

PROTOCOLOS DE COMUNICACIÓN DE RED

CLASE 6: CAPA DE APLICACIÓN

Lic. **Rubén G. Apolloni**

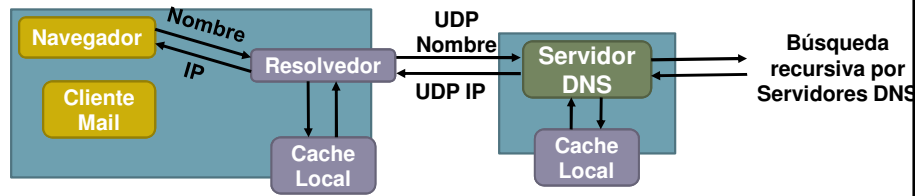
Área de Sistemas de Computación
Universidad Nac. De San Luis
(5700) San Luis – San Luis

Sistema de Nombres de Dominio (DNS).

- ⦿ Todos los dispositivos, host y recursos son identificados por medio de direcciones de red (IP).
- ⦿ Es difícil recordar direcciones de red.
- ⦿ Se crean los nombres ASCII para diferenciar los nombres de las máquinas de las direcciones de máquina.
- ⦿ Esquema de nombre jerárquico, basado en dominios.
- ⦿ Bases de datos distribuidas.

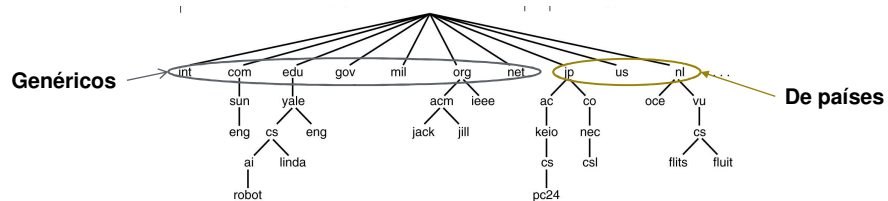
Sistema de Nombres de Dominio (DNS).

- Funcionamiento:



Sistema de Nombres de Dominio (DNS).

- Internet se divide en 200 dominios de nivel superior.
- Cada dominio se divide en subdominios, y así sucesivamente.
- Cada dominio controla los subdominios que se encuentran por debajo.



Sistema de Nombres de Dominio (DNS).

- ◉ Dominio del nivel superior:

Dominios	Significado
com	Comercial.
edu	Instituciones educativas.
gov	Gobierno de EEUU.
mil	Militar de EEUU.
int	Organizaciones Internacional.
net	Proveedores de red.
org	Organizaciones no gubernamental.
biz	Negocios.
info	Información.
name	Nombre de personas.
pro	Profesionales.
aero	Industria aeroespacial.
coop	Cooperativas.
museum	Museos.

Sistema de Nombres de Dominio (DNS).

- ◉ Cada dominio se expresa por la ruta “ascendente” completa hasta la raíz, comenzando por el dominio mismo.
- ◉ Paso absoluto: termina con punto (“raíz”).
dirinfo.unsl.edu.ar.
info.unlp.edu.ar.
- ◉ Paso relativo: sin punto al final.
- ◉ No se distingue entre mayúsculas y minúsculas.
- ◉ Cada componente no debe superar los 63 caracteres de longitud
- ◉ La longitud de la ruta no debe superar los 255 caracteres.

Sistema de Nombres de Dominio (DNS).

- Un DNS es una base de datos distribuida constituida de registros.
- Registros de Recursos (RR).
- Cada dominio contiene un grupo de RR.
- Cada RR contiene 5 campos de información.

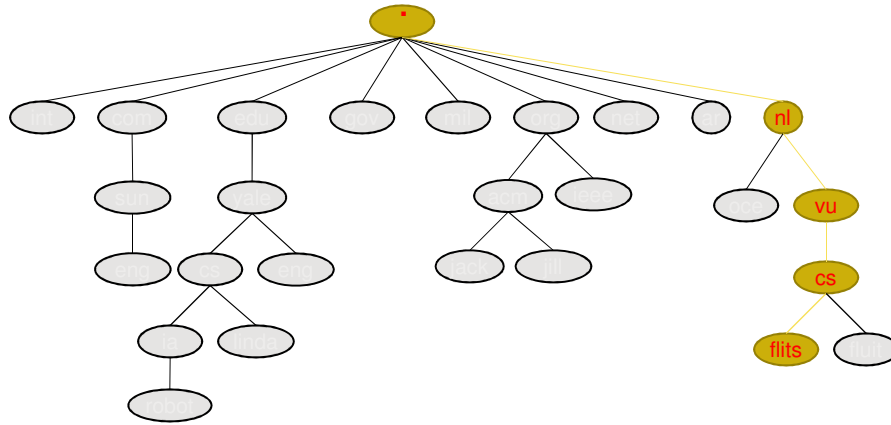
Nombre_dominio Tiempo_de_vida Clase Tipo Valor

- Nombre del dominio: indica el dominio al que pertenece el registro.
- Tiempo de Vida (TTL): indicación de la estabilidad del registro.
- Clase: IN si el RR contiene información de Internet.
- Tipo: define el tipo de recurso descrito por el registro.

