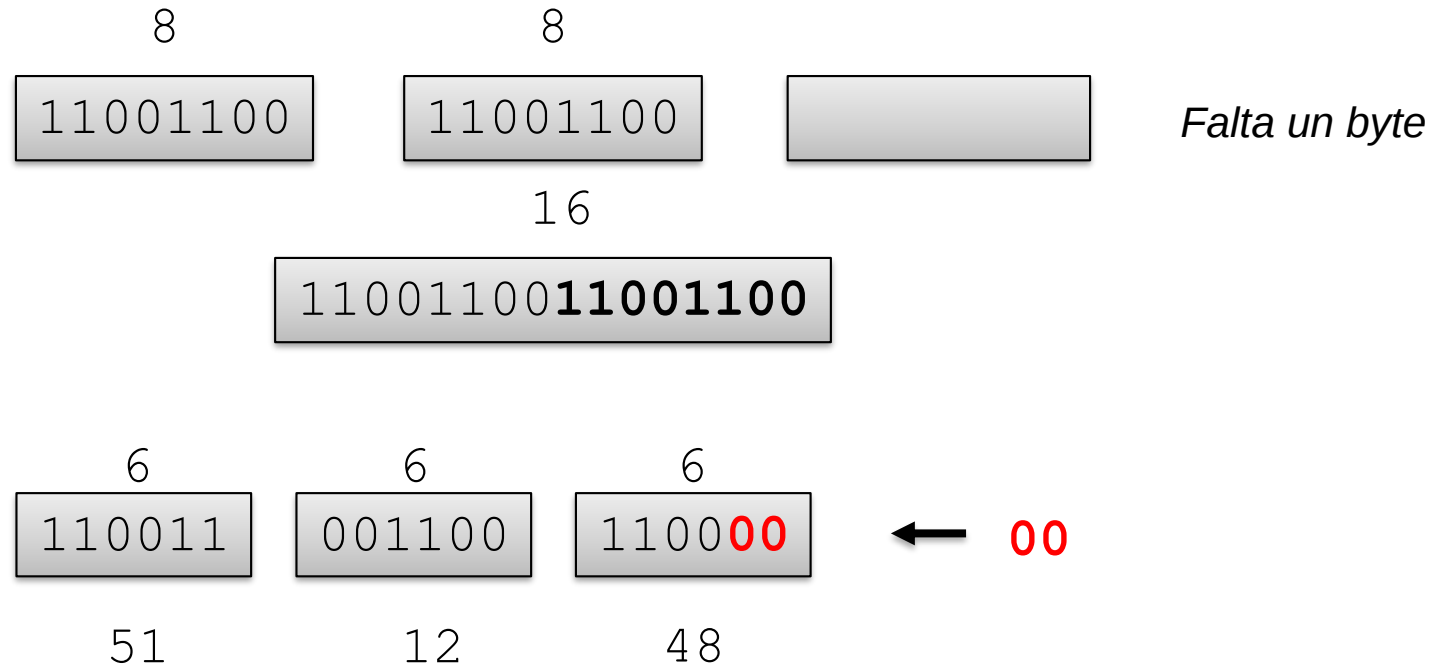
Resultado: **zMzM**

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Codificación BASE 64.

Ejemplo de codificación para 2 bytes



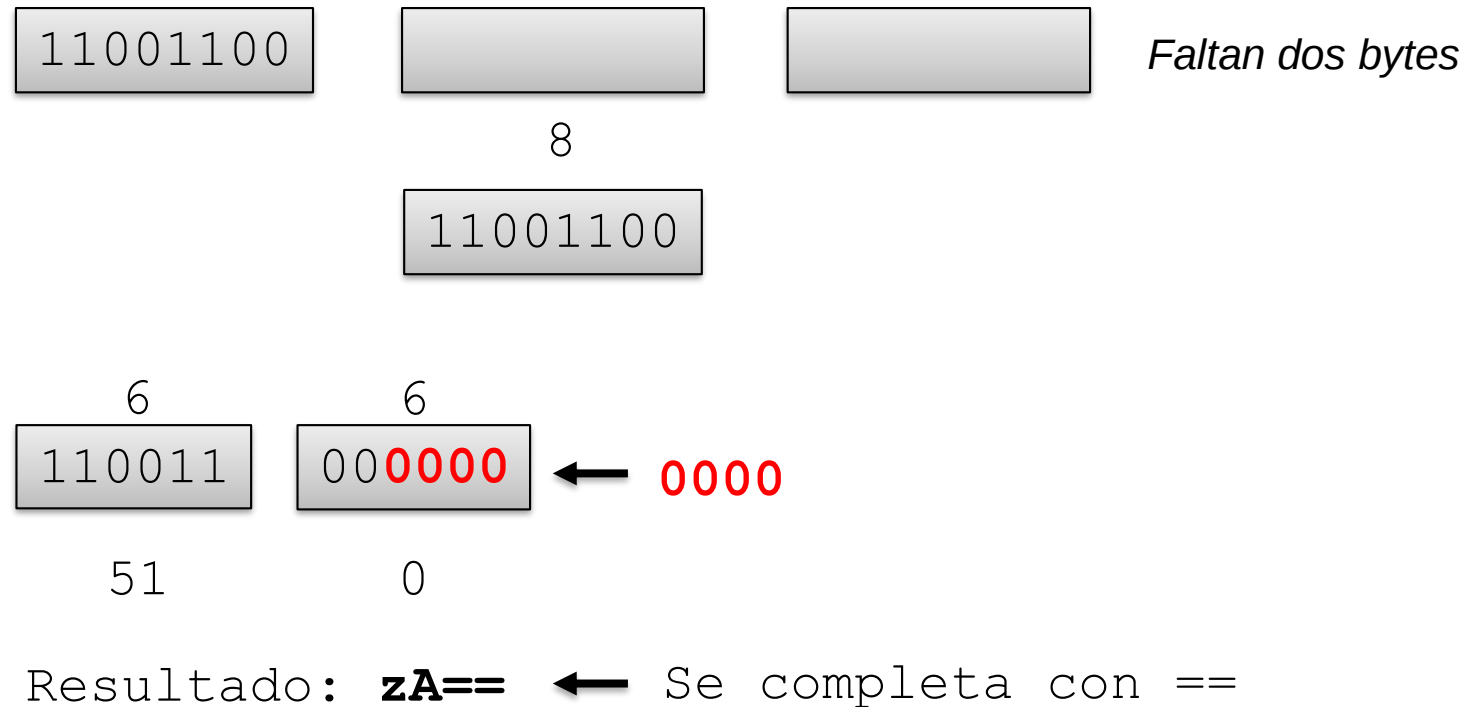
Resultado: **zMW=** ← Se completa con =

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Codificación BASE 64.

Ejemplo de codificación para 1 byte



3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Cabecera Content-Transfer-Encoding - Quoted-printable

- Para textos que contienen caracteres que pueden no ser ASCII, como las vocales con tildes.
- Algunas características:
 - ASCII 7 bits
 - Los caracteres >127 se codifican como un signo = seguido del valor del carácter en dos dígitos hexadecimales en mayúscula.
 - Ejemplo: palabra **sesión** → **sesi=C3=B3n**, ya que C·B· es el código de la 'ó'.
 - Las líneas codificadas no pueden tener mas de 76 caracteres (terminan con = , si hay que dividir la línea).
 - Mayor eficiencia cuando los mensajes están formados casi en su totalidad por caracteres ASCII.

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Cabecera Content-ID

- Permite que el contenido MIME tenga un identificador único.
- Es análoga a la cabecera Message-ID de la RFC 822 pero se refiere al propio contenido MIME.
- Es opcional y es usada con más frecuencia en los mensajes MIME multi-parte.

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Cabecera Content-Description

- Esta cabecera es opcional.
- Es una cadena de texto sin un formato específico para añadir una descripción de la entidad MIME.
- En un mensaje multi-parte cada entidad puede tener una cabecera Content-Description para informar al receptor de lo que realmente contiene.

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Otras cabeceras

- **Content-Language** (RFC 3282): permite, si se desea, especificar el idioma del texto en mensajes tipo RFC822, entidades MIME o recursos web.
- **Content-Disposition** (RFC 2183): se creó para su uso con la extensiones MIME, y es válida para cualquier entidad MIME. Su uso es opcional. Puede tomar los valores de `inline` y `attachment`, aunque deja abierto el uso a nuevas disposiciones y cabeceras «X-». Algunos de los parámetros que puede llevar son: `filename`, `creation-date`, `modification-date`, `read-date` o `size`, estando abierta la inclusión de otros parámetros.

3. Correo electrónico

MIME (Multipurpose Internet Mail Extensions)

Ejemplo de uso de MIME

```
Return-Path: <jccuevas@ujaen.es>
Delivered-To: jccuevas@vbuzon.ujaen.es
Reply-To: <jccuevas@ujaen.es>
From: =?iso-8859-1?Q?Juan_Carlos_Cuevas_Mart=EDnez?=
<jccuevas@ujaen.es>
To: <jccuevas@ujaen.es>
Subject: Uso de cabecera MIME
Organization: =?iso-8859-1?Q?Universidad_de_Ja=E9n?=
Message-ID: <003a01cf9b4a$52010f00$f6032d00$@ujaen.es>
```

MIME-Version: 1.0

Content-Type: multipart/mixed;

boundary="-----_NextPart_000_003B_01CF9B5B.158A0610"

Content-Language: es

This is a multipart message in MIME format.

-----_NextPart_000_003B_01CF9B5B.158A0610

Content-Type: text/plain;

charset="iso-8859-1"

Content-Transfer-Encoding: 7bit

Este es un mensaje MIME multiparte.

-----_NextPart_000_003B_01CF9B5B.158A0610

Content-Type: image/gif;

name="ESCUDO UJAEN_peq.gif"

Content-Transfer-Encoding: base64

Content-Disposition: attachment;

filename="ESCUDO UJAEN_peq.gif"

PSDPFRVc+An0Qwn2x7f1STOFC6stEHg/7rk+2MakWUuCJ1GRDAQEA0w==

-----_NextPart_000_003B_01CF9B5B.158A0610--

La codificación en Base64 del fichero ha sido recortada por motivos de espacio.

3. Correo electrónico

Modelos de acceso y recuperación de correos electrónicos

- **Modelo de acceso online:** en parte es una situación ideal en la que todas las máquinas están permanentemente conectadas a Internet con sus servidores SMTP activos.
- **Modelo de acceso offline:** el usuario establece una conexión al servidor donde se encuentra su buzón, descarga los correos y los elimina del servidor. La lectura u organización de los correos se hace offline una vez se han descargado estos.
- **Modelo de acceso desconectado:** es un híbrido de los dos anteriores. Los correos pueden descargarse, pero se quedan en el buzón del servidor. Además, la operaciones realizadas en offline, se sincronizan una vez se vuelve a conectar.

3. Correo electrónico

Protocolo de correo POP3

- Post Office Protocol v.3 (RFC 1939).
 - Hace uso de conexiones TCP (puerto 110).
 - Usando TLS es el puerto 995 (RFC8314).
 - El servidor ofrece un servicio de almacén.
 - POP descarga los mensajes y los almacena localmente: MODELO OFFLINE.
 - Requiere autenticación.
 - Se basa en preguntas y repuestas, similar a SMTP.
 - Actualizado por las RFCs: 1957, 2449, 6186 y 8314.



3. Correo electrónico

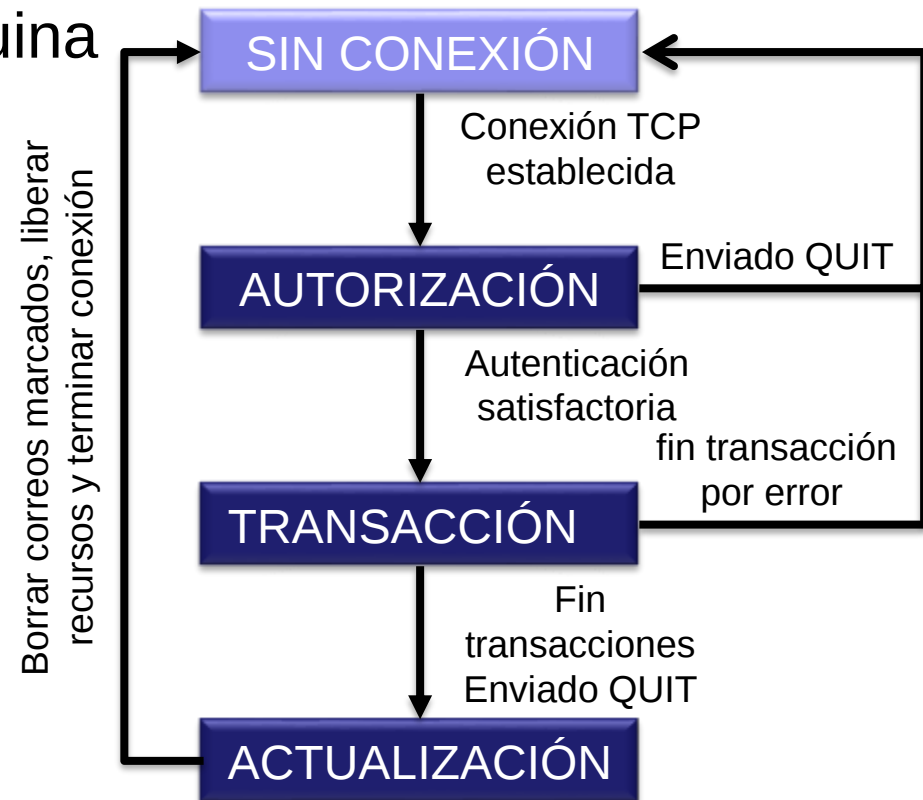
Protocolo de correo POP3

- Diálogo alternativo:
 - El cliente envía comandos y el servidor responde con un mensaje.
 - El orden de los comandos es importante.
 - Los comandos y respuestas son textos en ASCII.
 - Todos los comandos y respuestas deben terminar con los caracteres CR y LF.
 - Las respuestas comienzan con “+OK” o “-ERR”.
 - Las respuestas pueden contener mas de una línea
- Muy utilizado por clientes: Outlook, Thunderbird, etc.

3. Correo electrónico

Protocolo de correo POP3

- El protocolo sigue una máquina de estados finita.
- Estados:
 - Autorización.
 - Transacción.
 - Actualización.



3. Correo electrónico

Protocolo de correo POP3

Comandos POP3

- Definición ABNF de los comandos de POP3 (RFC 2449)
command = keyword * (SP param) CRLF ; 255 octets maximum
keyword = 3*4VCHAR
param = 1*VCHAR
- La máxima longitud de un comando (incluyendo el CRLF) se establece en 255 bytes (en la RFC 1939 eran 47 con el CRLF).
- La máxima longitud para la primera línea de respuesta es 512 bytes, incluyendo el CRLF.

