

# *Tema 6*

# *La Capa de Aplicación*



# Tema 6: La Capa de Aplicación

---

1. Introducción
2. DNS- El sistema de nombres de dominio
3. El correo electrónico
4. Word Wide Web
5. Audio y Video
6. Bibliografía

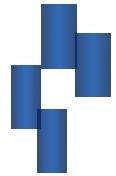
- La capa de aplicación se encarga de:
  1. Definir los protocolos de aplicaciones que el usuario requiere con frecuencia
  2. La capa de aplicación utiliza a las capas inferiores para proporcionar un transporte confiable
  3. Aún en esta capa de aplicación, se necesitan protocolos de ayuda que permitan el funcionamiento de las aplicaciones reales



# Tema 6: La Capa de Aplicación

---

1. Introducción
2. DNS- El sistema de nombres de dominio
3. El correo electrónico
4. Word Wide Web
5. Audio y Video
6. Bibliografía



## ***2. DNS-El sistema de Nombres del dominio***

---

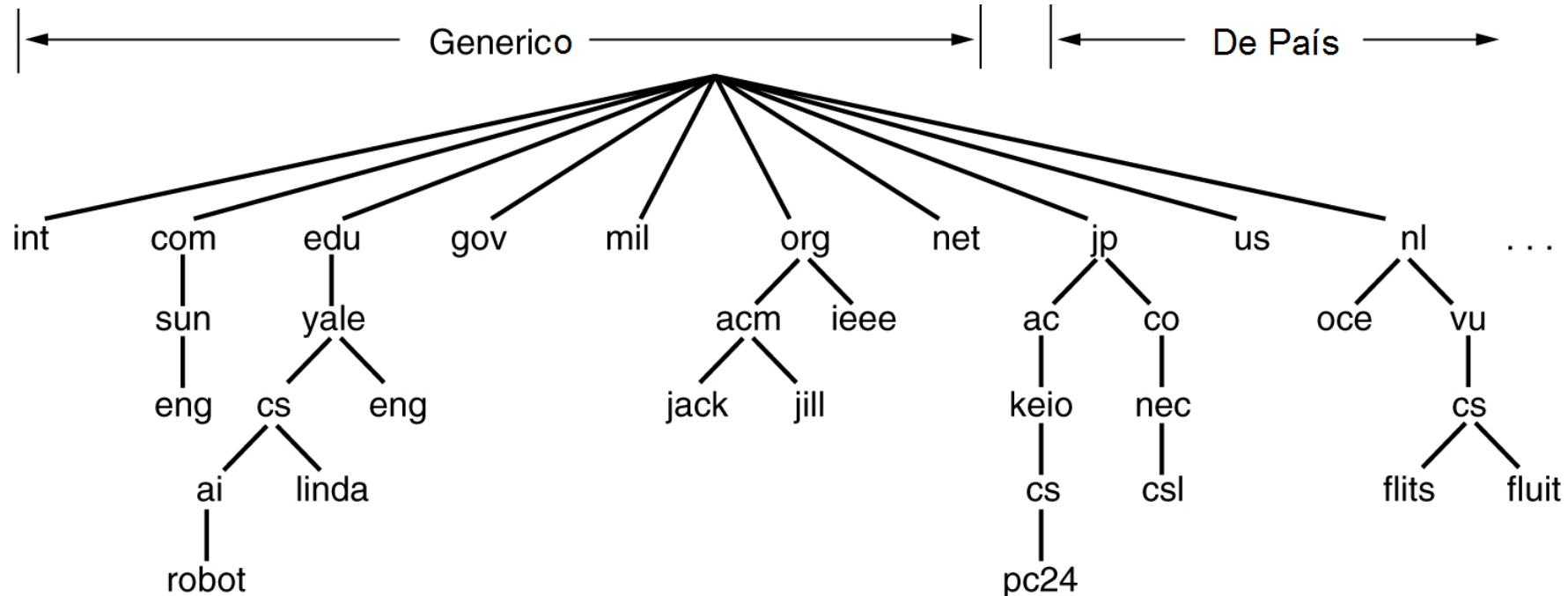
2.1 Introducción

2.2 Registro de recursos

2.3 Servidores de Nombre

- DNS se introduce para facilitar a las personas recordar una dirección IP.
- Enviar correo electrónico a [usuario@128.111.24.41](mailto:usuario@128.111.24.41) significa que si el ISP cambia de dirección IP, la dirección de correo debe cambiar.
- Debido a esto se introdujeron los nombres ASCII, para representar las direcciones IP, sin embargo la red sólo entiende direcciones IP, por lo que se requieren algunos mecanismos para la traducción.
- En los tiempos de ARPANET, sólo había un archivo, “hosts.txt”, en el que se listaban todos los hosts y su correspondencia.

- Cuando creció la red, se inventó **DNS**.
- La esencia de **DNS** es la invención de un **esquema de nombres jerárquicos** basados en dominios y un sistema de base de datos distribuida para implementar este esquema de nombres
- La forma en que se utiliza **DNS** es la siguiente:
  - Para relacionar un nombre con una dirección IP, un programa de aplicación llama a un procedimiento de biblioteca llamado **resolvedor** y le pasa el nombre como parámetro (gesthostbyname).
  - El **resolvedor** envía un paquete UDP a un **servidor DNS** local, que busca el nombre y devuelve la dirección IP al resolvedor, que entonces lo devuelve al solicitante.
  - Una vez que el programa tiene la dirección IP, el programa puede establecer una conexión TCP con el destino o enviarle paquetes UDP.



### Parte del espacio de nombres de dominio de Internet

Conceptualmente **Internet se divide en 200 dominios de nivel superior**, cada uno de los cuales abarca muchos hosts. Cada dominio se divide en subdominios y así sucesivamente. Los dominios de nivel superior se dividen en dos categorías: **genéricos** (com, edu, gov, mil, net, org.) y los **de país** que incluyen una entrada para cada país (es, us, jp, cu, nl, etc)

- En noviembre de 2000, ICANN aprobó cuatro nuevos dominios de nivel superior: biz (negocios), info (información), name (nombres de persona) y pro (profesionales como doctores y abogados)
- Cada dominio se nombra por la ruta hacia arriba desde él a la raíz. Los componentes se separan con puntos. Por ejemplo eng.sun.com.
- Los nombres de dominio no hacen distinción entre mayúsculas y minúsculas, para él significan lo mismo.
- Los nombres de componentes pueden ser de hasta 63 caracteres y la ruta completa de hasta 255 caracteres
- Para crear un nuevo dominio, se requiere el permiso del dominio en que se incluirá.
- Los nombres representan límites organizacionales, no las redes físicas.

- Cada dominio sea un host individual o un dominio superior, puede tener un grupo de registros de recursos asociados a él  
En un host individual, el registro de recursos más común es su dirección IP.
- Cuando un resovedor da un nombre de dominio al DNS, lo que recibe son los registros de recursos asociados a ese nombre. Por tanto, la función real del DNS es relacionar los dominios de nombres con los registros de recursos
- Un registro de recursos tiene 5 tuplas:

Nombre\_dominio | Tiempo\_vida | Clase | Tipo | Valor

- Nombre\_dominio: indica el dominio al que pertenece el registro
- Tiempo\_vida: indica la estabilidad del registro
- Clase: para Internet es siempre IN

Tipo	Significado	Valor
SOA	Inicio de autoridad	Parámetros para esta zona
A	Dirección IP de un host	Entero de 32 bits
MX	Intercambio de correo	Prioridad, dominio dispuesto a aceptar correo electrónico
NS	Servidor de nombres	Nombre de un servidor para este dominio
CNAME	Nombre canónico (alias)	Nombre de dominio
PTR	Apuntador	Alias de una dirección IP
HINFO	Descripción del host	CPU y SO en ASCII
TXT	Texto	Texto ASCII no interpretado

### Principales tipos de registros de recursos DNS

- El tipo de registro más importante es el A que contiene una dirección IP de 32 bits del host. Algunos hosts pueden tener más de una IP
- MX: indica el nombre de dominio para aceptar correo electrónico
- CNAME: permite la creación de Alias, PTR: ping
- HINFO: tipo de máquina y sistema operativo
- TXT: más información sobre el dominio no obligatoria

```
; Authoritative data for cs.vu.nl
cs.vu.nl.      86400  IN  SOA   star boss (952771,7200,7200,2419200,86400)
cs.vu.nl.      86400  IN  TXT   "Divisie Wiskunde en Informatica."
cs.vu.nl.      86400  IN  TXT   "Vrije Universiteit Amsterdam."
cs.vu.nl.      86400  IN  MX    1 zephyr.cs.vu.nl.
cs.vu.nl.      86400  IN  MX    2 top.cs.vu.nl.

flits.cs.vu.nl. 86400  IN  HINFO Sun Unix
flits.cs.vu.nl. 86400  IN  A    130.37.16.112
flits.cs.vu.nl. 86400  IN  A    192.31.231.165
flits.cs.vu.nl. 86400  IN  MX   1 flits.cs.vu.nl.
flits.cs.vu.nl. 86400  IN  MX   2 zephyr.cs.vu.nl.
flits.cs.vu.nl. 86400  IN  MX   3 top.cs.vu.nl.
www.cs.vu.nl.   86400  IN  CNAME star.cs.vu.nl
ftp.cs.vu.nl.   86400  IN  CNAME zephyr.cs.vu.nl

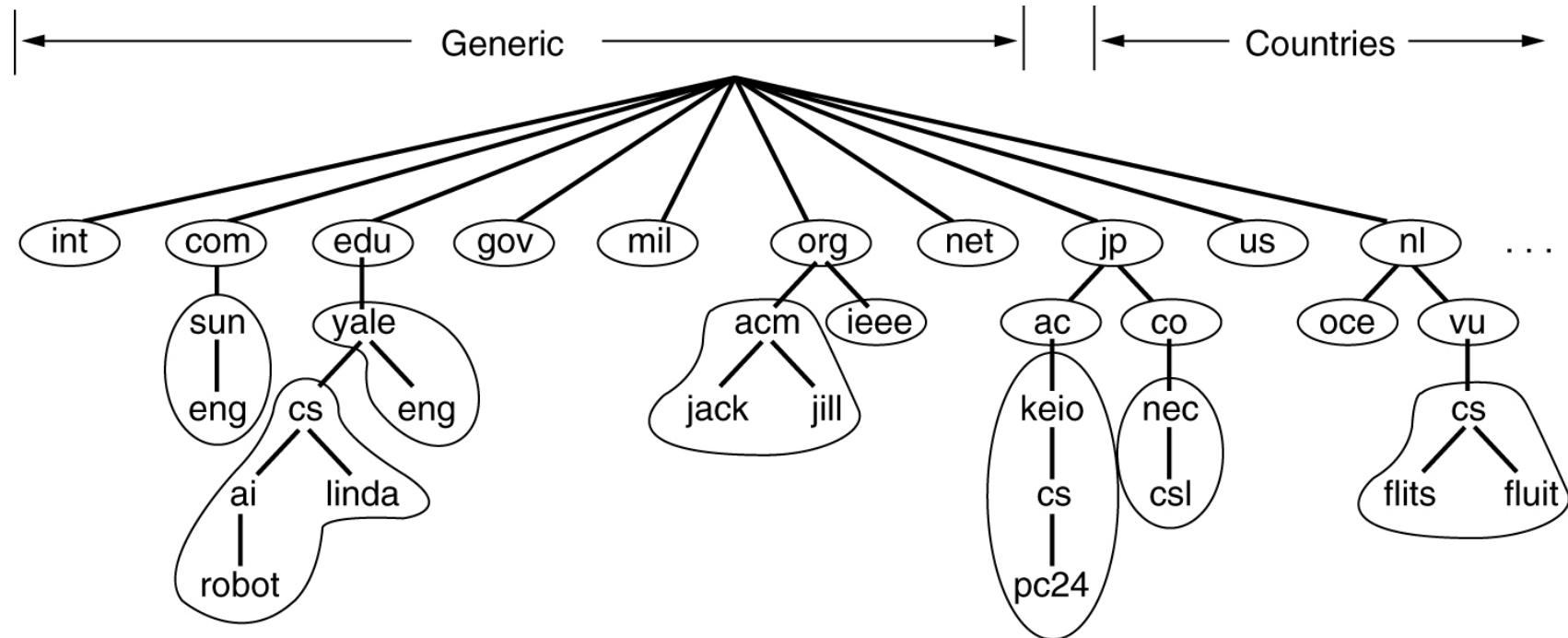
rowboat          IN  A    130.37.56.201
                  IN  MX   1 rowboat
                  IN  MX   2 zephyr
                  IN  HINFO Sun Unix

little-sister    IN  A    130.37.62.23
                  IN  HINFO Mac MacOS

laserjet         IN  A    192.31.231.216
                  IN  HINFO "HP Laserjet IISi" Proprietary
```