Sistema de Nombres de Dominio (DNS).

Tipo	Significado	Valor
SOA	Inicio de autoridad, permite la descripción del servidor de nombre de dominio con autoridad en la zona.	Parámetro para la zona.
A	Hace coincidir el nombre canónico con la dirección IP.	Dirección IP.
MX	Servidor de correo electrónico.	Prioridad, Dominio dispuesto a aceptar correo electrónico.
NS	Servidor de nombres de dominio con autoridad sobre el dominio.	Nombre de un servidor.
CNAME	Nombre Canónico, permite definir un alias para el nombre canónico.	Nombre del dominio
PTR	Puntero, alias de una dirección IP. Puntero hacia otra parte del espacio de nombres del dominios.	Nombre del dominio.
HINFO	Descripción del hardware del host y del sistema operativo.	CPU y Sistema Operativo
TXT	Texto	Texto no interpretado.

Sistema de Nombres de Dominio (DNS).



Sistema de Nombres de Dominio (DNS).

● Ejemplo: base de datos para el dominio cs.vu.nl

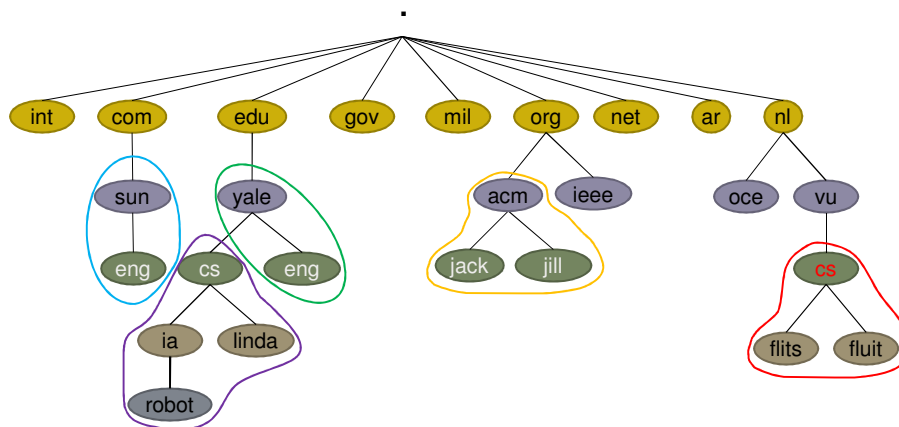
Información del		; Authoritative data for cs.vu.nl			
Información textual de la ubicación del dominio.	cs.vu.nl.	86400	IN	SOA	star boss (952771,7200,7200,2419200,86400)
	cs.vu.nl.	86400	IN	TXT	"Divisie Wiskunde en Informatica."
	cs.vu.nl.	86400	IN	TXT	"Vrije Universiteit Amsterdam."
	cs.vu.nl.	86400	IN	MX	1 zephyr.cs.vu.nl.
Donde entregar los mails destinados a persona@cs.vu.nl	cs.vu.nl.	86400	IN	MX	2 top.cs.vu.nl.
	flits.cs.vu.nl.	86400	IN	HINFO	Sun Unix
Direcciones alternativas para Alias para el servidor web y ftp	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	
Entradas para un impresora del dominio.	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

Sistema de Nombres de Dominio (DNS).

- El espacio de nombres DNS se divide en zonas no solapadas.
- Cada zona:
 - Contiene una parte del árbol.
 - Servidores de nombre con la información de autorización correspondiente.
 - Servidor de nombre primario: que obtienen la información desde un archivo en su disco.
 - Servidores de nombres secundarios: obtienen la información desde el servidor primario.
 - Algunos servidores se ubican fuera de lo zona.

Sistema de Nombres de Dominio (DNS).

- Ejemplo de zonas.

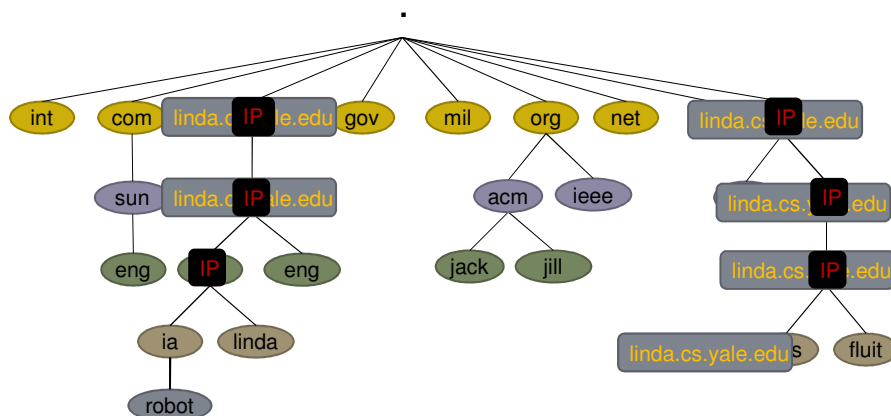


Sistema de Nombres de Dominio (DNS).