3. Correo electrónico

Protocolo de correo POP3

Comandos POP3

NEMÓNICO	DESCRIPCIÓN	RESTRICCIONES
USER <user>	Especifica un nombre de usuario	Solamente en el estado de AUTORIZACIÓN o tras un USER o PASS infructuoso.
PASS <pass>	Especifica contraseña	Solamente en el estado de AUTORIZACIÓN y tras un USER con éxito.
STAT	Estado del almacén (bandeja) de entrada: número total de mensajes y el total de bytes ocupados.	Solamente en el estado de TRANSACCIÓN
LIST [<M>]	Proporciona la lista de los mensajes y el tamaño de todos si no se aporta el parámetro M, uno por línea. La última línea contiene '.', Si no, se lista el tamaño de solo el mensaje M	Solamente en el estado de TRANSACCIÓN

3. Correo electrónico

Protocolo de correo POP3

Comandos POP3

NEMÓNICO	DESCRIPCIÓN	RESTRICCIONES
RETR <M>	Recupera un mensaje	Solamente en el estado de TRANSACCIÓN
DELE <M>	Marca un mensaje para ser borrado del almacén (bandeja) de entrada	Solamente en el estado de TRANSACCIÓN
NOOP	Devuelve un asentimiento positivo (+ok)	Solamente en el estado de TRANSACCIÓN
RSET	Todas las marcas de borrado se desactivan	Solamente en el estado de TRANSACCIÓN
QUIT	Borra los mensajes marcados y devuelve el resultado de esta operación. Además cierra la conexión TCP	Ninguna

3. Correo electrónico

Protocolo de correo POP3

Comandos opcionales de POP3

- Los comandos siguientes no es obligatorio incluirlos en las implementaciones mínimas de servidores de POP3:
 - **TOP <M> <L>**: Muestra la cabecera del mensaje **M** y **L** líneas del cuerpo del mensaje. Final con <CRLF>.<CRLF>. Solamente en el estado de TRANSACCIÓN.
 - **UIDL [<M>]**: Muestra el identificador del mensaje **M** o de todos los mensajes si no se aporta el parámetro **M**. Solamente en el estado de TRANSACCIÓN.
 - **APOP nombre MD5(marca de tiempo+clave)**: Autenticación del usuario usando el algoritmo MD5. Se envía el MD5 de la clave con una cadena de texto con el instante de la conexión (timestamp) enviada por el servidor en el saludo inicial (si este soporta APOP). Solamente en el estado de AUTORIZACIÓN o tras un USER o PASS erróneos.
 - Realmente los comandos **USER** y **PASS** son también opcionales, pero un servidor debe implementar al menos un mecanismo para identificar al usuario.
- Además, en la RFC se recomienda la implementación de todos estos comandos.

3. Correo electrónico

Protocolo de correo POP3

Ejemplo de intercambio POP3

```
S: +OK Hello there.  
C: USER mailuser  
S: +OK Password required.  
U: PASS 1234  
S: +OK logged in.
```

AUTORIZACIÓN

```
C: STAT  
S: +OK 2 320  
C: LIST  
S: +OK 2 messages (320 octets)  
S: 1 120  
S: 2 200  
S: .  
C: LIST 2  
S: +OK 2 200  
C: LIST 3  
S: -ERR Invalid message number  
C: RETR 1  
S: +OK 120 octets follow.
```

TRANSACCIÓN

```
S: From: sender  
S: To: mailuser  
S: Subject: test  
S:  
S: <Mensaje de correo>  
S: .  
C: DELE 1  
S: +OK message 1 deleted  
C: DELE 1  
S: -ERR message 1 already deleted  
C: RSET  
S: +OK  
C: LIST 1  
S: +OK 1 120  
C: NOOP  
S: +OK  
C: QUIT  
S: +OK Bye
```

TRANSACCIÓN

ACTUALIZACIÓN

3. Correo electrónico

Protocolo de correo POP3

Extensiones de POP3

- Este protocolo se ha pretendido siempre que sea simple, con pocas extensiones y variaciones.
- Si se desea mayor variedad de opciones de manejo del correo y el buzón, se debe usar IMAP, el cual se verá más tarde.
- Algunas de las extensiones más importantes son:
 - Autenticación SASL (Simple Authentication and Security Layer) descrita en la RFC 5034.

3. Correo electrónico

Protocolo de correo POP3

Extensiones de POP3

- Las capacidades opcionales que soporta un servidor las devuelve el comando **CAPA**.
- Los servidores que soportan el comando **CAPA** deben soportar el tamaño de comando mínimo de 255 octetos, así como la longitud mínima de cualquier extensión que soporte.

```
<conexión con el servidor POP3>
+OK Hello there.
Capa
+OK Here's what I can do:
STLS
TOP
USER
LOGIN-DELAY 10
PIPELINING
UIDL
IMPLEMENTATION Courier Mail Server
.
```

3. Correo electrónico

Protocolo de correo POP3

Extensiones de POP3

- Lista de capacidades devuelta por **CAPA**:
 - TOP: el comando opcional TOP está soportado.
 - USER: los comandos USER y PASS están disponibles, a pesar de que pueden no estar disponibles a todos los usuarios.
 - SASL: está disponible la capa de autenticación SASL a través del comando AUTH. A continuación de SASL vendrá él o los mecanismos permitidos.
 - RESP-CODES: Indica que el servidor responderá cada mensaje de texto con un corchete “[” con información adicional del código de respuesta extendida.
C: APOP mrose c4c9334bac560ecc979e58001b3e22fb
S: -ERR **[IN-USE]** Do you have another POP session running?
 - LOGIN-DELAY: indica los segundos que, como mínimo, se esperará entre intentos de autenticación. Afecta a los comandos APOP, AUTH, USER y PASS.

3. Correo electrónico

Protocolo de correo POP3

Extensiones de POP3

- Lista de capacidades devuelta por **CAPA**:
 - PIPELINING: permite agrupar varios comandos en una petición sin tener que esperar a la respuesta de cada uno. No se recomienda hacerlo con comandos como RETR o DELE.
 - EXPIRE: el parámetro informa de los mínimos días que el servidor mantendrá un correo leído. Si es 0, no se permite dejar correos que se hayan leído con RETR en el servidor. Si el valor es NEVER, el servidor nunca los borrará. Puede tener otro parámetro (el usuario) si la política es particular de cada usuario.
 - UIDL: Informa de si el servidor implementa el comando UIDL.
 - IMPLEMENTATION: Es una cadena que informa sobre la implementación del servidor (nombre, versión, etc.).
 - STLS: Informa de la disponibilidad del comando STLS para establecer una conexión segura con TLS (SSL).

3. Correo electrónico

Protocolo de correo IMAP

Evolución

- **Interactive Mail Access Protocol v.4rev2** (RFC 9051 de agosto de 2021, deja obsoleta a la v.4rev1 RFC 3501).
 - Su evolución fue algo turbulenta:
 - Hubo versiones que no se reconocieron (IMAP2 – RFC1064)
 - Que quedaron como experimentales (IMAP2 – RFC1176)
 - Otra que nunca llegó a RFC y se usó (IMAP2bis)
 - La versión 3 fue relegada al estado de «histórica» (RFC 1203).
 - Finalmente la versión definitiva hasta que llegó la versión 4 (RFC 1730).



3. Correo electrónico

Protocolo de correo IMAP

Evolución

- IMAP v4.rev1 sufrió una importante serie de actualizaciones:
 - RFC 4466 Collected Extensions to IMAP4 ABNF, Abril 2006.
 - RFC 4469 Internet Message Access Protocol (IMAP) CATENATE Extension, Abril 2006.
 - RFC 4551 IMAP Extension for Conditional STORE Operation or Quick Flag Changes Resynchronization, Junio 2006.
 - RFC 5032 WITHIN Search Extension to the IMAP Protocol, Septiembre 2007.
 - RFC 5182 IMAP Extension for Referencing the Last SEARCH Result. Marzo 2008.
 - RFC 5738 IMAP Support for UTF-8. Marzo 2010.
 - RFC 6186 Use of SRV Records for Locating Email Submission/Access Services. Marzo 2011.
 - RFC 6858. Simplified POP and IMAP Downgrading for Internationalized Email. Marzo 2013.
 - RFC 7817. Updated Transport Layer Security (TLS) Server Identity Check Procedure for Email-Related Protocols. Marzo 2016.
 - RFC 8314. Cleartext Considered Obsolete: Use of Transport Layer Security (TLS) for Email Submission and Access, enero 2018.
 - RFC 8474. IMAP Extension for Object Identifiers, septiembre 2018.
 - RFC 8996. Deprecating TLS 1.0 and TLS 1.1, marzo 2021.

3. Correo electrónico

Protocolo de correo IMAP

Evolución

Comparativa v4rev2 vs v4.rev1

- En general la versión IMAPv4.rev2 es compatible hacia adelante con las versiones IMAP4rev1, IMAP2, y la no publicada IMAP2bis.
- La version IMAPv4.rev2 es en gran medida compatible con la IMAPv4.rev1 (RFC 3501) y la especificación IMAP4 (RFC 1730), excepto por algunas características añadidas en estas últimas que resultaron problemáticas y han sido sustituidas por alternativas más apropiadas.
 - También se han dejado obsoletos algunos comandos, respuestas y formatos de mensaje.
- Algunos cambios son (Anexo E RFC 9051):
 - Se añade soporte a mensajes de 64 bits.
 - Se añade la respuesta CLOSED.
 - Las respuestas UNSEEN y RECENT a SELECT/EXAMINE se declara obsoleta.
 - Se elimina el comando CHECK, los clientes deben usar NOOP.
 - Los nombres de los buzones y las cabeceras de los mensajes permiten UTF-8.

3. Correo electrónico

Protocolo de correo IMAP

Características

- Hace uso de conexiones TCP (puerto 143).
 - Usando TLS es el puerto 993.
- El servidor ofrece un servicio de almacén.
- Permite la gestión remota del almacén.
 - Gestiona carpetas (crea, borra, renombra).
 - Mueve, lista, lee, busca, marca mensajes.
 - Descargas selectivas.
- El coste de todas estas nuevas funcionalidades, con respecto a POP3, es mucha mayor complejidad.
- Un cliente IMAP **debe estar siempre dispuesto a recibir datos del servidor** (al contrario que POP3 donde el servidor solo envía datos como respuesta a un comando). La información recibida de esta manera debería ser almacenada para evitar solicitar esta información más tarde al servidor (existen casos en que obligatoriamente debe ser recordada).

3. Correo electrónico

Protocolo de correo IMAP

Características

- IMAP se rige por una máquina de estados finitos.
- Los cambios de estados no son lineales como en POP3.
- Dependiendo del estado, hay un juego de comandos concreto permitido.
- Tiene cuatro estados (en vez de tres como POP3):
 - No autenticado.
 - Autenticado.
 - Seleccionado.
 - Salida (logout).

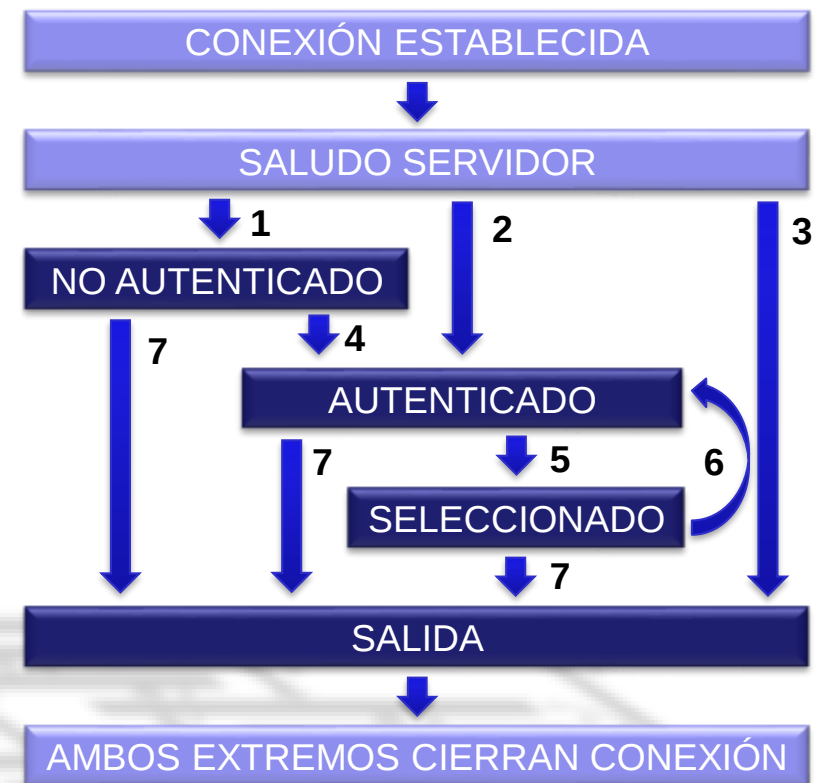
3. Correo electrónico

Protocolo de correo IMAP

Máquina de estados finitos

- Transiciones de estado:

1. Conexión sin pre-autenticación (Saludo con OK).
2. Conexión pre-autenticada (Saludo PREAUTH).
3. Conexión rechazada (Saludo BYE).
4. Comando LOGIN o AUTHENTICATE exitoso.
5. Comando SELECT o EXAMINE exitoso.
6. Comandos CLOSE o UNSELECT, código de respuesta no solicitado CLOSED o fallo en los comandos SELECT o EXAMINE.
7. Comando LOGOUT, apagado del servidor o cierre de la conexión.



3. Correo electrónico

Protocolo de correo IMAP

Comandos, resultados y respuestas

- Usan caracteres ASCII de 7 bits y por eso se puede usar un cliente TELNET para enviarlos.
- Tanto los comandos como las respuestas se terminan con CRLF.
- Los comandos normalmente no se abrevian como se hacía en POP3 (de STAT en POP3 a STATUS en IMAP).
- IMAP permite que el cliente envíe varios comandos sin tener que esperar a la respuesta del servidor (siempre que el resultado de estos sea independiente).

3. Correo electrónico

Protocolo de correo IMAP

Comandos, resultados y respuestas - Etiquetado

- IMAP aporta un sistema de etiquetado de comandos con etiquetas alfanuméricas (tags), únicas para una conexión (a0001, a0002, ...).
 - El servidor puede responder explícitamente a un comando usando la etiqueta con la que se envió.
 - No todas las respuestas llevarán etiquetas.

3. Correo electrónico

Protocolo de correo IMAP

Comandos, resultados y respuestas

- Los **resultados** (*server completion result response*) son respuestas a comandos que indican el estado o disposición del mismo. Pueden ser una respuesta concreta con la etiqueta del comando al que responde, o una respuesta general sin etiqueta.
- Una **respuesta** es cualquier información que envía el servidor al cliente, ya sea como resultado de un comando del cliente o enviado unilateralmente por el servidor. No están etiquetadas y no están especialmente indicadas para indicar el estado del servidor. Se preceden del carácter “*”. Se denominan “*untagged responses*” o respuestas sin etiquetar.

3. Correo electrónico

Protocolo de correo IMAP

Comandos, resultados y respuestas

- La información enviada al cliente por el servidor, que no implica que un comando se ha llevado a cabo, se inicia por el carácter '*'.
• Cuando un comando necesita más información para ser completado (como sucede con AUTHENTICATE o LOGIN) el servidor informa de esto enviando en primer lugar el carácter '+'.
• Uso de las llaves «{n}»: permite indicar la cantidad de caracteres que se van a enviar a continuación (tanto por parte del cliente como por el servidor).