Parte de una posible base de datos DNS para *cs.vu.nl.*

- Lo que no se muestra en el archivo de ejemplo anterior son las IP a usar para buscar los dominios de nivel superior. Tales direcciones son suministradas por los servidores raíz, cuyas direcciones IP están presentes en un archivo de configuración del sistema y se cargan en la caché del DNS cuando se arranca.
- Hay aproximadamente **una docena de servidores raíces en todo el mundo** y cada uno conoce las direcciones IP de todos los servidores de dominio de nivel superior.
- Por tanto, si una máquina conoce la dirección IP de por lo menos un servidor raíz, puede buscar cualquier nombre DNS.

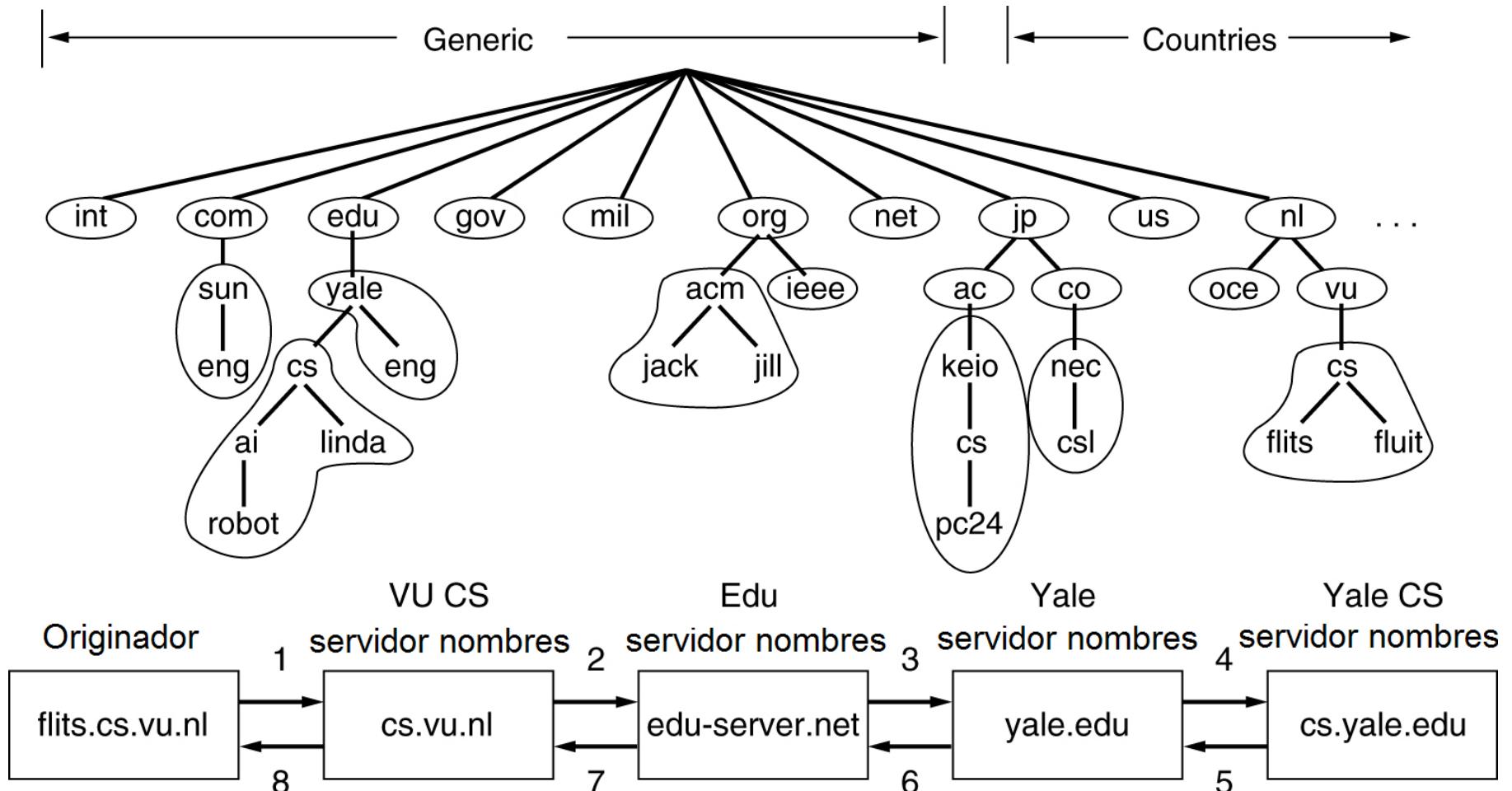


**Parte del espacio de nombres DNS, donde se muestra la división de zonas**

- Para evitar los problemas asociados a tener una sola fuente de información, el **espacio de nombres DNS** se divide en **zonas**
- Cada zona tiene dominios que no son compartidos por otras zonas.

# La capa de Aplicación

## 2.3 Servidores de Nombres



**Manera en que un resovedor busca un nombre remoto en 8 pasos**

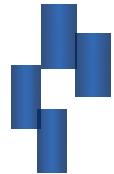
- Cada zona contiene una parte del árbol, y también contiene servidores de nombres que tienen la información de autorización correspondiente a esa zona. Si el dominio que se busca cae bajo la jurisdicción de esa zona se devuelve un **registro autorizado**, sino el servidor de nombres envía un mensaje de consulta al servidor de nombres de nivel superior.
- Este proceso se conoce como protocolo de consulta recursiva → existe también la posibilidad de si el servidor no conoce el dominio, enviar la dirección del siguiente servidor que se debe consultar.
- Cuando un cliente DNS no recibe una respuesta y salta los temporizadores, se manda a otro servidor → se consideran servidores inactivos antes de pérdidas de los paquetes.
- LDAP (protocolo ligero de acceso al directorio), DNS lo que hace es relacionar nombres simbólicos de máquinas con sus direcciones IP. LDAP ayuda a localizar personas, recursos y servicios, permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.  
16



# Tema 6: La Capa de Aplicación

---

1. Introducción
2. DNS- El sistema de nombres de dominio
3. El correo electrónico
4. Word Wide Web
5. Audio y Video
6. Bibliografía



## 3. Correo Electrónico

---

3.1 Introducción

3.2 Arquitectura y Servicios

3.3 El agente usuario

3.4 Formatos del mensaje

3.5 MIME: Extensiones Multipropósito de Correo

3.6 Internet

3.7 Transferencia de mensajes

3.8 Entrega Final

- El correo electrónico o e-mail lleva un largo trayecto. En la década del 90 se dio a conocer al gran público
- Los primeros sistemas de correo electrónico simplemente consistían en protocolos de transferencias de archivos, con la convención de que la primera línea del archivo contenía la dirección del destinatario
- Se utilizaba mucho la simbología ASCII

Smiley	Meaning	Smiley	Meaning	Smiley	Meaning
:)	I'm happy	= :-)	Abe Lincoln	:+)	Big nose
:(	I'm sad/angry	=):-)	Uncle Sam	:))	Double chin
:	I'm apathetic	*<:-)	Santa Claus	:-{)	Mustache
;-)	I'm winking	<:-()	Dunce	#:-)	Matted hair
:-O	I'm yelling	(-:	Australian	8-)	Wears glasses
:-(*)	I'm vomiting	:-)X	Man with bowtie	C:-)	Large brain

**Algunos emoticonos ASCII.**

Estos primeros sistemas de correo tenían varios inconvenientes:

- El envío de un mensaje a un grupo de personas era laborioso
- Los mensajes no tenían estructura interna
- El remitente nunca sabía si su mensaje había llegado o no
- La interfaz de usuario muy pobre
- No era posible crear y enviar mensajes que contuvieran mezcla de texto, dibujos, y voz

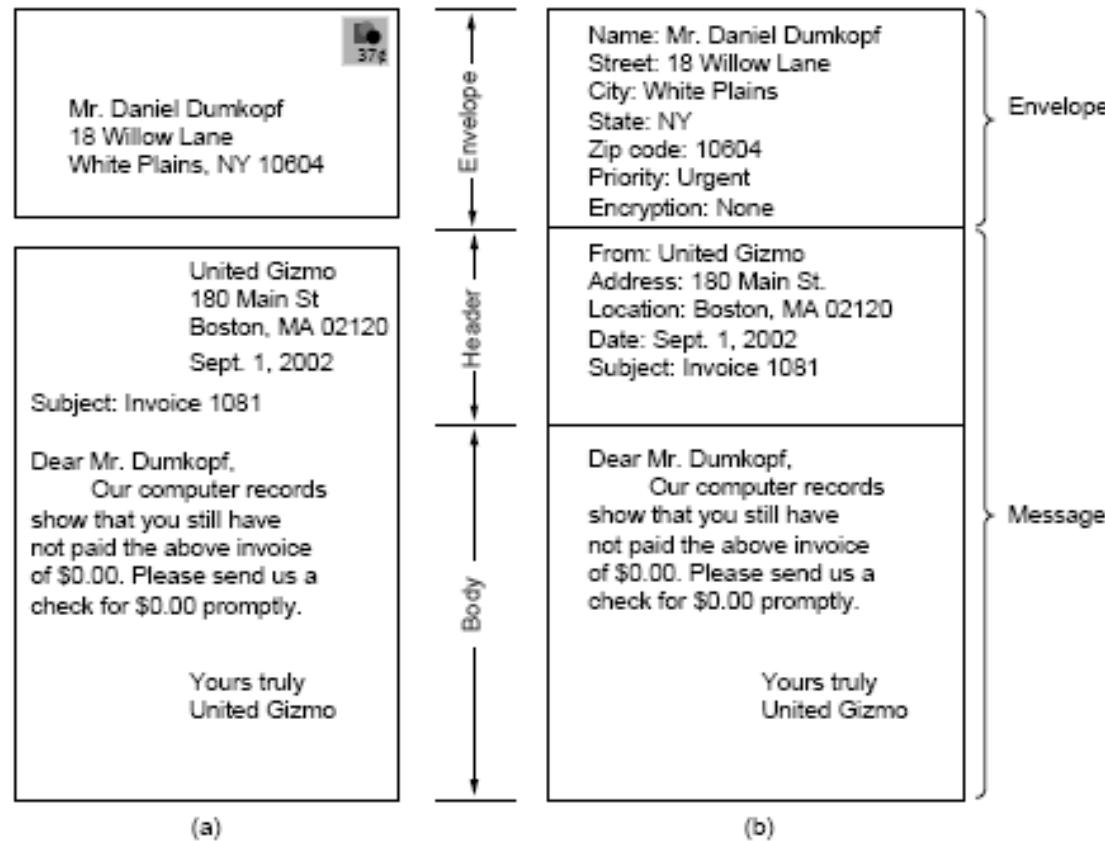
En 1982, se publicaron [RFC 821](#) (protocolo de transmisión) y [RFC 822](#) (formato de mensaje) en los cuales se basan los actuales sistemas de correo