- Resolución de nombres:
- Es un proceso cliente/servidor.
- La función del cliente (el "*resolver*" o "*name resolver*") es transparente para el usuario.
- Full resolver: programa de usuario que puede realizar consulta recursivas e iterativas.
- Ejemplo: `nslookup <dir_text>|<IP>`
- Stubs resolver: rutina enlazada con el programa de usuario, que envía las peticiones a un servidor de nombres. En UNIX se implementa con dos funciones de librería: `gethostbyname()` y `gethostbyaddr()` para convertir nombres de hosts a direcciones IP y viceversa.

Sistema de Nombres de Dominio (DNS).

- Consulta recursiva: demanda que el servidor lance a su vez una consulta para determinar la información buscada y luego devolvérsela al cliente.



Sistema de Nombres de Dominio (DNS).

- ◉ Consulta iterativa: el servidor de nombres debería devolver la información que disponga además de una lista de servidores adicionales con los que el cliente puede contactar para completar su consulta.

Sistema de Nombres de Dominio (DNS).

- ◉ Respuesta:
- ◉ Con Registro Autorizado: proviene de la autoridad que administra el registro, siempre es correcta.
- ◉ Con Registro no Autorizado: si proviene de un servidor que no es autoridad del dominio o de una cache.

Correo Electrónico.

- ⦿ Es un servicio de red que permite a los usuarios enviar y recibir mensajes.
- ⦿ El correo electrónico antecede a la Internet.
- ⦿ El correo electrónico comenzó a utilizarse en 1965 en una supercomputadora .
- ⦿ Inicialmente consistía en un protocolo de transferencia de archivos.
- ⦿ Con la convención de que la primera línea contenía la dirección del destinatario.

Correo Electrónico.

- ⦿ En 1982 se publico la primera propuesta de correo electrónico con el RFC 821 (protocolo de transmisión) y el RFC 822 (formato del mensaje).
- ⦿ El estándar actual de correo electrónico se definen en los RFC 2821 (protocolo de transmisión) y el RFC 2822 (formato del mensaje).
- ⦿ En 1984 ls CCITT redacto las recomendaciones X.400, pero nunca se popularizo.

Correo Electrónico. Funciones.

- ⦿ Redacción: proceso para crear mensajes y respuestas. Puede utilizarse un simple editor de texto.
- ⦿ Transferencia: mover el mensaje del remitente al destinatario. Implica establecer una conexión con el destino o con algún servidor intermedio, enviar el mensaje y cerrar la conexión.
- ⦿ Generación de Informes: indicar al remitente lo que ocurrió con el mensaje.
- ⦿ Visualización: leer el correo electrónico entrante. En algunos casos se requieren conversiones o invocar visores específicos.
- ⦿ Disposición: lo que hace el destinatario con el mensaje una vez que lo recibe (borrarlo, almacenarlos, responder, etc.).

Correo Electrónico. Arquitectura

- ⦿ Agente de usuario:
- ⦿ También denominados lectores de correo.
- ⦿ Leer , escribir y enviar correo electrónico.
- ⦿ Proveen métodos para interactuar con el sistema de correo electrónico.
- ⦿ Por medio de comandos, menús o interface gráfica.

Correo Electrónico. Arquitectura

- ◉ Agente de usuario: Ejemplos.

De escritorios	Webs	Texto
Eudora Kmail Microsoft Outlook MozillaThunderbird Novell Evolution Novell GroupWise Opera Mail Windows Live Mail Apple Mail	SquirrelMail WebPine 24SevenOffice Alpine Bongo	Pine Gnus mailx getmail

Correo Electrónico. Arquitectura.

- ◉ Agente de usuario: Opciones avanzadas.
- ◉ Buzones: proveen comando para almacenar el correo.
- ◉ Se requieren comandos para crear y borrar buzones. Inspeccionar el contenido, insertar y borrar mensajes
- ◉ Lista de correo: lista de direcciones de correo electrónico.
- ◉ Cuando se envía correo a una lista se entrega copias idéntica a todas las direcciones de la lista.
- ◉ Copias ocultas, correo de alta prioridad, correo electrónica secreto (encriptado), destinatarios alternativos.

Correo Electrónico. Arquitectura.

- ⦿ Agentes de Transferencia:
- ⦿ Mueve los mensajes de origen a destino.
- ⦿ Un programa demonio (daemons) de sistema que funcionan en segundo plano.
- ⦿ Mueven el correo electrónico de origen a destino y viceversa.

Correo Electrónico. Arquitectura.

- ⦿ Formato del mensaje:
- ⦿ Definido en el RFC 822.
- ⦿ Sobre: Definido en el RFC 821.
 - ⦿ Encapsula el mensaje.
 - ⦿ Contiene toda la información para transportar el mensaje.
 - ⦿ Dirección de destino, prioridad, nivel de seguridad.
 - ⦿ Los agentes de transporte lo utilizan para routing el mensaje.
- ⦿ Contenido:
 - ⦿ Lo que va dentro del sobre. Dos partes:
 - ⦿ Encabezado: contiene información de control para los agentes de usuario.
 - ⦿ Línea en blanco
 - ⦿ Cuerpo: el mensaje.