Ejemplo

```
C: A001 LOGIN {4}
S: + Ready for additional command text
C: fred {10}
S: + Ready for additional command text
C: flintstone
S: A001 OK LOGIN completed
C: A002 STATUS mbox (UIDNEXT MESSAGES)
S: * STATUS mbox (MESSAGES 2 UIDNEXT 42)
S: A002 OK STATUS completed
```

3. Correo electrónico

Protocolo de correo IMAP

Grupos de comandos

- Comando generales (cualquier estado).
 - Comandos del estado “No autenticado”.
 - Comandos del estado “Autenticado”: estos comandos también pueden ser usados en el estado seleccionado.
 - Comandos del estado “Seleccionado”: comandos destinados a manipular mensajes que tan solo se deben usar en el estado Seleccionado (lo que se ha seleccionado es un buzón de correo).
-
- NOTA: Los comandos no están obligados a ser enviados en mayúsculas (son «*case-insensitive*») pero tradicionalmente se envían en mayúsculas.

3. Correo electrónico

Protocolo de correo IMAP

Códigos de resultado (result)

CÓDIGO	DESCRIPCIÓN
OK	Resultado positivo ante una operación. Suele incorporar la etiqueta del comando al que corresponde. A veces puede no llevar la etiqueta, como en las respuestas iniciales
NO	Resultado negativo. Si lleva etiqueta es que el comando en cuestión no se ha podido llevar a cabo, y si no la lleva, se refiere a un problema con algún aspecto general del servidor
BAD	Informa de un error. Si lleva etiqueta se refiere concretamente al comando enviado (como error de sintaxis o comando desconocido), de otra manera no la lleva
PREAUTH	Es un mensaje sin etiqueta enviado al inicio de la sesión que indica que no se necesita autenticación, haciendo que se pase automáticamente al estado AUTENTICADO. La autenticación previa puede ser por muchas vías, las cuales no dependen de IMAP
BYE	Sin etiqueta, enviada cuando se va a cerrar la conexión, ya sea por la recepción de un LOGOUT o cuando la conexión tiene que ser cerrada por cualquier otro motivo

3. Correo electrónico

Protocolo de correo IMAP

Códigos de respuesta (response)

- Las respuestas (también denominadas mensajes) se usan para comunicar una amplia variedad de información al cliente, en contra de lo que es un resultado.
- Las respuestas suelen incluir un texto explicativo, además de diversos atributos.
- Pueden ser enviadas en respuesta a un comando o ante una incidencia (como la recepción de un nuevo correo) sin que esté vinculada al comando en curso.

3. Correo electrónico

Protocolo de correo IMAP

Códigos de respuesta (response)

CÓDIGO	DESCRIPCIÓN
ALERT	Un mensaje con información importante que el cliente debe facilitar al usuario humano
BADCHARSET	Búsqueda fallida debido a un carácter no permitido
CAPABILITY	Una lista de las características del servidor que se puede enviar al inicio de la sesión y así evitar el uso del comando CAPABILITY
PARSE	Informa de un error al procesar las cabeceras o extensiones MIME de un mensaje
PERMANENTFLAGS	Comunica una lista de flags de mensaje que el usuario puede manipular
READ-ONLY	Informa al cliente que el buzón es de solo lectura
READ-WRITE	Informa al cliente de que el buzón es de lectura/escritura

3. Correo electrónico

Protocolo de correo IMAP

Códigos de respuesta (response)

CÓDIGO	DESCRIPCIÓN
TRYCREATE	Enviado cuando se ha intentado el comando APPEND o COPY en un buzón que no existe, informando de que se pruebe a crear éste previamente
UIDNEXT	Se envía con un número entero que especifica el identificador único para que sea utilizado en la próxima operación. Esto permite que cada mensaje sea identificado de manera única
UIDVALIDITY	Seguido de un número entero, informa de la validez de los identificadores únicos de mensaje en una sesión. Es un valor que complementa los identificadores únicos. Si los UID se actualizan, UIDVALIDITY debe ser mayor que el anterior (una forma recomendada es que sea la hora en formato 32 bits)
UNSEEN	Enviado con el número del mensaje que está marcado y aún no ha sido visto

3. Correo electrónico

Protocolo de correo IMAP

Comandos generales

COMANDO	PARÁMETROS	DESCRIPCIÓN
CAPABILITY	No	Pregunta al servidor que capacidades y características tiene implementadas
NOOP	No	No hace nada.
LOGOUT	No	Informa al servidor de que el cliente ha terminado y que está listo para terminar la sesión. Esto hace que se evolucione al estado SALIDA (logout)

Ejemplo de LOGOUT:

```
C: A023 LOGOUT
S: * BYE IMAP4rev2 Server logging out
S: A023 OK LOGOUT completed
(El servidor y el cliente cierran la conexión)
```

3. Correo electrónico

Protocolo de correo IMAP

Comandos del estado NO AUTENTICADO

COMANDO	PARÁMETROS	DESCRIPCIÓN
LOGIN	Nombre de usuario y clave	Indica el usuario y su clave para iniciar la sesión
AUTHENTICATE	Nombre del mecanismo de autenticación	Mecanismo de autenticación basado en la especificación Simple Authentication and Security Layer (SASL) RFC 4422.
STARTTLS	No	Informa al servidor IMAP de que use Transport Layer Security (TLS) para la autenticación pero solamente en el puerto que emplea texto en claro. No cambia al estado autenticado.

3. Correo electrónico

Protocolo de correo IMAP

Comandos del estado AUTENTICADO

- Su función fundamental es la selección y gestión de los buzones de correo del usuario.
- También son válidos en el estado SELECCIONADO.
- Algunos de los comandos más significativos son:
 - **SELECT <buzón>**: selecciona un buzón para poder acceder a sus mensajes.
 - **EXAMINE <buzón>**: es idéntico a SELECT pero el buzón se marca como de solo lectura, por lo que no se pueden hacer cambios en él.
 - **CREATE <buzón>**: crea un buzón con el nombre del parámetro.
 - **DELETE <buzón>**: borra de manera permanente un buzón. Si se devuelve un OK también se borrarán todos los mensajes de dicho buzón.
 - **RENAME <buzón>**: cambia el nombre de un buzón.
 - **LIST <buzón> o <cadena de búsqueda>**: devuelve un conjunto de nombres de buzones disponibles para el cliente.
 - **STATUS <buzón>**: requiere el estado de un buzón concreto. No cambia el buzón seleccionado ni afecta al estado de ningún correo del buzón solicitado.

3. Correo electrónico

Protocolo de correo IMAP

Comandos del estado AUTENTICADO – Ejemplo de uso de SELECT

```
C: A142 SELECT INBOX
S: * 172 EXISTS
S: * OK [UIDVALIDITY 3857529045] UIDs valid
S: * OK [UIDNEXT 4392] Predicted next UID
S: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
S: * OK [PERMANENTFLAGS (\Deleted \Seen \*)] Limited
S: * LIST () "/" INBOX
S: A142 OK [READ-WRITE] SELECT completed
```

3. Correo electrónico

Protocolo de correo IMAP

Comandos del estado AUTENTICADO – Ejemplo de uso de SELECT

```
C: A142 SELECT INBOX
S: * 172 EXISTS
S: * OK [UIDVALIDITY 3857529045] UIDs valid
S: * OK [UIDNEXT 4392] Predicted next UID
S: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
S: * OK [PERMANENTFLAGS (\Deleted \Seen \*)] Limited
S: A142 OK [READ-WRITE] SELECT completed
  [...some time later...]
C: A143 SELECT Drafts
S: * OK [CLOSED] Previous mailbox is now closed
S: * 5 EXISTS
S: * OK [UIDVALIDITY 9877410381] UIDs valid
S: * OK [UIDNEXT 102] Predicted next UID
S: * LIST () "/" Drafts
S: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
S: * OK [PERMANENTFLAGS (\Deleted \Seen \Answered
  \Flagged \Draft \*)] System flags and keywords allowed
S: A143 OK [READ-WRITE] SELECT completed
```

Selección de un buzón,
cuando se tenía otro
seleccionado.



3. Correo electrónico

Protocolo de correo IMAP

Comandos del estado SELECCIONADO

- Su función principal es la gestión de los mensajes.
- Algunos de los comandos más significativos son:
 - **SEARCH <parámetros>**: busca en el buzón los mensajes que coincidan con los parámetros de búsqueda.
 - **FETCH <parámetros>**: obtiene los datos asociados con un mensaje.
 - **STORE <parámetros>**: altera los datos almacenados en un mensaje como los flags.
 - **CLOSE**: borra permanentemente los correos marcados con el flag \Deleted en el buzón actual y vuelve al estado AUTENTICADO desde SELECCIONADO.
 - **UNSELECT**: como CLOSE pero no se borran los correos.
 - **EXPUNGE**: borra permanentemente los correos marcados con el flag \Deleted en el buzón actual.
 - **COPY <parámetros> <buzón>**: copia un mensaje al final del buzón designado. Los flags y la fecha interna deberían ser preservados.
 - **MOVE <parámetros> <buzón>**: mueve un mensaje al final del buzón designado. EL mensaje será eliminado del buzón origen y en el destino se le creará un nuevo UID. Los flags y la fecha interna deberían ser preservados.

3. Correo electrónico

Protocolo de correo IMAP

Ejemplo usando un cliente Telnet (v4rev1)

```
telnet: > telnet imap.example.com imap
telnet: Trying 192.0.2.2...
telnet: Connected to imap.example.com.
telnet: Escape character is '^]'.
s: * OK Dovecot ready.
c: a1 LOGIN MyUsername MyPassword
s: a1 OK Logged in.
c: a2 LIST "" "*"
s: * LIST (\HasNoChildren) "." "INBOX"
s: a2 OK List completed.
c: a3 EXAMINE INBOX
s: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
s: * OK [PERMANENTFLAGS ()] Read-only mailbox.
s: * 1 EXISTS
s: * 1 RECENT
s: * OK [UNSEEN 1] First unseen.
s: * OK [UIDVALIDITY 1257842737] UIDs valid
s: * OK [UIDNEXT 2] Predicted next UID
s: a3 OK [READ-ONLY] Select completed.
```

```
c: a4 FETCH 1 BODY[]
s: * 1 FETCH (BODY[] {405}
s: Return-Path: sender@example.com
s: Received: from client.example.com ([192.0.2.1])
s: by mx1.example.com with ESMTP
s: id
<20040120203404.CCCC18555.mx1.example.com@client.examp
e.com>
s: for <recipient@example.com>; Tue, 20 Jan 2004
22:34:24 +0200
s: From: sender@example.com
s: Subject: Test message
s: To: recipient@example.com
s: Message-Id:
<20040120203404.CCCC18555.mx1.example.com@client.examp
e.com>
s:
s: This is a test message.
s: )
s: a4 OK Fetch completed.
c: a5 LOGOUT
s: * BYE Logging out
s: a5 OK Logout completed.
```

4. Protocolo Telnet

- Proporcionar un servicio de comunicación bidireccional orientado a byte.
- Definido en las RFC 854 y 855 (Mayo de 1983) por J. Postel y J.K. Reynolds.
- Es el estándar 8 de Internet (categoría *Internet Standard*).
- Objetivo
 - Permitir un método estándar para comunicar terminales entre sí de manera bidireccional y orientado a byte (ocho bits).
 - Comunicar procesos orientados a terminal unos con otros.
- Completan el protocolo las RFCs 856 a 861 (y otras) que tienen asignados otros números de estándar.

4. Protocolo Telnet

Definición y evolución del protocolo TELNET

RFC	Estándar	Título	Evolución
RFC 854	STD 8	Telnet Protocol Specification	Obsoletes RFC 764, Updated by RFC 5198
RFC 855	STD 8	Telnet Option Specifications	Obsoletes NIC 18640
RFC 856	STD 27	Telnet Binary Transmission	Obsoletes NIC 15389
RFC 857	STD 28	Telnet Echo Option	Obsoletes NIC 15390
RFC 858	STD 29	Telnet Suppress Go Ahead Option	Obsoletes NIC 15392
RFC 859	STD 30	Telnet Status Option	Obsoletes RFC 651
RFC 860	STD 31	Telnet Timing Mark Option	Obsoletes NIC 16238
RFC 861	STD 32	Telnet Extended Options: List Option	Obsoletes NIC 16239
RFC 5198		Unicode Format for Network Interchange	Obsoletes RFC 698

4. Protocolo Telnet

Características

- El protocolo TELNET hace uso de conexiones TCP al puerto 23.
- Se basa en 3 ideas principales:
 - El concepto de un "Terminal virtual de Red" (NVT, *Network Virtual Terminal*).
 - El principio de opciones negociadas.
 - Una visión simétrica de los extremos de conexión.

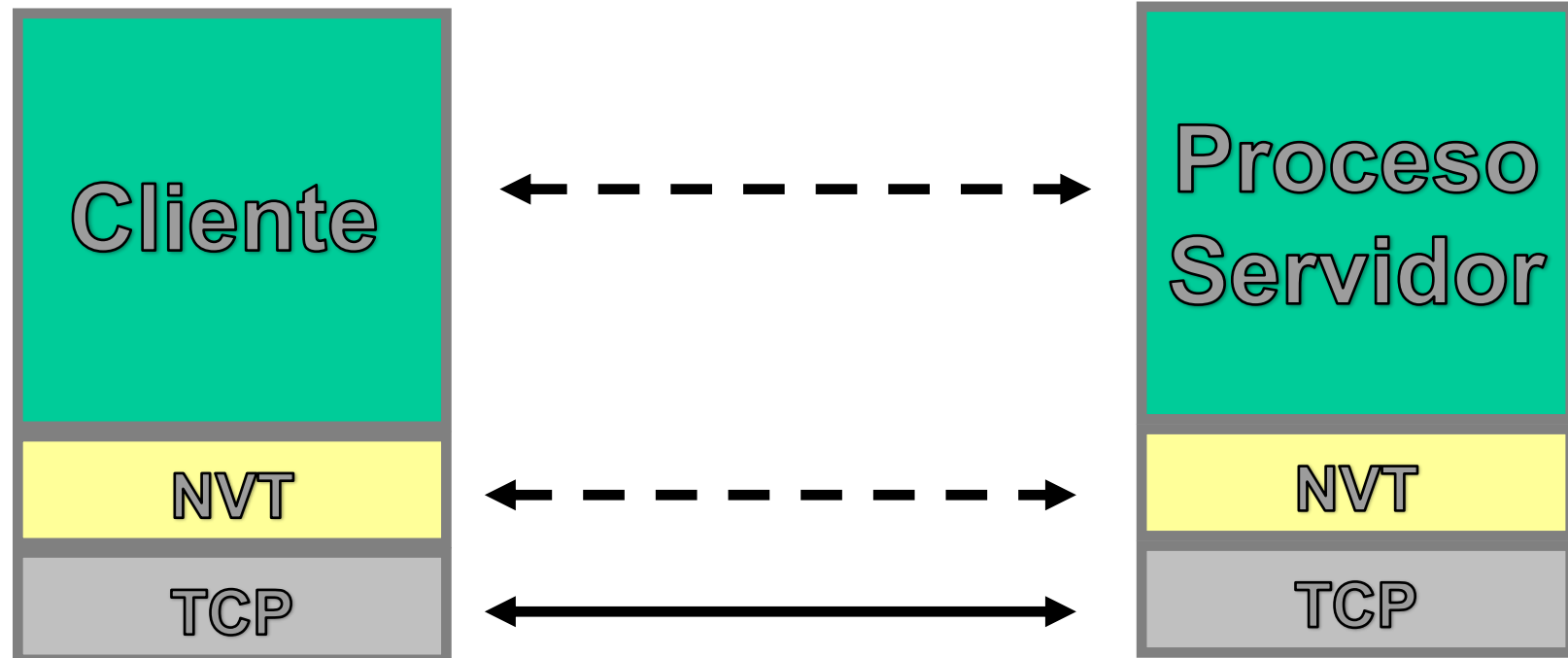
4. Protocolo Telnet

Terminal Virtual de Red

- Dispositivo imaginario que proporciona una representación intermedia de un terminal.
- Elimina la necesidad para los ordenadores SERVIDOR y CLIENTE de guardar información de las características del terminal del otro.
- Se proporciona un lenguaje estándar para la comunicación de funciones de control de terminal.