- Los sistemas de correo electrónico normalmente se clasifican en dos subsistemas:
  1. Los agentes de usuario: permiten leer y enviar correo electrónico
  2. Los agentes de transferencia de mensajes: mueven los mensajes del origen al destino

Los sistemas de correo electrónico desempeñan **cinco funciones básicas**

- **Redacción:** se refiere al proceso de crear mensajes y respuestas.
- **Transferencia:** se refiere a mover mensajes del remitente al destinatario, requiere establecer una conexión con el destino o alguna máquina intermedia, enviar el mensaje y liberar.
- **Generación del informe:** tiene que ver con indicar al remitente que ocurrió con el mensaje.
- **Visualización:** de los mensajes.
- **Disposición:** tiene que ver con lo que el destinatario hace con el mensaje una vez que lo recibe.

- Además de estos servicios básicos, los sistemas de correo proporcionan más servicios avanzados
  - Reenvío de correo automático.
  - Buzones de correo.
  - Lista de correos.
  - Copias ocultas.
  - Correo electrónico de alta prioridad.
  - Correo encriptado.
- Debe establecerse división entre sobre y contenido. El contenido se divide en encabezado y cuerpo.



- Un **agente de usuario** normalmente es un programa (a veces llamado lector de correo) que acepta una variedad de comandos para redactar, recibir y contestar los mensajes, así como para manipular los buzones de correo
- Algunos agentes de usuario tienen una interfaz elegante operada por menús o por iconos que requieren ratón y otros esperan comandos desde el teclado.
- Las tareas llevadas a cabo son:
  - Envío de correo electrónico
  - Lectura de correo electrónico

- **Envío de correo electrónico**
  - Para enviar un mensaje de correo electrónico se debe proporcionar la dirección de destino (además de otras parámetros de configuración).
  - La dirección de destino debe estar en un formato manejado por el agente de usuario: direcciones DNS, direcciones X.400, listas de correos, alias, ...
  - El mensaje puede redactarse en cualquier editor de textos o por uno incorporado en el agente de usuario

- **Lectura de correo electrónico**

- Al iniciar un agente de usuario, se encarga de buscar en el buzón de usuario el correo recibido.
- Muestra el correo recibido como la cantidad recibida o indicando una línea de cada uno de los correos especificando información obtenida de las cabeceras de los correos.
- Mostrados los mensajes, el usuario podrá desplegar o eliminar el/los mensaje/s.

Encabezado	Significado
To:	Direcciones de correo de los destinatarios primarios
Cc:	Direcciones de correo de los destinatarios secundarios
Bcc:	Direcciones de correo para las copias ocultas
From:	Persona o personas que crearon el mensaje
Sender:	Direcciones de correo del remitente
Received:	Línea agregada por cada agente de transferencia en la ruta
Return-Path	Puede usarse para identificar una ruta de regreso al remitente

En términos de entrega no hay diferencia entre los destinatarios primarios y secundarios. Es una diferencia por entero psicológica. Cc viene de copia al carbón. El campo Bcc es como el Cc, excepto que esta línea se borra de todas las copias enviadas a los destinatarios primarios y secundarios<sup>28</sup>

Encabezado	Significado
Date:	Fecha y Hora de envío del mensaje
Reply-To:	Dirección de correo a la que deben enviarse las contestaciones
Message-Id:	Número único para referencia posterior a este mensaje
References:	Otros identificadores del mensaje pertinente
Keywords:	Claves seleccionadas por el usuario
Subject:	Resumen corto del mensaje para desplegar en una línea

Algunos campos usados en el encabezado de mensaje RFC 822

RFC 822 indica explícitamente que los usuarios pueden inventar encabezados nuevos para uso privado, siempre y cuando comiencen con una X-, se garantiza que en el futuro no habrá encabezados que comiencen con X-

### Problemas con lenguajes internacionales:

- Mensajes en idiomas con acento  
(Ej. Español, Francés, Alemán).
- Mensajes en alfabetos no latinos  
(Ej. Hebreo y ruso).
- Mensajes en idiomas sin alfabetos  
(Ej. Chino y japonés).
- Mensajes que no contienen texto  
(Ej. Audio e imágenes).

- Se propuso una solución en el RFC 1341 y se actualizó en RFCs 2045-2049. Esta solución se le llamó **MIME** y se usa ampliamente.
- La **idea básica** de **MIME** es continuar usando el formato RFC 822, pero **agregar una estructura al cuerpo del mensaje** y **definir reglas de codificación** para los mensajes no ASCII.
- **MIME define cinco nuevos encabezados de mensaje.**

Encabezado	Significado
MIME-Version:	Identifica la versión de MIME
Content-Description:	Cadena de texto que describe el contenido
Content-Id	Identificador único
Content-Transfer-Encoding	Cómo se codifica el mensaje para su transmisión
Content-Type	Naturaleza del mensaje

- **Content-Transfer-Encoding:** indica la manera en que se codifica el mensaje para su transmisión a través de una red donde se podrían tener problemas con la mayoría de los caracteres distintos de letras, números y signos de puntuación
  1. **Texto ASCII:** los caracteres usan 7 bits
  2. **Texto ASCII:** los caracteres usan 8 bits
  3. **Codificación base64:** para codificar mensajes binarios, se dividen grupos de 24 bits en unidades de 6 bits y cada unidad se envía como un carácter ASCII legal. “A” para 0, “B” para 1, ..., y por último + y / para el 62, 63 respectivamente
  4. **Codificación entrecomillada imprimible:** mensajes casi completamente ASCII, se usa ASCII de 7 bits, con todos los caracteres por encima de 127, con un =valorhexdec

- **Content-Type:** especifica la naturaleza del cuerpo del mensaje. En el RFC 2045 hay siete tipos definidos, cada uno de los cuales tiene uno o más subtipos:

Tipo	SubTipo	Descripción
Texto	Plano	Texto sin formato
	Enriquecido	Texto con comandos de formato sencillos
Imagen	Gif	Imagen fija en formato GIF
	Jpeg	Imagen fija en formato JPG
Audio	Básico	Sonido
Video	MPEG	Película en formato MPEG
Aplicación	Octect-stream	Secuencia de bytes no interpretada
	PostScript	Documento imprimible en postScript
Mensaje	RFC 822	Mensaje MIME RFC 822
	Parcial	Mensaje dividido para su transmisión
	Externo	El mensaje mismo debe obtenerse en la red
Multipartes	Mezclado	Partes independientes en el orden especificado
	Alternativa	Mismo mensaje en diferentes formatos
	Paralelo	Las partes deben verse en forma simultánea
	Compendio	Cada parte es un mensaje RFC 822 completo

- **Tipo Texto:**
  - text: para texto normal
  - text/enriched: texto enriquecido, forma para indicar negritas, cursivas, tamaños, sangrías, justificaciones y esta basado en SGML, que es el lenguaje estándar genérico de marcado de etiquetas y que se usa en HTML. Ej. Ha llegado el <bold> momento </bold> dijo José
  - text/html (RFC 2854): para permitir que las páginas HTML, se enviaran en un correo RFC 822
- **Tipo Imagen:** se usan muchos formatos para almacenar y transmitir imágenes fijas, tanto con compresión como sin ella.
- **Tipos audio y video:** son para sonido e imágenes en movimiento respectivamente. El primer formato de video definido fue el diseñado por el modesto grupo MPEG (Grupo de expertos en imágenes en movimiento). Además de audio/basic, se agregó audio/mpeg para permitir enviar archivos mp3

- Tipo Aplicación:
  - Es un tipo general para los formatos que requieren procesamiento externo no cubierto por ninguno de los otros tipos. **Octet-stream** es simplemente una secuencia de bytes no interpretados.
  - **Tipo mensaje:** permite que un mensaje esté encapsulado por completo dentro de otro, por ejemplo para el reenvío de mensajes o **externo** que debe traerse de la red
  - **Tipo multiparte:** permite que un mensaje contenga más de una parte con el comienzo y fin de cada parte claramente delimitados
    - El subtipo mezclado permite que cada parte sea diferente, sin ninguna estructura adicional impuesta. La mayoría de los programas de correo actuales permiten que el usuario agregue uno o más archivos adjuntos a un mensaje de texto. **Estos archivos adjuntos se envían mediante el tipo multiparte**

# MIME

From: elinor@abcd.com  
To: carolyn@xyz.com  
MIME-Version: 1.0  
Message-Id: <0704760941.AA00747@abcd.com>  
Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm  
Subject: Earth orbits sun integral number of times

This is the preamble. The user agent ignores it. Have a nice day.

--qwertyuiopasdfghjklzxcvbnm  
Content-Type: text/enriched

Happy birthday to you  
Happy birthday to you  
Happy birthday dear <b>Carolyn</b>  
Happy birthday to you

--qwertyuiopasdfghjklzxcvbnm  
Content-Type: message/external-body;  
access-type="anon-ftp";  
site="bicycle.abcd.com";  
directory="pub";  
name="birthday.snd"

content-type: audio/basic  
content-transfer-encoding: base64  
--qwertyuiopasdfghjklzxcvbnm--

Mensaje multiparte que contiene alternativas de texto enriquecido y  
audio. Felicitación de cumpleaños como texto y como canción

- **El agente de transferencia de mensajes** se ocupa de transmitir del remitente al destinatario. La manera más sencilla de hacer esto es establecer una conexión de transporte de la máquina de origen a la de destino y transferir el mensaje.
- En Internet el correo electrónico se entrega al hacer que la máquina origen establezca una conexión TCP con el **puerto 25** de la máquina destino. Escuchando en este puerto está un demonio de correo electrónico que habla con el SMTP (Protocolo Simple de Transporte de Correo).
- Este demonio acepta conexiones de entrada y copia mensajes de ellas a los buzones adecuados.
- SMTP es un protocolo ASCII sencillo, después de establecer la conexión TCP con el puerto 25, **el servidor comienza enviando una línea de texto que proporciona identidad** e indica si está preparado o no para recibir correo.