Correo Electrónico. Arquitectura.

- Formato del mensaje:
- Encabezado: compuesto de varios campos
- Campo:
 - Línea en formato ASCII.
 - <Nombre_del_campo>: <Valor>
 - Los emplea el agente de transferencia para construir el sobre.

Correo Electrónico. Arquitectura.

- Formato del mensaje:
- Encabezado, Campo principales:

Encabezado	Significado
To:	Dir. de correo electrónico de los destinatarios primarios.
Cc:	Dir. de correo electrónico de los destinatarios secundarios.
Bcc:	Dir. de correo electrónico de los destinatarios de copia oculta.
From:	Persona/s que crearon el mensaje.
Sender:	Dir. de correo electrónico del remitente.
Received:	Una línea por cada agente de transferencia por el cual paso a través de la ruta hacia el destino final.
Return-Path:	Ruta de regreso al remitente.

Correo Electrónico. Arquitectura.

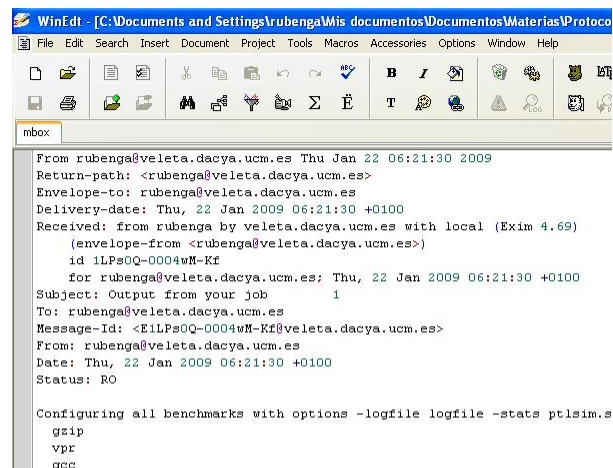
- Formato del mensaje:
- Encabezado, Campo adicionales:

Encabezado	Significado
Date:	Fecha y hora del envío del mensaje.
Reply-To:	Dir. de correo electrónico a la que se debe responder.
Message-Id:	Número único para referencias posteriores al mensaje.
In-Reply-To:	Identificador del mensaje al que éste responde.
References:	Otros identificadores de mensajes pertinentes.
Keywords:	Claves seleccionadas por el usuario.
Subjects:	Resumen corto del mensaje.

- El usuario puede agregar encabezados para uso privado. El nombre debe comenzar con X-
- Cuerpo del mensaje: el contenido de la carta.

Correo Electrónico. Arquitectura.

- Formato del mensaje:
- Ejemplo:



```
WinEdit - [C:\Documents and Settings\rubenga\Mis documentos\Documentos\Materias\Protoco
File Edit Search Insert Document Project Tools Macros Accessories Options Window Help
From rubenga@veleta.dacya.ucm.es Thu Jan 22 06:21:30 2009
Return-path: <rubenga@veleta.dacya.ucm.es>
Envelope-to: rubenga@veleta.dacya.ucm.es
Delivery-date: Thu, 22 Jan 2009 06:21:30 +0100
Received: from rubenga by veleta.dacya.ucm.es with local (Exim 4.69)
(envelope-from <rubenga@veleta.dacya.ucm.es>)
id 1LPs0Q-0004vH-Kf
for rubenga@veleta.dacya.ucm.es; Thu, 22 Jan 2009 06:21:30 +0100
Subject: Output from your job 1
To: rubenga@veleta.dacya.ucm.es
Message-Id: <E1LPs0Q-0004vH-Kf@veleta.dacya.ucm.es>
From: rubenga@veleta.dacya.ucm.es
Date: Thu, 22 Jan 2009 06:21:30 +0100
Status: RO

Configuring all benchmarks with options -logfile logfile -stats ptlsim.s
gzip
vpr
gcc
```

Correo Electrónico. Arquitectura.

- ◉ Formato del mensaje:
- ◉ MIME (Extensiones Multipropósito de Correo Internet).
- ◉ Definido en los RFC 1341 y actualizado en los RFCs 2045-2049
- ◉ Agrega estructura al cuerpo del mensaje .
- ◉ Define reglas de codificación para mensajes no ASCII.
- ◉ Define 5 nuevos encabezados:

Encabezados	Significados
MIME-Version:	Identifica la versión del MIME.
Content-Descriptions:	Cadena de texto que describe el contenido.
Content-Id:	Identificador único.
Content-Transfers-Encoding:	Cómo se envuelve el mensaje para su transmisión.
Content-Type:	Naturaleza del mensaje

Correo Electrónico. Arquitectura.

- ◉ Formato del mensaje: MIME (Extensiones Multipropósito de Correo Internet).
- ◉ Content-Type: inicialmente el RFC 2045 definió 7 tipos básicos:
- ◉ Formato:

Content-Type: <tipo>/<subtipo>

Correo Electrónico. Arquitectura.