4. Protocolo Telnet

Terminal Virtual de Red



4. Protocolo Telnet

Terminal Virtual de Red

- Mínimo denominador común de los terminales existentes en el mercado:
 - Bidireccional.
 - Desplazamiento vertical de las líneas en la pantalla (scroll).
 - Longitud ilimitada de línea/página.
 - Caracteres US-ASCII de 7 bits codificados en octetos.
- Ya que distintos sistemas utilizan distintos caracteres de control, TELNET tiene una definición propia de su significado para sacar información en la pantalla.

4. Protocolo Telnet

Terminal Virtual de Red

Códigos de caracteres especiales

Nombre	Nemónico	Código	Descripción
Null	NUL	0	No operación
Line Feed	LF	10	Mover a la siguiente línea de salida
Carriage Return	CR	13	Mover al principio de la línea de impresión
Bell	BEL	7	Señal sonora
Back Space	BS	8	Mover el cursor hacia la izquierda para escribir
Horizontal Tab	HT	9	Salto a un tabulador horizontal
Vertical Tab	VT	11	Salto a un tabulador vertical
Form Feed	FF	12	Mover al principio de una página

Estos son los únicos caracteres que ocasionan un efecto en la salida por pantalla del NVT (*printer*).

4. Protocolo Telnet

Terminal Virtual de Red

Comandos de control

- Se insertan órdenes en el flujo de datos entre el cliente y el servidor.
 - Estas órdenes se preceden por el octeto IAC ("*Interpret as Command*")
 - Para enviar un octeto FF (255 en decimal) como dato se necesita enviar IAC FF
- | | | |
|-----|-----|---|
| SE | 240 | Fin de los parámetros de negociación. |
| NOP | 241 | No operación. |
| DM | 242 | Marca de datos. |
| BRK | 243 | Break. |
| IP | 244 | Interrumpir proceso. |
| AO | 245 | Abortar salida. |
| AYT | 246 | Comprueba si un sistema está funcionando (<i>Are You There?</i>). |
| EC | 247 | Borrar último carácter. |
| EL | 248 | Borrar última línea. |

4. Protocolo Telnet

Terminal Virtual de Red

Comandos de control

GA	249	Continuar.
SB	250	Sub-negociación de la opción indicada.
WILL	251	Se solicita el poder iniciar el uso de una opción o confirmación de que ya se está usando.
WONT	252	Denota la negativa de usar o continuar usando la opción indicada.
DO	253	Es una solicitud para que el otro lado use una opción o la confirmación de que se espera que el otro lado la use.
DONT	254	Pide a la otra parte que deje de usar una opción o indica que ya no se espera que la use más.
IAC	255	Byte con valor 255.

4. Protocolo Telnet

Negociación de opciones

- El NVT proporciona un conjunto mínimo de servicios.
- Pero algunos terminales pueden soportar servicios más sofisticados.
- Los elementos terminales de la comunicación van a negociar un conjunto de opciones añadidas.
- Algunas opciones son:
 - Transmisión binaria (TRANSMIT-BINARY) [RFC 856].
 - Activación modo de echo (ECHO) [RFC 857].
 - Supresión de los envíos de la señal GO-AHEAD (SUPPRESS-GO-AHEAD) [RFC 858].
 - Modo línea, frente al modo carácter.
 - Cambiar conjunto de caracteres usado.

4. Protocolo Telnet

Negociación de opciones

Opción Transmisión en Binario (RFC 856)

- Nombre de opción: TRANSMIT-BINARY.
- Cuando esta opción está activa, todos los bytes recibidos deben ser tratados como un dato de 8 bit, a excepción de un IAC seguido de otro IAC que representa el valor FFh.
- De igual forma la recepción de un IAC seguido de un comando Telnet, deberá ser procesado como el comando que es (si el valor que sigue no es un comando válido, se asumirá que es un NOP, aunque este caso debería evitarse en el envío).

4. Protocolo Telnet

Aplicación de cliente TELNET

- Programa que soporta el protocolo TELNET sobre TCP
- Es un cliente TCP genérico
 - Envía al cliente a través del socket TCP todo lo que se escribe.
 - Imprime todo lo que recibe a través del socket.
 - Es útil para comprobar aplicaciones basadas en servidores TCP como HTTP, SMTP, POP3 o IMAP, aunque no todos.

4. Protocolo Telnet

Aplicación de cliente TELNET

Cliente telnet de Windows 7

```
telnet> open localhost 80
```

```
Get / http/1.1
```

```
Host:localhost
```

```
HTTP/1.1 200 OK
```

```
Date: Mon, 28 Jul 2014 11:24:33 GMT
```

```
Server: Apache/2.2.17 (Win32)
```

```
PHP/5.3.5
```

```
X-Powered-By: PHP/5.3.5
```

```
Content-Length: 4240
```

```
Content-Type: text/html
```

```
<?xml version="1.0" encoding="iso-
```

CLIENTE

```
8859-1"?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD  
XHTML 1.1//EN"
```

```
"http://www.w3.org/TR/xhtml11/DTD/x  
html11.dtd">
```

```
<html lang="en" xml:lang="en">
```

```
...
```

SERVIDOR

5. Protocolo FTP

- **File Transfer Protocol** (Descrito en la RFC 959 en octubre de 1985): Facilita la posibilidad de compartir ficheros.
- Es el estándar 9 de Internet.
- Uso de servidores de ficheros: almacenamiento para ordenadores sin grandes capacidades de almacenamiento.
- Transparencia respecto a la forma en que almacenan los datos las máquinas remotas (sistema de ficheros).
- Proporciona una transferencia de datos eficiente.
- Para uso directo de usuarios y programas.

5. Protocolo FTP

Evolución de FTP. Actualizaciones

- RFC 2228, FTP Security Extensions, Octubre de 1997
- RFC 2640, Internationalization of the File Transfer Protocol, julio de 1999.
- RFC 2773, Encryption using KEA and SKIPJACK, febrero de 2000. Experimental.
- RFC 3659, Extensions to FTP, marzo de 2007.
- RFC 5797, FTP Command and Extension Registry, marzo de 2010.
- RFC 7151, File Transfer Protocol HOST Command for Virtual Hosts, marzo de 2014.

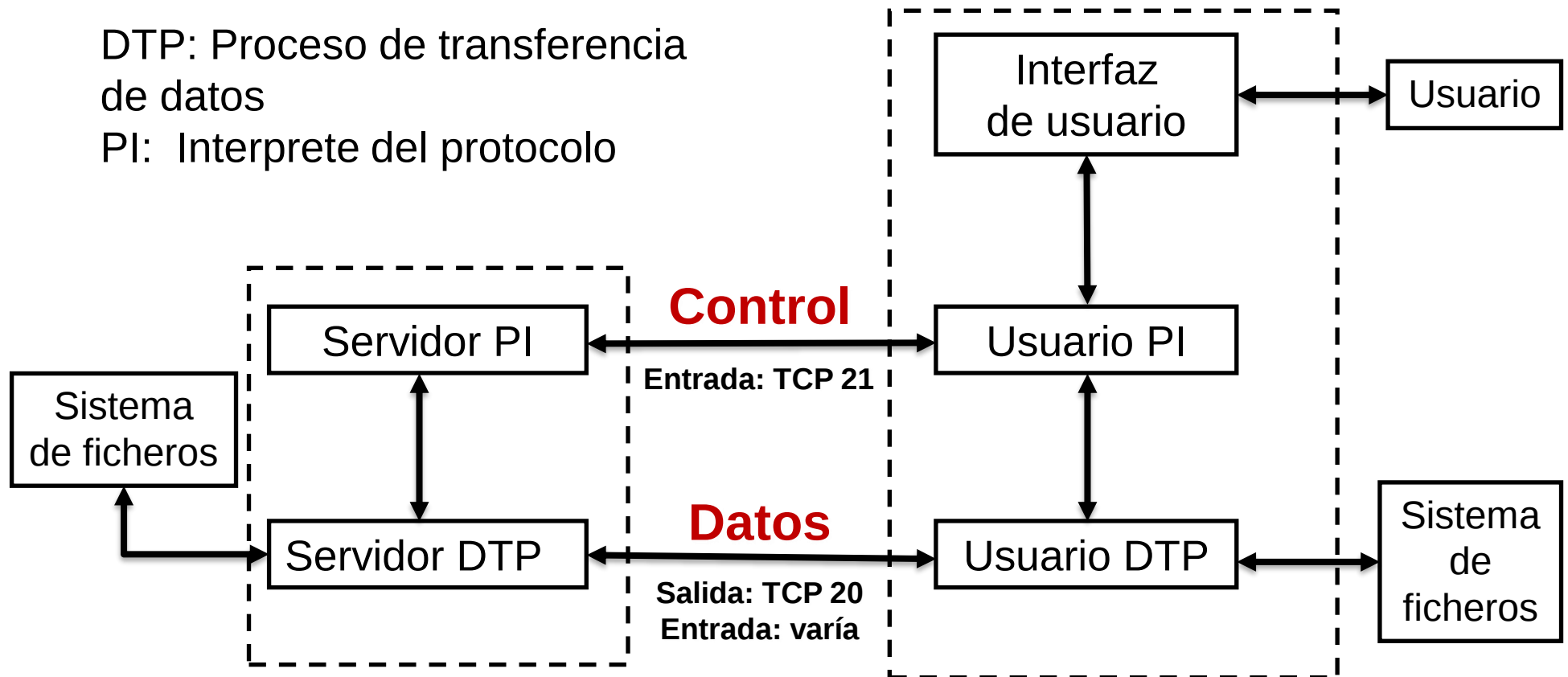


5. Protocolo FTP

Modelo FTP

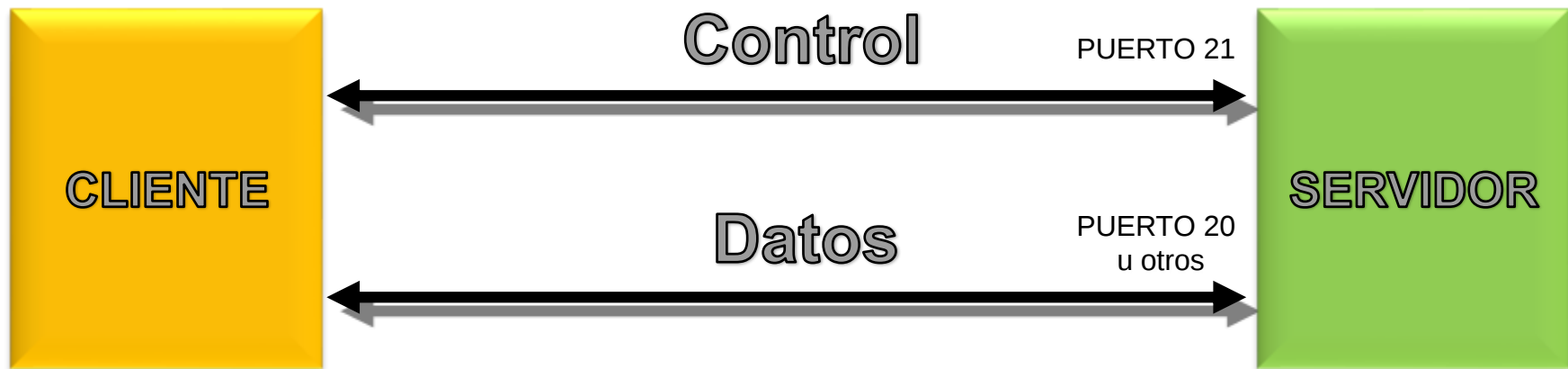
DTP: Proceso de transferencia de datos

PI: Interprete del protocolo



5. Protocolo FTP

Modelo FTP



5. Protocolo FTP

Conexiones de control y datos

- Las funciones de control (comandos) y los códigos de respuesta se transfieren usando la conexión de control.
- Los datos usan la conexión de datos (tanto archivos como listados). Esta conexión, y el proceso de los datos, se generan cuando se necesita.
- Tipo de datos por defecto: ASCII.
- La conexión de control debe estar activa mientras la transferencia tiene lugar y permanece durante toda la sesión.
- La conexión de control usa TELNET.

5. Protocolo FTP

Comandos FTP

Comandos de control de acceso

- USER <user>: especificar usuario.
- PASS <pass>: especificar contraseña.
- CWD <ruta>: cambiar directorio a <ruta> (como el comando cd).
- CDUP: cambia al directorio padre (como cd ..).
- QUIT: Cerrar la sesión y salir.
- Otros: ACCT (especifica una cuenta de usuario), SMNT (monta un nuevo sistema de archivos) y REIN (reinicializar la sesión).

5. Protocolo FTP

Comandos FTP

Comandos de control de transferencia

- TYPE: establece representación de datos del fichero a transmitir (ASCII, EBCDIC, image o local).
 - Al formato *image* (que viene de ser una imagen exacta del fichero y no de su posible contenido gráfico) también se le conoce como BINARY
- STRU: especifica la estructura del fichero (file, record o page).
- MODE: establece el modo de transferencia del archivo (*stream*, *block* o *compressed*).

5. Protocolo FTP

Comandos FTP

Comandos de control de transferencia

- **PORT a1,a2,a3,a4,p1,p2**
 - Especifica la IP y puerto a la que el servidor debería conectarse para la siguiente transferencia de datos.
 - El puerto de salida usado por el servidor es el 20. La interpretación es la siguiente:
Dirección IP: a1.a2.a3.a4
Puerto: $p1 \cdot 256 + p2$
- **PASV: Hace que el servidor entre en modo pasivo.**
 - Esto obliga al servidor a esperar en un puerto concreto para recibir una conexión del cliente. La respuesta del servidor da la información de en qué puerto esperará:
227 Entering Passive Mode (a1,a2,a3,a4,p1,p2)
Dirección IP: a1.a2.a3.a4
Puerto: $p1 \cdot 256 + p2$

5. Protocolo FTP

Comandos FTP

Comandos de servicio

- RETR: recupera un fichero (get).
- STOR: envía un fichero (put).
- APPE: envía un fichero y anexar a uno remoto.
- ABOR: aborta el comando previo.
- PWD: imprime el directorio de trabajo del equipo remoto.
- LIST: transfiere la lista de ficheros del directorio remoto. Emplea la conexión de datos.