# Transferencia de mensajes

Transfiriendo un mensaje de  
[elinore@abc.com](mailto:elinore@abc.com) a  
[carolyn@xyz.com](mailto:carolyn@xyz.com).

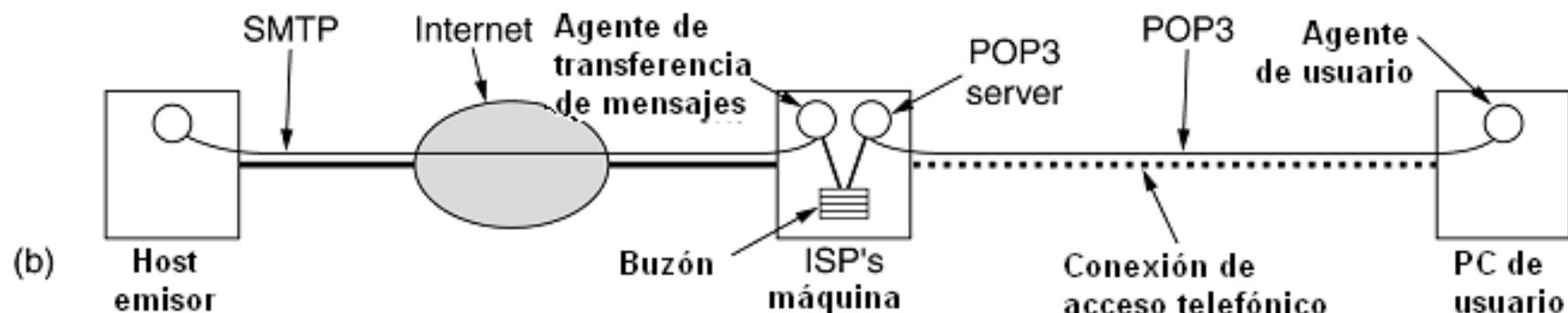
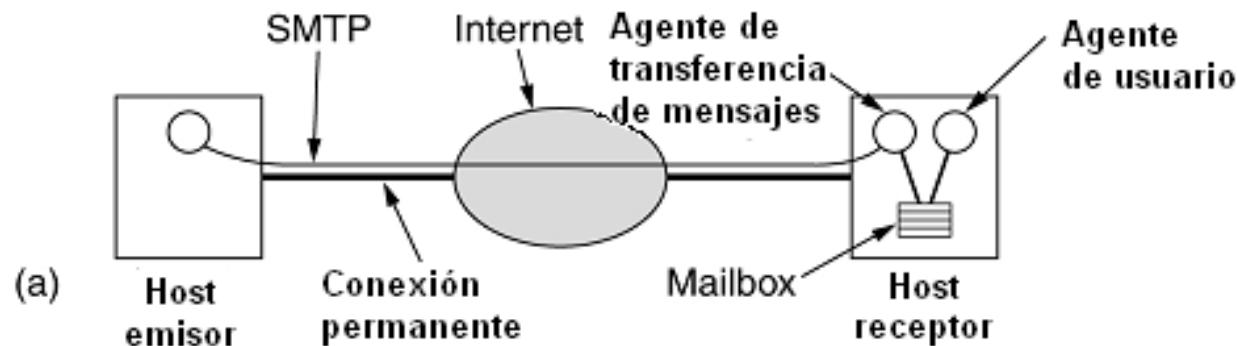
Ejemplo en una máquina UNIX  
telnet mail.isp.com 25

```
S: 220 xyz.com SMTP service ready
C: HELO abcd.com
    S: 250 xyz.com says hello to abcd.com
C: MAIL FROM: <elinor@abcd.com>
    S: 250 sender ok
C: RCPT TO: <carolyn@xyz.com>
    S: 250 recipient ok
C: DATA
    S: 354 Send mail; end with "." on a line by itself
C: From: elinor@abcd.com
C: To: carolyn@xyz.com
C: MIME-Version: 1.0
C: Message-Id: <0704760941.AA00747@abcd.com>
C: Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
C: Subject: Earth orbits sun integral number of times
C:
C: This is the preamble. The user agent ignores it. Have a nice day.
C:
C: --qwertyuiopasdfghjklzxcvbnm
C: Content-Type: text/enriched
C:
C: Happy birthday to you
C: Happy birthday to you
C: Happy birthday dear <bold> Carolyn </bold>
C: Happy birthday to you
C:
C: --qwertyuiopasdfghjklzxcvbnm
C: Content-Type: message/external-body;
C: access-type="anon-ftp";
C: site="bicycle.abcd.com";
C: directory="pub";
C: name="birthday.snd"
C:
C: content-type: audio/basic
C: content-transfer-encoding: base64
C: --qwertyuiopasdfghjklzxcvbnm
C: .
    S: 250 message accepted
C: QUIT
    S: 221 xyz.com closing connection
```

- Hasta ahora hemos supuesto que todos los usuarios trabajan en máquinas capaces de enviar y recibir correo electrónico. Este modelo funcionó bien durante décadas cuando todos los hosts de ARPANET estaban en línea
- Con el advenimiento de los ISP, el modelo dejó de usarse. El problema es el siguiente: ¿qué sucede en el ejemplo anterior si carolina no está en línea en ese momento?. Como no se puede establecer una conexión TCP, no se puede ejecutar el protocolo SMTP
- La solución es que un agente de transferencia de mensajes en una máquina ISP, acepte correo electrónico para sus clientes y lo almacene en sus buzones en la máquina ISP. Como el agente puede estar en línea todo el tiempo, el correo electrónico puede enviarse las 24 horas del día
- Para esto se creó el **Protocolo de Oficina de Correos Versión 3 (POP3)**,

# La capa de Aplicación

## 3.7 Entrega Final



- (a) Envío y recepción de correo cuando el receptor tiene una conexión permanente a Internet y el agente de usuario se ejecuta en la misma máquina que el agente de transferencia de mensajes. (b) Lectura de correo cuando el receptor tiene una conexión de acceso telefónico a una ISP

```
S: +OK POP3 server ready
C: USER carolyn
    S: +OK
C: PASS vegetables
    S: +OK login successful
C: LIST
    S: 1 2505
    S: 2 14302
    S: 3 8122
    S: .
C: RETR 1
    S: (sends message 1)
C: DELE 1
C: RETR 2
    S: (sends message 2)
C: DELE 2
C: RETR 3
    S: (sends message 3)
C: DELE 3
C: QUIT
    S: +OK POP3 server disconnecting
```

Uso de POP3 para obtener tres mensajes

- Muchas personas tienen una sola cuenta de correo a la cual quieren acceder desde varios sitios: trabajo, casa, cibercafé. Aunque POP3 permite esto debido a que descarga todos los mensajes de correo en cada contacto, el resultado es que los mensajes de correo quedan esparcidos en múltiples máquinas
- Esta desventaja dio lugar a un protocolo alternativo, IMAP que se define en RFC 2060
- A diferencia de POP3 que asume que el usuario vaciará el buzón en cada contacto y trabajará sin conexión, IMAP supone que todo el correo electrónico permanecerá en el servidor de manera indefinida en múltiples buzones de correo
- IMAP proporciona mecanismos para crear, destruir y manipular múltiples buzones de voz en el servidor y el usuario puede mantener un buzón para cada uno de sus contactos y colocar ahí mensajes de la bandeja de entrada una vez leídos
- El servidor IMAP escucha en el puerto 143. No todos los ISP, ni todos los programas de correo soportan IMAP.

# La capa de Aplicación

## 3.7 Entrega Final – IMAP

Característica	POP3	IMAP
En dónde se define el protocolo	RFC 1939	RFC 2060
Puerto TCP utilizado	110	143
En dónde se almacena el correo electrónico	PC del usuario	Servidor
En dónde se lee el correo electrónico	Sin conexión	En línea
Tiempo de conexión requerido	Poco	Mucho
Uso de recursos del servidor	Mínimo	Amplio
Múltiples buzones	No	Sí
Quién respalda los buzones	Usuario	ISP
Bueno para los usuarios móviles	No	Sí
Control del usuario sobre la descarga	Poco	Mucho
Descargas parciales de mensajes	No	Sí
¿Es un problema el espacio en disco?	No	Con el tiempo
Sencillo de implementar	Sí	No
Soporte amplio	Sí	Creciendo

43

Una comparación entre POP3 e IMAP

- Independientemente de si se usa POP3 o IMAP, muchos sistemas permiten establecer filtros a los correos electrónicos.
- Estos filtros son reglas que se verifican cuando llega el correo electrónico o cuando se inicia el agente de usuario.
- Cada regla especifica una condición y una acción.
- Algunos ISP proporcionan un **filtro** que clasifica de manera automática el correo electrónico como importante o como publicidad no deseada (correo basura) y almacena cada mensaje en el buzón correspondiente.
- Los filtros funcionan verificando primero si el origen es una **spammer** (persona o entidad que envía correo basura) conocido. A continuación examinan la línea de asunto. Si **cientos de usuarios** han recibido un mensaje **con la misma línea de asunto**, probablemente sea correo basura.

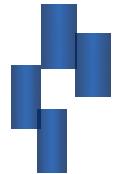
- **Correo Web:** Algunos sitios Web como Yahoo, Hotmail, etc, proporcionan servicio de correo, para esto tienen a agentes de transferencias normales que escuchan el **puerto 25 para conexiones SMTP entrantes**. La salida es formateada como página Web en HTML y el buzón reside en el servidor.



# Tema 6: La Capa de Aplicación

---

1. Introducción
2. DNS- El sistema de nombres de dominio
3. El correo electrónico
4. Word Wide Web
5. Audio y Video
6. Bibliografía