- Formato del mensaje: MIME (Extensiones Multipropósito de Correo Internet).
- Content-Type:

Tipo	Subtipo	Etiquetas	Descripción
Texto	Plano	text/plain	Texto sin formato.
	Enriquecido	text/enriched, text/html, text/xml	Texto con formato.
Imagen	Gif	image/gif	Imágenes gif.
	Jpeg	image/jpeg	Imagen jpeg.
Audio	Básico	audio/basic	Sonido.
Video	Mpeg	video/mpeg	Películas, videos.
Aplicación	Octet-stream	application/octet-stream	Secuencia de bytes no interpretados
	Postscript	application/postscript	Documentos.

Correo Electrónico. Arquitectura.

- Formato del mensaje: MIME.
- Content-Type:

Tipo	Subtipo	Etiquetas	Descripción
Mensaje	RFC822	message/rfc822	Mensajes MIME 822.
	Parcial	message/partial	Mensajes divididos para su transmisión.
	Externo	message/external-body	El mensaje debe obtenerse de la red.
Multiparte	Mezclado	multipart/mixed	Partes independientes en el orden especificado.
	Alternativa	multipart/alternative	Mismo mensaje en formatos diferentes.
	Paralelo	multipart/parallel	Las partes deben verse en forma simultánea
	Compendio	multipart/related	Cada parte es un mensaje RFC 822 completo.

- [Listado mas extenso de etiquetas MIME](#)

Correo Electrónico. Arquitectura.

- ◉ Formato del mensaje: MIME (Extensiones Multipropósito de Correo Internet).
- ◉ Problemas solucionados:
 - ◉ Mensajes en idiomas con acento.
 - ◉ Mensajes en alfabetos no latinos.
 - ◉ Mensajes en idiomas sin alfabetos.
 - ◉ Mensajes con formato.
 - ◉ Mensajes que contienen audios, fotos, videos, etc.

Correo Electrónico. Arquitectura.

- ◉ Dirección:
- ◉ Tiene la forma:

<usuario>@<direccion-DNS>|<IP>

- ◉ Existen otros formatos: por ejemplo X.400.
- ◉ Se componen de pares <atributo> = <valor>
/C=US/ST=MASSACHUSETTS/L=CAMBRIDGE/PA=360 MEMORIAL DR./CN=KEN SMITH

Correo Electrónico. Transferencia de Mensajes.

- Protocolo Simple de Transporte de Correo:
- Se ocupa de transmitir el/los mensaje/s del remitente a el/los destinatario/s.
- Protocolo ASCII.
- Modelo cliente/servidor.
- Cliente: La computadora origen establece una conexión TCP con el puerto 25 de la computadora destino.
- Servidor: el demonio de correo electrónico se encuentra escuchando en el puerto 25 a la espera de conexiones, y que entiende el protocolo SMTP.
- El demonio acepta la conexión de entrada y copia el mensaje al buzón que corresponde.
- Si no puede entregarse el mensaje devuelve un mensaje de error al remitente.

Correo Electrónico. Transferencia de Mensajes.

- Transferencia de un mensaje desde *elinore@abc.com* to *carolyn@xyz.com*.

```
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
^
```

Correo Electrónico. Transferencia de Mensajes.

- Transferencia de un mensaje desde *elinore@abc.com* to *carolyn@xyz.com*.

```
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
```

Correo Electrónico. Transferencia de Mensajes.

- Problema:
- Si la computadora del servidor no esta constantemente on-line.
- No se puede establecer una conexión TCP.
- No se ejecuta el demonio en el puerto 25.
- Solución:
- Que el agente de transferencia de mensajes se ejecute en una computadora que se encuentre siempre encendida.
- Los agentes de usuarios se obtengan los mensajes desde la computadora que ejecuta el agente de transferencia de mensajes.

Correo Electrónico. Transferencia de Mensajes.

- ⦿ Protocolo de Oficina de Correo Versión 3 (POP3).
- ⦿ Cada vez que arranca el programa de lectura de correo electrónico se conecta con el agente de transferencia.
- ⦿ Transfiere los mensajes del agente de transferencia y los almacena en la computadora donde ejecuta el programa de lectura de correo electrónico.
- ⦿ Establece una conexión TCP con el agente de transferencia en el puerto 110.
- ⦿ Establecida la conexión atraviesa 3 estados:
 1. Autorización: solicitud de usuario y clave.
 2. Transacciones: copia localmente los mensaje y los marca para borrar en el servidor.
 3. Actualización: se borran los mensajes del servidor.
- ⦿ Se cierra la conexión.

Correo Electrónico. Transferencia de Mensajes.

- ⦿ Protocolo de Oficina de Correo Versión 3 (POP3).

```

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
```

Correo Electrónico. Transferencia de Mensajes.

- ⦿ Protocolo de Oficina de Correo Versión 3 (POP3).
- ⦿ Ventajas:
 - ⦿ Se puede operar con el correo sin estar conectado.
- ⦿ Desventaja:
 - ⦿ Se almacenan en el disco local de la computadora del usuario.
 - ⦿ No es posible acceder a los mails desde diferentes computadoras.

Correo Electrónico. Transferencia de Mensajes.