5. Protocolo FTP

Respuestas FTP

- Las respuestas se envían usando la conexión de control.
- Las respuestas están formadas por:
 - Un código de estado de 3 dígitos (xyz - 3 caracteres numéricos)
 - 1yz: Operación preliminar correcta, esperar otra respuesta antes de proseguir con otro comando.
 - 2yz: Operación completada con éxito.
 - 3yz: El comando ha sido aceptado pero se necesita más información para proseguir con la operación.
 - 4yz: El comando no se ha aceptado, pero el error es transitorio y se puede reintentar.
 - 5yz: Negativa permanente a completar la petición realizada.
 - Un mensaje de texto.
 - El segundo dígito en la respuesta corresponde con un grupo de respuestas, como x0z para errores de sintaxis, x1z información, x2z conexiones, x3z cuenta y autenticación y x5y sistema de archivos.

5. Protocolo FTP

Transferencia de datos en FTP

- Modo de flujo de datos (*stream mode*).
 - Los datos/ficheros se envían tal cual, sin ninguna información adicional de control, sesión, cabeceras, etc.
 - Se confía en el trabajo en la capa de transporte.
 - El final de la transmisión se marca con el cierre de la conexión de datos.
 - Es el modo por defecto y el más usado.
- Modo de bloque.
 - La información se divide en bloques con una cabecera de 3 bytes para llevar su longitud (2 bytes) y el control sobre los bloques enviados (1 byte).
 - En este caso un algoritmo especial se encarga de este trabajo dentro de FTP.
- Modo comprimido.
 - Se comprime usando un algoritmo relativamente sencillo denominado *run-length encoding* que busca patrones de repeticiones del mismo byte en los datos.

6. Sistema de Nombres de Dominio (DNS)

Necesidad del servicio

- En las redes IP:
 - Las direcciones IP facilitan el encaminamiento
 - Sin embargo son difíciles de manejar para los humanos
- ¿Qué es más fácil?:
 - labtelema.ujaen.es o 150.214.179.118.
- Además son imposibles de intuir:
 - Con respecto al buscador Google es lógico pensar que podemos encontrar su web en español en www.google.es, e imposible de saber que su ipv4 es 172.217.168.163 y su ipv6 2a00:1450:4003:80a::2003.



6. Sistema de Nombres de Dominio (DNS)

Necesidad del servicio

- Para facilitar el manejo de las direcciones IP, a cada dirección se le asocia un nombre:
150.214.179.118 labtelema.ujaen.es
- El nombre es independiente de identificadores de red, direcciones, rutas, etc.
ujaen.es organización
sabiote.ujaen.es máquina
www.ujaen.es servicio
- Será necesario un mecanismo de conversión.

6. Sistema de Nombres de Dominio (DNS)

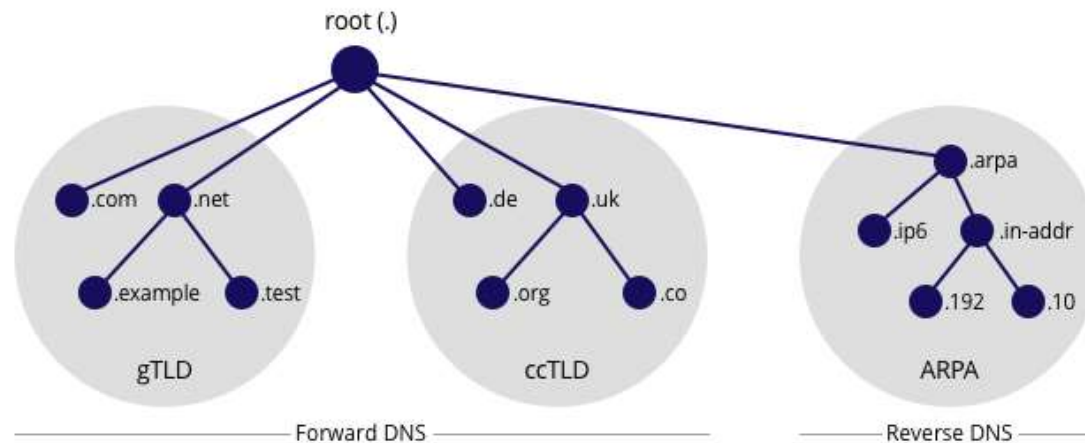
Primeros pasos en la resolución de nombres

- A mediados de los 70:
 - Comunidad pequeña (cientos de máquinas).
 - Un archivo mantiene la información (similar a hosts).
 - Mapeo *nombre – dirección* de las máquinas de la comunidad.
- Problemas:
 - **Consistencia.**
 - Archivo en continuo crecimiento.
 - El archivo ya era obsoleto cuando llegaba a algunas máquinas.
 - Aumento de la carga y tráfico de red.
 - **Colisiones de nombres.**
 - No se puede garantizar que se asignen nombres idénticos a distintas máquinas → dominio plano.
 - La autoridad que gestiona la incorporación de nuevos nombres ha de estar centralizada → carga de trabajo insostenible.

6. Sistema de Nombres de Dominio (DNS)

Primeros pasos en la resolución de nombres

- **SITUACIÓN:**
 - Gran conjunto de nombres.
 - En constante crecimiento.
 - Necesidad de administración no centralizada.
- **SOLUCIÓN:** Sistema de Nombres de Dominio (DNS).



6. Sistema de Nombres de Dominio (DNS)

Sistema de Nombres de Dominio (DNS)

- Es el estándar 13 del IETF.
 - Establece la **correspondencia entre nombres y direcciones IP**
 - Se definen la **sintaxis de los nombres y reglas** para la delegación de autoridad.
 - Implementa una **base de datos distribuida y jerárquica** que traduce nombres y direcciones.
 - **Servicio distribuido.**
 - Gestión descentralizada.
 - Orientado al uso por parte de aplicaciones.
 - **Almacena información** sobre recursos.
 - Esquema jerárquico de nombres basados en dominio.
 - Asegura que los nombres son únicos y fáciles de recordar.
- Es un protocolo que ha requerido un gran esfuerzo normalizador dado su carácter crítico, centrando su evolución en la **eficiencia, robustez y seguridad**.
 - Curiosidad: se tardó 10 años en sacar la primera versión RFC sobre seguridad en DNS (de 1987 a 1997).

6. Sistema de Nombres de Dominio (DNS)

Sistema de Nombres de Dominio (DNS)

Documentos de referencia

Número	Título	Autores	Fecha	Status
RFC 1034, STD 13	Domain names - concepts and facilities	P.V. Mockapetris	Noviembre 1987	Internet Standard
	Obsoletes RFC 973, RFC 882, RFC 883, Updated by RFC 1101, RFC 1183, RFC 1348, RFC 1876, RFC 1982, RFC 2065, RFC 2181, RFC 2308, RFC 2535, RFC 4033, RFC 4034, RFC 4035, RFC 4343, RFC 4035, RFC 4592, RFC 5936, RFC 8020, RFC 8482, RFC 8767 RFC 9471, Errata			
RFC 1035, STD 13	Domain names - implementation and specification	P.V. Mockapetris	Noviembre 1987	Internet Standard
	Obsoletes RFC 973, RFC 882, RFC 883, Updated by RFC 1101, RFC 1183, RFC 1348, RFC 1876, RFC 1982, RFC 1995, RFC 1996, RFC 2065, RFC 2136, RFC 2181, RFC 2137, RFC 2308, RFC 2535, RFC 2673, RFC 2845, RFC 3425, RFC 3658, RFC 4033, RFC 4034, RFC 4035, RFC 4343, RFC 5936, RFC 5966, RFC 6604, RFC 7766, RFC 8482, RFC 8490, RFC 8767, RFC 9619, Errata			

6. Sistema de Nombres de Dominio (DNS)

Sistema de Nombres de Dominio (DNS)

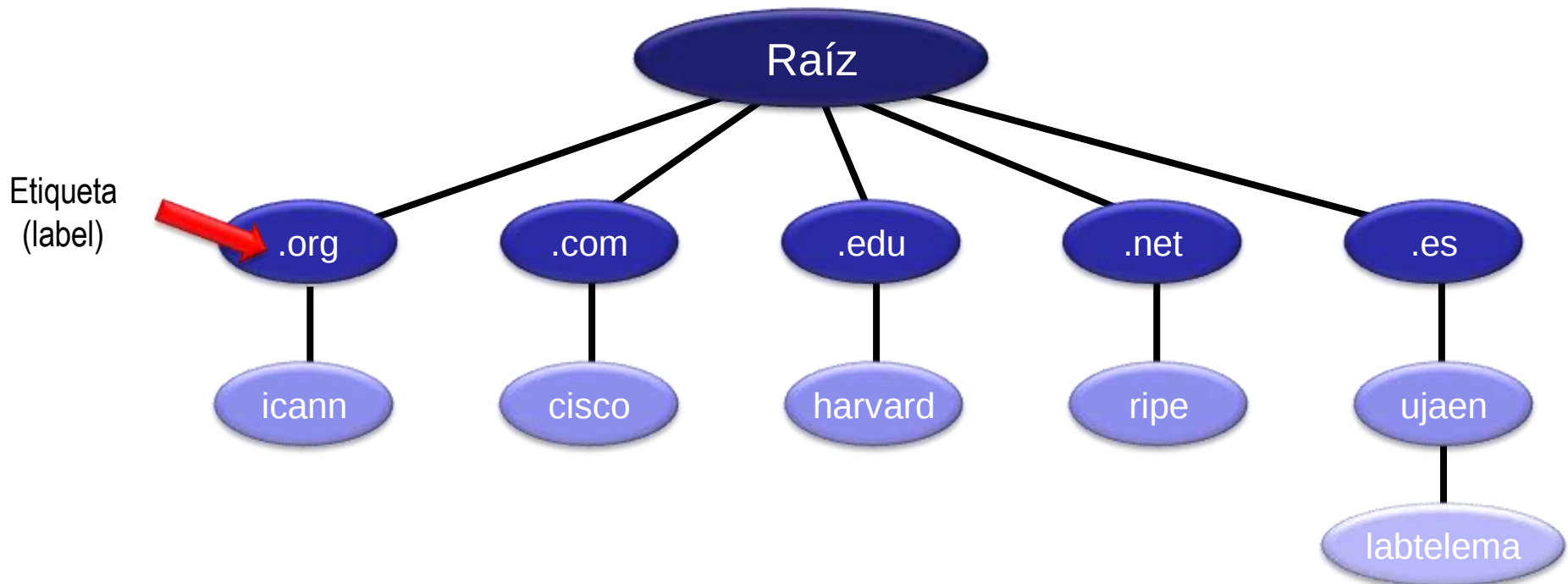
Extensiones sobre seguridad

- RFC 4033, *DNS Security Introduction and Requirements*
- RFC 4034, *Resource Records for the DNS Security Extensions*
- RFC 4035, *Protocol Modifications for the DNS Security Extensions*
- RFC 4509, *Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records*
- RFC 4470, *Minimally Covering NSEC Records and DNSSEC On-line Signing*
- RFC 5011, *Automated Updates of DNS Security (DNSSEC) Trust Anchors*
- RFC 5155, *DNS Security (DNSSEC) Hashed Authenticated Denial of Existence*
- RFC 5702, *Use of SHA-2 Algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC*
- RFC 5910, *Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP)*
- RFC 5933, *Use of GOST Signature Algorithms in DNSKEY and RRSIG Resource Records for DNSSEC*
- RFC 7858, *Specification for DNS over Transport Layer Security (TLS)*
- RFC 8310, *Usage Profiles for DNS over TLS and DNS over DTLS*

6. Sistema de Nombres de Dominio (DNS)

Sistema de Nombres de Dominio (DNS)

Espacio de nombres

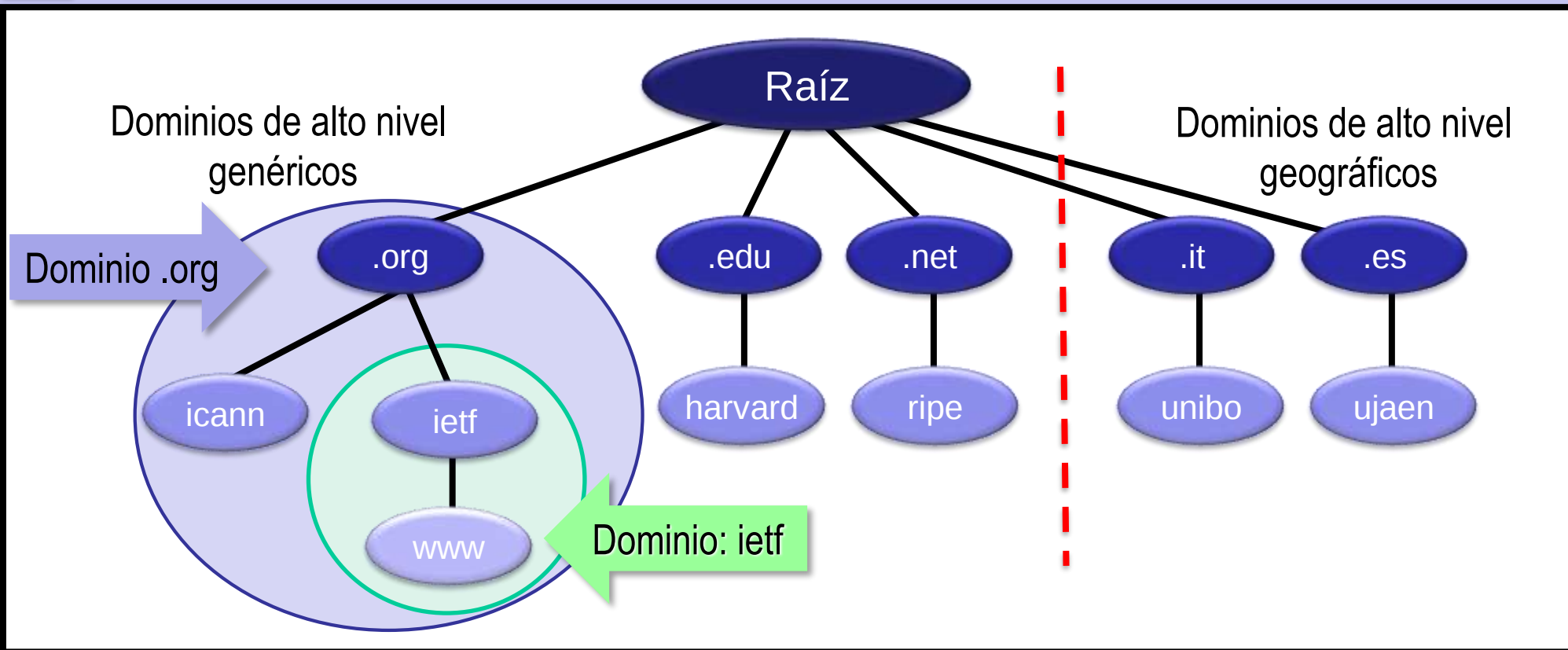


Este árbol puede tener tan solamente 128 niveles, el nivel 0 es el raíz (root) y el más profundo sería el 127.