## 4. *Word Wide Web*

---

4.1 Introducción

4.2 Panorama de la arquitectura

4.3 URLs: Localizadores Uniformes de Recursos

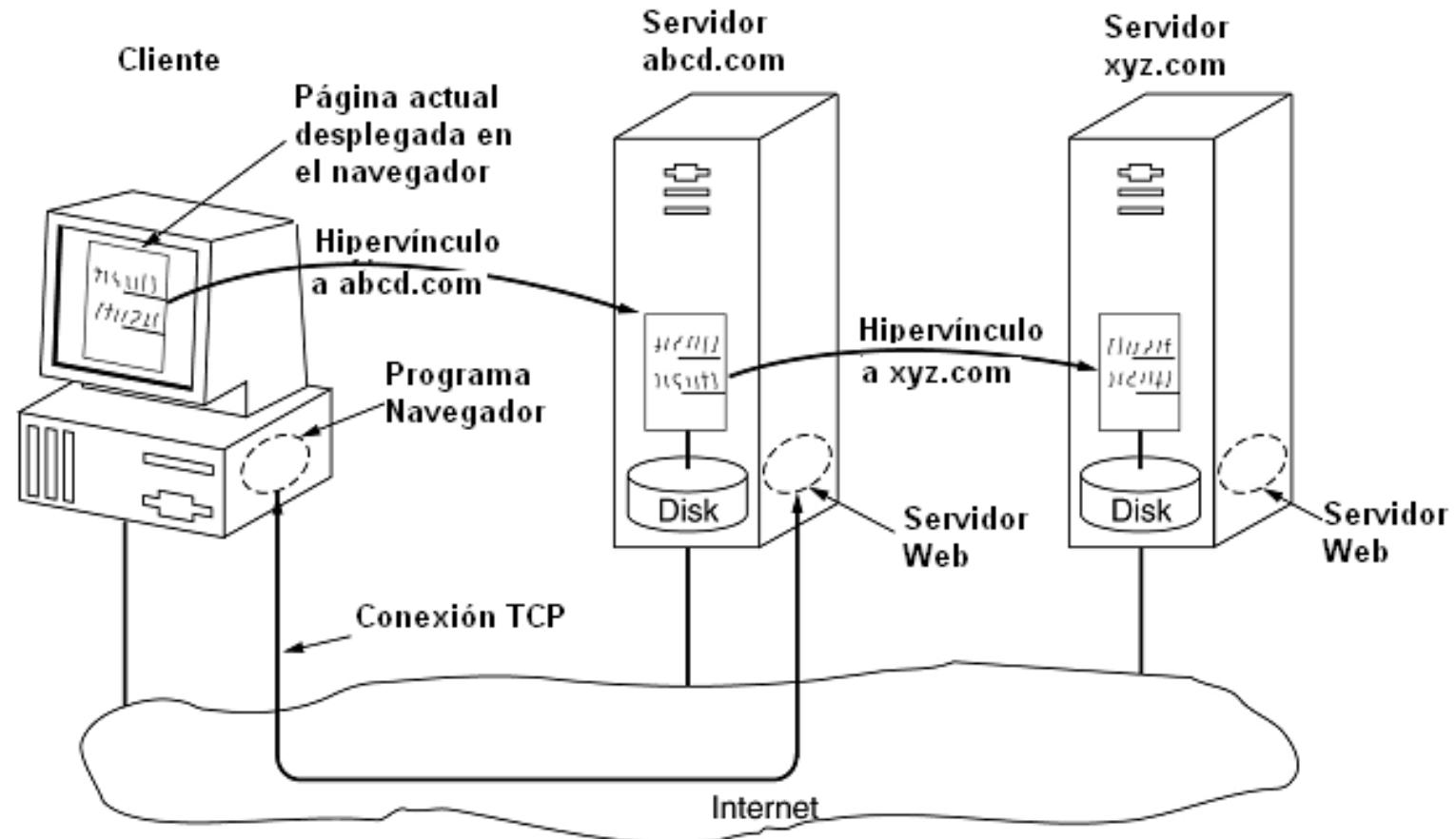
4.4 Documentos Web Estáticos

4.5 Documentos Web Dinámicos

4.6 Protocolo HTTP

- World Wide Web (WWW) es un armazón arquitectónico para acceder a documentos vinculados distribuidos en miles de máquinas de toda Internet.
- Web comenzó en 1989 en el CERN, Centro Europeo de Investigación Nuclear, con grandes grupos de investigadores de varios países colaborando en experimentos complejos.
- La propuesta inicial de una red de documentos vinculados surgió del físico del CERN Tim Berners-Lee en marzo de 1989. El primer prototipo (basado en texto). En Diciembre de 1991 se hizo una demostración pública en la conferencia Hypertext' 91 en San Antonio Texas.
- Posteriormente Marc Andreessen de la Universidad de Illinois comenzó a desarrollar el primer navegador gráfico Mosaic, que fue tan popular que hizo que su dueño montara su propia compañía, Netscape Communication Corp. cuya meta era desarrollar clientes, servidores y software para Web.

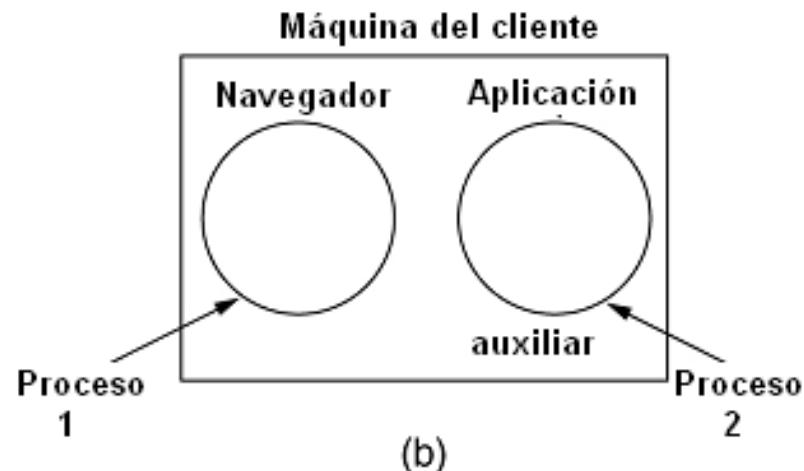
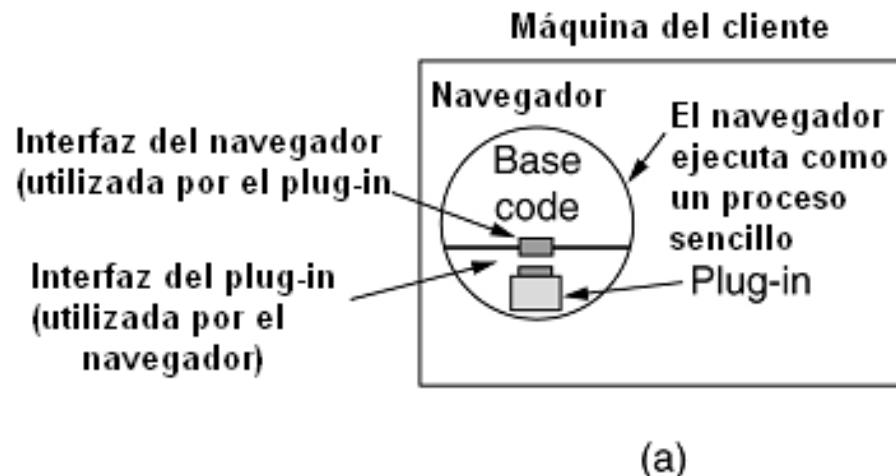
- Desde el punto del usuario, la Web consiste en un enorme conjunto de documentos a nivel mundial, llamados páginas Web. Cada página puede contener vínculos a otras páginas relacionadas en cualquier lugar del mundo.
- La idea de hacer que una página apunte a otra, lo que se conoce como hipertexto, fue inventada por un profesor visionario de ingeniería eléctrica del MIT, Vannevar Bush en 1945, mucho antes de que se inventara Internet.
- Un Navegador es un programa que puede desplegar una página Web y atrapar los clics que se hacen en los elementos presentes en la misma. Ej. IE, Netscape, Mozilla
- Cuando se selecciona un elemento que vincula a otra página (hipervínculo), el navegador obtiene la página solicitada, interpreta el texto y los comandos de formateo y la despliega, por tanto el hipervínculo necesita una manera de nombrar cada página en la Web. Las páginas se nombran utilizando URLs (Localizadores Uniformes de Recursos).



Las partes del modelo Web

- La URL tiene tres partes:
  1. Nombre del protocolo (http).
  2. DNS de la página donde se localiza la página web (www\_UCO.es).
  3. Nombre de archivo que contiene la página Web (index.html).
- Cuando se selecciona un hipervínculo (<http://www.itu.org/home/index.html>) se realiza una serie de pasos:
  1. El navegador determina el URL (enviando lo seleccionado).
  2. El navegador pide el DNS la dirección IP de www.itu.org.
  3. DNS responde con 156.106.192.32.
  4. El navegador realiza una conexión TCP con el puerto 80 de 156.106.192.32.
  5. Despues envía un mensaje en el que solicita el archivo /home/index.html.
  6. Se libera la conexión TCP.
  7. El navegador despliega todo el texto de /home/index.html.
  8. El navegador obtiene y despliega todas las imágenes del archivo.

- Los navegadores actuales tienen muchas facilidades para navegar, con botones funcionales para retroceder, avanzar, actualizar, buscar, etc.
- No todas las páginas contienen texto HTML estándar, pueden tener PDF, imágenes JPG, GIF, MP3 (Audio), MPEG (Video).
- Puesto que las páginas HTML estánrdar pueden incluir los contenidos anteriores, el navegador tiene un problema cuando encuentra una página que no puede interpretar.
- En lugar de agrandar cada vez más los navegadores incorporándoles intérpretes para una colección creciente de tipos de archivos, cuando un servidor devuelve una página también devuelve alguna información adicional acerca de ella. Dicha información incluye el tipo MIME de la página.
- Si el tipo MIME no es de los integrados, el navegador consulta su tabla de tipos MIME que le indica cómo desplegar la página.