- ⦿ Protocolo de Acceso a Mensajes de Internet (IMAP):
 - ⦿ Los mails son mantenidos en el servidor.
 - ⦿ Proporciona comandos para interactuar con los mensajes en el servidor: leer, crear, borrar , manipular buzones, etc.
 - ⦿ El servidor escucha en el puerto 143.
 - ⦿ El cliente debe estar en línea para poder acceder y manipular los mails.

Protocolo de Transferencia de Hipertexto (HTTP).

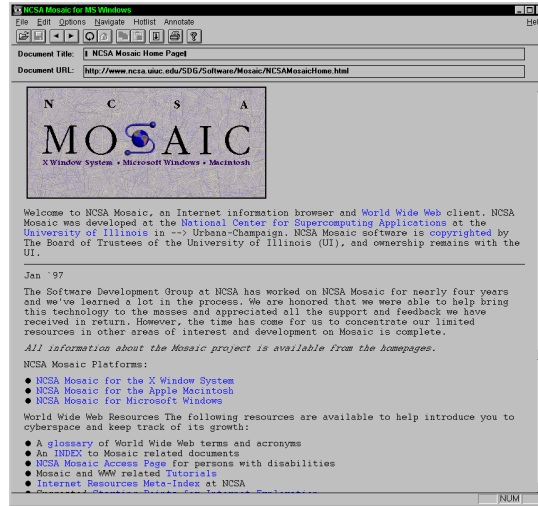
- ⦿ Es el protocolo base de World Wide Web (WWW) .
- ⦿ Puede emplearse en cualquier aplicación cliente – servidor que suponga la utilización de hipertexto.
- ⦿ Protocolo para transferir información con la eficiencia necesaria para efectuar saltos de hipertexto.
- ⦿ Datos transferidos: texto, hipertexto, audio, imágenes, etc.

Protocolo de Transferencia de Hipertexto (HTTP).

- ⦿ WWW o Web:
- ⦿ Basado en textos con hipervinculos.
- ⦿ Idea de Vannevar Bush en 1945.
- ⦿ Pensado y desarrollado en 1989 por el físico Tim Berners-Lee en el CERN (Centro Europeo de Investigación Nuclear).
- ⦿ Originalmente se diseño para el intercambio de información (texto, imágenes, etc.) entre los científicos.
- ⦿ Permite acceder a millones de documentos vinculados.
- ⦿ Distribuidos en miles de máquinas en todo el mundo.
- ⦿ En 1993 se libera el primer navegador gráfico, Mosaic.
- ⦿ En 1994 el CERN y el MIT firmaron un acuerdo para crear el World Wide Web Consortium (W3C).

Protocolo de Transferencia de Hipertexto (HTTP).

- Mosaic:
- Creado por Marck Andreessen
- Creador de la empresa Netscape Communications Corp.



Arquitectura: Direccinamiento.

- Localizador Uniforme de Recursos (URL):
- Cada URL hace referencia a una página.
- Consta de 3 partes
- <protocolo>://<Nombre DNS host>/<directorio.../ archivo>
- Ejemplos:
- <http://www.unsl.edu.ar/mapa.htm>
- <http://www.unsl.edu.ar>
- <http://www.unsl.edu.ar/~noticias/>

Arquitectura: Direccionamiento.

- Protocolo:

Nombre	Uso	Ejemplo
http	Hipertexto	http://www.unsl.edu.ar
ftp	Transferencia de archivos	ftp://ftp.sunet.se/pub/Linux/distributions/
file	Archivos locales o en la LAN	file:///C:/Documents%20and%20Settings/rubenga/Mis%20documentos/videos/V011109_12.07.3gp
news	grupos de noticias Usenet	news:comp.os.linux.hardware
mailto:	direcciones e-mail	mailto:rubenga@unsl.edu.ar

Arquitectura:

- La web consiste de millones de páginas de textos, imágenes, audios, videos, hipervínculos, etc.
- Un hipervínculo es una puntero a otra página ubicada en el mismo servidor o en otro servidor.
- Los navegadores obtienen la página solicitada, interpreta el texto y los comando de formateo que contiene y despliegan la página en la pantalla adecuadamente formateada.

Arquitectura: Cliente o Navegadores.

- ⦿ Programa que puede interpretar y mostrar páginas webs.
- ⦿ Interpretar las acciones de los usuarios.
- ⦿ Cuando un usuario selecciona un hipervínculo se realizan las siguientes acciones:
 1. Determina la página a la que hace referencia, mediante el Localizador Uniforme de Recursos (URL).
 2. Solicita al DNS la dirección IP.
 3. Obtiene la respuesta.
 4. Establece una conexión TCP con el puerto 80 del servidor.
 5. Solicita el archivo que contiene la página web.
 6. El servidor envía el archivo.
 7. Se libera la conexión TCP.
 8. Despliega la página contenida en el archivo.
 9. Obtiene y muestra todos los contenidos adicionales (imágenes, etc.)

Cliente: Navegadores.

- ⦿ No todas las páginas contienen HTML.
- ⦿ Puede ser un documento PDF, una imagen JPG, un video MPEG, etc.
- ⦿ El navegador no lo puede interpretar.
- ⦿ Cuando el servidor envía el archivo incluye información de tipo MIME.
- ⦿ El navegador consulta su tabla para determinar que programa ejecutar para visualizar el contenido.
- ⦿ Visores:
 - Plug-in:
 - Aplicaciones auxiliares:

Cliente: Navegadores.