6. Sistema de Nombres de Dominio (DNS)

Sistema de Nombres de Dominio (DNS)

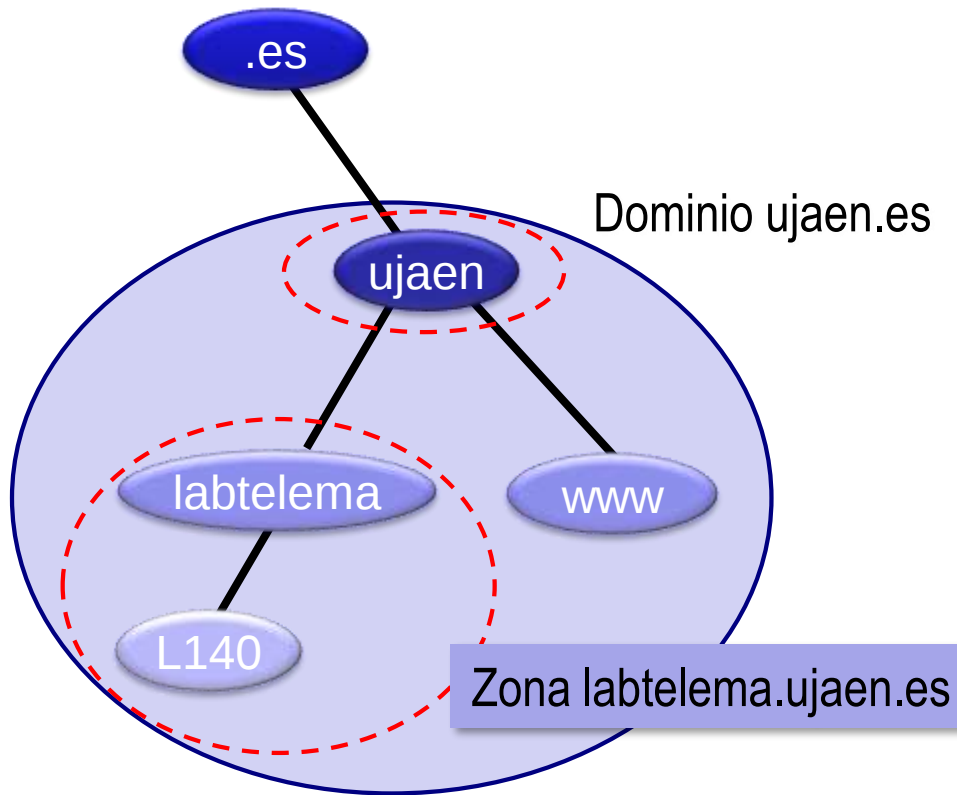
Espacio de nombres, dominios



6. Sistema de Nombres de Dominio (DNS)

Sistema de Nombres de Dominio (DNS)

Dominios y zonas



- Una zona es sobre lo que tiene autoridad un servidor.
- Si un dominio no tiene subdominios, la zona y el dominio coinciden.

6. Sistema de Nombres de Dominio (DNS)

Sistema de Nombres de Dominio (DNS)

Dominios de primer nivel (TLDs)

- TLD: *Top level domain*.
- Los hay genéricos y geográficos.
- Son el nivel más alto de nombre de dominio accesible (parte más a la derecha de un nombre de dominio).
- Los servidores Raíz guardan una referencia a todos ellos.



6. Sistema de Nombres de Dominio (DNS)

Sistema de Nombres de Dominio (DNS)

Dominios de primer nivel (TLDs)

Dominios Genéricos (gTLDs)

- Antes de 2012 había 22 dominios genéricos.
- Entre otros:
 - .edu: instituciones educativas
 - .gov: gobierno federal de EE. UU.
 - .mil: organismos militares EE. UU.
 - .int: organismos internacionales
 - .com: para empresas con actividad comercial de todo tipo.
 - .org: organizaciones no lucrativas.
 - .net: proveedores de red.
 - .biz: firmas comerciales, negocios.
 - .info: organizaciones que proporcionan información.
 - .name: Aquellos que desean una nomenclatura personal.
 - .museum: Museos y organizaciones acordes.
 - .coop: Cooperativas.
 - .aero: Compañías aéreas y de transporte aéreo.
 - .pro: Para profesionales como médicos, abogados, etc.

6. Sistema de Nombres de Dominio (DNS)

Sistema de Nombres de Dominio (DNS)

Dominios de primer nivel (TLDs)

Dominios Geográficos (ccTLD)

- Geográficos (*Country Codes – cc*)

.es	España
.jp	Japón
.de	Alemania
.uk	Reino Unido
.fr	Francia
.it	Italia



- Están gestionados y controlados por The Internet Corporation for Assigned Names and Numbers (ICANN) y la Domain Name Supporting Organization (DNSO).
- Más información en: www.icann.org

6. Sistema de Nombres de Dominio (DNS)

Sistema de Nombres de Dominio (DNS)

Dominios de primer nivel (TLDs)

Nuevos gTLDs

- El ICANN lanzó en 2011 después de muchos años de estudio el **New gTLD Program** que permite la solicitud directa de nuevos gTLDs.
- El programa se abrió en Enero de 2012 y para el 17 de diciembre de ese año ya había recibido 1.930 solicitudes (fuente icaan.org).
 - Datos a 10 de septiembre de 2024: 1930 solicitudes de las cuales 1241 han sido aceptadas e introducidas en Internet.
 - Ejemplos de nuevos gTLDs: IBM, CASA, WORLD, PHARMACY, PIZZA, YOUTUBE, GMAIL, BUSINESS, NETWORK, LACAIXA, JUEGOS, FUTBOL, HOTEL, MAP, SEARCH, HAIR, MOTO, RADIO, ...

6. Sistema de Nombres de Dominio (DNS)

Sistema de Nombres de Dominio (DNS)

Dominios de primer nivel (TLDs)

Nuevos gTLDs

1930 total number of applications received

911
North America

675
Europe

24
South America

17
Africa

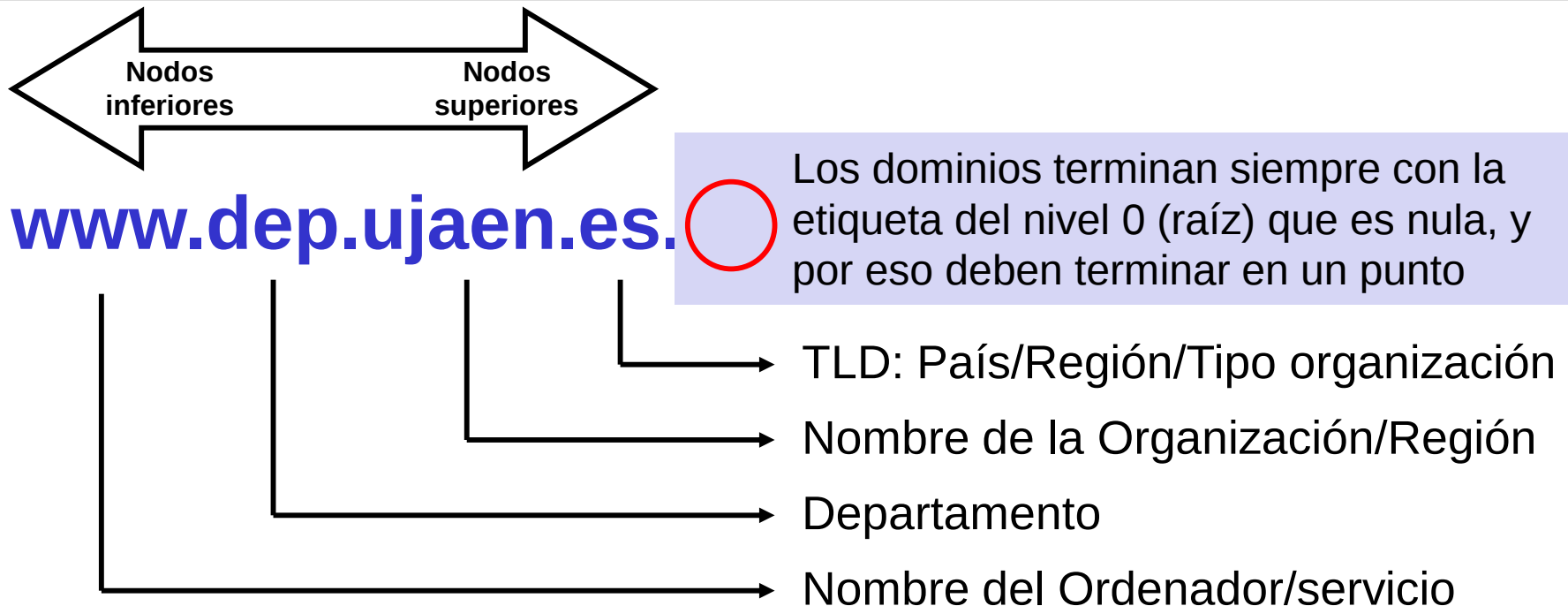
303
Asia Pacific

Fuente: newgtlds.icann.org/en/program-status/statistics

6. Sistema de Nombres de Dominio (DNS)

Sistema de Nombres de Dominio (DNS)

Estructura de un nombre de dominio



6. Sistema de Nombres de Dominio (DNS)

Sistema de Nombres de Dominio (DNS)

Estructura de un nombre de dominio

[www.dep.ujaen.es.](http://www.dep.ujaen.es)

Es un nombre de dominio cualificado completamente o FQDN
(siglas del inglés *fully qualified domain name*)

- Los nombres de dominio son independientes de las mayúsculas y tradicionalmente solo permitían caracteres ASCII.
 - A partir de 2003 se comenzó a permitir el uso de otros caracteres en las etiquetas a través del uso de UNICODE.
 - RFC 3490 *Internationalizing Domain Names in Applications* (IDNA), actualmente obsoleta por las RFCs 5890 y 5891).

6. Sistema de Nombres de Dominio (DNS)

Sistema de Nombres de Dominio (DNS)

Estructura de un nombre de dominio

- Por motivos de simplicidad en las implementaciones, el número total de octetos para representar un dominio está limitado a 255, incluyendo las etiquetas y los bytes de longitud de etiqueta.
- Las etiquetas como máximo pueden tener 63 caracteres.
- Ejemplo F.ISI.ARPA. (RFC 1035):

20	1	F
22	3	I
24	S	I
26	4	A
28	R	P
30	A	0

00xxxxxx: los dos bits MSB deben ser 0 en las etiquetas. Si son 11 es un puntero a un mensaje

6. Sistema de Nombres de Dominio (DNS)

Elementos del DNS

Servidores de nombres

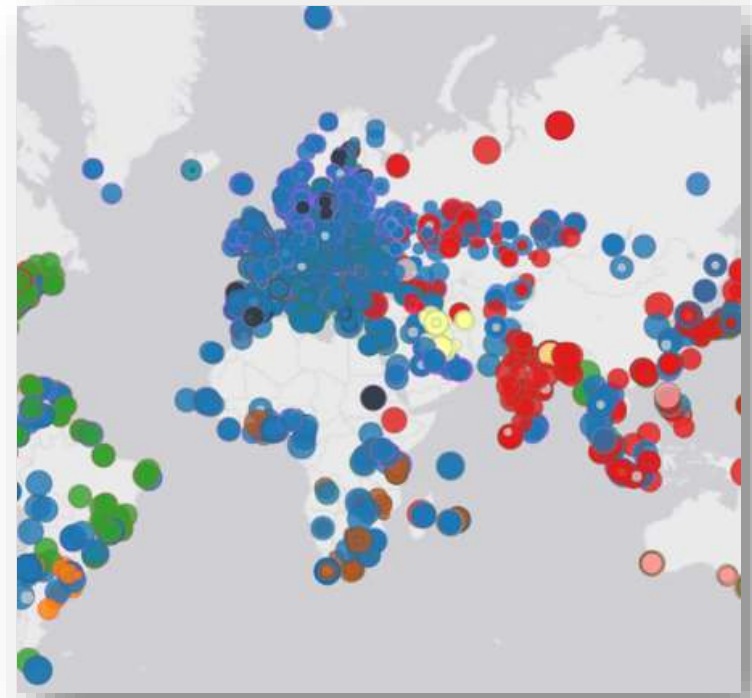
- Contienen la información sobre los dominios y debe procesar peticiones de forma concurrente.
- Cada servidor mantiene la autoridad sobre un subconjunto del espacio de nombres.
- Hay servidores primarios y secundarios, pero ambos tienen la misma autoridad sobre la zona:
 - **Servidor primario:** almacena localmente los datos de la zona sobre la que tiene autoridad, por lo que es el responsable de crear, mantener y actualizar el fichero de zona.
 - **Servidor secundario:** este servidor transfiere desde el primario toda la información de la zona y la almacena localmente. No crea o modifica registros, eso lo hace el servidor primario, solamente se descarga una nueva versión cuando sea oportuno.
 - La transferencia de información desde el primario al secundario se denomina “transferencia de zona” o “zone transfer”.
- La razón de que exista un servidor secundario es por motivos de redundancia y protección ante fallos.

6. Sistema de Nombres de Dominio (DNS)

Elementos del DNS

Servidores de nombres

- Los servidores raíz contienen información sobre quienes son los servidores de nombres de los dominios de primer nivel (TLD's).
- Los servidores de los TLD's contienen información de quienes son los servidores de los dominios de siguiente nivel.
- Todos los servidores de nombres saben quiénes son los raíz.