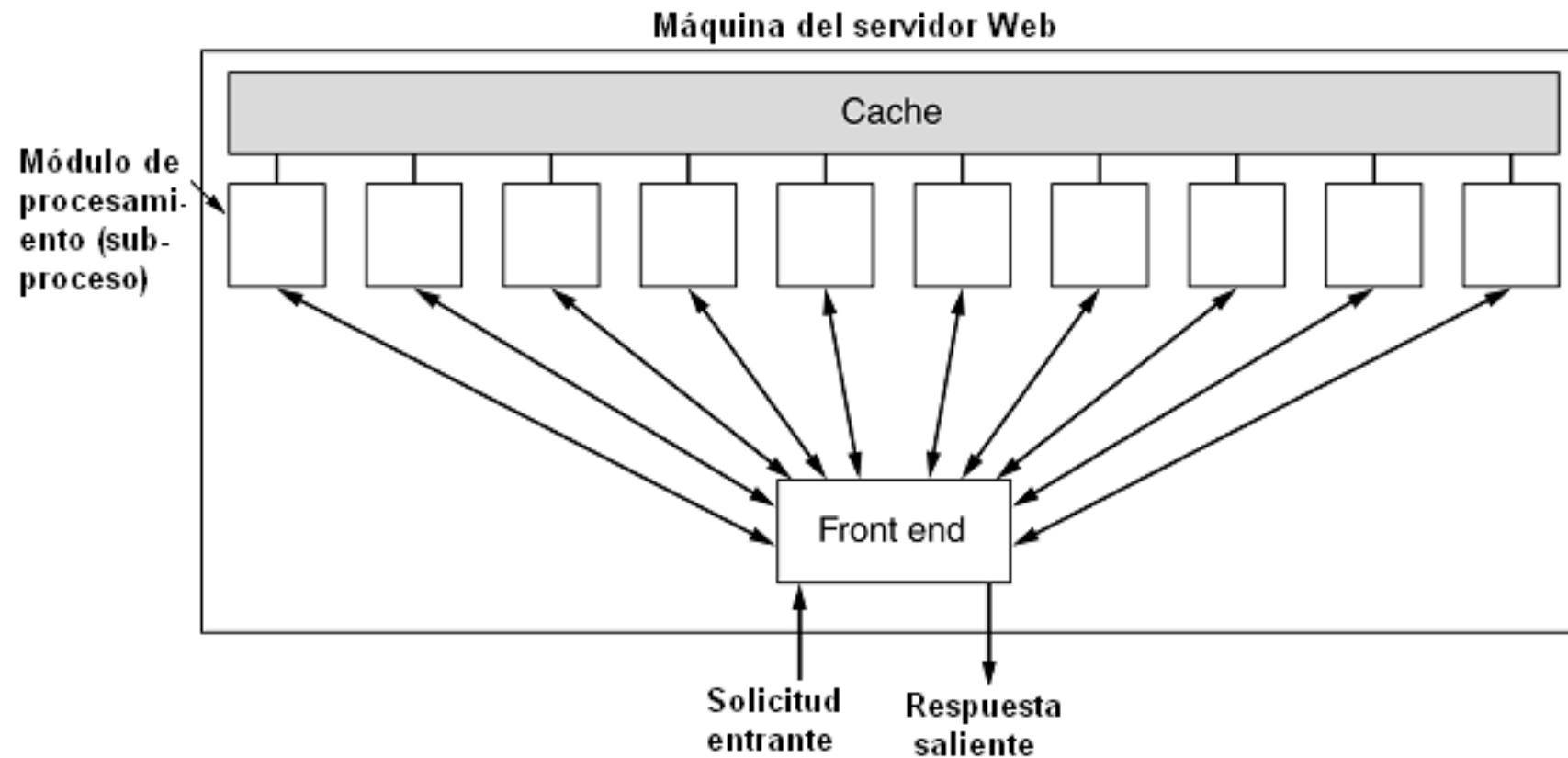


(a) Un plug-in de navegador. (b) Una aplicación auxiliar

- Hay dos posibilidades para interpretar:
  1. **Plug-in:** que se instala en un directorio especial del disco y lo instala como extensión del mismo. Ej. Plug-in de Flash.
  2. **Aplicación auxiliar:** es un programa completo que se ejecuta como un proceso independiente. No ofrece interfaz con el navegador y no utiliza los servicios de éste. Muchas aplicaciones utilizan el tipo MIME aplicación y se han definido varios subtipos: aplicación/pdf, aplicación/msword, aplicación/x-photoshop, audio/mp3.

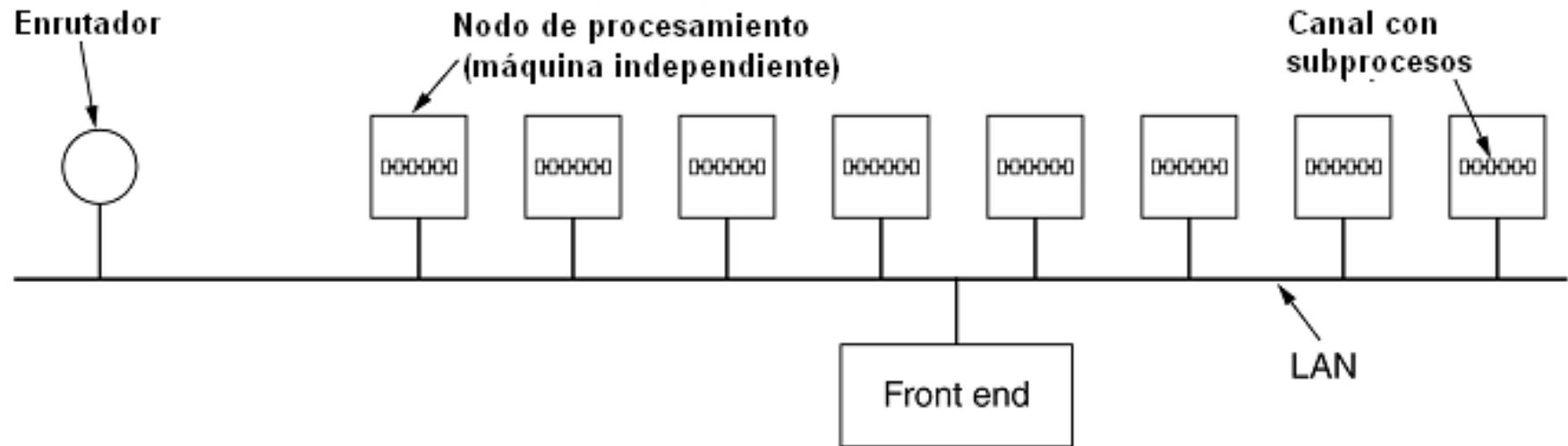
En Windows, cuando se instala un programa, la computadora registrará los tipo MIME que necesita manejar

- Cuando el usuario teclea una URL o hace clic en una línea de hipertexto, el navegador lo analiza e interpreta la parte entre http:// y la siguiente diagonal como un nombre DNS a buscar, identificando el servidor.
- Los pasos que da el servidor en su ciclo principal son:
  1. Acepta una conexión TCP de un cliente (un navegador)
  2. Obtiene el nombre del archivo solicitado
  3. Obtiene el archivo (del disco)
  4. Regresa el archivo al cliente
  5. Libera la conexión TCP
- Existen varios tipos de diseño para atender las conexiones:
  1. **Un disco, un proceso:** Un disco SCSI de alta calidad tiene un tiempo de acceso promedio de 5 mseg, lo que limita el servidor a 200 solicitudes/segundo o menos.
  2. **Mejora sobre anterior:** mantener una caché en la memoria de los archivos más recientemente utilizados .



Un servidor Web con múltiples subprocessos con un frontend y módulos de procesamiento

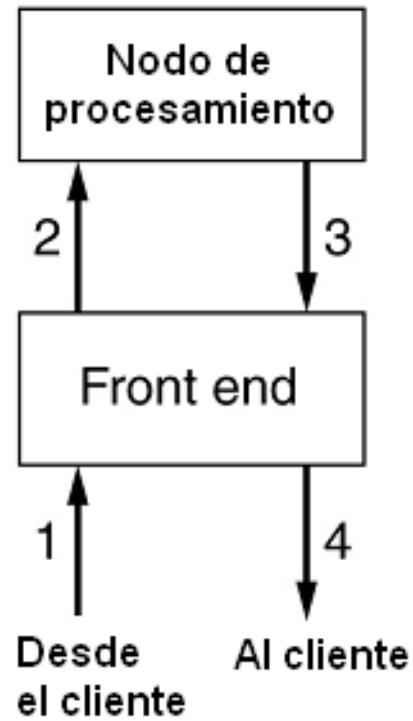
3. **Múltiples subprocessos y 1 disco:** El servidor consiste en un módulo de front end que acepta todas las solicitudes entrantes y k módulos de procesamiento. Mientras unos procesos están accediendo a disco, otros trabajan en otras solicitudes. **Con k módulos y k discos la velocidad puede ser k veces mayor**



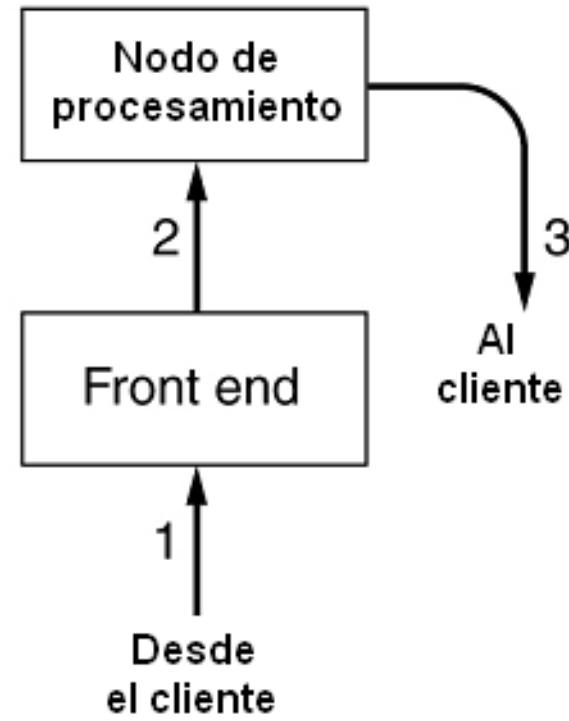
Granja de servidores

4. Si llegan demasiadas solicitudes cada segundo, la CPU no será capaz de manejar la carga de procesamiento, sin importar cuántos discos se pongan en paralelo. La solución es agregar más nodos (computadoras) con discos replicados (Granja de servidores). El front end aún acepta múltiples solicitudes entrantes, pero las distribuye en múltiples CPUs en lugar de múltiples subprocessos.

- El problema con la granja de servidores es que no hay caché compartido debido a que cada nodo de procesamiento tiene su propia memoria, a menos que se utilice un multiprocesador que son costosos.
- Una solución sería que el frontend registre el lugar a donde envía cada respuesta y que envíe las solicitudes de las páginas subsecuentes al mismo nodo.
- Otro problema con las granjas de servidores es que la conexión TCP del cliente pasan a través de front-end. Para solucionar este problema, algunas veces se utiliza un truco llamado **transferencia TCP (TCP handoff)**. Con ésta el punto final de la conexión TCP se pasa al nodo de procesamiento a fin de que pueda contestar directamente al cliente.



(a)



(b)

- (a) Secuencia normal de mensajes solicitud-respuesta.
- (b) Secuencia cuando se utiliza la **transferencia TCP**.

- Partes de una URL (Localizador Uniforme de Recursos).
  1. Protocolo
  2. DNS
  3. Nombre de la página
- Al seleccionarse el texto, el navegador busca el nombre del host usando DNS, ya con la IP del servidor establece una conexión TCP con el host y envía por esa conexión el nombre del archivo usando protocolo HTTP por el puerto 80.
- Este esquema de URL es abierto en el sentido de que es posible hacer que los navegadores utilicen múltiples protocolos para obtener diferentes recursos.

Nombre	Usado para	Ejemplo
http	Es el lenguaje nativo de la Web, el que hablan los servidores web	<a href="http://wwwuco.es">http://wwwuco.es</a>
ftp	Se usa para acceder a archivos mediante FTP, protocolo de transferencia de archivos de Internet. Descarga archivos	<a href="ftp://ftp.eru.es">ftp://ftp.eru.es</a>
File	Para acceder a un archivo de forma local	<a href="file:///usr/prog/p.c">file:///usr/prog/p.c</a>
News	30.000 grupo de noticias USENET, en los que se discuten una variedad de temas	<a href="news:comp.os.minix">news:comp.os.minix</a>
News	Puede utilizarse para llamar a un artículo de noticias como si fuera página web	<a href="news:AA0123456@cs.uta.edu">news:AA0123456@cs.uta.edu</a>
mailto	Permite a los usuarios enviar correo desde un navegador web, el navegador abre gestor de correo	<a href="mailto:i23mogab@uco.es">mailto:i23mogab@uco.es</a>
telnet	Establecer una conexión en línea con una máquina remota	<a href="telnet://www.w3.org:80">telnet://www.w3.org:80</a>

- En HTTP no existe el concepto de inicio de sesión. El navegador envía una solicitud a un servidor y obtiene un archivo.
- Para que los servidores puedan guardar información relativa a gustos, preferencias, patrones de navegación del usuario, personalización de sitios Web (los usuarios pueden establecer una página de inicio detallada que contenga sólo la información que desea).
- Para resolver este problema Netscape diseñó una técnica muy criticada llamada cookies. Las cookies se personalizaron en RFC 2109.
- Cuando un cliente solicita una página Web, el servidor puede proporcionar información adicional junto con la página solicitada. Esta información puede incluir una cookie que es un pequeño archivo (como mucho 4KB).
- Los navegadores almacenan las cookies en un directorio de cookies en el disco duro de la máquina del cliente, a menos que el usuario las deshabilite. Los cookies son simplemente archivos, no ejecutables.