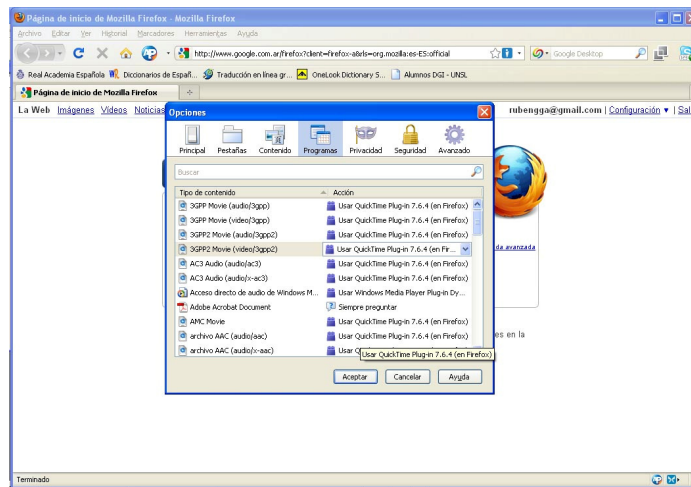
- ⦿ Plug-in (complementos):
- ⦿ Módulo de código que obtiene de un directorio preconfigurado del disco.
- ⦿ Se instala como una extensión del navegador.
- ⦿ Desarrollos externos.
- ⦿ Amplían las funciones
- ⦿ Se ejecutan dentro del navegador.
- ⦿ Se visualizan dentro de la página.
- ⦿ El plug-in proporciona un conjunto de procedimientos (interface), para que el navegador pueda invocarlo.
- ⦿ El navegador proporciona un conjunto de procedimientos para que los puedan invocar los plug-ins

Cliente: Navegadores.

- ⦿ Programas auxiliares:
- ⦿ Es un programa completo.
- ⦿ Se ejecuta como un proceso independiente.
- ⦿ No ofrece interface con el navegador.
- ⦿ No utiliza servicios del navegador.
- ⦿ El navegador deposita el archivo en un directorio.
- ⦿ Invoca la aplicación auxiliar y le pasa como parámetro el directorio y el archivo.
- ⦿ Se ejecuta la aplicación, la cual se encarga de abrir el archivo y visualizar el contenido.

Cliente: Navegadores.

- Los navegadores permiten configurar que aplicaciones se encargaran de visualizar cada uno de los contenidos.
- Asocian tipos MIMEs con aplicaciones.



Servidor.

- Atiende y entrega los archivos a los clientes que lo solicitan.
- Los pasos que se ejecutan en el servidor son:
 1. Acepta conexiones TCP de los clientes.
 2. Obtiene el nombre del archivo solicitado.
 3. Obtiene el archivo desde el disco.
 4. Libera la Conexión TCP.
- Problemas:
- Cada solicitud del cliente requiere acceder al dispositivo de almacenamiento (disco).

Servidor.

- ◉ Soluciones:
- ◉ Cache en memoria principal:
- ◉ Mantener la últimas páginas solicitadas en memoria principal.
- ◉ Antes de acceder al disco se verifica la cache de memoria.
- ◉ Desventajas:
 - Requiere mucha memoria.
 - Consumo de tiempo en las verificaciones.
 - Consumo de tiempo en el mantenimiento.

Servidor.

- ◉ Soluciones:
- ◉ Múltiples subprocesos:
- ◉ El servidor está constituido por:
 - ◉ Un proceso *front-end*, que acepta todas las solicitudes y las deriva a un módulo de procesamiento.
 - ◉ n módulos de procesamiento.
 - ◉ Todos los subprocesos pertenecen al mismo proceso.
 - ◉ Tienen acceso a la misma cache.
- ◉ Otro diseño:
 - ◉ No tener el *front-end*.
 - ◉ Cada módulo de procesamiento trata de capturar sus propias solicitudes.

Servidor.

- ⦿ Soluciones:
- ⦿ Múltiples subprocesos:
- ⦿ Cuando un módulo se boquea esperando un archivo desde el disco.
- ⦿ Otro proceso puede estar atendiendo otras solicitudes
- ⦿ Otra mejora: Tener varios discos.

Servidor.

- ⦿ Soluciones:
- ⦿ Granja de servidores:
- ⦿ Cada proceso ejecutará en un procesador diferente.
- ⦿ Funciona de la misma manera que el esquema de múltiples subprocesos.
- ⦿ Combinar ambos mecanismos, teniendo en cada procesador, múltiples subprocesos.
- ⦿ Problemas:
 - No se pueden emplear caches compartidas, todos tienen memoria local.
 - Conexiones TCP se establecen entre el *front-end* y el cliente. Se puede solucionar transfiriendo las conexiones al nodo de procesamiento (TCP handoff). Contesta el módulo de procesamiento directamente al cliente.

Protocolo de Transferencia de Hipertexto (HTTP).