6. Sistema de Nombres de Dominio (DNS)

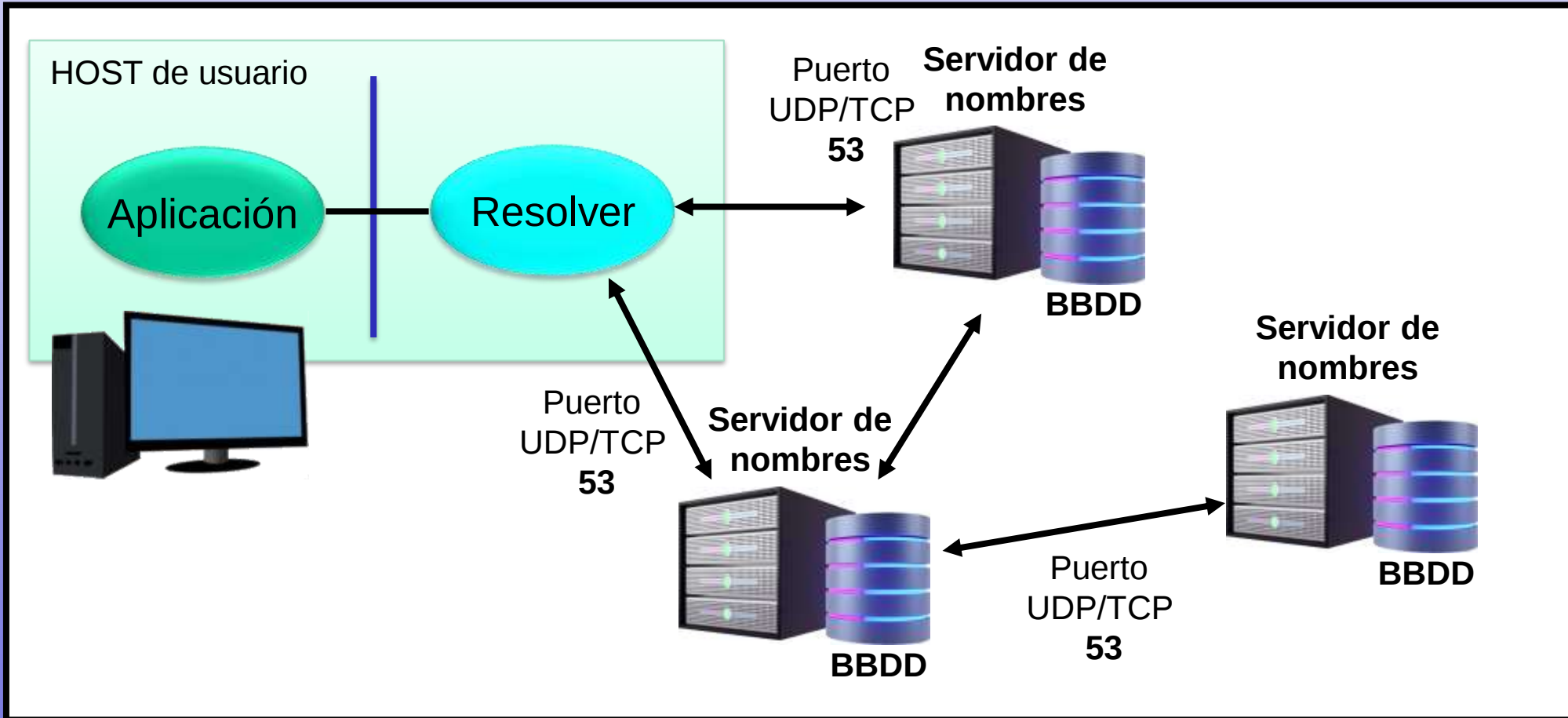
Elementos del DNS

Resolvers

- Se describe en sus versiones iniciales en las RFC 1034 y RFC 1035.
- Realizan peticiones a los servidores ante consultas de las aplicaciones clientes.
 - Estas peticiones son fundamentalmente UDP, pero en la RFC 1035 se permitía que los resolver o los servidores recursivos emplearan TCP.
 - En la RFC 7766 (previamente 5966) se describen como operar las peticiones DNS sobre TCP.
 - El hecho de que se pueden emplear peticiones sobre TCP evita los problemas que ocurren en redes con tráfico UDP filtrado o de respuestas por encima de los 512 bytes.

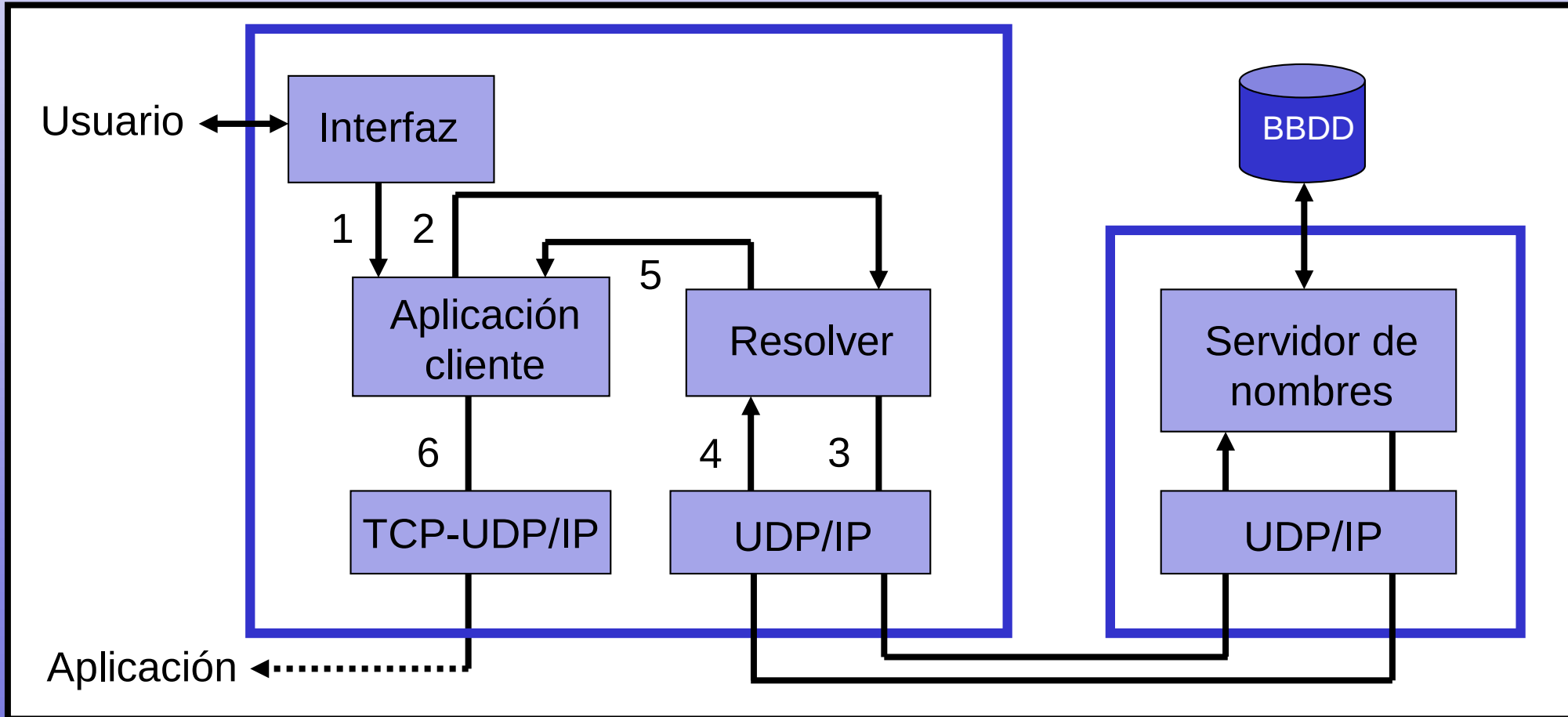
6. Sistema de Nombres de Dominio (DNS)

Modelo funcional



6. Sistema de Nombres de Dominio (DNS)

Proceso de resolución de nombres



6. Sistema de Nombres de Dominio (DNS)

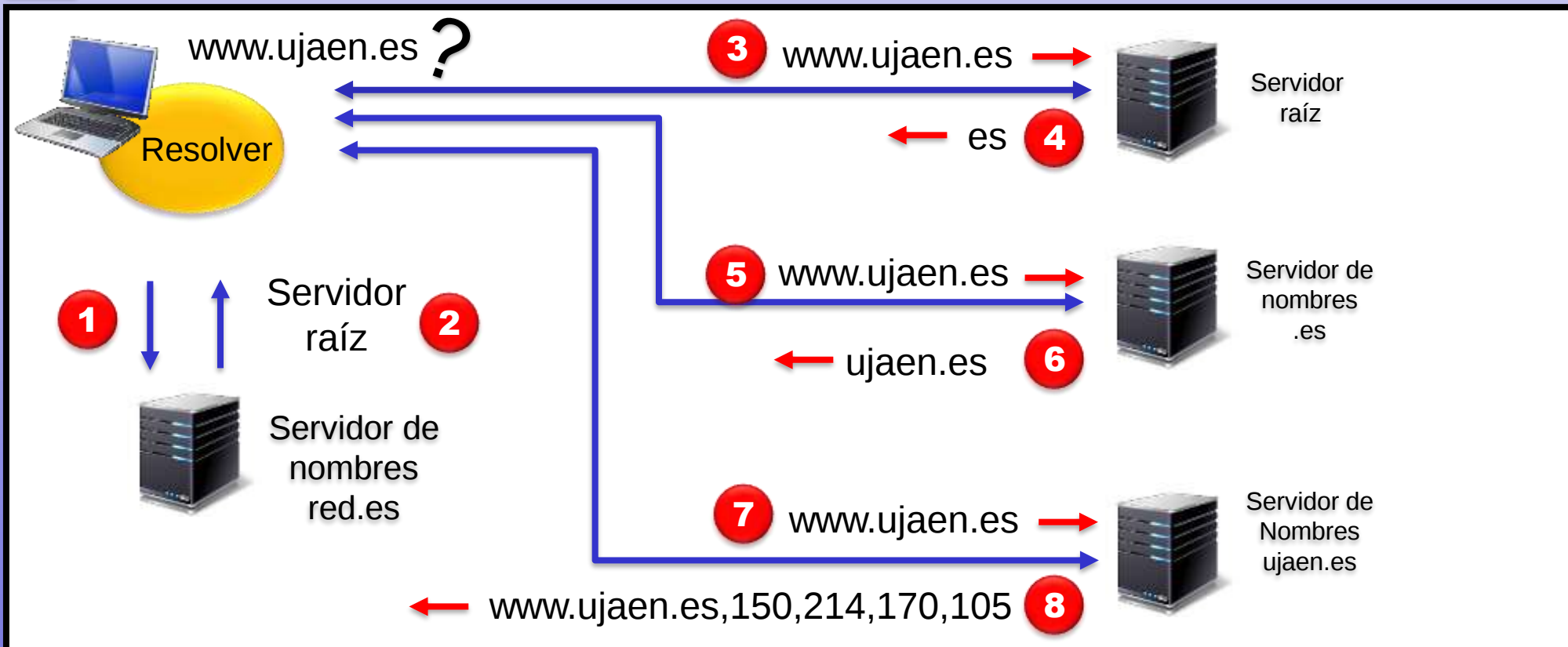
Proceso de resolución de nombres

- Un resolver debe conocer al menos un servidor local.
- Cada servidor debe conocer al menos un servidor raíz y a los servidores de dominio jerárquicamente inferiores.
- Los servidores deben conocer todos los dominios de nivel inferior.
- El cliente envía una petición al servidor:
 - Si el servidor no puede resolverla: contacta con otro servidor.
 - O devuelve una referencia.

6. Sistema de Nombres de Dominio (DNS)

Proceso de resolución de nombres

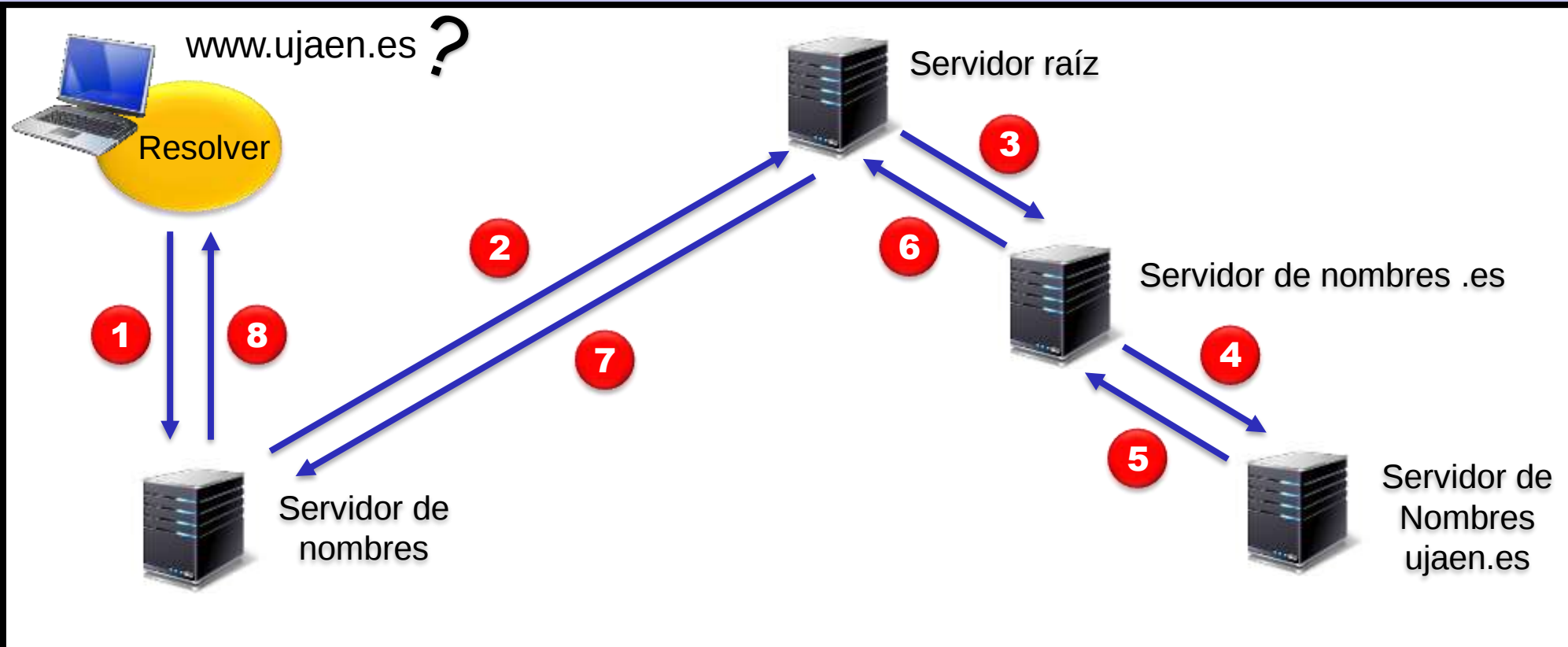
Consulta iterativa



6. Sistema de Nombres de Dominio (DNS)

Proceso de resolución de nombres

Consulta recursiva



6. Sistema de Nombres de Dominio (DNS)

Estructura de la información del sistema DNS

- Cada nombre de dominio tiene su información asociada en forma de Registros de Recursos (*Resource Records – RR*)
- Formato de un registro:
[nombre] [ttl] IN <tipo de registro> <valor>
- Donde:
 - **[nombre]** es el nombre del objeto referenciado por el RR. Puede ser un nombre de estación o un nombre de dominio. La cadena especificada es relativa al dominio actual a no ser que termine con un punto “.”. Si el nombre es omitido, el registro aplica al último objeto que tenga un nombre definido.

6. Sistema de Nombres de Dominio (DNS)

Estructura de la información del sistema DNS

- **[ttl]** es el tiempo de vida del registro. Define la cantidad de segundos que la información sobre este registro puede ser mantenida en la memoria de un sistema remoto. Si el ttl es omitido usa el ttl del RR SOA.
 - Un valor de 0 indica que el valor solo se debe usar para una transacción y no debe ser almacenado en caché.
- **IN** identifica el registro como de Internet. Existen otras clases de registros, pero casi no se usan.
- **<tipo de registro>** identifica el tipo de RR de acuerdo a la tabla que sigue.
- **<valor>** es la información específica al tipo de RR. Puede ser:
 - Un número, como una dirección IPv4 (4 octetos) o una dirección IPv6 (16 octetos).
 - Un nombre de dominio.
 - Un puntero (11xx xxxx xxxx xxxx).
 - Una cadena de caracteres.