### Campos de una cookie.

- **Dominio:** indica de dónde viene la cookie.
- **Ruta:** es la ruta en la estructura del directorio del servidor que identifican qué partes del árbol de archivos del servidor pueden utilizar la cookie.
- **Contenido:** toma la forma nombre=valor, es donde se almacena el contenido de la cookie. El servidor escribe lo que deseé.
- **Expira:** especifica cuando caduca la cookie. Si este campo está ausente. Para eliminar una cookie del disco duro del cliente, el servidor la envía nuevamente con la fecha caducada.
- **Seguro:** para indicar que el navegador puede devolver la cookie a un servidor seguro. Para comercio electrónico, actividades bancarias, etc.
- Justo antes de que un navegador solicite una página a un sitio Web, verifica su directorio de cookies para ver si el dominio al cual está solicitando la página ya colocó alguna cookie. De ser así todas las cookies colocadas por ese dominio se incluyen en el mensaje de solicitud. Cuando el servidor las obtiene puede interpretarlas en la forma que deseé.

Dominio	Ruta	Contenido	Expira	Seguro
toms-casino.com	/	CustomerID=497793521	15-10-02 17:00	Yes
joes-store.com	/	Cart=1-00501;1-07031;2-13721	11-10-02 14:22	No
aportal.com	/	Prefs=Stk:SUNW+ORCL;Spt:Jets	31-12-10 23:59	No
sneaky.com	/	UserID=3627239101	31-12-12 23:59	No

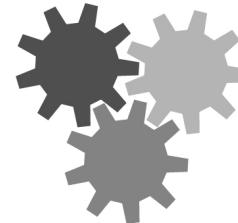
- En el ejemplo
  - La primera cookie fue establecida por toms-casino.com y **se utiliza para identificar al cliente**. Cuando este inicia una sesión, el navegador envía la cookie de forma que el servidor sepa quien es. Con el ID el servidor busca en su BD información sobre el cliente y en dependencia de los hábitos conocidos le muestra una página de inicio
  - La segunda fila **para hábitos de compra en una tienda virtual**.
  - La tercera **cookie es un portal Web**. Cuando el cliente entra al portal el navegador envía la cookie, que le indica al portal que construya una página que contenga los precios de las acciones de Sun y Oracle y los resultados de los Jets de Nueva York.
  - Los usuarios **pueden configurar su navegador para que rechacen las cookies**, o instalan programas que aceptan o rechazan las cookies en dependencia del sitio del que provenga.

- La base de la Web es la transferencia de páginas Web desde el servidor al cliente.
- En la forma más simple, las páginas Web son estáticas, son simplemente archivos que se encuentran en algún lugar del servidor esperando a ser recuperados.
- En este sentido, incluso un video es una página web estática porque es sólo un archivo.
- Las páginas Web estáticas se escriben en lenguaje llamado HTML (Lenguaje de Marcado de Hipertexto). HTML es un lenguaje de marcado que sirve para describir como se formatea un documento a través de etiquetas predefinidas. También se pueden emplear XML (Lenguaje de Marcado Extensible) y XSL (Lenguaje de Hojas de Estilo Extensible) para dar estructura a la página desarrollada y separar contenido de formato.

```
<html>
<head><title> AMALGAMATED WIDGET, INC. </title> </head>
<body> <h1> Welcome to AWI's Home Page</h1>
 <br>
We are so happy that you have chosen to visit <b> Amalgamated Widget's </b>
home page. We hope <i> you </i> will find all the information you need here.
<p> Below we have links to information about our many fine products.
You can order electronically (by WWW), by telephone, or by fax. </p>
<hr>
<h2> Product information </h2>
<ul>
  <li> <a href="http://widget.com/products/big"> Big widgets </a>
  <li> <a href="http://widget.com/products/little"> Little widgets </a>
</ul>
<h2> Telephone numbers</h2>
<ul>
  <li> By telephone: 1-800-WIDGETS
  <li> By fax: 1-415-765-4321
</ul>
</body>
</html>
```

(a)

## Welcome to AWI's Home Page



We are so happy that you have chosen to visit **Amalgamated Widget's** home page. We hope you will find all the information you need here.

Below we have links to information about our many fine products. You can order electronically (by WWW), by telephone, or by FAX.

---

**Product Information**

- [Big widgets](http://widget.com/products/big)
- [Little widgets](http://widget.com/products/little)

**Telephone numbers**

- 1-800-WIDGETS
- 1-415-765-4321

(a) HTML para una página Web de ejemplo. (b) Página con formato.

Etiqueta	Descripción
<html> ... </html>	Declara que la página Web está escrita en HTML
<head> ... </head>	Delimita el encabezado de la página
<title> ... </title>	Delimita el título (no se presenta en la página)
<body> ... </body>	Delimita el cuerpo de la página
<hn> ... </hn>	Delimita un encabezado de nivel n
<b> ... </b>	Pone en negritas
<i> ... </i>	Pone en cursivas
<center> ... </center>	Centra ... en la página horizontalmente
<ul> ... </ul>	Cochetes de una lista desordenada (con viñetas)
<li> ... </li>	Cochetes de un elemento de una lista ordenada o numerada
 	Obliga salto de línea aquí
<p>	Inicia un párrafo
<hr>	Inserta una regla horizontal
	Carga una imagen aquí
<a href="..."></a>	Define un hipervínculo

Selección de etiquetas HTML comunes.

- **HTML con o sin formularios**, no proporciona estructura alguna para las páginas Web. Además **mezcla el contenido con el formato**. Conforme el comercio electrónico y otras aplicaciones se vuelven más comunes hay una necesidad cada vez mayor de dar estructura a las páginas Web y separar el contenido del formato.
- Por esta razón el W3C ha mejorado HTML, para permitir que las páginas Web tengan estructura bien formada para su procesamiento automatizado.
- Se han desarrollado 2 nuevos lenguajes:
  1. **XML (Lenguaje de Marcado Extensible)**, que describe el contenido Web de forma estructurada
  2. **XSL (Lenguaje de Hojas de Estilo Extensible)** que describe el formato independientemente del contenido

Una página Web simple en XML.

Todo lo que hace el archivo es definir una lista de libros, que contiene tres libros. **No dice nada de cómo desplegar la página Web.**

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="b5.xsl"?>

<book_list>
  <book>
    <title> Computer Networks, 4/e </title>
    <author> Andrew S. Tanenbaum </author>
    <year> 2003 </year>
  </book>
  <book>
    <title> Modern Operating Systems, 2/e </title>
    <author> Andrew S. Tanenbaum </author>
    <year> 2001 </year>
  </book>
  <book>
    <title> Structured Computer Organization, 4/e </title>
    <author> Andrew S. Tanenbaum </author>
    <year> 1999 </year>
  </book>
</book_list>
```

Una hoja de estilo XSL.

Para proporcionar información de formato necesitamos un segundo archivo

**Book\_list.xsl.**

Este archivo es una hoja de estilo que indica cómo desplegar la página.

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">

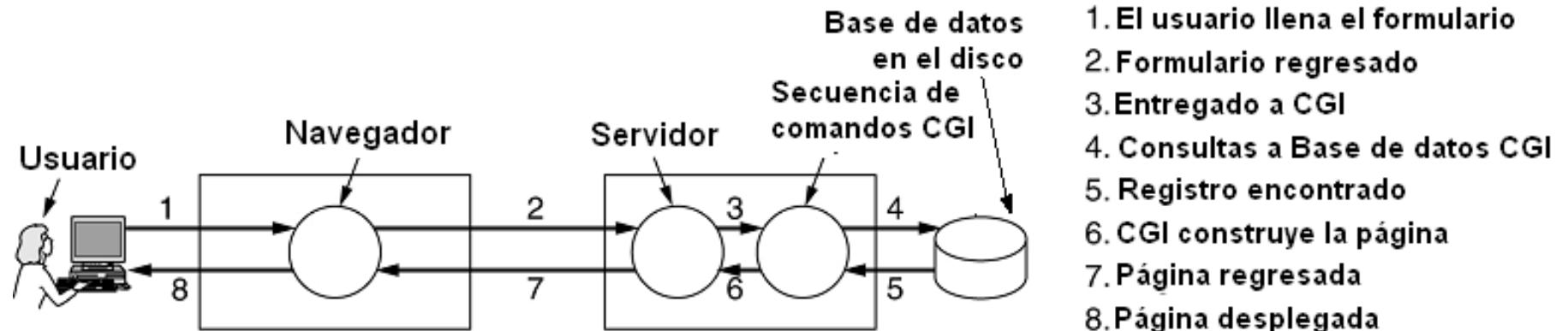
<html>
<body>
<table border="2">
<tr>
<th> Title</th>
<th> Author</th>
<th> Year </th>
</tr>

<xsl:for-each select="book_list/book">
<tr>
<td> <xsl:value-of select="title"/> </td>
<td> <xsl:value-of select="author"/> </td>
<td> <xsl:value-of select="year"/> </td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

- Además de describir páginas Web, XML sirve para otros propósitos. Por ejemplo se puede utilizar como lenguaje de comunicación entre programas de aplicación.
- En particular SOAP (Protocolo Simple de Acceso a Objetos) es una forma de hacer RPCs entre aplicaciones de una forma independiente del lenguaje y de la aplicación.
- En SOAP el cliente construye la solicitud como un mensaje XML y lo envía al servidor, utilizando HTTP. El servidor envía una respuesta otro mensaje XML pudiendo comunicarse aplicaciones heterogéneas.

- HTML sigue evolucionando para satisfacer las nuevas demandas, con la introducción de las PDAs se necesitaba una adaptación a estos nuevos dispositivos. Estos dispositivos tienen memoria limitada para navegadores grandes que tratan de lidiar con las páginas web incorrectas sintácticamente.
- De ahí que se ha especificado **XHTML ( Lenguaje de Marcado de Hipertexto Extendido)** es HTML 4.0 reformulado en XML.
- Algunas de las características de XHTML respecto a HTML
  - Las páginas XHTML y los navegadores deben seguir estrictamente el estándar. Si una página es de mala calidad se debe señalar no corregir.
  - Todas las etiquetas y los atributos deben estar en minúsculas <HTML> por <html>.
  - Se requiere siempre el uso de etiquetas de cierre incluso para <p>, <br>, <hr>, <img>.
  - Todos los atributos deben estar entre comillas, incluso los números .
  - Las etiquetas deben estar anidadas de manera apropiada

- Hasta ahora el modelo que hemos utilizado es el **cliente envía un nombre de archivo al servidor**, el cual regresa el archivo. Este era el modelo imperante en los primeros días de la Web.
- En los años más recientes, **cada vez más el contenido es dinámico**, es decir se genera a solicitud, en lugar de almacenarlo en disco.
- La generación del contenido **puede suceder ya sea en el servidor o en el cliente**.



Pasos en el procesamiento de información de un formulario HTML

```
<html>
<body>

<h2> This is what I know about you </h2>
<?php echo $HTTP_USER_AGENT ?>

</body>
</html>
```

*Una página HTML de ejemplo con PHP incrustado*

# Documentos Web dinámicos en el servidor