- ⦿ Transferencias de la Web Wide Web.
- ⦿ Especifica los mensajes que se intercambian entre el cliente y el servidor.
- ⦿ Orientado a transacciones.
- ⦿ Es un protocolo sin estados: cada operación se trata independiente de las restantes.
- ⦿ HTTP 1.0: Para cada transacción se establece una nueva conexión TCP entre el cliente y el servidor.
- ⦿ HTTP 1.1: conexiones TCP persistentes: es posible realizar más de una transacción..

Protocolo de Transferencia de Hipertexto (HTTP).

- ⦿ Conexión:
 - El navegador establece una conexión TCP con el puerto 80 del servidor.
- ⦿ Métodos:
 - Soporta operaciones con páginas web y con objetos.
 - Cada solicitud consiste en una o más líneas de texto ASCII.
 - La primera palabra es el método solicitado.
 - Cada solicitud obtiene una respuesta que consiste en una línea de estado, e información adicional (p. ej. una página web).

Protocolo de Transferencia de Hipertexto (HTTP).

- ⦿ Métodos:
- ⦿ **GET**: solicitud para obtener la página identificada en la URL.
- ⦿ **OPTIONS**: solicitud de información disponible.
- ⦿ **HEAD**: solicitud de la cabecera de la página web.
- ⦿ **PUT**: solicitud para aceptar la entidad adjunta y almacenarla bajo la URL proporcionada.
- ⦿ **POST**: solicitud para aceptar la entidad adjunta como una nueva subordinada a la URL identificada.
- ⦿ **PATCH**: como PUT pero transfiere las diferencias.
- ⦿ **DELETE**: solicita al servidor que elimine el recurso identificado por la URL.

Protocolo de Transferencia de Hipertexto (HTTP).

- ⦿ Métodos:
- ⦿ Ejemplo:
GET nombre_archivo HTTP/1.1

Protocolo de Transferencia de Hipertexto (HTTP).

- Encabezados:
- Encabezados de solicitud:
- A la línea de solicitud le puede seguir una o mas líneas adicionales que contienen información.

Encabezados	Significado
User-Agent	Información del navegador y su plataforma.
Accept	Tipo de pagina que se pueden manejar.
Accept-Charset	Conjunto de caracteres aceptables.
Accept-Encoding	Codificación que acepta.
Accept-Language	Lenguajes que maneja.
Host	Nombre DNS del servidor.
Authorization	Lista de credenciales del cliente.
Cookie	Envía al servidor una cookie preestablecida.
Date	Fecha y hora en que se envió el mensaje
Upgrade	El protocolo al que el emisor desea cambiar.

Protocolo de Transferencia de Hipertexto (HTTP).

- Encabezados:
- Encabezados de respuesta:
- Información de respuesta desde el servidor al cliente.

Encabezados	Significado
Server	Información del servidor.
Accept	Tipo de pagina que se pueden manejar.
Content-Encoding	Codificación del contenido.
Content-Language	Lenguajes utilizado.
Content-Length	Longitud de la página en bytes.
Content-Type	Tipo MIME de la pagina.
Last-Modified	Ultima modificación de la página.
Location	Envía al servidor una cookie preestablecida.
Accept-Range	Rangos de bytes que aceptara el servidor.
Set-Cookie	Cookie para que guarde el cliente.
Date	Fecha y hora en que se envió el mensaje.
Upgrade	El protocolo al que el emisor desea cambiar.

Protocolo de Transferencia de Hipertexto (HTTP).

- Ejemplo:

```
$ telnet www.ietf.org 80 >pag_web.log
GET /rfc.html HTTP/1.1
Host: www.ietf.org
```

Protocolo de Transferencia de Hipertexto (HTTP).

- Ejemplo:

```
Trying 64.170.98.32...
Connected to www.ietf.org.
Escape character is '^]'.
HTTP/1.1 200 OK
Date: Tue, 24 Nov 2009 13:14:58 GMT
Server: Apache/2.2.4 (Linux/SUSE) mod_ssl/2.2.4 OpenSSL/0.9.8e PHP/5.2.6 with Suhosin-Patch mod_python/3.3.1 Python/2.5.1 mod_p
Last-Modified: Mon, 23 Nov 2009 18:45:01 GMT
ETag: "1b50042-2f19-3914a540"
Accept-Ranges: bytes
Content-Length: 12057
Vary: Accept-Encoding
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin template="/Templates/GeneralPage.dwt" codeOutsideHTMLOutsideLocked="tr
<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<!-- InstanceBeginEditable name="title" -->
<title>Internet Engineering Task Force</title>
<!-- InstanceEndEditable -->
<script type="text/javascript" language="JavaScript" src="css/ietf.js"></script>
<script type="text/javascript">
document.write('<link id="sheet4" href="/css/ietf4.css" rel="stylesheet" type="text/css" />');
document.write('<link id="sheet3" href="/css/ietf3.css" rel="stylesheet" type="text/css" />');
document.write('<link id="sheet2" href="/css/ietf2.css" rel="stylesheet" type="text/css" />');
</script>
<link id="sheet1" href="css/ietf.css" rel="stylesheet" type="text/css" />
</head>
<body onload="setStyle(0);">
<div id="container">
```