6. Sistema de Nombres de Dominio (DNS)

Estructura de la información del sistema DNS

Algunos tipos de Registros

Nombre del RR	Tipo de Registro	Función
Inicio de Autoridad	SOA	Indica el inicio de los datos para una zona y define parámetros que afectan a todos los registros para la zona
Servidor de Nombres	NS	Identifica el servidor de nombres para el dominio
Dirección	A	Convierte un nombre de estación en una dirección de IPv4
Dirección IPv6	AAAA	Convierte un nombre de estación en una dirección de IPv6 (RFC 3596)
Puntero	PTR	Convierte una dirección de IP a un nombre de estación
Oficina de Correos	MX	Identifica hacia donde se debe enviar el correo electrónico para el dominio o estación
Nombre Canónico	CNAME	Define un alias para una estación ya definida
Información de Estación	HINFO	Describe el hardware y el sistema operativo de una estación
Servicios Ofertados	WKS	Anuncia servicios de redes ofertados
Texto	TXT	Almacena cualquier información arbitraria

6. Sistema de Nombres de Dominio (DNS)

Estructura de la información del sistema DNS

Estructura del registro SOA

Campo	Longitud	Descripción
MNAME		El <domain-name> del servidor de nombres que es el original o fuente primaria para esta zona.
RNAME		El <domain-name> que especifica el buzón de correo de la persona responsable de la esta zona.
SERIAL	32 bits	Número de versión de la copia original de esta zona.
REFRESH	32 bits	Intervalo de tiempo, en segundos, en el que la zona debe ser actualizada.
RETRY	32 bits	Intervalo de tiempo, en segundos, que debe transcurrir ante de reintentar una petición fallida.
EXPIRE	32 bits	Intervalo de tiempo, en segundos, que especifica el límite superior del tiempo que puede transcurrir antes de que la zona deje de ser autoridad.
MINIMUM	32 bits	El valor mínimo en el campo TTL que debe exportarse en cualquier RR de esta zona.

6. Sistema de Nombres de Dominio (DNS)

Estructura de la información del sistema DNS

Ejemplo

- telematica.net.zone

```
@      IN      SOA      @  admin.telematica.net (
                        2 ; serial
                        28800 ; refresh
                        7200 ; retry
                        604800 ; expire
                        86400 ; ttl )

@      IN      NS      ns.telematica.net.
Uno    IN      A        150.214.179.46
Ns     IN      A        150.214.179.46
www    IN      CNAME    uno.telematica.net.
```

6. Sistema de Nombres de Dominio (DNS)

Estructura de la información del sistema DNS

Ejemplo

- 179.214.150.in-addr.arpa.zone

```
@      IN      SOA      @ admin.telematica.net (
                        4 ; serial
                        28800 ; refresh
                        7200 ; retry
                        604800 ; expire
                        86400 ; ttl )
@      IN      NS      ns.telematica.net.
46     IN      PTR     uno.telematica.net.
```

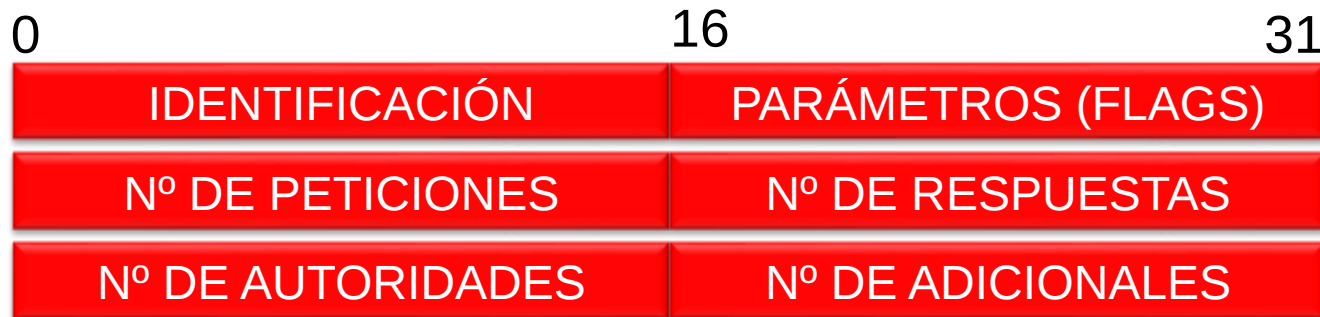
6. Sistema de Nombres de Dominio (DNS)

Formato de los mensajes

Cabecera

- Es común a peticiones y respuestas (RFC 6895).

- Formato:



- El número de respuestas, autoridades y adicionales en un mensaje de petición van a 0.

6. Sistema de Nombres de Dominio (DNS)

Formato de los mensajes

Cabecera - Flags



- QR: petición (0) o respuesta (1)
- OpCode, Tipo de petición o respuesta (RFC 6195). Lo pone quien realiza la petición y se copia en la respuesta.

opCode	Name	Reference
0	Query	[RFC1035]
1	IQuery (Inverse Query, Obsolete)	[RFC3425] Era como si se solicita una IP a través del RR PTR
2	Status	[RFC1035]
3	available for assignment	
4	Notify	[RFC1996] Un servidor maestro comunica un cambio a un servidor esclavo
5	Update	[RFC2136] Actualiza un recurso sin tener que descargar toda la zona.
6-15	available for assignment	

- AA (respuesta de autoridad): si la respuesta proviene de un servidor que es una autoridad su valor es 1.
- TC (truncado): su valor es 1 cuando la respuesta se ha truncado a 512 bytes usando UDP.
- RD (recursividad deseada): A 1 cuando el cliente solicita la resolución por recursividad, se repite en la respuesta.
- RA (recursividad disponible): Cuando está a 1 en la respuestas indica que la recursividad está disponible (solo para respuestas).
- AD (Authenticated Data) y CD (Checking Disabled) [RFC 4033] para DNS seguro.

6. Sistema de Nombres de Dominio (DNS)

Formato de los mensajes

Cabecera - Flags



- rCode (4 bits): Informa del error en la respuesta. También puede usarse dentro de algunos RR: OPT RRs [RFC2671], TSIG RRs [RFC2845], y TKEY RRs [RFC2930].

0	NoError	No Error	[RFC1035]
1	FormErr	Format Error	[RFC1035]
2	ServFail	Server Failure	[RFC1035]
3	NXDomain	Non-Existent Domain	[RFC1035]
4	NotImp	Not Implemented	[RFC1035]
5	Refused	Query Refused	[RFC1035]
6	YXDomain	Name Exists when it should not	[RFC2136]
7	YXRRSet	RR Set Exists when it should not	[RFC2136]
8	NXRRSet	RR Set that should exist does not	[RFC2136]
9	NotAuth	Server Not Authoritative for zone	[RFC2136]
10	NotZone	Name not contained in zone	[RFC2136]
11 - 15	Available for assignment		

Hay más códigos (hasta 16 bits) para el uso en los RRs antes mencionados.

6. Sistema de Nombres de Dominio (DNS)

Formato de los mensajes

Tipos de mensajes



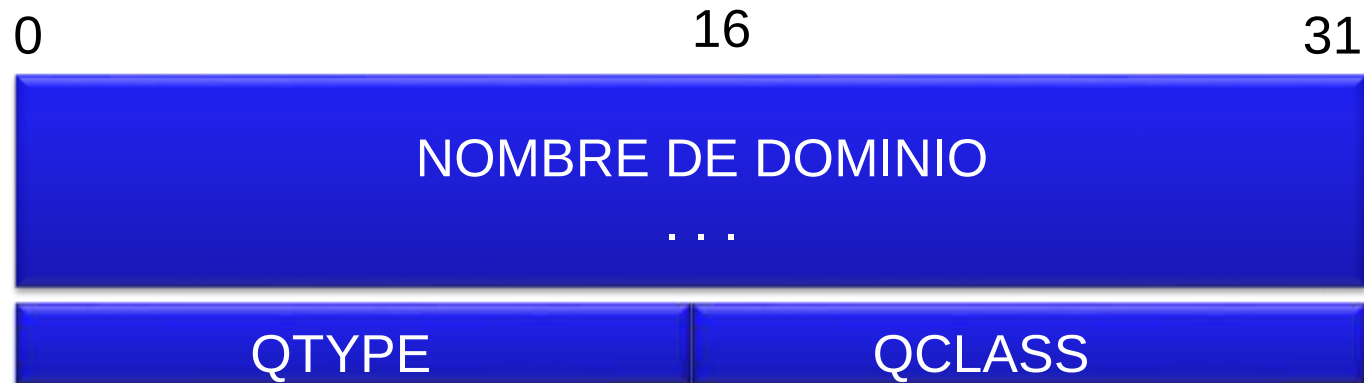
Peticiones



Respuestas

6. Sistema de Nombres de Dominio (DNS)

Formato de los mensajes



QTYPE: **TIPO DE RR**

QCLASS: **CLASE (IN)**

6. Sistema de Nombres de Dominio (DNS)

Formato de los mensajes

RR de respuesta



6. Sistema de Nombres de Dominio (DNS)

Operar con el DNS

- En UNIX y Windows se puede emplear el comando nslookup para hacer peticiones de resolución de nombres y direcciones:

```
>nslookup www.ujaen.es
DNS request timed out.
    timeout was 2 seconds.
Servidor:  UnKnown
Address:  192.168.1.1
```

```
Respuesta no autoritativa:
Nombre:  sabiote.ujaen.es
Address:  150.214.170.105
Aliasas:  www.ujaen.es
```

```
>nslookup 150.214.170.105
DNS request timed out.
    timeout was 2 seconds.
Servidor:  UnKnown
Address:  192.168.1.1
```

```
Nombre:  red.ujaen.es
Address:  150.214.170.105
```

6. Sistema de Nombres de Dominio (DNS)

Ventajas del DNS

- Espacio de nombre jerárquico:
 - Elimina problema de nombres repetidos.
 - Permite subdominios iguales en dominios distintos.
- Elimina problema de carga y tráfico de red:
 - Información distribuida.
- Asegura consistencia:
 - Actualización de la información de forma automática.
 - Gestión descentralizada.
- El tráfico que genera es pequeño:
 - Generalmente son paquetes UDP de 512 bytes como máximo.

6. Sistema de Nombres de Dominio (DNS)

Ventajas del DNS

Uso de TCP

- En la RFC 1035 se permitía que los resolver o los servidores recursivos emplearan TCP.
- En la RFC 7766 (previamente 5966):
 - Todas las implementaciones deben soportar tanto UDP como TCP.
 - Con TCP se evita el truncado de información de 512 bytes.
 - Se emplea fundamentalmente para la transferencia de datos de zona.

6. Sistema de Nombres de Dominio (DNS)

Inconvenientes del DNS

- Tiene un inconveniente: si fallan los servidores raíz falla TODO el sistema DNS.
- Necesidad de réplicas:
 - 13 servidores raíz replicados y repartidos por todo el mundo (del A al M).
 - Difícil que fallen todos a la vez.
 - Lista completa en <http://www.root-servers.org/>
- Cada servidor raíz tiene a su vez varias réplicas.
 - Los servidores C, D, E, F, K y L tienen una réplica en Madrid (9 en total), el E, F, J y K en Barcelona (7 en total), el E en Málaga, y el L en El Prat de Llobregat.
- Aspectos clave en servidores raíz:
 - copias de seguridad, redundancia del hardware y seguridad.

6. Sistema de Nombres de Dominio (DNS)

Servidores raíz

- Imágenes conseguidas a través de www.root-servers.org



Ejemplo de uso sobre Wireshark

*Ethernet

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

dns

No.	Time	Source	Destination	Protocol	Length	Info
109	3.116976	192.168.1.133	87.216.1.65	DNS	78	Standard query 0x9925 A labtelema.ujaen.es
128	3.180864	87.216.1.65	192.168.1.133	DNS	94	Standard query response 0x9925 A labtelema.ujaen.es A 150.214.179.118

> Frame 109: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface \Device\NPF_{FA535B7F-599C-44A8-A31C-2BC973962CA0}, id 0

> Ethernet II, Src: Giga-Byt_6a:11:de (18:c0:4d:6a:11:de), Dst: Comtrend_5d:a8:fc (1c:64:99:5d:a8:fc)

> Internet Protocol Version 4, Src: 192.168.1.133, Dst: 87.216.1.65

> User Datagram Protocol, Src Port: 52177, Dst Port: 53

Source Port: 52177

Destination Port: 53

Length: 44

Checksum: 0x1b84 [unverified]

[Checksum Status: Unverified]

[Stream index: 4]

> [Timestamps]

UDP payload (36 bytes)

Domain Name System (query)

Transaction ID: 0x9925

> Flags: 0x0100 Standard query

Questions: 1

Answer RRs: 0

Authority RRs: 0

Additional RRs: 0

Queries

> labtelema.ujaen.es: type A, class IN

[Response In: 128]

0000 1c 64 99 5d a8 fc 18 c0 4d 6a 11 de 08 00 45 00 .d.]....Mj....E.

0010 00 40 6d f6 00 00 80 11 00 00 c0 a8 01 85 57 d8 .@m.....W.

0020 01 41 cb d1 00 35 00 2c 1b 84 99 25 01 00 00 01 .A..5.,...%...

0030 00 00 00 00 00 00 09 6c 61 62 74 65 6c 65 6d 61l abtelema

0040 05 75 6a 61 65 6e 02 65 73 00 00 01 00 01 .ujaen.e s....

Ejemplo de uso sobre Wireshark

*Ethernet

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

dns

No.	Time	Source	Destination	Protocol	Length	Info
109	3.116976	192.168.1.133	87.216.1.65	DNS	78	Standard query 0x9925 A labtelema.ujaen.es
128	3.180864	87.216.1.65	192.168.1.133	DNS	94	Standard query response 0x9925 A labtelema.ujaen.es A 150.214.179.118

> Frame 128: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface \Device\NPF_{FA535B7F-599C-44A8-A31C-2BC973962CA0}, id 0

> Ethernet II, Src: Comtrend_5d:a8:fc (1c:64:99:5d:a8:fc), Dst: Giga-Byt_6a:11:de (18:c0:4d:6a:11:de)

> Internet Protocol Version 4, Src: 87.216.1.65, Dst: 192.168.1.133

> User Datagram Protocol, Src Port: 53, Dst Port: 52177

▼ Domain Name System (response)

Transaction ID: 0x9925

> Flags: 0x8180 Standard query response, No error

Questions: 1

Answer RRs: 1

Authority RRs: 0

Additional RRs: 0

▼ Queries

> labtelema.ujaen.es: type A, class IN

▼ Answers

▼ labtelema.ujaen.es: type A, class IN, addr 150.214.179.118

Name: labtelema.ujaen.es

Type: A (Host Address) (1)

Class: IN (0x0001)

Time to live: 86400 (1 day)

Data length: 4

Address: 150.214.179.118

[Request In: 109]

[Time: 0.063888000 seconds]

DNS

0000 18 c0 4d 6a 11 de 1c 64 99 5d a8 fc 08 00 45 00 ..Mj...d.]....E.

0010 00 50 d3 f3 40 00 3b 11 50 63 57 d8 01 41 c0 a8 .P...@;.PcW..A..

0020 01 85 00 35 cb d1 00 3c 9d f8 99 15 81 80 00 01 ...5...<...%....

0030 00 01 00 00 00 00 09 6c 61 62 74 65 6c 65 6d 61l abtelema

0040 05 75 6a 61 65 6e 02 65 73 00 00 01 00 01 c0 0c .ujaen.e s.....

0050 00 01 00 01 00 01 51 80 00 04 96 d6 b3 76Q...v