```
<html>
<body>
<form action="action.php" method="post">
<p> Please enter your name: <input type="text" name="name"> </p>
<p> Please enter your age: <input type="text" name="age"> </p>
<input type="submit">
</form>
</body>
</html>
```

(a)

```
<html>
<body>
<h1> Reply: </h1>
Hello <?php echo $name; ?>.
Prediction: next year you will be <?php echo $age + 1; ?>
</body>
</html>
```

(b)

```
<html>
<body>
<h1> Reply: </h1>
Hello Barbara.
Prediction: next year you will be 25
</body>
</html>
```

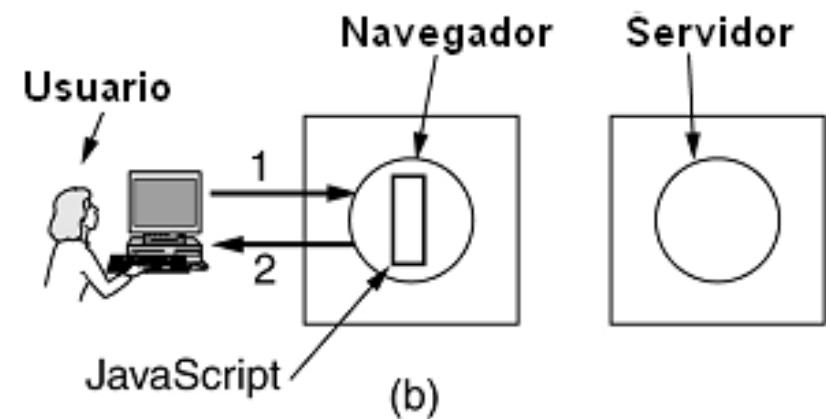
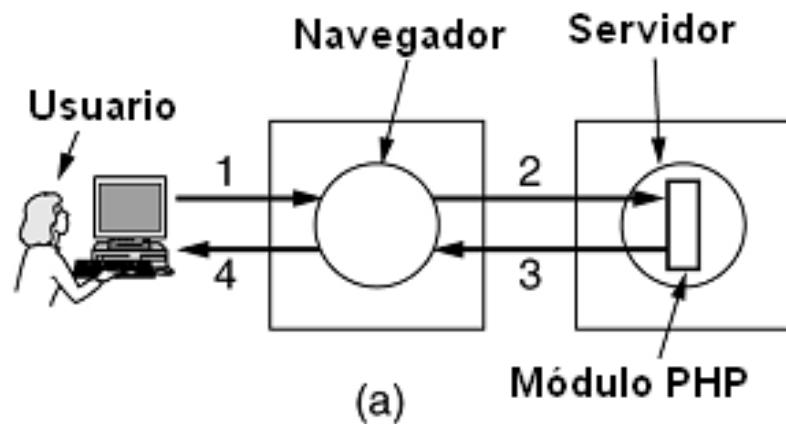
(c)

- (a) Una página Web que contiene un formulario. (b) Un script en PHP para manejar la salida del formulario. (c) Salida de la secuencia de comandos PHP cuando las entradas son “Barbara” y 24.

### Uso de JavaScript para procesar un formulario

```
<head>
<script language="javascript" type="text/javascript">
function response(test_form) {
    var person = test_form.name.value;
    var years = eval(test_form.age.value) + 1;
    document.open();
    document.writeln("<html> <body>");
    document.writeln("Hello " + person + ".<br>");
    document.writeln("Prediction: next year you will be " + years + ".");
    document.writeln("</body> </html>");
    document.close();
}
</script>
</head>

<body>
<form>
Please enter your name: <input type="text" name="name">
<p>
Please enter your age: <input type="text" name="age">
<p>
<input type="button" value="submit" onclick="response(this.form)">
</form>
</body>
</html>
```



- (a) Secuencia de comandos en el servidor PHP.
- (b) Secuencia de comandos en el cliente con JavaScript.

```
<html>
<head>
<script language="javascript" type="text/javascript">

function response(test_form) {
    function factorial(n) {if (n == 0) return 1; else return n * factorial(n - 1);}
    var r = eval(test_form.number.value);      // r = typed in argument
    document.myform.mytext.value = "Here are the results.\n";
    for (var i = 1; i <= r; i++)           // print one line from 1 to r
        document.myform.mytext.value += (i + "!" + factorial(i) + "\n");
}
</script>
</head>

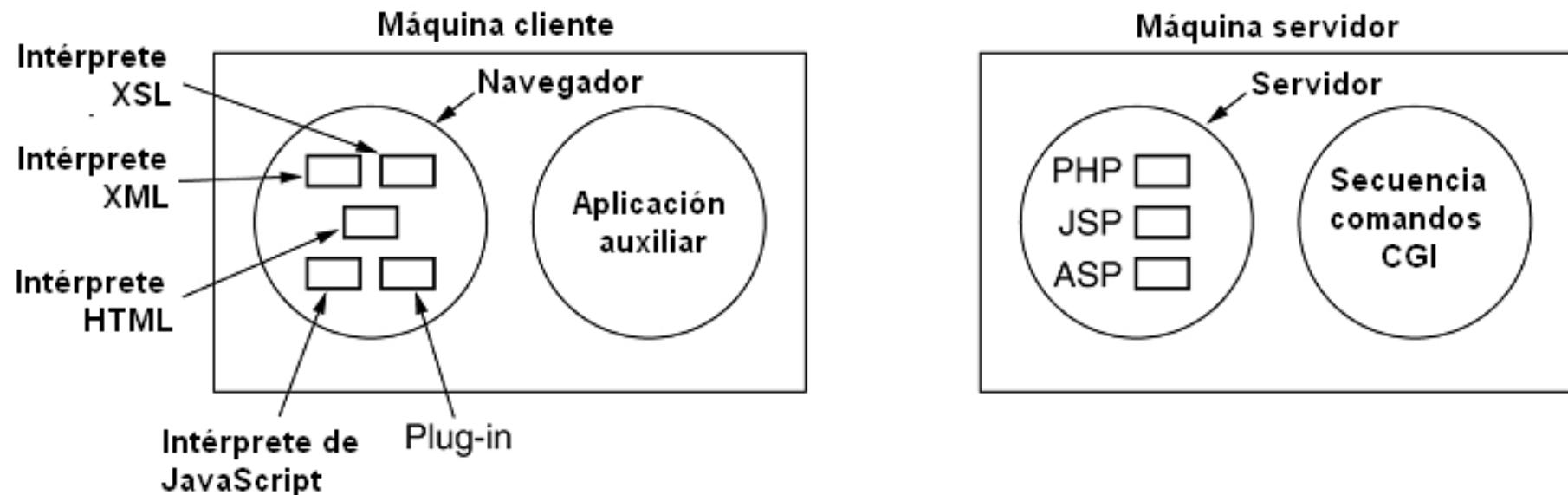
<body>
<form name="myform">
Please enter a number: <input type="text" name="number">
<input type="button" value="compute table of factorials" onclick="response(this.form)">
<p>
<textarea name="mytext" rows=25 cols=50> </textarea>
</form>
</body>
</html>
```

Un programa en JavaScript para calcular e imprimir factoriales.

```
<html>
<head>
<script language="javascript" type="text/javascript">
if (!document.myurl) document.myurl = new Array();
document.myurl[0] = "http://www.cs.vu.nl/~ast/im/kitten.jpg";
document.myurl[1] = "http://www.cs.vu.nl/~ast/im/puppy.jpg";
document.myurl[2] = "http://www.cs.vu.nl/~ast/im/bunny.jpg";
function pop(m) {
    var urx = "http://www.cs.vu.nl/~ast/im/cat.jpg";
    popupwin = window.open(document.myurl[m],"mywind","width=250,height=250");
}
</script>
</head>

<body>
<p> <a href="#" onMouseover="pop(0); return false;"> Kitten </a> </p>
<p> <a href="#" onMouseover="pop(1); return false;"> Puppy </a> </p>
<p> <a href="#" onMouseover="pop(2); return false;"> Bunny </a> </p>
</body>
</html>
```

Una página Web interactiva que responde al movimiento del ratón.



Las diversas formas de generar y desplegar contenido.

- El protocolo de transferencia utilizado en la World Wide Web es **HTTP**. Este protocolo especifica cuáles mensajes pueden enviar los clientes a los servidores y qué respuestas obtienen. [Utiliza el puerto 80](#).
- Cada interacción consiste en una solicitud **ASCII**, seguida por una respuesta tipo **MIME RFC 822**. Todos los clientes y servidores deben obedecer este protocolo definido en RFC 2616.
- HTTP 1.1 soporta [conexiones persistentes](#), con ellas es posible establecer una conexión TCP, enviar una solicitud y obtener una respuesta y después enviar solicitudes adicionales evitando la sobrecarga de TCP por solicitud.

Método	Descripción
GET	Solicita al servidor la lectura de una página Web. Ej. GET nombre_archivo HTTP/1.1
HEAD	Solicita la lectura del encabezado de una página Web
PUT	Solicita el almacenamiento de una página Web en un servidor remoto. El cuerpo de la solicitud contiene la página, que puede contener encabezados MIME
POST	Igual que PUT, pero los datos se insertan en un formato específico. Enviar un mensaje a un grupo de noticias
DELETE	Elimina la página Web
TRACE	Para depuración. Indica al servidor que regrese la solicitud
CONNECT	Reservado para uso futuro
OPTIONS	Consulta ciertas opciones

Código	Significado	Ejemplos
1xx	Información	100 = el servidor está de acuerdo en manejar la solicitud del cliente
2xx	Éxito	200 = la solicitud es exitosa; 204 = no hay contenido
3xx	Redirección	301 = página movida, 304 = la página en caché es aún válida
4xx	Error del cliente	403 = página prohibida, 404 = página no encontrada
5xx	Error del servidor	500 = error interno del servidor, 503 = trata más tarde

### Los grupos de respuesta del código del estado

Cada solicitud obtiene una respuesta que consiste en una línea de estado e información adicional. La línea de estado contiene 3 dígitos que indica si la solicitud fue atendida y si no el por qué.

# Ejemplo de uso de HTTP

El inicio de la salida de  
*www.ietf.org/rfc.html*.  
telnet *www.ietf.org* 80 >log  
GET /rfc.html HTTP/1.1

```
Trying 4.17.168.6...
Connected to www.ietf.org.
Escape character is '^].
HTTP/1.1 200 OK
Date: Wed, 08 May 2002 22:54:22 GMT
Server: Apache/1.3.20 (Unix) mod_ssl/2.8.4 OpenSSL/0.9.5a
Last-Modified: Mon, 11 Sep 2000 13:56:29 GMT
ETag: "2a79d-c8b-39bce48d"
Accept-Ranges: bytes
Content-Length: 3211
Content-Type: text/html
X-Pad: avoid browser bug
```

```
<html>
<head>
<title>IETF RFC Page</title>

<script language="javascript">
function url() {
    var x = document.form1.number.value
    if (x.length == 1) {x = "000" + x }
    if (x.length == 2) {x = "00" + x }
    if (x.length == 3) {x = "0" + x }
    document.form1.action = "/rfc/rfc" + x + ".txt"
    document.form1.submit
}
</script>

</head>
```